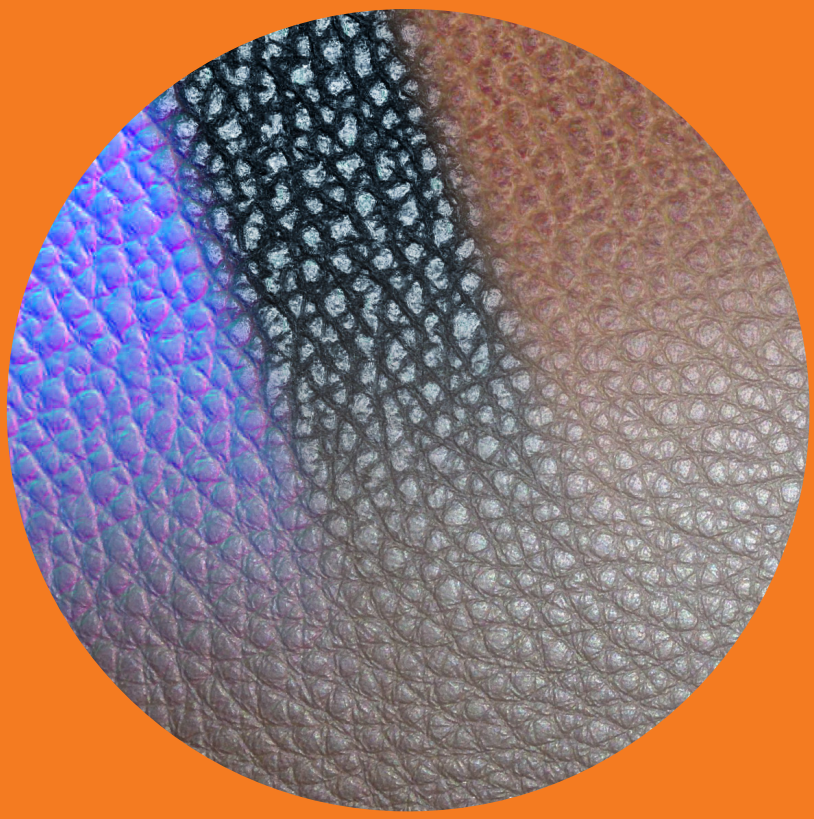
Department of Computer Science

Computational Methods for Capture and Reproduction of Photorealistic Surface Appearance

Miika Aittala



Aalto University

DOCTORAL DISSERTATIONS

Aalto University publication series
DOCTORAL DISSERTATIONS 199/2016

Computational Methods for Capture and Reproduction of Photorealistic Surface Appearance

Miika Aittala

A doctoral dissertation completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Science, at a public examination held at the lecture hall T2 of the school on 28 October 2016 at 12.

Aalto University
School of Science
Department of Computer Science

Supervising professor
Professor Jaakko Lehtinen, Aalto University, Finland

Preliminary examiners
Professor Szymon Rusinkiewicz, Princeton University, USA
Professor Todd Zickler, Harvard University, USA

Opponent
Professor Steve Marschner, Cornell University, USA

Aalto University publication series
DOCTORAL DISSERTATIONS 199/2016

ISBN 978-952-60-7048-3 (printed)
ISBN 978-952-60-7047-6 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-60-7047-6

Unigrafia Oy
Helsinki 2016

Finland

Publication orders (printed book):
miika.aittala@iki.fi

441 697
Printed matter

Aalto University

Abstract

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Author
Miika Aittala

Name of the doctoral dissertation
Computational Methods for Capture and Reproduction of Photorealistic Surface Appearance

Publisher School of Science

Unit Department of Computer Science

Series Aalto University publication series DOCTORAL DISSERTATIONS 199/2016

Field of research Computer Graphics

Manuscript submitted 8 June 2016 **Date of the defence** 28 October 2016

Permission to publish granted (date) 1 September 2016 **Language** English

☐ **Monograph** ☒ **Article dissertation** ☐ **Essay dissertation**

Abstract

This thesis addresses the problem of capturing and reproducing surface material appearance from real-world examples for use in computer graphics applications. Detailed variation of color, shininess and small-scale shape is a critically important factor in visual plausibility of objects in synthetic images. Capturing these properties relies on measuring reflected light under various viewing and illumination conditions. Existing methods typically employ either complex mechanical devices, or heuristics that sacrifice fidelity for simplicity. Consequently, computer graphics practitioners continue to use manual authoring tools.

The thesis introduces three methods for capturing visually rich surface appearance descriptors using simple hardware setups and relatively little measurement data. The specific focus is on capturing detailed spatial variation of the reflectance properties, as opposed to angular variation, which is the primary focus of most previous work. We apply tools from modern data science — in particular, principled optimization-based approaches — to disentangle and explain the various reflectance effects in the scarce measurement data.

The first method uses a flat panel monitor as a programmable light source, and an SLR camera to observe reflections off the captured surface. The monitor is used to emit Fourier basis function patterns, which are well suited for isolating the reflectance properties of interest, and also exhibit a rich set of mathematical properties that enable computationally efficient interpretation of the data.

The other two methods rely on the observation that the spatial variation of many real-world materials is stationary, in the sense that it consists of small elements repeating across the surface. By taking advantage of this redundancy, the methods demonstrate high-quality appearance capture from two photographs, and only a single photograph, respectively. The photographs are acquired using a mobile phone camera.

The resulting reflectance descriptors faithfully reproduce the appearance of the surface under novel viewing and illumination conditions. We demonstrate state of the art results among approaches with similar hardware complexity. The descriptors captured by the methods are directly usable in computer graphics applications, including games, film, and virtual and augmented reality.

Keywords computer graphics, surface appearance, materials, reflectance, rendering, texture, inverse problems, optimization

ISBN (printed) 978-952-60-7048-3 **ISBN (pdf)** 978-952-60-7047-6

ISSN-L 1799-4934 **ISSN (printed)** 1799-4934 **ISSN (pdf)** 1799-4942

Location of publisher Helsinki **Location of printing** Helsinki **Year** 2016

Pages 201 **urn** http://urn.fi/URN:ISBN:978-952-60-7047-6

Aalto-yliopisto

Tiivistelmä

Aalto-yliopisto, PL 11000, 00076 Aalto www.aalto.fi

Tekijä
Miika Aittala

Väitöskirjan nimi
Laskennallisia menetelmiä pintamateriaalien ulkonäön kaappaamiseen ja toisintamiseen

Julkaisija Perustieteiden korkeakoulu

Yksikkö Tietotekniikan laitos

Sarja Aalto University publication series DOCTORAL DISSERTATIONS 199/2016

Tutkimusala Tietokonegrafiikka

Käsikirjoituksen pvm 08.06.2016 **Väitöspäivä** 28.10.2016

Julkaisuluvan myöntämispäivä 01.09.2016 **Kieli** Englanti

☐ **Monografia** ☒ **Artikkeliväitöskirja** ☐ **Esseeväitöskirja**

Tiivistelmä

Tämä väitöskirja käsittelee pintamateriaalien ulkonäön automaattista kaappaamista ja toistamista tietokonegrafiikan sovelluksia varten. Ulkonäkö muodustuu värien, kiiltävyyden ja pinnanmuotojen vaihtelusta, ja se on keskeisen tärkeä osa esineiden visuaalista uskottavuutta synteettisessä kuvanmuodostuksessa. Näiden ominaisuuksien kaappaaminen edellyttää heijastuneen valon määrän mittaamista lukuisissa valaistus- ja katseluolosuhteissa. Pääosa olemassaolevista menetelmistä hyödyntää joko monimutkaisia mekaanisia laitteita tai yksinkertaistettuja heuristiikkoja, jotka eivät toista pintojen ulkonäköä uskollisesti. Tämän seurauksena suurin osa käytännön sisällöntuotantotyöstä tehdään edelleen käsin.

Tässä väitöskirjassa esitellään kolme menetelmää visuaalisesti rikkaiden pintamateriaalimallien kaappaamiseksi käyttäen yksinkertaisia laitteistoja ja suhteellisen vähälukuisia mittauksia. Erityinen huomio kohdistuu yksityiskohtaisen, pinnalla vaihtuvan rakenteen mallintamiseen, siinä missä aikaisemmassa tutkimuksessa on usein keskitytty ensisijaisesti katselukulman vaikutuksen mallintamiseen. Esiteltävät menetelmät hyödyntävät modernin data-analyysin työkaluja — erityisesti hyvin määriteltyjä optimointitehtäviä — erotellakseen ja selittääkseen havaitut heijastusilmiöt vähäisessä mittausdatassa.

Ensimmäinen menetelmä hyödyntää litteää monitoria ohjelmoitavana valonlähteenä ja järjestelmäkameraa pinnasta heijastuneen valon määrän mittaamiseen. Monitorilla näytetään Fourier-kantafunktioita, jotka soveltuvat hyvin heijastusfunktioiden matemaattiseen käsittelyyn ja tulkitsemiseen, ja joiden lukuisat matemaattiset erityisominaisuudet mahdollistavat tehokkaan laskennallisen ratkaisumenetelmän muodostamisen. Jälkimmäiset kaksi menetelmää hyödyntävät todellisen maailman pinnoille tyypillistä stationaarista rakennetta, jossa keskenään samankaltaiset pienet elementit toistuvat koko pinnan yli. Yhdistämällä mittaushavaintoja toistuneiden elementtien kesken menetelmät saavuttavat korkealaatuisia kaappaustuloksia vain kahdesta ja yhdestä valokuvasta. Valokuvien ottamiseen käytetään matkapuhelimen kameraa.

Kaapatut heijastusmallit toistavat pintojen ulkonäön uskollisesti uusissa katselu- ja valaistusolosuhteissa. Tulokset vertautuvat edullisesti aiempiin vastaavia kevyitä laitteistoja hyödyntäviin menetelmiin. Ne ovat suoraan käytettävissä useissa tietokonegrafiikan sovelluksissa, mukaanlukien pelit, elokuvat sekä virtuaali- ja lisätty todellisuus.

Avainsanat tietokonegrafiikka, pintamateriaalit, heijastavuus, synteettinen kuvantaminen, tekstuuri, käänteisongelmat, optimointi

ISBN (painettu) 978-952-60-7048-3 **ISBN (pdf)** 978-952-60-7047-6

ISSN-L 1799-4934 **ISSN (painettu)** 1799-4934 **ISSN (pdf)** 1799-4942

Julkaisupaikka Helsinki **Painopaikka** Helsinki **Vuosi** 2016

Sivumäärä 201 **urn** http://urn.fi/URN:ISBN:978-952-60-7047-6

Preface

The work presented in this thesis was carried out between the years 2011 and 2016 at the Department of Computer Science (and its predecessor Department of Media Technology) at Aalto University School of Science, and during a brief but memorable visit as a research intern at NVIDIA Corporation's Helsinki offices from September 2012 to January 2013.

First and foremost, I want to express my gratitude to my advisor Prof. Jaakko Lehtinen. It was he who invited me to pursue a doctoral degree in the first place, and offered to act as my advisor before even having received his own professorship. Without his encouragement I would surely never have started on this path, and what more, never even have entertained the possibility of publishing my work at a venue like SIGGRAPH. The fact that I did — three times, no less — is in no small part thanks to Jaakko's knowledge, guidance and enthusiasm, as well as the patience and confidence he had in me as I pursued the sometimes rather vague and ambitious ideas that led to these results.

I would also like to offer my gratitude to the remaining small group of co-authors in these papers, namely Prof. Tim Weyrich and Dr. Timo Aila, for fruitful collaboration. I also want to thank Prof. Lauri Savioja, who acted as my primary advisor during the first years of my doctoral studies, and gave me the opportunity to work freely on my chosen topics.

I further wish to thank my co-workers Markus Kettunen and Ari Silvennoinen at the Computer Graphics group in Aalto, as well as Dr. Samuli Laine and Tero Karras (and Jaakko and Timo) at the Research team at NVIDIA Helsinki, and the numerous other colleagues I've had the pleasure of interacting with over the years, for good times, illuminating discussions, as well as collaboration in various publications that are not included in this thesis.

I also wish to acknowledge the generous financial support offered to me

during this work by the HeCSE doctoral programme.

Finally, I am grateful to my friends and family — my mother Maarit and my brothers Tommi and Joona — for their unconditional support in my endeavours, and wish to acknowledge the memory of my father Kari, whose influence is most likely the reason I gravitated towards this path in my formative years.

Nice, France, September 26, 2016,

Miika Aittala

Contents

List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.

I Miika Aittala, Tim Weyrich, and Jaakko Lehtinen. Practical SVBRDF Capture in the Frequency Domain. *ACM Transactions on Graphics*, Volume 32, Issue 4, Article No. 110, July 2013.

II Miika Aittala, Tim Weyrich, and Jaakko Lehtinen. Two-shot SVBRDF Capture for Stationary Materials. *ACM Transactions on Graphics*, Volume 34, Issue 4, Article No. 110, August 2015.

III Miika Aittala, Timo Aila, and Jaakko Lehtinen. Reflectance Modeling by Neural Texture Synthesis. *ACM Transactions on Graphics*, Volume 35, Issue 4, Article No. 65, July 2016.

Author's Contribution

Publication I: "Practical SVBRDF Capture in the Frequency Domain"

The author designed and implemented the methods, conducted the experiments, and participated in writing the manuscript.

Publication II: "Two-shot SVBRDF Capture for Stationary Materials"

The author designed and implemented the methods, conducted the experiments, and participated in writing the manuscript.

Publication III: "Reflectance Modeling by Neural Texture Synthesis"

The author designed and implemented the methods, and conducted the experiments (with the exception of the introductory toy examples), and participated in writing the manuscript.

1. Introduction

1.1 Overview and goals

Computer graphics is a field of art and science concerned with computer-assisted creation of visual imagery. Photorealistic image synthesis, in particular, aims to reproduce the visual appearance of reality by simulating the interaction of light and matter in a scene, so as to mimic the image formation process that gives rise to our visual sensations. This process is called *rendering*. The task is difficult, as humans are accustomed to viewing the real world, and hence quick to spot poor imitations of reality. Nevertheless, the behavior of light is well understood theoretically, and highly accurate practical rendering algorithms have been known for decades [73, 125]. These methods are capable of producing images that are indistinguishable from photographs. In recent years, they have found widespread adoption in film and visualization industry, as advances in computational capabilities of hardware have made their use feasible. Real-time applications such as games and virtual reality must still resort to approximations and shortcuts for performance reasons, but the field is advancing rapidly [62, 122].

The results from these methods are, however, only as good as the input data: one also needs high-quality content as an input to the renderer. Roughly speaking, a renderer uses *geometry*, *materials* and *lighting* to produce the image, as illustrated in Figure 1.1. This content is typically created by skilled artists in a time-consuming manual modeling process. A typical goal is to create high-quality virtual replicas of real-world scenes. With this in mind, it would make sense to bypass the manual work by directly capturing this content from real-world examples. Indeed, a large body of research exists on capturing each of the types of content

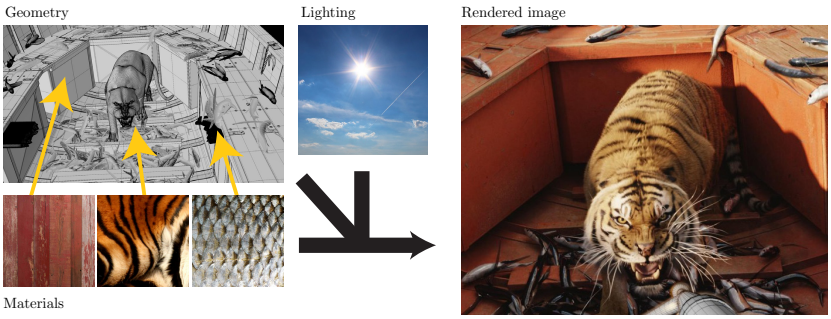


Figure 1.1. A renderer computes a photorealistic synthetic image out of the description of geometry, materials and lighting. Image © 2012 20th Century Fox.

enumerated above.

In this thesis, we are concerned with capturing and reproducing the appearance of real-world materials. In other words, we are looking to capture mathematical descriptors that predict how a given surface looks when seen under arbitrary lighting and viewing conditions. This includes effects such as color, shininess, bumpiness and translucency of a material, and the spatial variation of these properties. While impressive results have been demonstrated in previous work on appearance capture, these methods generally require complex physical devices and capture procedures, or are limited in their fidelity and applicability. Consequently, they find limited use among practitioners.

Our goal in this thesis is to extend this work by simplifying the task for the user. In particular, we are looking to design low-cost physical setups with simple capture procedures and no custom hardware or moving parts. However, this limits the quantity and type of the data we can collect: the raw data no longer directly reveals the information we are looking to recover. The major theme in this work is the use of advanced data analysis techniques for extracting material appearance descriptors out of scarce measurement data — in effect, shifting the complexity from the measurement *acquisition* stage to the measurement *interpretation* stage. In particular, steps are made towards solving for rich material properties from a single photograph alone — an elusive long-term goal in the field.

1.2 Materials

All solid objects are composed of molecules bound together. In everyday situations, it is convenient to distinguish between material and shape: material is the "continuous" substance from which an object is built, whereas

a) b)

Figure 1.2. Examples of real-world materials.

shape describes the macroscopic form into which it is arranged. The material determines the chemical and many physical properties of the object: for example, at what temperature does it melt, how it responds to mechanical stress, and how it interacts with electromagnetic radiation — in particular, visible light. The latter determines the visual appearance of the surface.

The exact division is context-dependent. For example, woven fabric might be considered as a material when designing clothes, but from an ant's point of view the individual threads are large-scale shapes. At an opposite extreme, a satellite might consider "forest" and "city" to be materials covering the Earth's surface. Most objects are composed of multiple materials with various degrees of heterogeneity. Consider the hammer in Figure 1.2a: it consists of a wooden handle and a steel head, and the head is partially rusted. Likewise, many materials are combinations of multiple sub-materials: tarmac consists of countless small rocks embedded in tar, as seen in Figure 1.2b. One typically considers any sufficiently repeating detail, such as microscopic porosity or surface roughness, or macroscopic texture, to be a property of the material.

Interaction between materials and light is of particular interest to computer graphics and vision. The very reason we are able to see objects is because light has scattered from them towards our eyes. The manner of this scattering gives strong clues about the identity of the material, in the form of effects like color, shininess, bumpiness, translucency, and spatial variations thereof. In computer graphics, these effects must be simulated, and their visual plausibility is of central interest.

The task of simulating these interactions is typically divided between light transport and appearance modeling. The former is concerned with keeping track of the global distribution of the scattered light in a scene. The latter is concerned with the local scattering events themselves. A typ-

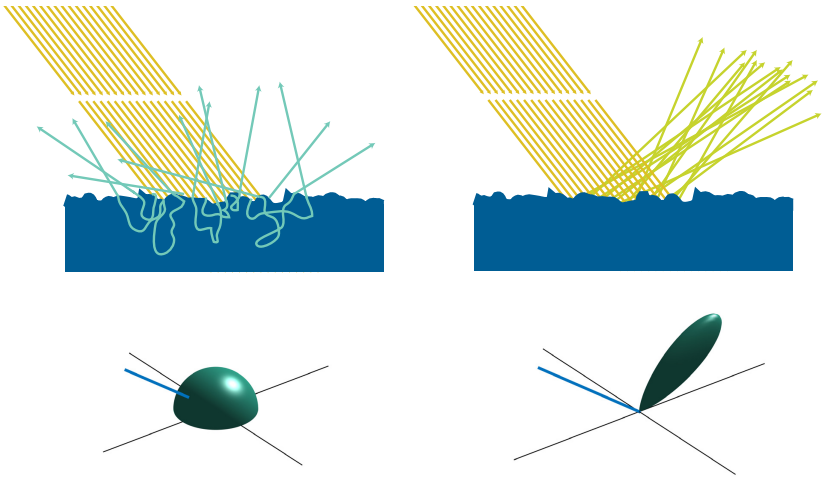


Figure 1.3. Mechanism of diffuse and specular reflectance at microscopic scale. Diffuse reflectance (left) arises when the light penetrates into the material, bounces around randomly, and emerges at a nearby location. The random walk within the material interior scrambles the exitant directions perfectly, giving rise to a uniform distribution of reflections (shown as a polar plot in bottom left). Specularity (right) is caused by immediate reflection at the surface boundary. The microscopic roughness of the surface randomly scrambles the reflection directions, giving rise to a distribution that is typically biased towards the perfect mirror direction (bottom right). A part of the rays are absorbed by the surface. The absorption probability is wavelength-dependent, and gives the surface its apparent color. Typical dielectric materials exhibit both specular and diffuse reflectance. Reflections from metallic surfaces are purely specular.

ical appearance model must be able to predict the distribution of outgoing scattered light from a surface, given a distribution of incoming light.

In most materials, light scatters at or very near the object's surface and does not penetrate deeply into the interior. Hence, in typical applications it suffices to model an object as its two-dimensional exterior surface, instead of a full three-dimensional solid. Similarly, it suffices to endow this surface with a *surface material*, which describes the material properties that are relevant for modeling local surface reflections and refractions. More general phenomena, such as non-local sub-surface scattering, are relevant in some important special cases such as human skin [69]. Similarly, complex volumetric structure of e.g. hair, fur and many fabrics requires specialized techniques for plausible visual reproduction [72, 134]. In this thesis, the focus is on surface reflectance, and we leave these generalizations out of our scope.

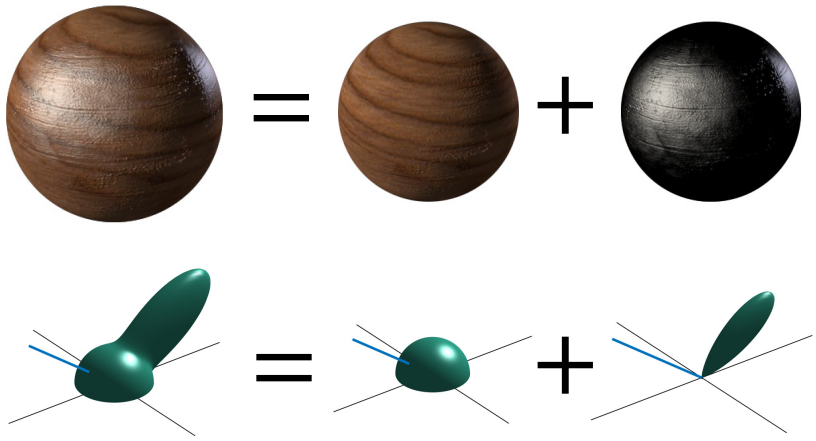


Figure 1.4. Top row shows the appearance of a material decomposed into the diffuse and specular components. The diffuse component is perfectly matte, and describes the base color of the surface. The specular component is responsible for the highlights against light sources, and (possibly blurry) reflections of the surrounding environment. The bottom row shows corresponding distributions of reflectance directions. Note however that the distribution of reflectance varies from point to point on the surface, as the material depicted is not homogeneous.

1.2.1 Modeling surface reflectance

Two mechanisms of surface reflection tend to dominate in most materials. These are illustrated in Figure 1.3. In *diffuse* reflectance, the light does penetrate into the material, but it re-emerges at practically the same position. However, as the light makes multiple random bounces within the material, its exitant direction becomes uniformly randomized, resulting in an appearance that does not depend on viewing direction. This gives rise to the "base color" of a surface. The second common mechanism is *specular* reflectance, where the reflection occurs directly at the surface boundary. The roughness of the surface scrambles the exitant directions, typically giving rise to a smoothened reflection distribution. Note in particular that the observed specular reflection does depend on the viewing direction. Intuitively, this gives the surface its "shininess". The appearance caused by these types of reflections is demonstrated in Figure 1.4.

These notions are formalized by a reflectance descriptor called the *bidirectional reflectance distribution function* (BRDF). It is a function that describes this angular distribution of reflections as depicted in Figure 1.3. It also varies with respect to the angle of incidence of the light. In total, the BRDF is a four-dimensional function, as its value depends on the incoming and outgoing light directions, each characterized by a pair of

angles.

This dimensionality is high. Exhaustively tabulating the BRDF value for every pair of angles is prohibitively expensive for most applications. Dividing a four-dimensional grid to 100 points along each dimension, for example, results in 10 million values that need to be specified. Fortunately, the space of naturally occurring reflectance functions is not arbitrary. They exhibit significant amounts of structure and redundancy, which suggests that a lower-dimensional characterization should suffice to describe the key features of any BRDF. As noted above, most BRDFs are superpositions of two simpler components, namely the diffuse and specular part. The diffuse component is characterized by its color and intensity (*albedo*). The main features of the specular component are likewise the albedo, and also the *glossiness* which characterizes the "opening angle" of the reflected lobe. Some materials also exhibits *anisotropy*, which results in elongated specular highlights such as seen in brushed metal. Typical *isotropic* materials do not have this property. These considerations have inspired a large body of research in *parametric BRDF models* [103, 9, 25, 130, 4, 84, 14], which model BRDFs using such low-dimensional characterizations.

Spatial variation The BRDF only describes the *angular variation* of the reflectance at a single point, or for a homogeneous material as seen in Figures 1.5a and 1.5b. Almost all real-world materials also exhibit significant *spatial variation*, as illustrated in Figure 1.5c. Arguably, it is often the most prominent feature of a surface material. Most everyday surfaces are well modeled by a small set of angular variation effects; it is the spatial variation of these properties that really sets different materials apart and gives them their distinctive characters.

The BRDF can be straightforwardly extended with two spatial dimensions, yielding the six-dimensional *spatially varying BRDF*, or SVBRDF. Exhaustive tabulation of these high-dimensional functions is out of the question for most practical applications. Instead, it is common to use "texture maps" that describe the variation of the parameters of a low-dimensional BRDF model across the surface. An additional *normal map* is often used to model small-scale surface shape variations. Figure 1.6 shows an example of such a representation. These kinds of surface appearance descriptors are widely used in industry [15]. Most software modeling packages and real-time rendering engines use them by default, although the specifics of the models vary.

a) b) c)

Figure 1.5. a) Object with a homogeneous plastic material with no spatial variation, sufficiently described by BRDF alone. b) A homogeneous metal material. However, on closer inspection, the surface has visible small-scale roughness and wear and tear. Such details are often critically important for visually plausible images. An SVBRDF or a similar spatially varying material descriptor is required for modeling these effects. c) A general spatially varying material.

The methods in this thesis are concerned with capturing SVBRDF maps of this kind.

1.3 Capturing surface reflectance

This thesis is concerned with *capturing* SVBRDFs from real-world surfaces. Given that the SVBRDF predicts the proportion of light reflected between each pair of incoming and outgoing angles at each surface point, capturing it is in fact straightforward in principle. One merely needs to translate a light source and a camera to each angle in turn, and record the amount of light reflected by each surface point by taking a photograph. Figure 1.7 illustrates this principle. A device built for this purpose is known as a *gonioreflectometer* [96, 27].

However, this approach is not very practical due to the high dimensionality of the functions: a very large number of photographs need to be captured in order to sample the angular space with sufficient resolution. Furthermore, the device requires precise robotic mechanical control and careful calibration to ensure the reliability of the measurements.

Fortunately, real-world reflectance exhibits significant structure, which can be exploited in order to extract the relevant information from a smaller amount of measurement data. As a very simple example, due to the reflection mechanisms described above, surfaces tend to reflect most strongly

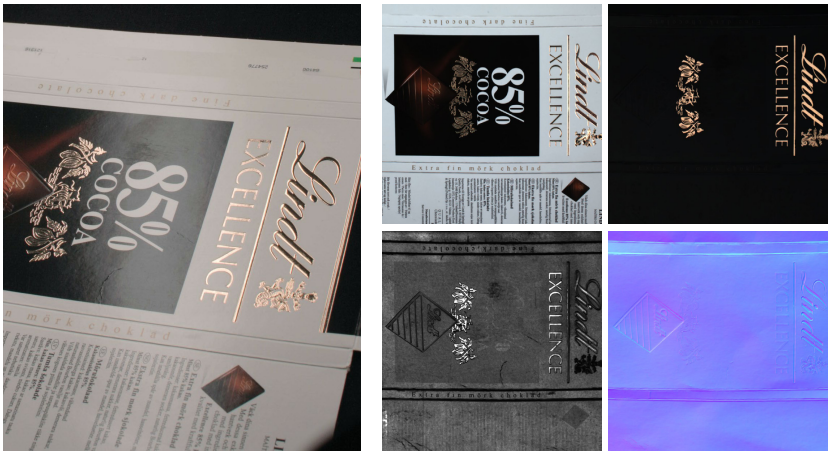


Figure 1.6. An example of the kind of parametric SVBRDFs we aim to capture. On the left is a photograph of the physical material. Notice the shininess hinted by the specular highlights, and the embossed gold lettering. On the right is an SVBRDF "texture map" representation of the material (in fact, captured using the method in Publication I). The diffuse albedo map (top left) describes the base diffuse color of the material. The specular albedo map (top right) describes the intensity of the specular highlight: there is some specularity across the entire material, but the gold letters stand out as bright yellow. The glossiness map (bottom left) describes the glossiness of the specular reflection. Note how the golden parts, again, are more mirror-like, and in particular the deeper creases are rather dull. Finally, the normal map (bottom right) describes the variations of the surface shape.

towards the perfect mirror direction, and the reflectance falls of smoothly towards other viewing angles. It is unlikely (if possible in theory) that one would find a pocket of strong reflectance in some completely unrelated direction. This suggests that certain directions may be sampled less densely, as well as the possibility of interpolating and extrapolating reflectance information from incomplete measurements. On the other hand, one does not necessarily need to make direct point measurements of individual BRDF values — for example, large area light sources illuminate the surface from a wide range of angles, and may help us to collect reflectance information from multiple angles simultaneously.

Besides angular variation, also the spatial variation is typically structured. For example, the surface of a given object typically only exhibits a small number of different reflectances, and consequently measured information can often be shared across surface locations. Consider e.g. Figure 1.6: the shininess properties of the cardboard are roughly similar across the surface, even though the specific spatial features vary.

Incorporating knowledge about such regularities into the design of the method, often in highly indirect and non-trivial ways, is a central under-

Figure 1.7. Photographs of a book cover under various viewing and illumination directions. Notice how different aspects of the surface color, glossiness, and shape are revealed under the different conditions. These photographs represent only a tiny fraction of the number of photographs required for exhaustive sampling of the reflectance functions. Careful calibration and mechanical control are required to ensure reliability of the measurements.

lying theme in this thesis. Indeed, similar consideration have inspired a variety of exotic capture devices (e.g. [44, 48, 64, 50, 35, 21]) that make strategic measurements most likely to reveal the desired reflectance information. For example, Gardner et al. [44] translate a linear light source (fluorescent tube) over a surface and infer the material properties from its reflections. Another problem arises with these approaches, however: the measurements often do not directly reveal the values of the SVBRDF. Instead, they need to be disentangled algorithmically.

In this thesis, we model the task of recovering the reflectance descriptors from indirect measurements as an *inverse problem*. The idea is to form a mathematical *predictive* (or *forward*) model, which is essentially a virtual replica of the real-world measurement setup. This model can be used to test different hypotheses about the reflectance of the underlying material. Specifically, we use a principled process of *optimization* to drive a search for a material descriptor that would produce the same measurements as those we observed in the real world. The assumption is that such a descriptor is the underlying explanation behind the real-world observations as well, and hence represents the true reflectance properties of the surface. Figure 1.8 illustrates this process. A canonical example is op-

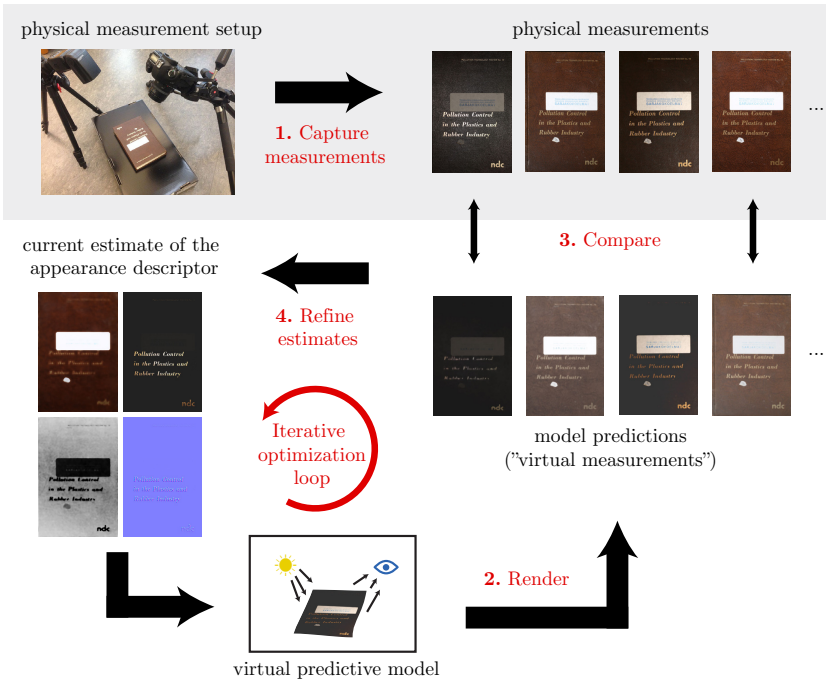


Figure 1.8. A schematic overview of the capturing and data fitting process by optimization. First, a set of measurements is captured by a physical setup that records images of the physical material sample under varying conditions. Then, a rough initial guess about the underlying appearance descriptor is made. A virtual predictive model is used to simulate the appearance of this descriptor under the same conditions as those used in the measurements. The predictions of the model are compared with the physical measurements, and the estimate of the descriptors is refined in a way that improves the match. This process of prediction and refining is repeated iteratively until it converges to a descriptor solution that accurately reproduces the physical input data. The physical capture setup depicted is fictional but reminiscent of a gonioreflectometer [96, 27].

timizing for the unknown surface color, shininess and bumpiness parameters (such as shown in Figure 1.6), so that the renderings of the surface end up matching the input photographs, the latter taken under various controlled lighting and viewing conditions (e.g. as in Figure 1.7).

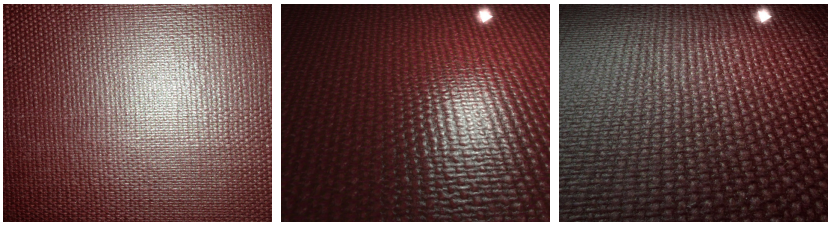
1.3.1 Mathematical challenges

A central theme in this thesis is the *joint design* of the physical measurement setup and the corresponding interpretation model. On one hand, the captured data must sufficiently well encode the reflectance information of interest, without being too cumbersome to acquire. On the other hand, it must also be interpretable using a tractable and reasonably efficient algorithm.

The ultimate goal of capturing material appearance is *reproduction*: we are looking to use the captured appearance under novel viewing and lighting conditions. Our desire to build practical low-cost physical setups prevents us from exhaustively measuring every possible combination of reflection directions. For example, if we constrain the camera and the material sample to fixed positions (as we do in all the methods in this thesis), we only obtain reflectance information from a single exitant angle at any given point. This leaves a large portion of the angles unexplored. Thankfully, as outlined above, the angular behavior of reflectance functions is somewhat predictable, and plausible extrapolations can be made from well-chosen slices of the functions. This requires care due to the high dimensionality and non-linearity of the functions involved.

The key problem is *ill-posedness*: the data is often ambiguous and admits to multiple explanations. A basic example is the difficulty of reflectance recovery from a single photograph. One can easily find an infinite number of different material models that precisely match any given photo. However, vast majority of them fail to *generalize* to novel viewing and lighting conditions, and without additional information there is no way to choose a good one. An example of a trivial solution is an entirely flat and diffuse surface, with the image of the input photograph printed on it. While this solution looks correct from the original angle, it falls apart when the camera and the light are moved: for example, any specular highlights remain fixed to their original positions. Similarly, the shading variations caused by surface bumps may also be interpreted as alternating dark and light regions on a flat surface. See Figure 1.9 for an illustration. This difficulty carries over to more complex setups — for example, it might be difficult to determine the relative amount of diffuse and specular reflectance at a given point, because both parameters may have a similar visual effect under the measurement setup used. In general, solutions to ill-posed problems can be much worse than that that decipted in Figure 1.9, as the optimizer is free to almost arbitrarily mix the various shading parameters unless care is taken.

A related difficulty is *non-convexity*. Optimization methods typically make greedy improvements to the solution in each iteration, and once they end up with a solution that cannot be improved by small nudges to the parameters (a “local minimum”), they finish. Ideally, this happens when the solution parameters correspond to the physical reality, and cannot be improved any further. Unfortunately, the mathematical form of



a) b) c)

Figure 1.9. Generalizing to novel viewing and lighting angles from a single photograph. a) An input photograph taken of a real-world surface with a mobile phone camera and flash. Notice the specular highlight, bumpiness, and the red diffuse color. b) Proper generalization to new viewing and lighting conditions using a material descriptor where the shading effects have been disentangled. In particular, the specular highlight position and the shading of the individual bumps have responded to the new position of the light source. c) A trivial material descriptor that merely paints the surface with the input photograph fails to generalize properly. Notice how the position of the specular highlight and the shading of the bumps still correspond to the original lighting conditions of the photo. However, for this same reason, the descriptor successfully reproduces the appearance of the input photograph under the original viewing conditions, and it is therefore a possible solution to the inverse problem unless we somehow rule it out. The difference is significantly more pronounced in motion.

most optimization problems also leads to the existence of additional spurious local minima. These minima are often clearly sub-optimal, i.e. they are a poor numerical fit to the measurements. Nonetheless, an optimizer is unable to escape them once it falls into them, because in doing so it would need to temporarily accept an even worse numerical fit. The underlying mathematical reason for this phenomenon is the non-convex multimodal shape of the function being optimized [11]. The nature of these local minima is often very difficult to reason about in practice — in the end, one tends to accept their existence, but finds that with careful design decisions and tuning a method becomes "good enough" at avoiding them.

While not always explicitly discussed in the publications, these considerations are highly relevant to all of the methods presented in this thesis. The final design of the algorithms, and the specific configurations of the details, are often products of a long period of experimentation. Unfortunately, quite little can be explicitly quantified about this procedure; the design space is very large, and intuition of the algorithm designer plays a definite role. Nevertheless, once a good configuration is found, it is often quite robust: same design choices and parameter values yield good results for a wide range of inputs. One of the goals in these introductory chapters is to shed some light on these issues, and how they relate to the

algorithms presented (often by necessity quite tersely) in the publications.

1.3.2 Natural materials

The problem of generalizing from incomplete and ambiguous measurements is closely related to the difficulty of incorporating "common sense" into algorithms. Humans are skilled at inferring material properties from scarce data: we often easily recognize the material of an object even from a single photo, and we can predict how it would look like under different conditions. Most candidate explanations are immediately dismissed as implausible. Returning to the example above, a human viewer rarely confuses specular highlights with white blotches of paint. We observe countless such bright blotches on objects every day, and they almost never turn out to stick to the surface when we shift our heads. Consequently, we have *learned* that the "painted-on" hypothesis is extremely unlikely, and apply this assumption to any similar blotches *a priori*. In contrast, computer algorithms based on mathematical reasoning lack this kind of knowledge, and consider either explanation to be equally likely until proven otherwise. While empirical studies in human visual perception (e.g. [123]) have found rather subtle and sophisticated patterns of reasoning involved in related tasks, arguably these processes are nonetheless automatic, in the sense that in vast majority of cases we do not need to stop and explicitly perform logical reasoning in order to interpret the scene presented to us.

We do have some computational tools at our disposal. By using so-called *priors*, we can assign a "plausibility score" to any proposed solution, and use it to resolve ill-posedness without having to capture more data. The idea is to guide the optimizer towards choosing a solution that simultaneously explains the data, and satisfies our *a priori* beliefs about what a good solution should be like. For example, most methods presented in this thesis use *smoothness priors* that favor solutions consisting of smoothly varying regions (as opposed to e.g. rapidly oscillating noise). This encodes our belief that surface points close to one another tend to also have similar properties. However, despite their usefulness, these tools are ultimately rather blunt.

Ideally, a prior would encode human-like understanding of what it means for a solution to be plausible, so that it might be used, for example, to choose a plausible generalization in the deeply ambiguous single-photograph capture problem demonstrated in Figure 1.9. The simple priors we presently

apply are far too weak for this task.

The *manifold* viewpoint posits that naturally occurring materials are concentrated on a tiny but extremely complicated subset of the space of all "mathematically valid" materials. In particular, a randomly chosen SVBRDF is overwhelmingly likely to depict random noise, and fall outside this manifold. Priors may be interpreted as tools for characterizing this manifold.

The modern machine learning approach to similar problems is to instead emulate human learning by repeated observation of real-world examples [54]. The use of *deep neural networks* has recently lead to breakthroughs in applications such as image [117, 119] and speech recognition [63]. These techniques hold a promise for material appearance capture as well. Publication III presents some first steps towards this direction by taking advantage of natural image understanding encoded into such networks.

1.4 Overview of methods

This thesis introduces three publications, each of which describes a method for capturing parametric SVBRDF maps, as illustrated in Figure 1.6.

To keep the methods practical, we aim to perform this task using only commodity hardware, in particular avoiding any moving parts that need to be robotically controlled. We aim to avoid fragile calibration procedures to the extent possible, often choosing to use algorithms that *tolerate* e.g. photometric distortions in the data and gracefully absorb them into the solutions, rather than taking laborous steps towards completely eliminating them. In a similar vein, we aim to produce appearance descriptors that *plausibly* explain and generalize from the scarce observations. While such extrapolations cannot always be an exact match to the photometric ground truth, they are in practice useful in many applications, and may also serve as useful starting points for manual editing and authoring.

In order to focus fully on reflectance, we make the common restriction of assuming that the captured surface is a flat plane, as opposed to general 3D model. Some methods do perform joint capture (e.g. [65, 124]), but this necessarily leads to either a significantly more complicated hardware setup, or compromises in both sub-tasks.

Let us briefly review the ideas behind the methods. They will be discussed more thoroughly in Chapters 5 and 6, as well as in the publica-

tions.

1.4.1 Publication I: Fourier basis measurements

The first publication presents a method for low-cost capture of a wide range of spatially varying materials, using only off-the-shelf commodity hardware in a simple physical setup with no moving parts. The method works by displaying a sequence of Fourier basis functions on an LCD monitor and photographing their reflections off the captured surface. These measurements can be viewed as pointwise measurements of the Fourier transforms of slices of the unknown reflectance functions. They are interpreted by an algorithm that directly renders the corresponding slices in the Fourier domain, and fits the predictions of this model to the data by optimization. The frequency domain data enables effective capture, as many of the interesting features of typical reflectance functions become readily apparent in this domain. The domain is also suited for capturing extremely sharp mirror-like reflections, which are challenging for traditional methods. State of the art results are demonstrated for a variety of example materials.

1.4.2 Publications II & III: Stationary materials

The key observation behind the two latter publications is that most real-world surface materials are *stationary*, or "textured", in the sense that same features keep repeating across a surface. This redundancy suggests an opportunity for tremendous reduction in the amount of required input data. By illuminating the surface using a near-field light source, the repeated features become observed under multiple lighting conditions within a single image. Hence, the single photograph contains information of dozens of traditional distantly viewed and lit photographs. The difficulty lies in combining the information from the different image regions, as identical pieces of material can no longer be directly identified by their pixel values due to the varying lighting.

The methods in both publications measure the reflectance information from a head-on flash photograph from a mobile phone. Aside from that, they take vastly different approaches to solving this problem, resulting in two-shot and single-shot methods, respectively.

In Publication II, this flash photograph is augmented with a second photograph taken under distant environment illumination. This *guide*

photo is used to find explicit matches between points in distinct regions of the surface. The linked points are considered to have the same material, which is solved for by finding a set of parameters that predicts the observed appearance by optimization.

While effective, this approach consists of a sequence of partly heuristic steps, specifically engineered for this particular setup and purpose. A more principled and flexible approach would be to simply optimize the visual match between renderings of the surface and the corresponding flash photograph regions. This would also eliminate the need for a separate guide photo. However, comparing similarity of textures is difficult. Naive pointwise image difference fails as a metric, because the textural features are most likely not aligned: for example, the lines in two images of a brick wall are unlikely to coincide when the images are overlaid, resulting in a large numerical difference. Indeed, at the time of writing of Publication II, no suitable high-quality method existed for this task.

Soon after the publication, Gatys et al. [46] introduced a texture synthesis method based on continuously optimizing the similarity of neural network activation statistics between the solution and a texture exemplar. This resulted in state of the art quality in parametric texture synthesis. The key component in their approach is a textural similarity metric which can be used directly as a part of general optimization problems. In Publication III we use this metric to directly optimize the similarity of our material solution and the input data, essentially synthesizing a small piece of an SVBRDF that summarizes the reflectance information in the flash photo. The approach of combining texture synthesis and material appearance acquisition is novel.

Another intersting aspect about Publication III is its unconventional use of a pre-trained convolutional neural network. In order to perform its original task of classifying images into categories, the network seems to have formed a strong internal understanding about the structure of natural images. Our algorithm takes advantage of this knowledge implicitly. This has potential implications in terms of modeling the space of natural materials.

2. Appearance modeling

We assume that the reader is familiar with the general concepts regarding light transport and material models. We will briefly review these topics from the viewpoint relevant for appearance capture, and in particular for the work in this thesis.

2.1 Radiometry

Cameras and eyes are sensitive to visible light, which is electromagnetic radiation with wavelengths from roughly 400 to 700 nanometers. *Radiometry* is a field of study concerned with measuring electromagnetic radiation.

Computer graphics and vision typically adopt the model of *geometric optics*, where radiation (light) is assumed to propagate along straight paths. Phenomena related to the wave and quantum nature of radiation are ignored, as their effect is negligible at visible wavelengths in vast majority of macroscopic scenes. Similarly, effects such as phosphorescence and fluorescence are ignored.

2.1.1 Radiometric quantities

Let us derive some key radiometric quantities by considering radiation as being composed of quantified "photons" (inspired but not exactly corresponding to the concept in physics), each traveling towards some direction in a straight line at a fixed speed, and each carrying some fixed amount of energy measured in Joules [J]. The actual radiometric quantities arise from a somewhat informal limit argument, where we consider the number and velocity of photons to approach infinity, so as to result in a "continuous" stream of energy. For a more thorough treatment from a similar viewpoint, see Veach [125].

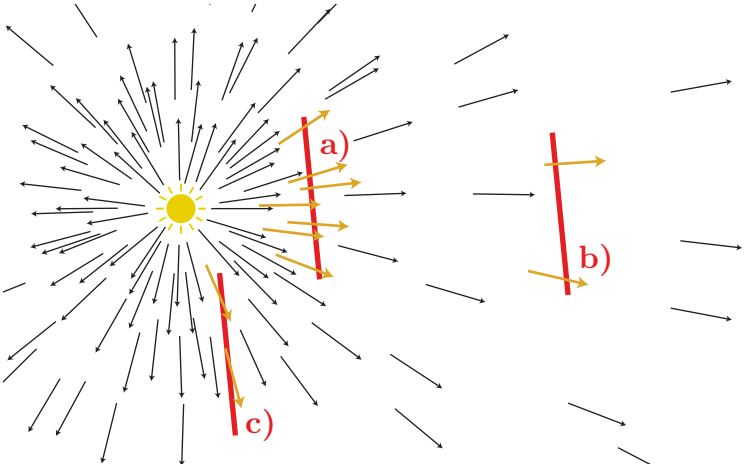


Figure 2.1. A visualization of radiation in terms of photons. The arrows depict the path taken by individual photons during a unit time interval. The light source emits new photons at a constant rate. As they travel away from the light source, they move further apart from one another, and their distribution becomes thinner. The irradiance on a surface is proportional to the expected number of photons traveling through the surface in unit time. Notice how the number of photons that intersected the surface patch (a) is higher than the corresponding number for a distant patch (b) and a patch that is oriented obliquely against the light source (c). In the continuous limit of "infinite amount of photons" and irradiance through an infinitesimally small patch, these effects explain the attenuation of irradiance according to the inverse square distance and to the cosine of the incidence angle.

Point light sources emit photons at a fixed rate per unit time. Hence, we may express the expected rate of emission as *power*, or *radiant flux* $\Phi = \frac{\mathrm{d}Q}{\mathrm{d}t}$ in the unit of Joules per second [J/s], or *Watts* [W].

Consider a virtual surface patch in space, as seen in Figure 2.1. The expected number of of photons per unit time traveling through a unit area of this surface is called *irradiance* and it is measured in units $[\frac{\mathrm{W}}{\mathrm{m}^2}]$. We typically consider the irradiance on an infinitesimal surface patch (which is hence characterized solely by its surface normal), $E = \frac{\mathrm{d}\Phi}{\mathrm{d}A}$. In particular, if the surface patch represents an infinitesimal region of a physical surface, irradiance expresses the radiant power hitting the surface point. Notice in particular that this quantity depends on the distance from the emitter, and the orientation of the surface normal.

Consider now a more selective version of irradiance, that only counts the photons that strike the virtual surface from some small cone of directions around the normal ω of the patch. The cone may be characterized as a region on the unit sphere, as in Figure 2.2. The size of the opening is called *solid angle*, and it is measured simply as the area of the region on the unit sphere. This unit is called *steradian* [sr]. Notice that it simply

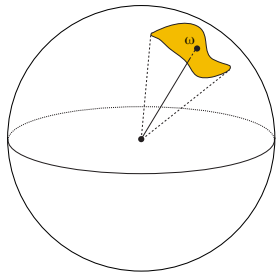


Figure 2.2. A cone of directions visualized on a unit sphere. The size of the opening of this cone is measured as a solid angle, and corresponds to the area of the region subtended by the cone on the unit sphere. Note that this definition is only concerned with the measure of the angles within the cone, not its shape.

generalizes the notion of angles measured in radians (i.e. arc length on unit circle). Our selective irradiance measures the power per unit area per unit solid angle. Hence, its units are $[\frac{\mathrm{W}}{\mathrm{sr\,m^2}}]$.

Letting the surface area and the solid angle shrink towards zero, we arrive at radiance $L = \frac{\mathrm{d}\Phi}{\mathrm{d}\omega\ \mathrm{d}A_\perp}$. Here, the notation $\mathrm{d}A_\perp$ highlights the fact that the infinitesimal area is oriented perpendicularly towards ω. Intuitively, radiance is proportional to the number of photons per unit time passing towards a given direction at a given position in space. This is visualized in Figure 2.3. Contrary to irradiance, radiance from a direction ω is conserved along the straight unoccluded line in that direction, because the same "pencil" of photons is responsible for the radiance at any position along the line.

Note that eyes and cameras are sensitive not only to the amount of light hitting them, but also its direction of arrival. Indeed, a typical perspective image is simply a map of the radiance incident towards the position of the camera from a cone of directions. The typical rendering procedure boils down to computing the distribution of radiance in the scene, and evaluating it at the camera pixels. To determine this distribution, one follows the radiance from light sources as it alternatingly travels to the visible surfaces in free space, and becomes reflected upon hitting these surfaces.

So far we have assumed monochromatic radiation. As the radiometric picture is essentially independent between different wavelengths, the quantities discussed above can be straightforwardly extended to consider the wavelength by adding it as a parameter λ. Hence, for example the spectral radiance is described by a function $L(x, \omega, \lambda)$. While in principle

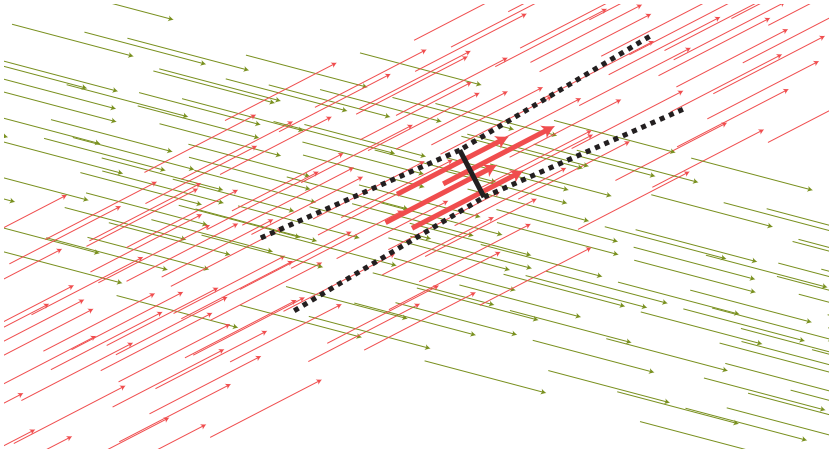


Figure 2.3. Visualization of radiance in terms of discrete photons (see Figure 2.1 for an explanation of the setup). In general scenes, any region of space typically contains photons traveling to all directions. In this simple scene, two collimated "beams" of photons are crossing at a region of space. They are marked with red and green for clarity. This scene might be physically arranged by crossing two laser beams in an otherwise perfectly dark room. Radiance is (again, in the limit) the rate of photons passing through an infinitesimal patch in space *from an infinitesimally small set of angles perpendicular to the patch*. Hence, no radiance is registered from the green photons by the patch depicted in the figure.

λ is an arbitrary positive real number, practical renderers typically use a discrete spectrum consisting of some chosen set of wavelengths — often only three, corresponding to the red, green and blue components. To avoid notational clutter, we will often assume monochromatic radiation in the formulas and discussion. The extension to the spectral case is generally straightforward.

2.2 Reflection and light transport

Consider Figure 1.3. The photons arriving at an infinitesimal surface region either become absorbed, or scatter towards a random direction. The scattering follows some probability distribution over the outgoing angle $\omega_o \in \Omega$, determined by the physical properties of a material, along with its microscopic shape variations. Here, Ω is the upper unit hemisphere above the surface point. The shape of the scattering distribution itself is a function of the incoming angle of the photon, $\omega_i \in \Omega$. In terms of the continuous radiometric quantities, the same distribution expresses the amount of radiance re-emitted towards each direction ω_o, in proportion to the irradiance received from each direction ω_i. The distribution is en-

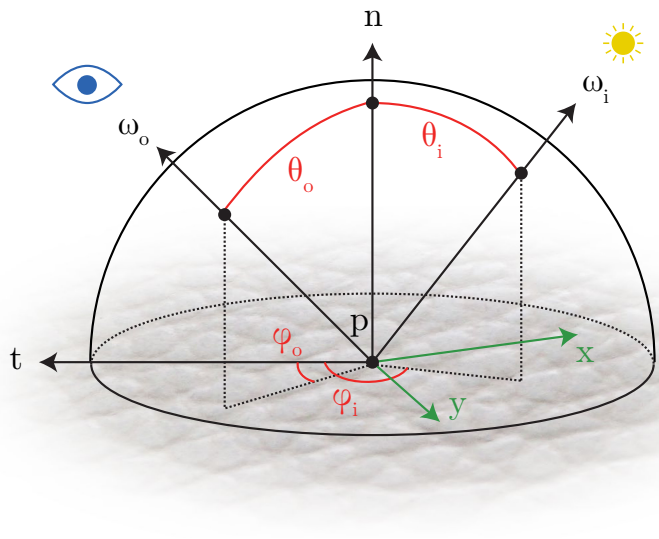


Figure 2.4. BRDF parameterization. The BRDF describes the amount of light reflected from any illumination direction ω_i towards any viewing direction ω_o. This is a function of four angles (highlighted in red). The surface normal n and the tangent t determine the local coordinate system. In a spatially varying BRDF (SVBRDF) the reflectance also varies as a function of the position p on a surface, i.e. as a function of the two spatial coordinates (green). This results in a total of 6 dimensions. For isotropic materials, the BRDF only depends on the *difference* of the azimuth angles, and hence one dimension may be dropped.

coded by the *bidirectional reflectance distribution function*, or BRDF at that surface point [100]:

$$f_r(\omega_i \to \omega_o) = \frac{\mathrm{d}L(\omega_o)}{\mathrm{d}E(\omega_i)} \tag{2.1}$$

Figure 2.4 illustrates the parameterization of this function. Figures 1.3 and 1.4 show some examples of BRDFs for a fixed incidence angle ω_i. Figure 2.5 illustrates several different angular slices of a same BRDF, corresponding to different incidence angles.

Physically valid BRDFs We may compute the proportion of power re-emitted and the power received by the surface from the direction ω_i by integrating over the outgoing angles:

$$\alpha(\omega_i) = \int_\Omega f_r(\omega_i \to \omega_o) \cos\omega_o \, \mathrm{d}\omega_o \tag{2.2}$$

(Recall that irradiance received by a surface depends on the angle of incidence. The cosine of the angle between ω_o and the surface normal in this formula follows from this foreshortening effect.) In order for the BRDF to be physically plausible, it must not reflect out more power than what it

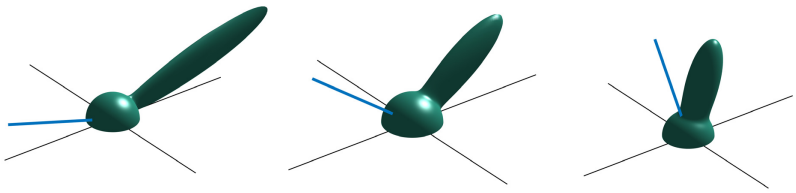


Figure 2.5. A BRDF visualized as a function of ω_o for three different incoming angles ω_i. The full BRDF cannot be visualized at once, due to its high dimensionality.

receives. In other words, $\alpha(\omega_i)$ must not exceed 1 for any ω_i. The value $1 - \alpha(\omega_i)$ represents the absorption probability.

The BRDF must also obey *Helmholtz reciprocity*: it must be symmetric with respect to the incoming and outgoing angles, i.e. $f_r(\omega_i \rightarrow \omega_o) = f_r(\omega_o \rightarrow \omega_i)$.

Reflection equation The BRDF can be used to predict reflections from arbitrary distributions of radiance incident upon a surface point. Because photons arriving from different directions do not interact with one another, their contributions to the distribution of outgoing radiance are independent and simply summed together. Integrating over the contributions of incident differential irradiance from the hemisphere above a point, and accumulating the exitant radiance according to Eq. 2.1, we obtain the *reflection equation*:

$$L_o(\omega_o) \quad = \quad \int_\Omega f_r(\omega_i \rightarrow \omega_o) \cos\omega_i \; L_i(\omega_i) \; \mathrm{d}\omega_i \tag{2.3}$$

The radiance function is divided into the incoming radiance L_i and exitant radiance L_o to model the distribution of light prior to and after the reflection. Essentially, this formula states that (for a fixed surface point p and surface normal n) the radiance towards an outgoing direction is the "continuous sum" of all radiance arriving at the point, weighted by the cosine of the incidence angle and the corresponding BRDF value. Note in particular that this is a *linear* transformation from incident radiance functions to exitant radiance functions:

$$\int_\Omega f_r(\omega_i \rightarrow \omega_o) \cos\omega_i \; [\alpha L_1(\omega_i) + \beta L_2(\omega_i)] \; \mathrm{d}\omega_i =$$

$$\alpha \int_\Omega f_r(\omega_i \rightarrow \omega_o) \cos\omega_i \; L_1(\omega_i) \; \mathrm{d}\omega_i + \beta \int_\Omega f_r(\omega_i \rightarrow \omega_o) \cos\omega_i \; L_2(\omega_i) \; \mathrm{d}\omega_i$$

Essentially, the equation is an infinite-dimensional analogue of a matrix product, where the BRDF supplies the entries of the matrix. In fact, due

to the typical shape of BRDFs, the operation can for some models be seen as a convolution [106]. This explains the intuitive observation that glossy reflections tend to blur (i.e. filter) the reflected image of the surrounding environment.

The equation is linear with respect to the BRDF as well. This is useful when the BRDF is represented as a sum of diffuse and specular sub-BRDFs: each component contributes to the exitant radiance additively.

We will often need to consider reflections at an explicitly specified surface position p and a corresponding normal orientation n. The formula is then:

$$L_o(p, \omega_o) \quad = \quad \int_\Omega f_r(p, \omega_i \to \omega_o)\ \max(0, \omega_i \cdot n)\ L_i(p, \omega_i)\ \mathrm{d}\omega_i \qquad (2.4)$$

Here, the BRDF f_r is taken to be rotated to the local coordinate system around the specified surface normal n (and if needed, an orthogonal surface tangent direction t that fixes the remaining axes).

This equation can be developed into the full rendering equation which models the global light transport in the scene by identifying the incoming and outgoing radiance functions, and adding a radiance emission term to model light sources. This makes the equation recursive and difficult to evaluate. Algorithms such as path tracing [73] and radiosity [56] are used to solve the equation by numerical means. We leave these developments out of our scope, as none of the methods we design involve complex multi-bounce light transport.

2.2.1 Primary reflections from light sources

We will, however, need to model single-bounce reflections for light emitted from different types of sources, as this is the means by which we gather information about the BRDFs of the physical surfaces.

Point light source A point light source is an infinitely concentrated light source, as depicted in Figure 2.1. In Publications II and III, we use it to model a camera flash.

Because a point light is infinitely concentrated (it may be modeled as a *Dirac delta distribution*), the integral in reflection equation (Eq. 2.4) reduces to a point evaluation:

$$L_o(p, \omega_o) \quad = \quad f_r(p, \omega_i \to \omega_o)\ \max(0, \omega_i \cdot n)\ \frac{P}{\|q - p\|_2^2} \qquad (2.5)$$

Here, P is the power of the emitter, ω_i is the unit vector $\frac{q-p}{\|q-p\|}$ from the surface point p towards the light source at q. The squared inverse distance

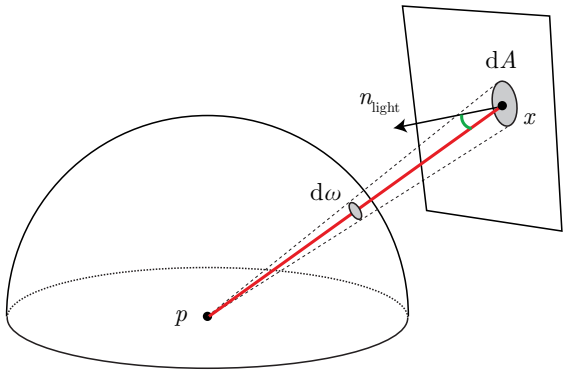


Figure 2.6. Change of variables from the hemisphere to the emitter plane. The geometric terms represent the ratio of the areas of an infinitesimal region $\mathrm{d}\omega$ on the hemisphere, and its projection $\mathrm{d}A$ onto the emitter plane. This boils down to the inverse square of the distance (red), times the cosine of the angle of incidence to the light source (green).

from the surface point to the light source models the distance decay of the irradiance as illustrated in Figure 2.1.

Area light source with spatially varying emission pattern In Publication I, we use a flat-panel monitor to emit a sequence of illumination patterns onto the material surface. We will briefly present a change of coordinate system that is underlies the mathematical formulation of the method. For a more thorough account, see for example Veach [125].

We model the monitor as an area light source with a spatially varying emission power pattern $E(x)$, where x indexes the coordinates on the emitter plane. Recall that the reflection equation Eq. 2.4 is an integral over the upper unit hemisphere Ω above the surface point. In these coordinates, the incoming radiance L_i from the area light source is represented by a rectangle with curved sides. This function is difficult to integrate over directly. Instead, we can perform a change of variables to the planar coordinates of the area light source, as depicted in Figure 2.6. The reflection equation becomes an integral over $\mathbb{R}^2$:

$$L_o(p, \omega_o) \quad = \quad \int_{\mathbb{R}^2} f_r(p, \omega_i(x) \to \omega_o) \ \max(0, \omega_i(x) \cdot n) \ E(x) \ G(x) \ \mathrm{d}x \tag{2.6}$$

Note that the incidence angle ω_i becomes a function of the emitter surface point x. The function $G(x)$ is the standard *Jacobian determinant* that arises in changes of variables in integration. It has a simple geometric interpretation in this case: it is the product of the inverse square distance from the emitter point x to the surface point p, and the cosine of the exitant angle from the emitter, as illustrated in Figure 2.6. They are often

called the *geometric terms*. Intuitively, they model the fact that distant and obliquely viewed emitters *appear* smaller when viewed from p, and hence contribute less. In general, whether this transformation helps us to evaluate the integral depends on context. It plays a key role in Publication I.

2.3 BRDF models

In theory, a BRDF can be almost arbitrary: any function that obeys energy conservation and reciprocity is a physically valid BRDF. Nevertheless, only a small subset of all physically valid BRDFs are encountered in real world. In practical applications BRDFs are chosen from parametric families of functions, which we refer to as *BRDF models*. These families are indexed by a finite-dimensional set of parameters, which is convenient for practical use on finite computers. Secondly, the family is typically chosen as to span as many different plausible real-world BRDFs as possible (and preferably, little else). Sometimes models are specifically designed for specific classes of materials — such as wood [92] or fabrics [113] — to model their unique characteristics. The functions should preferably also have a mathematically convenient form, for example to enable effective importance sampling for Monte Carlo rendering applications.

2.3.1 Tabulated BRDFs

A naive BRDF model is obtained by extensive tabulation of the BRDF values: the four dimensions of a BRDF are subdivided into a fine grid, and the "parameters"[1] of the model are the values of the BRDF at each grid point. The BRDF is evaluated at any pair of angles by interpolating between the supplied parameter values at the nearest grid points.

While highly expressive, this model suffers from various shortcomings. As discussed in Section 1.2.1, fine subdivision of the four-dimensional function potentially results in millions of parameters for a single BRDF. While significant portions of the parameters may be eliminated by restriction to isotropic BRDFs (3-dimensional functions), dropping the mirrored parts due to reciprocity, and concentrating grid points at typical angles with high-frequency content, the representation remains unwieldy

[1] Tabulation is often considered to be a *non-parameric* model as it directly specifies the BRDF values; however, we use this term here in reference to the previous section.

for most practical applications.

Furthermore, the model is *too expressive*: while it can accurately represent any real-world BRDF imaginable, the vast majority of choices for the grid values result in a nonsensical BRDF. Analogous to natural images, one is overwhelmingly unlikely to ever obtain a plausible real-world BRDF by choosing the parameters at random. This makes authoring and editing BRDFs in tabulated format difficult. For similar reasons, tabulated format is of little help in extrapolation of reflectance data from incomplete measurements (which is one of the goals of the methods in this thesis), because a vast space of nonsensical solutions are often compatible with the measurements.

Certain devices based on brute-force point sampling [42, 132] produce BRDF data in a tabulated format. In some applications, this data is used directly for rendering. However, the tabulated data is most commonly fitted to a lower-dimensional model instead.

Carefully chosen changes of angular parameterization [112] often reveal significant redundancy in the BRDF values along the parameter axes. For example, Lawrence et al. [81] exploit this by tabulating BRDF values as one-dimensional slices along chosen axes in a factored representation, significantly reducing the dimensionality of the model while partly retaining it expressiviness.

2.3.2 Low-dimensional parametric models

As discussed in Section 1.2.1, plausible real-world BRDFs constitute a small subset of the space of physically valid BRDFs.

Everyday experience indicates that most BRDFs have some common characteristics. As demonstrated in Figures 1.3 and 1.4, they tend to consist of two roughly distinct parts: the *diffuse component* that looks the same regardless of the viewing angle and gives the surface its base color, and the *specular component* that describes the shiny highlights seen against light sources. This effect is modeled by a peak in the BRDF around the ideal mirror direction. The color, intensity and spread of this peak, or *lobe*, may vary. The tighter the spread of the lobe, the more *glossy* (shiny) the material appears. Some materials, such as brushed metal, exhibit *anisotropy*, which manifests as elongation of the specular highlight. Certain physical properties, such as the index of refraction, also affect the appearance of the material. These high-level observations suggest that most BRDFs are characterized by a few distinct features, to which we

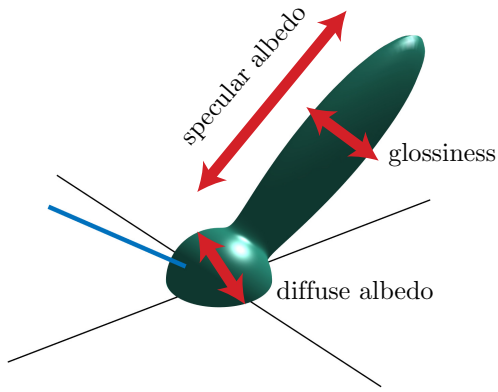


Figure 2.7. A typical parametric model controls the magnitudes of the diffuse and the specular lobes via albedos, and the opening angle of the specular lobe via glossiness. Some parametric models also model effects such as anisotropy [130]. This would correspond to a flattening of the specular lobe along some axis.

could assign numerical parameterization. Indeed, this approach is widely used (*e.g.* [103, 9, 25, 130, 4, 84, 14]).

Phong model One of the earliest parametric BRDF-like models is due to Phong [103]:

$$f_{\text{Phong}}(\omega_i, \omega_o; \rho_d, \rho_s, \alpha) = \frac{1}{\pi}\rho_d + \rho_s \max(0, R(\omega_i) \cdot \omega_o)^\alpha \tag{2.7}$$

where $R(\omega)$ reflects the vector ω with respect to the surface normal vector.

The Phong model is based on the simple empirical considerations above. For a given user-supplied choice of a diffuse albedo (intensity and color) ρ_d, specular albedo ρ_s and glossiness α, it is a function of the incoming and outgoing angle. Note that the dot product reaches value 1 when the incoming angle is at a perfect reflection configuration towards the outgoing angle, and smoothly falls off towards zero elsewhere. The exponentiation by α controls the sharpness of this falloff. Figure 2.7 illustrates the effect of these parameters on a representative slice of the BRDF.

The model was very popular in early days of computer graphics. Visually it has a somewhat plastic-like appearance. However, it is flawed as a BRDF: it does not obey reciprocity, and only satisfies energy conservation within certain combinations of parameters.

In general, energy conservation of a BRDF model can be enforced by normalizing the specular lobe with the integral in Eq. 2.2 and requiring that $\rho_d + \rho_s \leq 1$ (this is also the reason for the factor $1/\pi$ in the diffuse component). For some models, this normalization constant can be computed analytically as a function of the model parameters.

Microfacet models The lack of reciprocity of the Phong model is addressed by the *Blinn-Phong model* [9]:

$$f_{\text{Blinn}}(\omega_i, \omega_o; \rho_d, \rho_s, \alpha) = \frac{1}{\pi}\rho_d + \rho_s \max(0, \mathbf{n} \cdot \mathbf{h})^\alpha \tag{2.8}$$

Here, $\mathbf{h} = \frac{\omega_i + \omega_o}{|\omega_i + \omega_o|}$ is the *halfway vector* between the incoming and outgoing angles. This seemingly odd formulation stems from physically inspired considerations: while the Phong model is based on empirical reasoning and mathematical convenience, the Blinn-Phong model is a simple instance of a physically-inspired idea known as *microfacet theory*.

As illustrated in Figure 1.3, a BRDF is a statistical representation of scattering events occurring below a cut-off scale between the microscopic and macroscopic worlds. Stochastically repeating surface structure that is much smaller than the pixel pitch has no visible spatially varying appearance. Its effects manifest as angular variations only. The idea of microfacet theory is to derive BRDF models based on this principle, without having to explicitly model the specific microgeometry of the surface, or simulate the full light transport within it.

The typical assumption, due to Cook and Torrance [25], is as follows. At microscopic scale, the surface has a very simple BRDF: typically it is assumed to be a mirror (i.e. a perfectly glossy surface, described by a Dirac delta function BRDF), with a physically-based model $F(\omega_i)$ that describes the intensity of this reflection as a function of incidence angle (F denoting *Fresnel*). However, the microscopic shape of the surface is rough. Specifically, it is a stationary height field, giving rise to a *microfacet distribution* $D(h)$ that describes the probability of each surface normal occurring. It is parameterized by the halfway vector: given an incoming and outgoing angle, whatever light is transmitted between them must have been reflected from facets that are exactly at halfway between them (assuming perfectly specular microfacets). Finally, a shadowing and masking function $G(\omega_i, \omega_o)$ encodes the self-shadowing and visibility effects. It is derived from the microgeometry. The full BRDF model is then

$$f_{\text{Cook-Torrance}}(\omega_i, \omega_o) = \frac{1}{\pi}\rho_d + \rho_s \frac{D(h)F(\omega_i)G(\omega_i, \omega_o)}{4(\omega_i \cdot n)(\omega_o \cdot n)} \tag{2.9}$$

This formulation is general: the specifics and the parameters of the D, F and G terms vary across models [25, 130, 4, 84, 14].

In this thesis, we generally use relatively simple models such as the Blinn-Phong, and include the Fresnel effects via the Schlick approximation [114].

Dimensionality of the BRDF space Matusik et al. [93] explored the underlying dimensionality of the BRDF space by using non-linear dimensionality reduction techniques on a wide selection measured tabulated BRDFs. The results indicate that isotropic real-world BRDFs lie on a roughly 10-dimensional non-linear manifold. In other words, 10 real numbers should suffice to uniquely describe any naturally occurring isotropic BRDF. This is in line with the fact that most hand-engineered parametric models use a comparable number of parameters. However, while the model itself is potentially useful for e.g. navigation and editing in the BRDF space, it is somewhat impractical for general rendering purposes.

2.3.3 Spatial variation

The BRDF models discussed in the previous subsection describe the angular variation of the reflectance only. Few real-world objects are covered by a perfectly homogeneous BRDF. Rather, the BRDF typically varies across the surface. Photographic examples of both cases can be seen in Figure 1.5. As seen, spatial variation is a critically important component of visual realism — arguably, in many cases more so than the angular variation.

Generally, any BRDF representation can be extended to handle spatial variation by simply introducing two new spatial dimensions. In the case of tabulated BRDFs, this exacerbates the storage requirements of the already unwieldy representation. Parametric models offer a good balance between expressiveness, computational cost and authoring effort. Indeed, in practical visual effects work, most material authoring efforts are directed towards so-called *texture maps*[2] that describe the spatial variation of the parameters of some chosen general-purpose BRDF model. This thesis is concerned with low-cost automatic capture of precisely this type of data. Figure 1.6 shows an example of this type of an SVBRDF.

Normal maps In addition to variations of the BRDF, most surfaces also exhibit geometric roughness which is spatially large-scale enough as to be visible to the eye, but small-scale enough as to be inconvenient to model as explicit geometry. This variation is very commonly modeled by a cheap and effective approximation to actual geometry, using a *normal map* that describes the orientation of the local surface normal at each point [10].

[2]Not to be confused with the property of *texturedness* in the sense of stationarity, discussed later in this thesis.

At render-time, the BRDF and the cosine in the rendering equation are simply evaluated with the local coordinate system rotated accordingly; no explicit tessellation is needed to create the appearance of the surface bumps. The illusion breaks down with extreme height variations, as the model cannot account for silhouette changes, self-shadowing and masking, or interreflections within the small-scale geometry. Nevertheless, the model is successful for a wide range of materials. Each of the methods in this thesis also captures a normal map of the surface. See the lower-right map in Figure 1.6 for an example.

One technical requirement related to normal maps is *integrability*. Any height field can be converted to a (tangent plane parameterized) normal map by differentiating it along the x- and y-axes. Conversely, a normal map may be considered plausible if there exists a height field that corresponds to it — this is not the case for most vector fields. This property may be enforced without explicit formation of a height field by requiring that the vector field have a vanishing *curl*.

Use in this thesis In summary, each publication in this thesis is concerned with capturing parametric SVBRDF maps, which are essentially multi-channel images in $\mathbb{R}^{w \times h \times c}$, of width w, height h, and c parameters per pixel. The parameters describe the diffuse and specular albedos, glossiness, normal orientation and other model parameters for each pixel. The publications vary in specific details, as each one uses a slightly different BRDF model and parameterization.

While the capture real-world targets and the representation of the solution are assumed to be planar, the maps can be used on arbitrary 3D objects using standard computer graphics techniques such as UV mapping. These questions are largely orthogonal to our problem of capturing the SVBRDF in the first place; we leave them outside our scope.

2.3.4 Generalizations

For completeness, let us briefly review some generalizations to (SV)BRDFs. While many techniques in literature focus on capturing these representations, they are beyond the scope of the publications in this thesis.

The framework reviewed above can also handle refractions in addition to reflections. The BRDF, defined on the upper hemisphere, can be generalized to the full sphere, giving rise to the BSDF or *bidirectional scattering distribution function*.

Strong three-dimensional surface shape variations and translucency may cause visible self-shadowing, interreflections, masking and long-range subsurface scattering effects, which are not well modelled by the combination of low-resolution geometry and BRDF and normal variations.

Bidirectional Texture Functions (BTFs) [27] use the same six-dimensional spatial and angular parameterization as SVBRDFs. However, instead of assigning an independent BRDF to each surface point, an entire patch of the material is lit and photographed from a large number of of angles (often using a spatial gonioreflectometer). At render-time, the patch with the appropriate viewing and lighting angles is placed on the surface. This enables non-local effects such as self-shadowing and masking to become "baked" into the representation. The representation can give highly realistic results for difficult materials. However, BTFs generally require a tabulated representations, and they are much more difficult to author than SVBRDFs. Consequently, they find much less use in practice.

BSSRDF [69] generalizes the BRDF to non-local subsurface scattering by adding extra spatial dimensions that describe the scattering to distant surface locations. This effect is particularly important for human skin, which appears unnaturally hard when rendered with a BRDF.

Considering the radiance in free space, aggregate light transport effects in scenes are sometimes encoded and captured in e.g. reflectance fields [30].

3. Mathematical preliminaries

In this chapter, we review key ideas from inverse problems, optimization and other mathematical tools employed in the methods of this thesis. The treatment is not comprehensive, and we assume the reader is generally familiar with the subject matter; the focus is on the topics relevant to our viewpoint.

3.1 Reflectance capture as an inverse problem

The formulas introduced in the previous chapter also govern light transport in the real world, assuming that the models of geometric optics and surface reflection hold. The general strategy in appearance capture is to arrange the physical sample, sensors and lights in a controlled setup which can be analyzed under this framework. In many cases, the quantities of interest cannot be directly read off the measurements; rather, involved mathematical techniques must be used to estimate them. In this section, we review a general mathematical framework for this purpose [121].

One can generally view the problem of recovering unknown scene information from indirect observations as an *inverse problem*. In this viewpoint, the model of light transport describes a *forward model* F that can be used to predict an image y, given the scene parameters x. Conceptually, rendering an image of a known scene means simply evaluating $y = F(x)$. The forward model is the predictive model discussed in Section 1.3

An inverse problem turns this problem on its head: given an *observation* (or *measurement*) y, for example a photo, an inverse problem asks what are the unknown parameters x that explain it, assuming that $y = F(x)$. In other words, we are looking to solve the equation $y = F(x)$ with respect to x instead of y.

More generally, we typically have K observations y_i. The scene parameters can be divided into two categories: the unknown parameters $x \in \mathbb{R}^L$, and auxiliary parameters q_i which we control during the measurement process to obtain a variety of different measurements. We then seek a solution x for the system of equations

$$\begin{cases} F(x; q_1) = y_1 \\ F(x; q_2) = y_2 \\ F(x; q_3) = y_3 \\ \dots \\ F(x; q_K) = y_K \end{cases} \tag{3.1}$$

Concretely, x might be a vector of unknown SVBRDF parameters, y_i might be observed intensities of pixels in various photographs (one for each pixel, potentially totaling millions of observations), and q_i might describe the light and camera position used in each measurement.

Overdetermined and underdetermined problems Solving the inverse problem is typically a much more difficult task than the (relatively) straightforward evaluation of the corresponding forward model. While the forward operator F may also be difficult to evaluate in practice (for example, it might represent a full Monte Carlo light transport simulation), it is nevertheless a unique and well-defined procedure. In contrast, there is no universal way of obtaining an inverse solution given a forward operator. Indeed, an exact solution often does not even exist: when there are more measurements than unknowns in Eq. 3.1 (i.e. it is *overdetermined*), there is generally no x that satisfies each of the equations simultaneously. Instead, one typically seeks a solution that satisfies Eq. 3.1 as closely as possible, according to some error metric:

$$\operatorname*{argmin}_x \|F(x; q_1) - y_1\| + \|F(x; q_2) - y_2\| + \dots + \|F(x; q_K) - y_K\| \tag{3.2}$$

When there are fewer observations than unknowns, the problem is *underdetermined*. This leads to another difficulty: Eq. 3.1 can often be satisfied exactly by an infinite number of different solutions. Typically the vast majority of them are mathematical artifacts instead of meaningful solutions to the underlying real-world problem. This type of a problem is *ill-posed*. To eliminate these excessive degrees of freedom, one typically introduces extra constraints, such as favoring the solution with the smallest norm. These "tie-breakers" can be interpreted as specifications of *a priori* beliefs, or *priors*, about the properties of favorable solutions.

Ill-conditioned problems and regularization Complex real-world problems often have characteristics of both: it is easy to collect an overdetermined number of measurements, but they might nevertheless be uninformative or ambiguous about some of the unknowns. For example, to uniquely identify shininess properties of a piece of surface, it must be observed in a suitable lighting angle as to cause a specular highlight (or a lack thereof). If measurement under such conditions is not provided, a solution to Eq. 3.2 may choose arbitrary values for the related parameters, as they have no effect on the error. In these cases the solution is sometimes extremely sensitive to distortions in the measurements: meaningless features in the data, such as noise, can often be reproduced by setting the "underdetermined" variables to extremely unnatural values. In this case the problem is said to be *ill-conditioned*.

Again, when the data contains insufficient information about the underlying reality, priors can be used to *regularize* the problem by explicitly specifying desirable properties a solution should have. The typical procedure involves simply adding a terms onto Eq. 3.2, with the aim of penalizing undesirable values of the unknowns independently of the data:

$$\underset{x}{\operatorname{argmin}} \sum_i \|F(x; q_i) - y_i\| + P(x) \tag{3.3}$$

For example, we might use a prior to penalize very large values of the unknowns, hence discouraging the solution from overfitting to noise. However, careful balancing is required: exceedingly strong priors tend to bias the solution, overriding subtle features present in the data. Ideally the priors should let the data decide when it is unambigous, but step in where the data is insufficient. We will discuss priors in more depth in Section 3.1.5.

Solving inverse problems The practical means for solving an inverse problem depend on its mathematical characteristics. Many interesting phenomena are modeled by functions that possess significant structure, which can be exploited by applying techniques aimed towards restricted classes of problems. In particular, many problems are linear or can be closely approximated as such. In this case a meaningful inverse problem often reduces to a matrix-vector equation that can be solved by standard tools in numerical linear algebra.

However, this is not always the case. A general methodology for solving inverse problems is *optimization*, where an initial guess of a solution is iteratively improved in a principled manner. These methods are dis-

cussed in more detail in Section 3.2. We briefly note an additional layer of difficulty introduced by optimization: in many cases the *globally optimal* solution to the inverse problem is difficult to find, as an optimizer may become stuck into locally optimal solutions that cannot be improved by any small changes. The existence of these spurious minima is a property of the objective function, and should be considered in the design of the problem.

3.1.1 An example

Let us sketch a concrete example of a canonical inverse problem in material appearance capture. Let us assume that we have photographed a material from multiple distant viewing angles under distant point light illumination, as shown in Figure 1.7. This sort of data might be acquired for example by using a spatial gonioreflectometer [96, 27]. We have performed thorough calibration, and know the viewing and lighting angles γ_j and δ_j, and the irradiances R_j of the light precisely for each of the J photographs indexed by j. Denote by $y_j \in \mathbb{R}^I$ a vectorized representation of the input photograph consisting of $I = \text{width} \times \text{height}$ observations (assuming monochromatic measurements for simplicity of exposition).

We also assume that we have *rectified* the input photographs into a common coordinate system by undoing any perspective distortions, as shown in Figure 3.1. Hence, the same pixel corresponds to the same surface point in each photo, and consequently, the same BRDF and surface normal.

We are looking to fit an SVBRDF using the Blinn-Phong BRDF model (Section 2.3.2) with normal variation to this data. The SVBRDF is described by a vector $x \in \mathbb{R}^{I \times 6}$, where the 6 parameters per pixel correspond to the diffuse albedo $\rho_d \in \mathbb{R}$, specular albedo $\rho_s \in \mathbb{R}$, the glossiness $\alpha \in \mathbb{R}$, and the normal orientation $n \in \mathbb{R}^3$ of each pixel. In other words, the goal is to solve for a set of parameter maps as in Figure 1.6.

Using the equation for reflections from point lights (Eq. 2.5) and the formula for the Blinn-Phong model (Eq. 2.8), we arrive at a forward model where each pixel is rendered using the formula:

$$f(\rho_d, \rho_s, \alpha, n; \gamma, \delta, R) = R \max(0, n \cdot \delta) \left[\rho_d + \rho_s \max\left(0, n \cdot \frac{\gamma + \delta}{|\gamma + \delta|}\right)^{\alpha} \right] \quad (3.4)$$

The full forward model $F(x; \gamma, \delta, R) : \mathbb{R}^{I \times 6} \mapsto \mathbb{R}^I$ simply evaluates this model at every pixel using the local values of the variables. In other words, it renders the SVBRDF into an image under the specified lighting and viewing conditions.

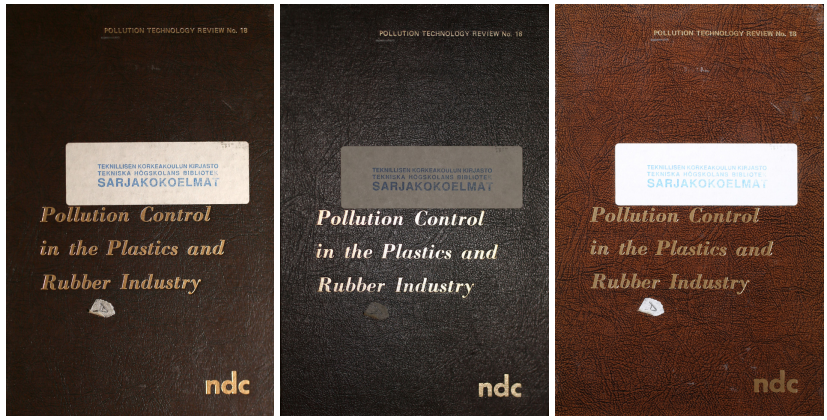


Figure 3.1. Examples of images y_j from the rectified input data. The perspective distortions in the input photographs (Figure 1.7) have been cancelled by a homography transformation [58] using known (calibrated) information about the geometric configuration of the camera and object. Each pixel now represents a fixed point on the surface across the set of the rectified images. Each image is associated with a viewing angle γ_j and a lighting angle δ_j.

The inverse problem, then, might be:

$$\underset{x}{\operatorname{argmin}} \sum_j \|y_j - F(x; \gamma_j, \delta_j, R_j)\|_2^2 \tag{3.5}$$

In other words, we are looking to find an SVBRDF F that, when rendered, reproduces the input photographs as accurately as possible, as measured by the pointwise squared difference of pixel values. We might also add a *smoothness prior*, for example:

$$\underset{x}{\operatorname{argmin}} \sum_j \|y_j - F(x; \gamma_j, \delta_j, R_j)\|_2^2 + \lambda \|\nabla x\|_2^2 \tag{3.6}$$

Here, $\nabla : \mathbb{R}^{I \times 6} \mapsto \mathbb{R}^{2I \times 6}$ is a linear operator that evaluates finite differences of the parameter maps along x- and y-directions. λ is a weighting that determines the relative importance of the data fit and the prior. The idea is to penalize differences between the values of the neighboring pixels. This discourages spurious oscillations and abrupt jumps in the solution maps.

In practice, we would also need some mechanism for constraining the surface normal n to be a unit vector. One possibility is to express it in a *tangent-plane* parameterization $\tilde{n} \in \mathbb{R}^2$, from which the unit vector can be recovered when needed as $n = [\tilde{n}_x, \tilde{n}_y, 1]^{\mathrm{T}} / (\tilde{n}_x^2 + \tilde{n}_y^2 + 1)$. Any choice of values for $\tilde{n}$ will then result in a valid unit vector.

We might also introduce a prior to penalize the curl of this vector field in order to enforce the integrability of the normal map (this approach is

used in Publications I and II). Alternatively, we may internally represent the entire normal map as a height field, and differentiate it to obtain a tangent-plane parameterized normal map that is by definition integrable. This approach is used in Publication III.

3.1.2 Probabilistic viewpoint

Besides the forward model, the typical specification of an inverse problem also includes the error metric to be used in Eq. 3.2, and as briefly alluded, the specification of prior beliefs. Thus far, we have justified the formulas heuristically, as "penalizing" the solution according to how much its predictions differ from the observations, or from preferred values. This viewpoint gives little insight into the justification, meaning and practical consequences of different choices.

Principled probabilistic considerations shed some light into these issues. The sum of independent measurement-wise deviations in Eq. 3.2 is typical; we will see that it naturally arises from a Maximum Likelihood estimation perspective. Augmenting this framework with Bayesian considerations, we arrive at Maximum A Posteriori estimation, which proposes a natural justification for specifying additional prior terms. [59, 87, 121, 8]

3.1.3 Maximum likelihood estimation

Consider the problem in Eq. 3.1. Let us assume that the forward model F and the parameters q_i perfectly model the physical reality in which the measurements were made. Let x^* denote the true values of the unknown parameters x we are seeking to recover. Further, let $y_i^* := F(x^*; q_i)$ be the values of the measurements predicted by this model, assuming perfect knowledge.

Assume now that the measurements y_i we possess are corrupted by noise: $y_i = y_i^* + n_i$, where n_i are mutually independent random variables with known distributions. In particular, let us for now assume that $n_i \sim \mathcal{N}(\cdot; 0, \sigma^2)$, i.e. each observation is corrupted by a zero-mean Gaussian random variate of known standard deviation σ.

Given these assumptions, the joint probability density of the observa-

tions y, given the variables x, is

$$\begin{aligned} p(y|x) &= \mathcal{N}(y|y^*, \sigma^2\mathbf{I}) && (3.7)\\ &= \prod_i \mathcal{N}(y_i|y_i^*, \sigma^2) && (3.8)\\ &= \prod_i \mathcal{N}(y_i|F(x,q_i), \sigma^2) && (3.9)\\ &= C\prod_i \exp\left\{-\frac{(y_i - F(x,q_i))^2}{2\sigma^2}\right\} && (3.10)\\ &= C\exp\left\{-\frac{1}{2\sigma^2}\sum_i (y_i - F(x,q_i))^2\right\} && (3.11) \end{aligned}$$

where C is a normalization constant that only depends on σ.

The probability density is a function of the observations y, given the unknowns x. We can also take an alternative perspective without changing anything about the function itself. Interpreted as a function of the parameters x, given the observations y, the function is called *likelihood*. This appears to be useful, as it is the observations y that are known to us. Indeed, *maximum likelihood* estimation is a commonly used method for finding the most likely x to explain y. It simply calls for finding an x^{ML} for which the likelihood is maximized:

$$\begin{aligned} x^{\mathrm{ML}} &= \operatorname*{argmax}_x p(y|x) && (3.12)\\ &= \operatorname*{argmax}_x C\exp\left\{-\frac{1}{2\sigma^2}\sum_i (y_i - F(x,q_i))^2\right\} && (3.13) \end{aligned}$$

Clearly, the maximizer is not affected by the constant factor C; hence it may be dropped. Similarly, the maximizer remains the same when the optimized function is mapped by a pointwise strictly monotonous function — specifically, the logarithm:

$$\begin{aligned} x^{\mathrm{ML}} &= \operatorname*{argmax}_x C\exp\left\{-\frac{1}{2\sigma^2}\sum_i (y_i - F(x,q_i))^2\right\} && (3.14)\\ &= \operatorname*{argmax}_x \log\exp\left\{-\frac{1}{2\sigma^2}\sum_i (y_i - F(x,q_i))^2\right\} && (3.15)\\ &= \operatorname*{argmax}_x -\frac{1}{2\sigma^2}\sum_i (y_i - F(x,q_i))^2 && (3.16)\\ &= \operatorname*{argmin}_x \frac{1}{2\sigma^2}\sum_i (y_i - F(x,q_i))^2 && (3.17)\\ & && (3.18) \end{aligned}$$

Observe that this final expression (often referred to as the *negative log-likelihood*) is of the form of Eq. 3.2; the above derivation explicitly identifies the underlying assumptions. While the initial assumptions are often

not exactly met in realistic settings, the probabilistic viewpoint does often offer a useful guideline in designing inverse problems. Other types of minimization tasks can also be derived from different assumptions — for example, weightings according to estimated reliability can be introduced in a principled manner by using a different noise variance for each measurement.

Error metrics Of specific interest is the emergence of the squared difference metric. It is inherited from the quadratic term inside the exponential in the density function of the normal distribution. Not coincidentally, minimization problems involving squared deviations enjoy a variety of special mathematical properties, making these problems generally easier to solve than those involving other metrics.

Certain alternative metrics enjoy some interesting properties, at the expense of more difficult optimization problems. In particular, squared error is known to be sensitive to outliers in data: a large corruption in a single data point can throw the entire solution off, as the squaring unduly magnifies their importance. A popular and effective remedy for this is the use of (non-squared) absolute value of the difference (the ℓ_1-*metric*). Indeed, this metric also enjoys surprising properties and applications related to robust estimation and the notion of *sparsity* [110, 37, 36, 19]. Working backwards from the result of above derivations, we can interpret the use of this metric as an implicit assumption that the errors are *Laplace distributed*, as this distribution has a density function of shape $\exp(-|x|)$. In Publications I and II we make use of a smoothed (i.e. differentiable) version of the ℓ_1 error metric, called the Huber loss [67].

3.1.4 Bayesian viewpoint

Viewing the parameter recovery problem from a *Bayesian* perspective, and leaving the associated philosophical implications out of our scope, we recover what is in practice an extension of the Maximum Likelihood estimation framework. Let us briefly review the underlying ideas. Figure 3.2 shows a visual example of the procedure.

The Bayesian viewpoint is based on quantifying uncertainty about the state of the world. Let $p(x, y)$ be a joint probability density over all possible values of the unknown parameters x and the measurements y. From the standpoint of p, there is nothing special about either group of variables. We make a distinction between them to highlight the assumed

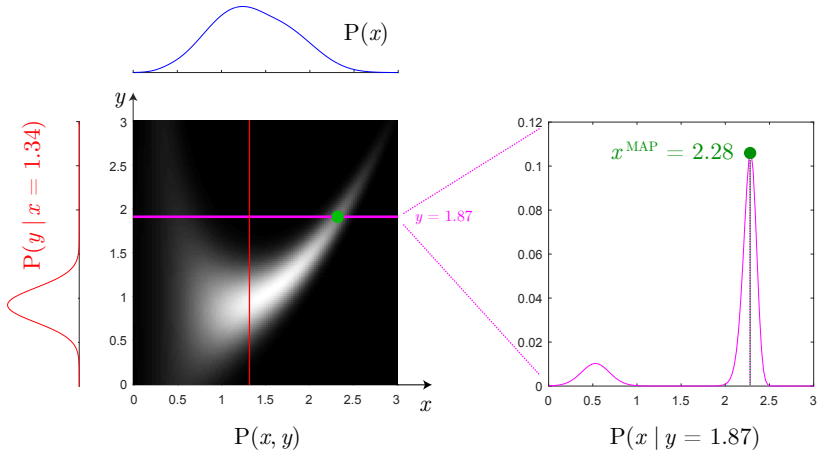


Figure 3.2. An example of Bayesian inference for one unknown variable x and one observation y. We specify the prior $P(x)$ that describes our *a priori* beliefs about the probabilities of the values of the unknown variables, and the conditionals $P(y|x)$ that for each possible value of x describe how the measurements are distributed (i.e. the mapping from the unknowns to the observations, plus the measurement noise; this is analogous to the likelihood function). Here, we show an example of a conditional at $x = 1.34$. Together, these distributions define the joint probability distribution $P(x, y)$. Notice how the prior is simply the joint probability density marginalized (integrated) over y, and the likelihoods are (normalized) vertical slices of this density. Let us now assume that we perform a measurement, and observe the value $y = 1.87$. The *posterior* distribution is the conditional $P(x|y = 1.87)$, and corresponds to an individual horizontal line in the joint probability density. It is visualized on the right. We find that this (multimodal) density peaks at $x = 2.28$: this is the Maximum A Posteriori estimate x^{MAP}.

causal structure (*i.e.* x having some underlying values, and subsequently giving rise to y). The Bayesian approach is to implicitly construct p assuming this causality and supplying the relevant probability distributions.

The marginal $p(x) = \int p(x, y)\mathrm{d}y$ is of special interest: it is the probability of the unknowns having a given values, when nothing is known about y. In the Bayesian interpretation, this marginal is known as the *prior distribution* of the unknowns — as in, prior to any measurements. The Bayesian approach calls for explicit specification of $p(x)$ (*without* explicitly specifying the integrand in the marginal); it encodes our *a priori* beliefs about the plausible values of x.

The main object of interest is the conditional probability distribution $p(x|y)$ of the unknowns, once particular values of the measurements have been made. The measurements are seen as narrowing down our uncertainty. This distribution is known as the *posterior* distribution: the state of our knowledge, *post*-measurement. From basic laws of probability, we

have

$$p(x|y) = \frac{p(x,y)}{p(y)} \tag{3.19}$$

Multiplying by $p(y)$ and applying the resulting formula in two different ways, we find:

$$p(x|y)p(y) = p(x,y) = p(y|x)p(x) \tag{3.20}$$

Rearranging, we arrive at the *Bayes' rule* for computing the posterior:

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \tag{3.21}$$

Here, $p(x)$ is the prior distribution discussed above, and $p(y|x)$ can be interpreted as describing the distribution of observations, given the values of the unknowns. Notice that this is precisely the same function as in Maximum Likelihood estimation. This distribution, too, must be specified. $p(y)$ also has a specific, less commonly used interpretation; we leave it outside our scope, as we find below that it vanishes in practical computations of our interest.

Maximum a posteriori estimation Despite of thoroughly encoding our state of knowledge about the world (assuming a model encoded by $p(x)$ and $p(y|x)$), the full posterior distribution is often difficult to interpret in meaningful ways. *Maximum a posteriori* estimation calls for finding the value of x that maximizes the posterior distribution, hence being the "most probable" explanation in some sense:

$$x^{\text{MAP}} = \operatorname*{argmax}_x \frac{p(y|x)p(x)}{p(y)} \tag{3.22}$$

In finding this minimum, we may apply the same trick as in ML estimation: the constant (with respect to x) factor $\frac{1}{p(y)}$ may be dropped, and the expression is transformed by the negative logarithm.

$$x^{\text{MAP}} = \operatorname*{argmin}_x \; -\log p(y|x) - \log p(x) \tag{3.23}$$

Observe that this minimization task is precisely the same as in ML estimation, but with an added prior term $-\log[p(x)]$. ML estimation can be seen as MAP estimation with a perfectly uninformative prior: in this case the prior term becomes constant and may be dropped from the minimization.

In particular, with the same assumptions for $p(y|x)$ as in Section 3.1.3, and $p(x)$ taken as a normal distribution with mean μ_p and covariance Σ_p, we find

$$x^{\text{MAP}} = \operatorname*{argmin}_x \frac{1}{2\sigma^2} \sum_i (y_i - F(x,q_i))^2 + \frac{1}{2}(x-\mu_p)^{\mathrm{T}} \Sigma_p^{-1} (x-\mu_p) \tag{3.24}$$

In summary, MAP estimation justifies the addition of data-independent penalty terms in Eq. 3.3. While the publications in this thesis generally state the prior terms without explicit reference to their probabilistic interpretations, the viewpoint provides valuable insights and guidance for design decisions.

3.1.5 Priors

Publications I and II make use of two types of priors in particular: pointwise specifications of plausible ranges of values, and *smoothness constraints* that bind the solutions at neighboring points together. Publication III introduces a special type of a prior that enforces *stationarity* of image statistics, which is to our knowledge novel. We discuss this prior in more detail in Section 6.3.3.

Pointwise priors We often have a some idea about what kind of values the solution variables should take. Even a very loose preference is often useful to specify, as highly ill-conditioned problems may lead to solutions with numerical values several orders of magnitude outside any reasonable range.

For example, the surface normals in typical surfaces we capture should vary within a few dozen degrees. Hence, it makes sense to favor surface normals with the x- and y-components close to zero. However, if the data overwhelmingly supports an extreme normal deviation, it should be allowed. This type of behavior is often sufficiently well enforced by a quadratic prior term, i.e. an assumption that the values are normally distributed with some specified mean and variance. Sometimes other norms, such as absolute difference, or a Huber norm, might be favored instead.

Smoothness priors Most real-world signals with a natural spatial arrangement (such as a pixel grid) exhibit continuity: neighboring values are not fully independent from each other. For example, a pixel in a natural image is highly likely to have a similar value with its neighbor. This also applies to parametric SVBRDF maps, which are the main object of interest in this thesis.

It often makes sense to explicitly enforce this behavior by priors when solving ill-conditioned estimation problems concerning image-like data. Solving the unknown parameters independently at each pixel may result in noise, as the rapidly varying measurement noise becomes amplified. Furthermore, extreme noise or structured spatial artifacts may

occur when the solution is not unique: neighboring pixels may randomly choose completely different but apparently equally good solutions. In our SVBRDF recovery problems, this sort of an ambiguity is particularly evident in the diffuse-specular separation, as the observed data can sometimes be explained by either component.

Priors explicitly favoring similar values for neighboring pixels significantly help with these problems by enabling a rough form of information sharing within neighborhoods: nearby pixels must jointly negotiate a solution that satisfies both their individual data and prior constraints, as well as the requirement of spatial consistency.

A basic smoothness prior is obtained by assuming a joint normal distribution between the neighboring variables. This is implemented by introducing two prior terms of the form of Eq. 3.24, with $\Sigma_p = D_x$ and $\Sigma_p = D_y$, respectively, and $\mu_p = 0$. Here, D_x and D_y are finite difference matrices. This encodes a preference that the difference between two neighboring pixels should be close to zero. The example presented in Section 3.1.1 used a prior of this kind.

Unsurprisingly, excessively strong smoothness priors lead to excessively blurry solutions lacking fine spatial detail. Natural images are in fact poorly modeled by the assumption of normally distributed neighbor differences, i.e. the quadratic norm. They are characterized by smooth regions interspersed with abrupt edges. The quadratic difference imposes a very large penalty on discontinuities, and strongly discourages their formation. A better model is obtained by the use of absolute ℓ_1 differences (or the smoothed version, Huber loss), which retain some mathematically convenient properties of the quadratic norm (in particular convexity) but do not incur the amplified penalty on large differences [110, 19].

Interpretation According to the probabilistic interpretation, priors such as above can be seen as defining a probability distribution over SVBRDFs. The pointwise priors alone assume independence between pixels: hence, they implicitly encode a "belief" that the space of plausible materials is the space of white noises of certain expectation and variance. While reasonable SVBRDFs fit this description in a statistical sense, the assumption is not very restrictive. The smoothness prior with a squared norm adds dependencies between neighbors (and by transitivity, global dependencies). The space of SVBRDFs assumed by this prior is that of Brownian noise, i.e. a generalized random walk. While more reasonable than white noise, this model is still very loose, and as discussed, does not con-

tain signals with sharp edges.

While often very useful, these priors are nevertheless very coarse characterizations of the space of naturally occurring SVBRDFs. They only manage to rule out the most grossly implausible solutions. While more advanced priors have been presented in literature — for instance, Barron and Malik [5] manage to recover surprisingly plausible geometry and reflectance data from a single photograph using a strongly prior-driven model — the problem is far from solved. We alluded to this issue in the introductory section, and we will discuss it further in the conclusions.

3.2 Optimization

Numerical optimization is a widely used methodology for solving inverse problems. Optimization algorithms aim to find minima of scalar-valued functions by starting from an initial guess, and iteratively improving the solution. The function is seen as a black box that can be queried at individual points for its value and derivatives. More concretely, they solve problems of type

$$\underset{x}{\operatorname{argmin}}\ F(x) \tag{3.25}$$

by producing a sequence of improved iterates $x_1, x_2, ...$, until further progress cannot be made. The scalar-valued function $F : \mathbb{R}^L \mapsto \mathbb{R}$ is called the *objective* or *loss* function. It typically measures the "badness" of the solution: high values indicate significant deviation from the data or the prior assumptions.

Notice that the typical inverse problem in Eq. 3.2 is precisely of this form. In restricted cases a solution can be obtained by more direct means — for example, if the minimization task reduces to a linear least squares problem, it may be solved using a set of standard tools in linear algebra [121, 53]. [1] However, problems involving complicated non-linear functions, such as parametric BRDF models, rarely yield to such approaches without significant simplifying assumptions and a corresponding reduction in solution quality.

The methods introduced in this thesis make heavy use of optimization. We will briefly review some particularly relevant algorithms and their common background. A thorough discussion of these and many other op-

[1] Internally, many linear algebra solvers also perform optimization in a special context. However, the problem types and solution methods are highly standardized and well understood.

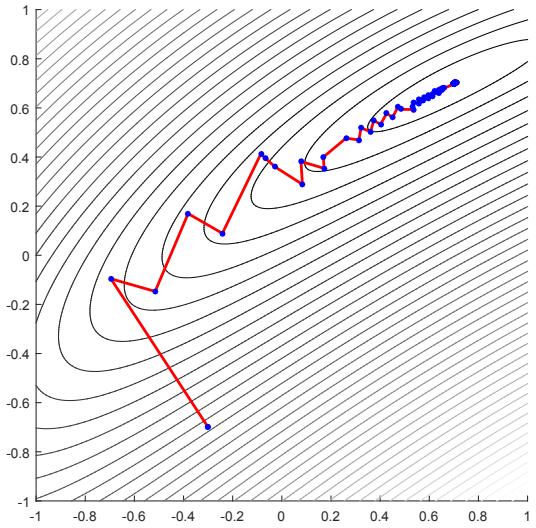


Figure 3.3. Progress of gradient descent optimization in a function of two variables, visualized as a contour plot. The initial guess is $[-0.3, -0.7]$. From there, the optimization proceeds by stepping along the direction of the gradient at the current iterate. Notice that the gradients are always perpendicular to the contour lines of the function. In an elongated valley, they rarely point towards the minimum, which leads to slow zig-zag convergence.

timization methods is presented by Nocedal and Wright [101].

3.2.1 Gradient descent

Most commonly used optimization strategies are based on using derivative information. The idea is natural: to find the bottom of a valley surrounded by hills, it makes sense to always walk towards a descent direction. The direction of the *steepest descent* is given by the gradient of the height field (i.e. the objective function). This suggests the *gradient descent algorithm*, where an initial guess x_0 and a differentiable objective function F are supplied, and the iterates are given sequentially by

$$x_{t+1} = x_t - \gamma \nabla F(x_t). \tag{3.26}$$

γ is a step length, which must be small enough to ensure that the step results in an actual reduction in the objective function, but large enough to give meaningful progress. In practice, one often needs to use adaptive step length choice with back-tracking line search. The sequence then has the property that the values $F(x_t)$ decrease monotonically with t. The iteration is terminated once the improvements become insignificant, by some specified threshold.

3.2.2 Second-order methods

Gradient descent is inefficient. Somewhat unintuitively at first glance, descending along the steepest direction usually does not lead to the shortest path to the bottom of a valley. This is clearly seen in Figure 3.3. *Second-order methods* use second-order partial derivatives of the objective function to determine improved step directions.

Gradient descent as a first-order method. Let us first view gradient descent in this context. Gradient descent can be seen as a procedure where the objective function is replaced by its *first-order* approximation at the current iterate $p := x_t$, which is then minimized. The idea is that minimizing this surrogate function is easier than minimizing the original. We start from the full Taylor expansion of the objective function:

$$F(x) = F(p) + \nabla F(p)^{\mathrm{T}}(x-p) + \frac{1}{2}(x-p)^{\mathrm{T}}\nabla^2 F(p)(x-p) + \ldots \tag{3.27}$$

Here, $\nabla^2 F(p) \in \mathbb{R}^{L\times L}$ is the symmetric Hessian matrix of second-order partial derivatives of F at p. We obtain the first-order approximation by dropping the higher-order terms:

$$\tilde{F}(x) = F(p) + \nabla F(p)^{\mathrm{T}}(x-p) \tag{3.28}$$

Minimizing this raw surrogate function with respect to x is fruitless, however: no minimum exists as the function obtains arbitrarily large negative values along the direction $-\nabla F(p)$. The magnitude of the jump can be controlled by introducing a *trust region* around p, for example via a quadratic form:

$$\tilde{\tilde{F}}(x) = F(p) + \nabla F(p)^{\mathrm{T}}(x-p) + \frac{1}{2\gamma}(x-p)^{\mathrm{T}}(x-p) \tag{3.29}$$

The idea is that the extra term "penalizes" the approximation at far-away distances from p (as controlled by the scaling parameter γ). The term becomes insignificant near p, where $\tilde{F}$ is believed to be an accurate approximation to F. Minimizing this function with respect to x by finding the critical point (which is necessarily a minimum) yields the gradient descent step in Eq. 3.26.

Newton's method. Second-order methods are based on the same reasoning, with the exception that also the second-order term in Eq. 3.27 is retained:

$$\hat{F}(x) = F(p) + \nabla F(p)^{\mathrm{T}}(x-p) + \frac{1}{2}(x-p)^{\mathrm{T}}\nabla^2 F(p)(x-p) \tag{3.30}$$

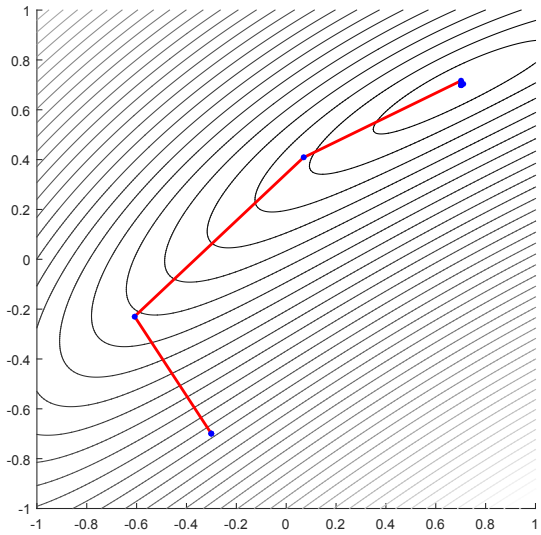


Figure 3.4. Visualization of the progress of Newton's method. Notice the significantly improved step directions compared to gradient descent (Figure 3.3).

Jumping to the critical point of this second-order surrogate function yields *Newton's method* (denoting $x_t := p$):

$$x_{t+1} = x_t - (\nabla^2 F(x_t))^{-1} \nabla F(x_t) \tag{3.31}$$

If the Hessian matrix is positive definite, this critical point is the bottom of a "bowl" defined by the quadratic form. Otherwise the step is meaningless. Furthermore, like in gradient descent, excessively long steps may result in increase of the objective value. This problem can similarly be solved by using a quadratic trust region with an adaptive size. With a large enough weight, it is guaranteed to raise the eigenvalues of the quadratic form enough as to make it positive definite. We will review this approach in more detail below. Figure 3.4 shows an example of the convergence of Newton's method; notice the significant improvement over gradient descent.

Gauss-Newton and Levenberg-Marquardt methods. *Least-squares optimization problems* are characterized by an objective function that is a sum of squared *residuals* provided by a function $r : \mathbb{R}^L \mapsto \mathbb{R}^K$:

$$F(x) = \frac{1}{2}\left[r_1(x)^2 + r_2(x)^2 + ... + r_N(x)^2\right] \tag{3.32}$$

$$= \frac{1}{2}\|r(x)\|_2^2 \tag{3.33}$$

$$= \frac{1}{2} r(x)^{\mathrm{T}} r(x) \tag{3.34}$$

Many interesting problems fall into this category: for example, Eq. 3.2 is a least-squares problem when the error metric is the squared ℓ_2-norm $\|\cdot\|_2^2$. As discussed in Section 3.1.2, this metric arises naturally in the maximum likelihood and maximum a posteriori estimation when it is assumed that the measurement noise is normally distributed. Similarly, Bayesian prior beliefs expressed as normal distributions result in squared terms.

Functions with this structure admit to a highly useful approximation to their Hessian matrices. Let $J(x) \in \mathbb{R}^{K \times L}$ denote the full Jacobian matrix of the function r at x. Then, $\nabla^2 F(x) \approx J(x)^{\mathrm{T}} J(x)$, and $\nabla F(x) = J(x)^{\mathrm{T}} r(x)$. This approximation becomes increasingly accurate as x approaches a minimum with the value 0 (which is not always the case; nevertheless, the approximation is often efficient). Computing the Jacobian is often significantly much easier than computing the Hessian. Using this matrix in place of the Hessian in Newton's method yields the *Gauss-Newton method*:

$$x_{t+1} = x_t - (J(x_t)^{\mathrm{T}} J(x_t))^{-1} (J(x_t)^{\mathrm{T}} r(x_t)) \tag{3.35}$$

There is no guarantee that this step results in a reduced objective value. Adding a soft quadratic trust region of weight λ as in Eq. 3.29 and rearranging, we arrive at the Levenberg-Marquardt step:

$$x_{t+1} = x_t - (J(x_t)^{\mathrm{T}} J(x_t) + \lambda I)^{-1} (J(x_t)^{\mathrm{T}} r(x_t)) \tag{3.36}$$

A refinement of this approach uses a trust region scaled according to the matrix diagonal $D := \operatorname{diag} J^{\mathrm{T}} J$ to roughly compensate for the magnitude of variation of the variables:

$$x_{t+1} = x_t - (J(x_t)^{\mathrm{T}} J(x_t) + \lambda D(x_t))^{-1} (J(x_t)^{\mathrm{T}} r(x_t)) \tag{3.37}$$

The trust region size parameter λ is adaptively adjusted according to the optimization progress: it is increased whenever a step fails to reduce the objective value, and decreased when a step succeeds. The step is guaranteed to succeed for high enough values of λ, unless the current iterate is already a local minimum. This approach has proven very successful for a wide range of problems; we use it in Publications I and II.

Structure of the Hessian The quadratic problem $H^{-1} g$ in second-order methods is typically solved directly by Cholesky decomposition, or by preconditioned Conjugate Gradient method [53]. The structure of the Hessian (or $J^{\mathrm{T}} J$) determines how difficult this sub-problem is. Because the

number of entries in these matrices is quadratic with respect to the number of variables in the problem, the matrix must in practice be *sparse* for large-scale problems. Because any direct interaction between two variables (i.e. occurrence in the same equation) creates an entry to the corresponding position in the Hessian, sparse connectivity should be preferred. Fortunately, a wide range of problems relating to 2D image topologies are of this type: often the equations concern local variables at pixels, or relations between neighboring pixels. The former induces a sparse block diagonal structure on the Hessian, and the latter a sparse set of "bands".

Sparse matrices often contain only a few non-zero elements per row, despite having dimensions in the millions. Many linear algebra software packages and libraries provide support for construction, storage and computations with sparse matrices.

Quasi-Newton methods In many problems, the computation of a full Hessian matrix, or even the $J^{\mathrm{T}}J$ approximation (when applicable), is practically impossible due to memory and performance constraints. In high-dimensional problems with dense variable interactions, the number of nonzero entries in these matrices might be orders of magnitude larger than the amount of available memory. It is common to fall back to first-order methods such as gradient descent in such cases. However, a hybrid approach has proven effective for many problems: *quasi-Newton methods* accumulate first-order gradient information to compute estimates of the (inverse) Hessian. The per-iteration input to these methods at each step is the same as in gradient descent. Quasi-Newton methods often achieve a similar asymptotic convergence rate as the regular Newton methods.

Publication III takes advantage of the L-BFGS method [102] to drive an extremely complex and unstructured optimization task. The method is based on carefully chosen low-rank updates to an estimate of the inverse Hessian. Furthermore, the application of the Hessian is based on stored gradients from previous steps, and the matrix is never formed explicitly. An example of the convergence of the L-BFGS method is shown in Figure 3.5.

3.2.3 Convexity

A fundamental property of gradient-based methods is that they only converge to a *local minimum* of the objective function — that is, the bottom of *some* valley, not necessarily the deepest one (the *global minimum*). The

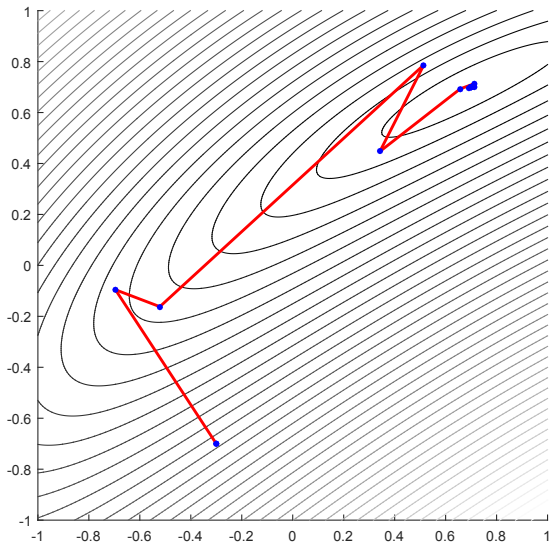


Figure 3.5. L-BFGS method often retains much of the favorable convergence behavior of Newton's method (Figure 3.4), but only uses the same first-order gradient information as the gradient descent method (Figure 3.3).

basin into which the solution falls is largely determined by the initial guess. It is sometimes also affected in unpredictable ways by the finite-length steps, as the iterates may inadvertently jump over a ridge of higher objective values.

There is, however, a wide and useful class of scalar-valued functions for which gradient descent based methods are guaranteed to find a global minimum. *Convex* functions are characterized by the property that connecting any two points on the (possibly high-dimensional) graph of a function by a line segment, it always lies at or above the graph:

$$\forall x_1, x_2, \forall \theta \in [0,1]: \qquad \theta F(x_1) + (1-\theta)F(x_2) \geq F(\theta x_1 + (1-\theta)x_2). \quad (3.38)$$

For a twice differentiable function, this is equivalent to the condition that the Hessian matrix of the function is nonnegative definite at every point. Intuitively, a convex function always "curves upwards". Any local minimum of a convex function is necessarily also the global minimum. Hence, a local optimizer cannot get stuck in a sub-optimal minimum. Due to this property, convex functions are often considered to be the class of "easy" problems: finding a good solution is guaranteed, and often an efficient special solver algorithm is available. Various interesting and surprisingly non-trivial problems can be formulated and effectively solved as convex optimization tasks. [11]

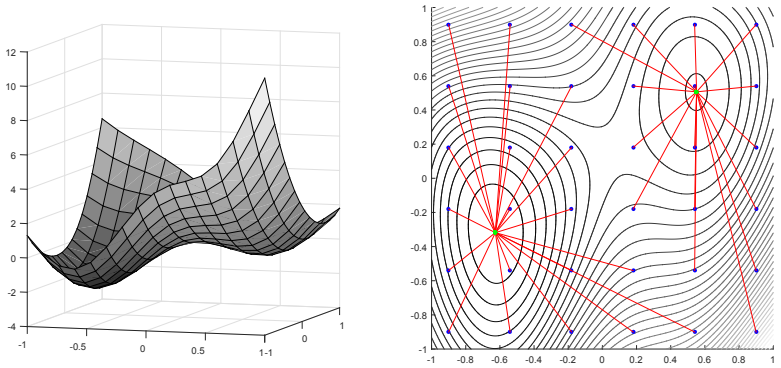


Figure 3.6. A non-convex function of two variables, visualized with surface and contour plots. Notice that the region above the graph of the function (the *epigraph*) is not a convex set; this is the origin of the name. The function has two local minima. The minimum on the left has the smaller value: it is the global minimum. Gradient-based optimization may converge to either minimum depending on the initial guess. An optimization task was started from each of the initial guesses marked by blue dots. The red lines indicate the minimum to which each of the tasks converged. Notice how a significant portion of the optimizations converged to the sub-optimal local minimum on the right.

Our interest in convexity is mainly negative: problems related to interpretation of appearance measurements are often non-convex. In particular, parametric BRDF models are generally not convex functions. Related optimization tasks tend to inherit this property. Figure 3.6 illustrates a non-convex function, and the behavior of gradient-based optimization in presence of multiple local minima. Designing the measurements and the objective functions in a way that admits to reliable optimization is a significant challenge.

3.2.4 Preconditioning

Preconditioning is a process of transforming a numerical problem into a form that is easier to solve, without changing the expected solution itself. It is often done by transforming the space that the unknown variables are presented in. [53, 121]

For a simple example, consider again the poor gradient descent steps shown in Figure 3.3. If we were to possess an estimate of how the objective function is elongated — say, we had an estimate about the axes and the amount of the elongation — we could form a linear transformation $P \in \mathbb{R}^{2\times 2}$ that approximates this stretch. Transforming the space by the inverse of P brings the objective function into a roughly isotropic

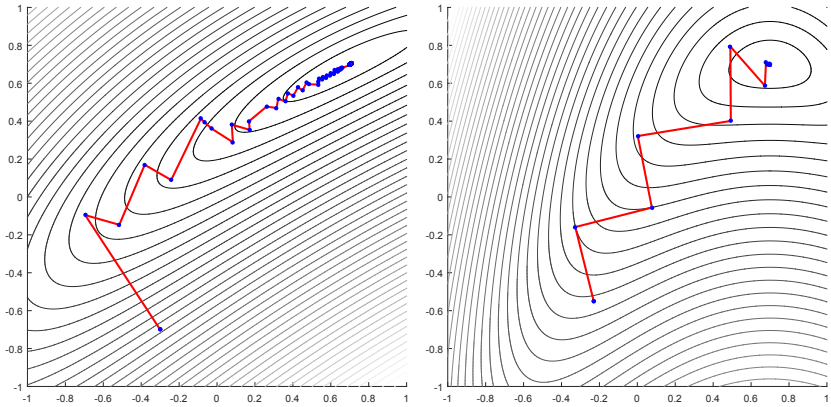


Figure 3.7. The original and the preconditioned version of the function in Figure 3.3, and the progress of gradient descent optimization in both. The function on the right is a diagonally squeezed version of the function on the left. Recall that the gradients of a function are always perpendicular to the contour lines. Intuitively, the goal of this operation is to make the contour lines more isotropic, so that they more readily point towards the minimum. This significantly improves the convergence behavior of the gradient descent method.

shape. Gradient descent steps in this transformed function point more directly towards the minimum, leading to rapid convergence, as shown in Figure 3.7. Hence, instead of solving the problem $\underset{x}{\operatorname{argmin}}\ F(x)$, we solve $\underset{\tilde{x}}{\operatorname{argmin}}\ F(P(\tilde{x}))$, and once converged, recover the un-preconditioned solution by $x = P(\tilde{x})$.

Many linear algebra algorithms take advantage of similar ideas. In particular, preconditioned Conjugate Gradient method, used to solve linear systems of equations $Ax = b$ (for positive-definite A), uses a carefully chosen estimate $P \approx A^{-1}$ as a preconditioner. The problem is transformed into an equivalent problem $PAx = Pb$ by left multiplication. Note that now $PA \approx A^{-1}A = I$, which makes it easier to find good steps. [53]

While not strictly about preconditioning, Newton's method (Section 3.2.2) can also be derived from similar ideas: at each step, the Hessian is used to form a linear estimate of the local elongation of the objective function. The Newton step is then simply a gradient descent step in a space where this elongation has been inverted. [11]

3.2.5 Constraints

One often needs to constrain the values of the variables in a minimization problem. For example, negative glossiness and albedo values are physically meaningless, but an optimizer might propose such values as solutions if it leads to lower values of the objective function. Various exten-

sions of the methods discussed above can explicitly enforce constraints [16, 17]. However, these methods tend to be either much more computationally intensive and difficult to use, or much slower to converge.

One approach for enforcing constraints is to transform the space of the variables in a way that prevents them from reaching the forbidden values. For example, the constrained problem

$$\begin{aligned} \underset{x}{\operatorname{argmin}} \quad & F(x) \\ \text{subject to} \quad & x \geq 0 \end{aligned}$$

might be transformed into the unconstrained problem

$$\underset{\tilde{x}}{\operatorname{argmin}}\, F(\exp(\tilde{x})) \tag{3.39}$$

Notice that no value of $\tilde{x}$ will result in a negative-valued argument to F. Given the solution $\tilde{x}^*$, the solution to the original problem is recovered as $x^* = \exp \tilde{x}^*$. Effectively, we are then optimizing for the logarithm of x instead of x itself. We use this kind of transformations in all of the publications to enforce different constraints.

3.2.6 Alternative methods

It should be noted that the class of gradient-based methods discussed here is not the only approach to optimization. In the interest of completeness, we briefly discuss a couple of examples of alternative methods below.

In the absence of available gradient information (or when it is not applicable, or it is essentially useless due to discreteness or extreme fluctuations of the objective function), a class of methods known as *metaheuristics* are sometimes used. For example, in *simulated annealing* [78] an optimization step is performed by making a random perturbation to the iterate according to some carefully chosen distribution, and accepting or rejecting the step based on a principled criterion. While theoretical guarantees about convergence do apply, in practice the behavior of the algorithm is very difficult to reason about.

Certain methods based on *combinatorial optimization* allow one to maintain a segmentation of an image into regions based on some well-defined objective function, and simultaneously maintain a set of per-region parameters. These are updated in an alternating fashion. Such approaches have found success in various related tasks, and could prove useful in our problems as well, as the spatial variation in many materials consists

of continuous regions separated by sharp transition boundaries. [79, 12, 120]

3.3 The Fourier transform

Publication I relies heavily on the *Fourier transform*. We review the relevant concepts directly in a general multivariate setting. A comprehensive review of the topic can be found in e.g. Bracewell [13].

The Fourier transform is a deeply fundamental object in mathematics. It decomposes a function into its constituent frequencies. This decomposition is also a function, with respect to the frequency variable. The representations are dual to each other: both the original *primal domain* function and its *frequency domain* counterpart are alternative views of the same object. Operations on a function often have a corresponding expression in the other domain: for example, differentiation in primal domain becomes pointwise multiplication in the frequency domain. Often, a problem that is difficult in one domain becomes easy in the other. This is the case in Publication I as well.

Definition and basic properties Concretely, the Fourier transform $\mathcal{F}$ is a mapping from a complex-valued function $f : \mathbb{R}^K \mapsto \mathbb{C}$ to its frequency representation $\hat{f} : \mathbb{R}^K \mapsto \mathbb{C}$ via an integral transform:

$$\hat{f}(\omega) = \mathcal{F}f(\omega) = \int e^{-i\omega^{\mathrm{T}}x} f(x) \mathrm{d}x. \tag{3.40}$$

Conversely, an inverse transform is obtained by a similar formula

$$f(x) = \mathcal{F}^{-1}\hat{f}(x) = \frac{1}{2\pi} \int e^{i\omega^{\mathrm{T}}x} \hat{f}(\omega) \mathrm{d}\omega. \tag{3.41}$$

The conventions with leading multipliers vary in literature. Note that these transforms are linear: for any two functions f, g and scalars α, β, we find $\mathcal{F}\{\alpha f + \beta g\}(\omega) = \alpha \mathcal{F}f(\omega) + \beta \mathcal{F}g(\omega)$.

For any fixed frequency $\omega \in \mathbb{R}^K$, the value of the Fourier transform can be seen as an *inner product* $< f, e^{-i\omega^{\mathrm{T}}x} >$ between f and a complex-valued plane wave basis function. Expanding the complex exponential, the plane waves are seen to be of the form $\cos(\omega^{\mathrm{T}}x) - i\sin(\omega^{\mathrm{T}}x)$. The real and imaginary parts of this function are unit-amplitude sine waves oscillating at wavelength $\frac{2\pi}{|\omega|}$ along the direction $\frac{\omega}{|\omega|}$, at 90 degree phase offset to one another. Alternatively, the complex magnitude is a constant 1, and the phase rotates at a constant rate.

The basis functions are orthogonal to one another. Hence, the Fourier transform is a unitary linear transformation, up to multiplicative constants: $\mathcal{F}^{-1} = \frac{1}{2\pi}\mathcal{F}^*$, where $\mathcal{F}^*$ is the adjoint of $\mathcal{F}$.

The frequency domain representation of a function is generally complex-valued. However, when the primal domain function is real-valued, the frequency representation has redundancy in the form of conjugate symmetry: $\hat{f}(\omega) = \overline{\hat{f}(-\omega)}$.

Frequency decomposition The basis functions, as mutually orthogonal waves of different frequencies, can be seen as measuring the "frequency content" of the primal-domain function. In particular, if the primal-domain function is a plane wave itself, the Fourier transform is merely a concentrated peak at the frequency of the wave. Superpositions of multiple waves result in multiple peaks. Such signals are particularly common in audio processing, where the Fourier transform and its variants are used to analyze the frequencies present in recorded signals, or conversely, to synthesize novel sounds. A particularly useful tool for these purposes is the *power spectrum* $|\mathcal{F}f|^2$, which reveals how the power in the signal is distributed across different frequencies.

In general, functions with slow variation consist mainly of low-frequency content. Conversely, rapidly varying functions also have high-frequency content. An audio analogy is again helpful: the rapidity of the vibration of an object determines the frequency (pitch) of the sound it emits.

Convolution theorem Perhaps the most important feature of the Fourier transform is *convolution theorem*. Convolution is an operation where a (mirrored) function is "slid" past another one, and their inner products are accumulated:

$$(f * g)(x) = \int f(y)g(x - y)\mathrm{d}y. \tag{3.42}$$

Convolutions are often used to express filtering operations: for example, for f an image and g a blur kernel, the convolution $f*g$ is a blurred version of the image. For one operand fixed, convolution is a linear transform on the other.

The convolution theorem states that convolution in primal domain corresponds to pointwise multiplication in frequency domain:

$$\mathcal{F}\{f * g\}(\omega) = \mathcal{F}f(\omega) \cdot \mathcal{F}g(\omega). \tag{3.43}$$

The latter operation is often significantly easier to handle both in theoretical analysis and numerical computations.

Dirac delta impulses An interesting and useful property of the Fourier transform is that it enables principled handling of Dirac delta impulses, i.e. "functions" $\delta(x)$ which have an infinitely tall and narrow spike at $x = 0$. The key property is that these functions perform point evaluation in inner products: $\int \delta(x-a)f(x)\mathrm{d}x = f(a)$. The related machinery can be formally derived using the theory of *tempered distributions*. Inserting the point evaluation formula into Eq. 3.40, we find that the FT of a shifted Dirac delta impulse is:

$$\mathcal{F}\delta_a(\omega) = \int e^{-i\omega^{\mathrm{T}}x}\delta(x-a)\mathrm{d}x = e^{-i\omega^{\mathrm{T}}a} \tag{3.44}$$

Note that this family of functions is the set of basis functions of the Fourier integral transform itself.

In particular, for an impulse at the origin, the FT is a constant function 1. This is indicative of the behavior of the transform in general: roughly speaking, it tends to convert "narrow" functions into "wide" ones, and vice versa. Indeed, one can show $\mathcal{F}\{f(Ax)\}(\omega) = \frac{1}{|A|}\mathcal{F}f(A^{-\mathrm{T}}\omega)$, where $\hat{f} = \mathcal{F}f$ and A is a non-singular square matrix. In other words, linearly stretching a function results in an inverse stretch to its frequency domain representation.

Shifting Fourier transform turns shifting into pointwise multiplication:

$$\mathcal{F}\{f(x-a)\} = e^{-ia^{\mathrm{T}}\omega}\mathcal{F}f(\omega) \tag{3.45}$$

This follows from considering the shift as a convolution with a Dirac delta impulse at a.

Differentiation Fourier transform also simplifies differentiation. The directional derivative along a vector q in frequency domain reduces to a pointwise multiplication[2]:

$$\mathcal{F}\{\nabla_q f\}(\omega) = iq^{\mathrm{T}}\omega\mathcal{F}f(\omega) \tag{3.46}$$

The underlying reason is that differentiating a complex exponential basis function results in another complex exponential, phase-shifted by 90 degrees and scaled by its frequency. Basis functions that oscillate rapidly along the the direction of the vector q obtain the full effect; conversely, oscillations in directions orthogonal to q are zeroed out.

[2]The pointwise multiplication suggests a convolutional nature for this operation. Indeed, differentiation may be seen as convolution with an "infinitely tightly spaced finite difference" of Dirac delta type. This notion can be made rigorous using the theory of tempered distributions.

Publication I makes an unusual application of a dual version of the differentiation formula to analytically evaluate Fourier transforms of a specific class of functions. Thanks to the close relation between the forward and inverse Fourier transforms, the differentiation formula can be turned on its head, stating instead that the pointwise product of an affine function and f corresponds to a directional differentation in the frequency domain:

$$\mathcal{F}\{q^{\mathrm{T}} x f\}(\omega) = i \nabla_q \mathcal{F} f(\omega) \tag{3.47}$$

Special cases Various related transforms can be seen as special cases of the full Fourier transform.

Fourier series arise when f is periodic, as the FT reduces to an infinite but discrete sequence of spikes.

Discrete Fourier Transform (DFT) arises when f is a finitely periodic sequence of discrete spikes. In this case its FT is of a similar form, and the entire transform reduces to a finite-dimensional linear transform (i.e. it is represented by a matrix). The Fast Fourier Transform (FFT) is an algorithm for computing the DFT in $O(n \log n)$ time, as opposed to $O(n^2)$ for a naive matrix-vector product.

In Publication I, we do not make use of these special cases, but rather work in terms of the full Fourier transform, i.e. assuming continuous functions in an infinite domain. In Publication III, we apply DFT and FFT to discrete pixel images for specific sub-tasks.

3.4 Gaussian functions

Another key component of Publication I is the use of *Gaussian functions* (or *Gaussians* in short). They arise in various contexts in mathematics and engineering. In particular, the density function of a (non-degenerate) normally distributed random variable is a Gaussian. Normally distributed random variables emerge from the Central Limit Theorem as limits of sums of independent random variables. Given this fundamental role, it is not surprising that the distribution enjoys various special mathematical properties. Publication I makes heavy use of the property that the Fourier transform of a Gaussian is another Gaussian. Ahrendt [2] presents an overview of the properties we use.

Definition and basic properties Intuitively, a Gaussian function is a fuzzy, possibly elongated "blob", examples of which are shown in Figure 3.8. Its

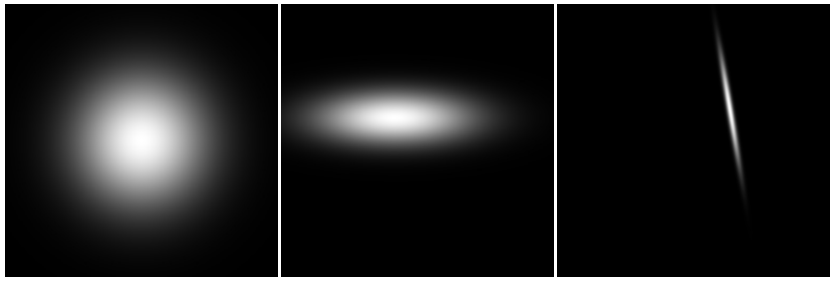


Figure 3.8. Bivariate Gaussian functions in $[-3,3] \times [-3,3]$: a standard zero-mean Gaussian with unit covariance; a non-zero mean Gaussian with diagonal covariance matrix (i.e. no "tilt"), and a general Gaussian.

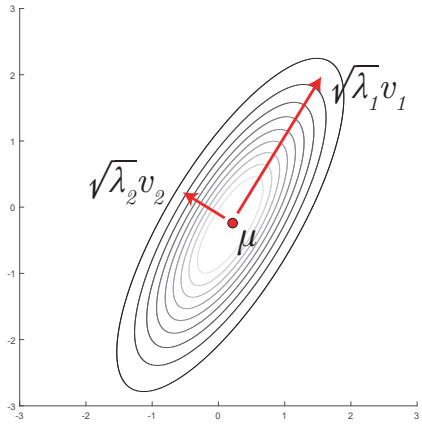


Figure 3.9. The level curves of a Gaussian are ellipsoids. The mean μ determines their centerpoint. The directions of the axes are determined by the eigenvectors of the covariance matrix Σ, and their lengths are proportional to the square roots of the respective eigenvalues.

level curves are elliptical. A k-dimensional Gaussian is defined by a centerpoint (*expectation* or *mean*) vector $\mu \in \mathbb{R}^K$, and a covariance matrix $\Sigma \in \mathbb{S}_k^+$, which defines its orientation and elongation along the axes. Here, $\mathbb{S}_k^+$ is the space of symmetric positive definite matrices of size $k \times k$. Given these, the Gaussian is a scalar-valued function

$$\mathcal{N}_{\mu,\Sigma}(x) = |2\pi\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\mu)^{\mathrm{T}}\Sigma^{-1}(x-\mu)\right) \tag{3.48}$$

Figure 3.9 illustrates the geometric interpretation of the parameters.

A Gaussian is essentially a negative-definite quadratic form ("downward opening paraboloid"), exponentiated and normalized as to integrate to 1. This makes it a probability density function. When considering the Gaussian as a general probability distribution, Σ may also be singular (i.e. only nonnegative definite); in this case the density formula is un-

defined, as the probability mass becomes "infinitely concentrated" on a lower-dimensional subspace.

We often consider weighted Gaussians with a leading multiplier z, and simply use the term Gaussian for these functions as well.

Fourier transform The Fourier transform[3] of a zero-mean Gaussian is another (unnormalized) zero-mean Gaussian with covariance inverted. Further applying Eq. 3.45 to shift the center to μ, we find:

$$\mathcal{F}\{\mathcal{N}_{\mu,\Sigma}\}(\omega) = \exp\left(-i\mu^{\mathrm{T}}\omega - \frac{1}{2}\omega^{\mathrm{T}}\Sigma\omega\right) \tag{3.49}$$

Pointwise product The set of Gaussian functions is closed under pointwise multiplication. The underlying reason is simple: under multiplication, the quadratic forms inside the exponentials are summed, yielding another quadratic form. The formulas work out to

$$\mathcal{N}_{\mu_a,\Sigma_a}(x) \cdot \mathcal{N}_{\mu_b,\Sigma_b}(x) = z_c \mathcal{N}_{\mu_c,\Sigma_c}(x) \tag{3.50}$$

where

$$\Sigma_c = (\Sigma_a^{-1} + \Sigma_b^{-1})^{-1}, \tag{3.51}$$

$$\mu_c = \Sigma_c(\Sigma_a^{-1}\mu_a + \Sigma_b^{-1}\mu_b), \tag{3.52}$$

$$z_c = |2\pi(\Sigma_a + \Sigma_b)|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mu_a - \mu_b)^{\mathrm{T}}(\Sigma_a + \Sigma_b)^{-1}(\mu_a - \mu_b)\right) \tag{3.53}$$

Linear mixtures A *Gaussian mixture model* (GMM) is a linear combination of Gaussian functions:

$$G(x) = \sum_{i=1}^{N} z_i \mathcal{N}_{\mu_i,\Sigma_i}(x) \tag{3.54}$$

Figure 3.10 shows an example of a GMM. These models are useful for approximating functions and probability distributions. The key observation in Publication I is that in a plane parameterization, a typical BRDF consisting of a specular and diffuse part is well approximated by such a mixture.

Gaussian mixtures inherit many of the useful properties of Gaussian functions. In particular, thanks to linearity the Fourier transform of a GMM is simply the weighted sum of the Fourier transforms of the component Gaussians. A product of two Gaussian mixtures is another Gaussian mixture with MN components (for mixtures of M and N components, respectively).

[3] In general, Fourier transform corresponds (up to sign and normalization conventions) to the notion of *characteristic function* in probability theory literature. Formulas for the latter can easily be translated to the usual Fourier transform conventions.

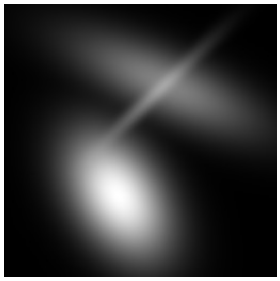


Figure 3.10. A mixture of three Gaussians.

Products with affine functions Publication I makes use of Gaussian functions modulated by *affine* functions of form $a^{\mathrm{T}}x+b$. While straightforward in primal domain, somewhat surprisingly these functions also have an explicit Fourier transform formula. Using Eq. 3.47, we find

$$\mathcal{F}\{(a^{\mathrm{T}}x+b)\mathcal{N}_{\mu,\Sigma}\}(\omega) = \left[(-i\Sigma a)^{\mathrm{T}}\omega + (-\mu^{\mathrm{T}}a+b)\right]\mathcal{F}\mathcal{N}_{\mu,\Sigma}(\omega) \tag{3.55}$$

3.5 Neural networks

Publication III uses a neural network based approach for comparing textured image patches. It is based on a texture synthesis method by Gatys et al. [46]. In this section, we will briefly review the key ideas and applications of convolutional neural networks. Goodfellow et al. [54] present a comprehensive recent overview on the topic.

Machine learning is a field of computer science and applied mathematics that studies algorithmic learning from examples. In particular, *supervised learning* is concerned with learning to map inputs to outputs, based on a finite set of examples. In contrast to traditional engineering, the idea is to sidestep the explicit manual construction of an internal rule-based model. It is replaced by a "black box" that is first trained using the example data, and thereafter used to make novel predictions. [59, 87, 8]

Some problems are well modeled by traditional engineering methods. For example, Newtonian physics is governed by a small set of unambiguous equations that predict the future positions and velocities of objects (for practical purposes) exactly. On the other hand, it is extremely difficult to hand-engineer a program that discriminates between photographs of cats and photographs of dogs: the raw pixel values provide scarce clues, and inventing suitable feature descriptors has proven challenging. Indeed, hand-engineered computer vision approaches have met with success only

on a limited set of problems.

Continuing to use the recognition task as an example, we can model it as that of building a function $f : \mathbb{R}^{M \times N \times 3} \mapsto \mathbb{R}^K$, where the input is an RGB photo, and the output is a list of probabilities of the image belonging to given K categories (e.g. "dog", "cat", "tractor", "ostrich", ...) The task of supervised learning is to formulate such an f from being shown a sequence of (possibly millions of) images accompanied by the expected output (i.e. a vector with the value 1 on the correct category and zero elsewhere).

General neural networks *Neural networks* are a framework for building functions such as this. The idea is to compose the function from a sequence of very simple operations (often called *layers*): mostly affine transformations alternating with simple non-linear pointwise *activations functions* such as $a(x) = \max(0, x)$ or $a(x) = \tanh x$. The parameters of a neural network are the weight matrices and the bias vectors of the affine transformations. Sufficiently deep compositions of these elementary operations can in principle approximate arbitrarily complicated functions, assuming that suitable parameters can be found.

A network is trained by optimizing the agreement of the predictions of the network with the desired values in the training set. In practice, this is done using variants of *stochastic gradient descent*, where only a small part of the training set (which is potentially huge, or even infinite) is used to compute the derivatives on each iteration. The derivatives themselves are evaluated using an algorithm known as *backpropagation*, which essentially consists of sequential application of the chain rule. The intermediate derivatives — and consequently the derivative of the entire network — are easy to compute thanks to the simple form of the elementary operations.

Convolutional neural networks General neural networks are ill-suited for most tasks involving images. First of all, the linear transformations become enormous, due to the large number of pixels. Aside from storage issues, the astronomical number of parameters leads to *overfitting*: even a large training set cannot sufficiently constraint it, and consequently the network generalizes poorly to previously unseen input. Such issues of capacity and generalization are central to the study of neural networks; however, they are beyond the scope of our discussion. Finally, image features tend to be *shift-invariant*: an edge is an edge, no matter which part of the image it resides in. The same holds for higher-level features, such as, say, circles, eyes, or ostriches. A general neural network is not required

to respect this property.

Convolutional neural networks are a restriction on the general neural network model that elegantly address all of these issues. The general affine transformations are replaced by *convolutions*. In this context, a convolution is understood as an affine transformation performed on each small image neighborhood in a sliding window fashion. The number of "channels", or *activations*, in the convolved image may change. Due to the small spatial extents of the convolution *kernels*, the dimensionality of the parameter space becomes manageable. Furthermore, the excessive capacity in the network is eliminated, and the shift invariance is enforced in a natural manner.

3.5.1 VGG-19 network

The texture synthesis method of Gatys et al. [46], and consequently Publication III, takes advantage of a specific convolutional network architecture known as VGG-19 [117]. It is designed for image recognition and trained using a large-scale image dataset based on ImageNet [32], consisting of 1.2 million images manually classified into 1000 categories. Most widely used convolutional neural networks follow similar principles.

The VGG network architecture is simple; it consists of a few types of elementary functions, connected as layers in a linear chain. The input to the network is an image $a^0 \in \mathbb{R}^{224\times224\times3}$. Each layer performs some function

$$f^l : \mathbb{R}^{S_{l-1}\times S_{l-1}\times n_{l-1}} \mapsto \mathbb{R}^{S_l\times S_l\times n_l} \tag{3.56}$$

on its input image stack, possibly modifying its dimensions. Hence, the *activation* of the l'th layer is computed as the composition

$$a^l = f^l \circ f^{l-1} \circ ... \circ f^2 \circ f^1(a^0). \tag{3.57}$$

The functions are chosen as to gradually contract the spatial dimensions S_l while expanding the number of activations n_l, so as to ultimately yield the vector $a^L \in \mathbb{R}^{1\times1\times n_L}$ of class membership probabilities (where $n_L = 1000$ for the ImageNet, and L is the total number of layers). The functions f^l are of four types:

- *Convolutions* compute the sliding-window affine transformation discussed above. Specifically, the formula for this is

$$a_i^l = \sum_j a_j^{l-1} * k_{ij}^l + b_i^l \tag{3.58}$$

Here, a_i^l denotes the i'th feature map of the l'th layer, $*$ is the discrete convolution operator as traditionally understood in signal processing, k_{ij}^l is a $m \times m$ convolution kernel (typically with $m = 3$), and b^l is a vector of per-activation biases. The number of kernels (and biases) determines the number of output activations n_l. *Fully connected* layers, towards the end of the network, can be seen as convolutions with 1×1 kernels. The kernels k and biases b are the free parameters of the network, subject to learning.

- *Rectified activation functions* are simple pointwise functions of the form $a^l = \max(0, a^{l-1})$ that always follow a convolution. They are the main source of nonlinearity in the network.

- *Pooling* layers are used to reduce the spatial dimensions of the activations. The network uses *max-pooling*, which simply cuts the spatial dimensions to half by retaining only the maximum value of each 2×2 region.

- The very final layer of the network is a *softmax* layer which normalizes the activations of the previous layer into a discrete probability distribution. We will not be using it in our application.

Figure 3.11 illustrates the arrangement of these operations in the VGG-19 network.

Significance of the activations The hope behind deep convolutional neural network models is that the layers learn a hierarchy of increasingly complex and meaningful feature detectors. Indeed, this behavior emerges in trained networks.

The first layers tend to represent simple filters, such as edge, corner and blob detectors, the results of which can be read in the different activation channels. They are somewhat analogous to traditional hand-engineered computer vision feature descriptors, such as SIFT [85] and BRIEF [18]. They are also reminiscent of filters found in visual cortices of mammals [66, 90, 1]. The higher layers combine this information to detect increasingly complex geometric and *semantic* features: first, corners of eyes, eyebrows, pupils — then, entire eyes, noses and ears — then, faces, arms, legs — from these, cats, dogs, humans — and ultimately even fine-grained sub-species of animals and objects.

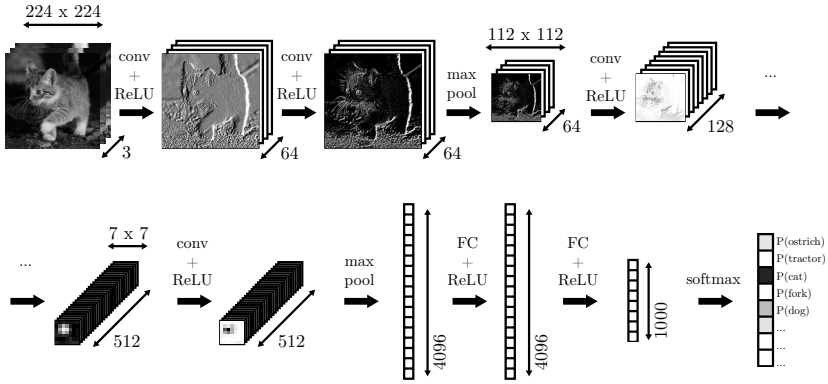


Figure 3.11. Structure of the VGG-19 network (with a few intermediate layers omitted). The input RGB image (top left) is seen as a $224 \times 224 \times 3$ activation map. It is fed through a sequence of 3×3 convolution filters, rectified linear activation units (ReLU), and max-pooling operations. The spatial dimensions are gradually shrunk by pooling, whereas the number of activations gradually grows in convolutions. In the last layers, the image is collapsed into a vector, which may be seen as an $1 \times 1 \times n$ image. The fully connected (FC) transformations are hence special cases of convolutions. Finally, the softmax layer normalizes the result into a probability distribution over classes.

3.5.2 Backpropagation

We do not train neural networks in any of the presented methods. However, we do take advantage of a component that is centrally important for the training procedure: the backpropagation algorithm [111].

Consider a chain of composited functions such as in Eq. 3.57. Let us assume that the last function f^L is scalar-valued, i.e. $a^L \in \mathbb{R}$ (in training it would evaluate a loss value that compares the prediction of the network with the training data).

As an illustration of the backpropagation algorithm, let us evaluate the partial derivatives of the network scalar output a^L with respect to each pixel in the input activation layer a^0. These are enumerated in the Jacobian matrix $J \in \mathbb{R}^{1 \times N}$ of the entire composite function, evaluated at a^0. Here, N is the number of entries in a^0, i.e. $224 * 224 * 3$ for the VGG-19 network. The chain rule states that J can be computed as the matrix product of the Jacobians of the functions in the composition (Eq. 3.57):

$$J = J_L \; J_{L-1} \; ... \; J_2 \; J_1 \tag{3.59}$$

Here, J_l is the Jacobian of f^l, evaluated at the point $a^{l-1} = f^{l-1} \circ ... \circ f^2 \circ f^1(a^0)$. These points are obtained in the first (*forward*) pass of the algorithm by simply evaluating the entire composition in sequence, and storing the intermediate results.

While this product can be used to compute the partial derivatives of interest, in practice constructing the full Jacobians and computing the matrix products is prohibitively expensive. Fortunately, it is unnecessary. The key trick behind the backpropagation algorithm is to instead consider the *transpose* of the Jacobian. This seemingly trivial modification flips the order of the matrix multiplications:

$$
\begin{aligned}
J^{\mathrm{T}} &= (J_L \; J_{L-1} \; J_{L-2} \; ... \; J_2 \; J_1)^{\mathrm{T}} && (3.60)\\
&= J_1^{\mathrm{T}} \; J_2^{\mathrm{T}} \; ... \; J_{L-2}^{\mathrm{T}} \; J_{L-1}^{\mathrm{T}} \; J_L^{\mathrm{T}} && (3.61)\\
&= J_1^{\mathrm{T}} \left(J_2^{\mathrm{T}} \left(\; ... \left(J_{L-2}^{\mathrm{T}} (J_{L-1}^{\mathrm{T}} \; J_L^{\mathrm{T}}) \right) ... \right) \right) && (3.62)
\end{aligned}
$$

Notice that the innermost matrix J_L^{T} is in fact a column vector, because the corresponding function f^L is scalar-valued. Hence, Eq. 3.62 is a sequence of matrix-vector products. In the second (*backward*) pass of the backpropagation algorithm, these matrix-vector products are computed by sequentially multiplying J_L^{T} from the left with the transpose Jacobians, starting from J_{L-1}^{T} and ultimately ending at J_1^{T}. Notice that the intermediate result is always a vector, and no matrix-matrix products need to be computed.

This procedure relies only on our ability to compute the transpose Jacobians of the elementary operations f^l in the network. This is (by design) straightforward for the kinds elementary operations discussed above. Furthermore, the transpose Jacobians are usually structured enough that the *effect* of the multiplication can be implemented as a suitable piece of program code, and no matrix ever needs to be built explicitly. Note that the code that implements multiplication by J^{T} is sometimes quite different than that for J.

As a side effect, this procedure also computes the partial derivative of the loss with respect to the intermediate layers. Training uses this information to update the parameters of the layers. More generally, the algorithm can be applied to networks where the layers are connected in a directed acyclic graph (DAG), as the chain rule also handles branching of this kind. We make use of this in Publication III, where the additional operations attached to the VGG network — in particular, rendering — are expressed as a composition of operations arranged in a DAG.

4. Related work in appearance capture

In this chapter, we will review relevant previous work on appearance capture. Weyrich et al. [131] present a relatively recent comprehensive overview on appearance acquisition.

4.1 Direct sampling

Classical reflectance capture methods are based on arranging point lights, sensors (cameras) and the material sample into geometric configurations that directly reveal BRDF values. Given known convex geometry, pinhole camera and a point light, the integral in the reflection equation (Eq. 2.4) for a given pixel reduces to the value of the BRDF at a pair of angles, times known constants from the light and exposure parameters. The remaining "inverse problem" is trivial: a raw point sample of the BRDF value is recovered by dividing away the constant. A dense tabulated representation of the BRDF can be obtained by collecting these measurements from a large number of view and light angles. The challenge in these methods lies in physical arrangements: a wide range of geometric configurations must be covered to obtain a good sampling of the BRDF, often requiring complex hardware and delicate calibration procedures. In some approaches the sampling is left incomplete, and other computational techniques are used to fill in the gaps; we will review such approaches in later subsections.

4.1.1 Gonioreflectometry

A classical device for this purpose is the *gonioreflectometer* [42, 132]. It typically consists of a robotically controlled gantry, with a camera and a light source attached to two arms. Pairs of directions are sampled exhaustively.

While highly accurate BRDFs can be captured (given careful calibration), the process is cumbersome. The number of samples required to get a sufficient covering of the angular space is very large. As a general BRDF is a four dimensional function of angles, one needs roughly 100^4, or 100 million samples to cover it with a relatively dense grid. Assuming one measurement per second, the process would take years. Reciprocity of the BRDF can be used to cut the required samples to half. In practice, by cutting down on the density of the sampling along dimensions and regions of (typical) low variation, and dropping the fourth dimension in case of isotropic materials, the capture time can be brought down to more reasonable numbers. Besides the slowness of the process, building and using the required hardware is challenging and is rarely undertaken by practitioners.

The gonioreflectometer in its classical form only measures the BRDF of a homogeneous piece of material. The same idea can be extended to capturing an SVBRDF, by photographing and illuminating a spatially varying surface from various angles [96, 27]. Figure 1.7 illustrates the kind of input captured by these devices. The setup inherits the difficulties of single-point capture, and poses additional calibration and storage challenges.

For practical purposes, the tabulated representations are often fitted to lower-dimensional parametric BRDF models such as those reviewed in Section 2.3.2. This comes at the cost of losing some accuracy. Section 3.1.1 sketched an example of this procedure. This procedure also interpolates and extrapolates the reflectance function to cover angles that were not included in the captured dataset — however, the caveats on generalization apply.

4.1.2 Alternative geometries

Significant savings in capture effort can be made by more clever physical arrangements of the measuring device and the measured surface. For example, suitably shaped and aligned curved mirrors [26, 28] or curved geometry [91] allow one to observe extended slices of a BRDF in a single image. This reduces the physical complexity of the motions needed to obtain samples from the relevant angles. Furthermore, it allows one to use each pixel of a photograph as a separate measurement of the BRDF, yielding a large number of tightly-spaced measurements in bulk. In particular, Matusik et al. [93, 94] applied the technique of Marschner et al. [91]

to measure a set of 100 isotropic BRDFs from spherical specimens. The resulting dataset, known by the name MERL[1], has since been used as the basis of a number of studies on material appearance. Homogeneous anisotropic BRDFs have been captured by attaching slices of differently oriented material samples onto a rotating cylinder [99].

4.2 Indirect sampling

Direct point sampling of the BRDF function results in readily usable measurements. The complexity lies in the physical setup, which must be carefully engineered to isolate individual light paths from the source to the sensor, and to cover a wide range and quantity of angles. The gonioreflectometer and related techniques are hence at one extreme of a tradeoff: their measurements are difficult to obtain, but simple to interpret.

The methods we present in this thesis are based on *indirect sampling* of the reflectance. Publication I uses area light sources with controlled illumination patterns. The measured values are no longer proportional to the value of the (SV)BRDF at any single pair of angles. Rather, they are complicated mixtures of these values. On the other hand, Publications II and III do use point samples, but because the structure of the spatial variation is unknown, they cannot be directly assigned to any individual surface point.

4.2.1 Extended light sources

The effect of the material is also apparent under other lighting conditions than point lighting. Under more general illumination, the reflected radiance from the surface is no longer directly proportional to an individual value of the BRDF. Rather, it is typically a weighted integral over the values — in other words, an inner product of a measurement function and the (cosine-weighted) BRDF. Recalling the reflection equation (Eq. 2.4), we have:

$$L_o(\omega_o) = \int_\Omega L_i(\omega_i)\ f_r(\omega_i \rightarrow \omega_o) \cos\omega_i \,\mathrm{d}\omega_i \tag{4.1}$$

$$= < L_i,\ f_r(\omega_i \rightarrow \omega_o) \cos\omega_i > \tag{4.2}$$

where L_i and L_o are the incident and exitant radiance functions, respectively. By controlling L_i, we can make linear "queries" into the content of

[1] Publicly available at http://www.merl.com/brdf/

the BRDF. In general L_i can be arbitrary.

Seen in this light, a direct point sample simply corresponds to an inner product with a Dirac delta distribution at the sample direction. Such a measurement gives the value of the BRDF (up to other known multiplicative factors) exactly at the measured angles, and no information about other parts of the function. A measurement with lighting angle ω_m and viewing angle ω_o is then given by the reflectance equation with a corresponding Delta distribution substituted for the incoming radiance:

$$\begin{aligned} L_o(\omega_o) &= \int_\Omega \delta_{\omega_m}(\omega_o)\ E(\omega_i)\ f_r(\omega_i \to \omega_o) \cos\omega_i \ \mathrm{d}\omega_i && (4.3)\\ &= <\delta_{\omega_m},\ f_r(\omega_i \to \omega_o) E(\omega_i) \cos\omega_i > && (4.4)\\ &= f_r(\omega_m \to \omega_o) E(\omega_i) \cos\omega_m && (4.5) \end{aligned}$$

As the known cosine factor $\cos\omega_m$ and irradiance from the light source $E(\omega_i)$ can be divided out, this directly reveals the value of the BRDF at the given angles. The same ideas generalize to spatially varying BRDFs by considering each surface position in isolation.

In contrast, a more general measurement against e.g. an area light source gives the average of the BRDF values over a finite region of the angular space. It provides information about the reflectance across a wider range of angles, but cannot directly pinpoint the exact value (which is ultimately of interest) at any individual point. However, because BRDFs exhibit many types of regularity and structure, a small number well chosen measurements often suffice to reveal the significant features. The tradeoff is that interpreting such measurements is harder. The framework in Section 3.1 is designed to deal with this type of problems. However, previous work often relies on collecting measurements that can be interpreted by more direct means, at the expense of added acquisition complexity.

The method of Gardner et al. [44] is an illustrative example. They propose a device that translates a linear light source over a flat surface sample. The camera is stationary. The temporal intensity profile of each pixel is recorded, and a spatially varying isotropic Ward BRDF [130] is fitted to the trace by a heuristic procedure. Surface normals can be estimated by performing a second pass with the surface rotated. Conceptually, the underlying idea is to exploit the angular redundancy in isotropic BRDFs: the linear light source essentially marginalizes the BRDFs along one of the angular dimensions. This sampling captures the width and the direction of the peak, while losing only little information since the BRDF is close to radially symmetric. Due to the fixed camera position, most

viewing angles are never observed. However, because the observed slice suffices to identify key properties of the BRDFs, the measurements are plausibly extrapolated to the missing angles by the parametric model fit.

4.2.2 Basis illumination

Carefully choosing the illumination used for sampling can help to ensure that the measurements reveal maximal amount of information about the reflectance while admitting to a tractable interpretation procedure. This is more likely when the illumination enjoys some special mathematical properties — in particular, area lights emitting different basis function patterns, such as spherical harmonics [106, 118], have been applied in literature.

Basis function transformations are generally used to obtain an alternative "perspectives" into functions in mathematics and engineering. Many of them change the role of local and global features in a signal. Globally supported basis functions, such as spherical harmonics and the Fourier plane waves, generate response to arbitrarily sharply concentrated peaks. The angular dimensions of SVBRDFs are often of precisely this type: a tight unimodal specular peak is pointing towards some direction, depending on the surface normal. In contrast, direct pointwise measurements almost always miss these features: without extensive sampling, sharply peaked specular lobes may fall between the measured angles, or cause aliasing patterns.

Simple examples of this idea are the constant and linear gradient basis function; the former measurement reveals the integral over the function, and the latter its centroid. Applying this principle, Ma et al. [86] estimate the surface normals of an object by surrounding it with a spherical dome that emits spherical gradients. The average direction of the reflections can be estimated from this data. Ghosh et al. [50] extend this to second-order gradients, which are used to simultaneously extract also spatially varying glossiness values. The method relies on polarization to separate the diffuse and specular components. The spherical gradient functions are closely related to spherical harmonics, which is the natural Fourier basis on a spherical domain. Ghosh et al. [48] use a curved mirror setup to emit zonal basis patterns closely related to spherical harmonics to accurately capture homogeneous BRDFs. Tunwattanapong et al. [124] emit spherical harmonic patterns using a rotating arc of LED lights; a full spherical pattern is emitted by modulating the LED intensities over a revolu-

tion, and integrated by long camera exposure. Full geometry and spatially varying BRDF is recovered. Spherical harmonics have also proven useful in analyzing reflectance phenomena in a signal processing context [106]. Malzbender et al. [89] capture visually rich re-lightable images of objects using a polynomial basis. However, their re-lightings are limited to the original viewing angle.

While effective at reducing the number of measurements, previous approaches in basis function measurements require complicated hardware in order to emit the patterns in a suitable domain — often, a spherical dome surrounding the sample. This complication stems from the desire to make the measurements in the spherical domain, which is most natural to BRDFs. In Publication I, we present an alternative approach: by representing the BRDFs on a plane-projected domain, we can use a standard LCD monitor to emit the patterns. While mathematically more complicated, this choice of domain admits to a natural use of the Fourier basis, which enjoys several useful properties that facilitate the interpretation.

4.3 Exploiting spatial redundancy

A significant number of approaches, based on both direct and indirect sampling, take advantage of spatial redundancy of surfaces to reduce the amount of measurements needed. Many surfaces consist of a small number of different BRDFs mixed and scattered across the surface. Hence, even if each surface location is insufficiently sampled in isolation, information from other parts of the surface can be used to fill in the missing data. The challenge in these approaches lies in identifying the redundancies. Publications II and III use this general approach to capture SVBRDFs from a very low number of input photographs.

The sampling of Marschner et al. [91] can be seen as a trivial example of this principle: the measurements are insufficient for any individual point on the sphere, but the knowledge that the sphere has a homogeneous BRDF allows us to combine the measurements across the spatial locations, yielding a complete sampling.

The presence of multiple BRDFs on a single object leads to a much more challenging task: simultaneously determining the BRDFs, and what points they are present at, is a difficult chicken-and-egg problem. Lensch et al. [83] obtain a sparse sampling of an object of known shape under varying point lighting. The assumption is that a few different paramet-

ric BRDFs are present on the surface. The algorithm alternatingly clusters the surface points to the current set of estimated BRDFs, and refines the BRDF estimates based on the most recent clustering. The resulting SVBRDF a linear combination of a global set of few representative BRDFs at each surface point. Goldman et al. [52] use a similar idea of a small set of basis BRDFs in combination with a fixed camera, to simultaneously estimate the surface shape along with the BRDFs. Alldrin et al. [3] extend these methods to support non-parametric BRDF models.

Dong et al. [35] explicitly measure a basis of high-fidelity representative BRDFs (including anisotropy) directly from a surface using a custom portable scanning device. A separate set of coarse measurements are made from the full surface, and this data is used to assign the basis BRDFs. Ren et al. [108] introduce a lightweight linear light source based SVBRDF capture method that uses a physical chart of exemplar materials that is assumed to represent the BRDFs present on the surface.

Wang et al. [128] recover SVBRDFs exhibiting anisotropic reflectance from a set of measurements that cover the angular space relatively densely, but only from a limited angular range. Measurements from surface points with same BRDF but different anisotropy orientation are combined to fill in the missing angular data at each point. The linear light source method of Gardner et al. [44] was extended by Chen et al. [21] to handle anisotropic materials using similar ideas.

Zickler et al. [136] exploit the joint redundancy in angular and spatial dimensions. The input to the method is a sparse set of variously illuminated images of a known geometric object. The SVBRDF is viewed as a 6-dimensional function with no hard division between spatial and angular dimensions, and the input data is interpolated in this domain using radial basis functions.

Wang et al. [127] estimate a coarse bi-scale roughness model (*i.e.* a microfacet BRDF with stochastic meso-scale normal variation) from a photograph of a stochastically repeating surface material under step-edge illumination. Hence, unlike most previous work, the method takes advantage of recurrence of spatial features in the material: the underlying assumption is that points with identical material properties are surrounded by similar neighborhoods. However, their model is only accurate for a very narrow class of Perlin noise type surface shapes with constant glossiness and albedo, and cannot be extended in any obvious way to support more general surfaces. Publications II and III apply a similar underlying

idea to recover full detailed SVBRDF maps from a much wider range of stochastically repeating surfaces.

4.4 Strong assumptions and heuristics

A number of approaches simplify the capture procedure by relying on strong assumptions, user input and heuristic tricks to extract plausible reflectance representations from the data. These approaches usually require only a small amount of input data. The stationarity assumption behind Publications II and III can be seen as falling into this category.

Depth-from-shading methods are based on the assumption that pixel intensities in a photograph of a surface are correlated with the surface depth due to shading effects. Glencross et al. [51] use this assumption to extract height and albedo maps from a flash/no-flash photograph pair of a diffuse surface. However, only a narrow range of surfaces is expected to accurately correspond to their assumptions, and the reconstructions rarely match the ground truth despite their visual plausibility.

A popular software package CrazyBump [24] extracts a normal map from a single photograph by proprietary heuristic algorithms, guided by user-controlled sliders. While sometimes surprisingly plausible visually, these reconstructions do not accurately match the input data or the underlying reality.

Dong et al. [34] propose a user-aided method for assigning BRDFs onto photographs of materials. The pixel values and a set of user-provided sparse strokes are used to compute blending weights. Finally, a heuristic normal recovery procedure is applied based on user's estimate of the lighting direction.

Barron and Malik [5] use a large set of carefully engineered priors to estimate shape and diffuse reflectance of arbitrary surfaces from a single image.

"Blind" methods recover BRDFs and SVBRDFs from objects viewed under unknown environment illumination [109, 33]. The idea is to find the most plausible explanation of the observed reflections by making assumptions about the space of typical environments (i.e. natural images with strong step edges) and the space of commonly occurring BRDFs.

4.5 Exploiting physical properties of reflectance

Helmholtz stereopsis [135] takes advantage of the reciprocity of BRDFs. Pairs of photographs of a scene, with the placements of the camera and the light source exactly swapped, can be used to extract the depth and the normal maps independently of surface BRDF, and subsequently to reconstruct the geometry of the object. This idea was used by Holroyd et al. [65] to jointly recover full geometry and reflectance of 3D objects.

Many methods (*e.g.* [86, 50]) rely on the fact that polarization of first-surface specular reflection differs from that of diffusely reflected light. This observation can be used to separate the diffuse and specular components. Ghosh et al. [49] explore the use of different types of polarization to extract full SVBRDF information from a few photographs of an object with known geometry. Having a readily separated mesurements of the diffuse and specular components of an SVBRDF greatly simplifies the data interpretation.

On the other hand, working with polarizers adds hardware complexity and manual steps to the capture procedure, and we choose to avoid it in our methods. The diffuse-specular separation is a relevant problem in this thesis, as the methods all use a fixed viewpoint, and hence clues from camera motion cannot be used. The methods perform the separation computationally, by considering the mixture as a part of the unknown parameters to be recovered. The main clue that we rely on is the fact that the specular component is often significantly more peaked than the diffuse component.

Our methods also make implicit use of the *dichromatic* model of reflectance [115], and hence benefit from colored measurements. The chromaticity of the diffuse and specular components in isolation typically do not depend on the viewing angle. Hence, the chromaticities of the observations of a surface point implicitly constraint the space of possible diffuse-specular decompositions. Furthermore, the specular component tends to be monochromatic across the surface. Publications II and III explicitly enforce this property on the solution in order to cut down the degrees of freedom in the problem.

5. Frequency domain measurements

Publication I continues the tradition of basis function measurement approaches [86, 48, 50, 124]. As discussed in Section 4.2.2, basis functions with global support are effective at isolating the relevant features of BRDFs. Successful SVBRDF capture has previously been demonstrated using natural basis functions of the spherical domain of BRDFs. However, emitting these patterns typically requires the sample to be surrounded by a custom spherical light dome — a major engineering effort. Ghosh et al. [50] also present a low-cost alternative, where an LCD monitor is placed above the sample, and a small slice of the distant spherical environment is approximately simulated by displaying projections of the spherical basis on the monitor plane. The approach is limited to glossy specular materials with limited normal variation. Furthermore, it requires the use of polarizers to separate the diffuse and specular components from one another. The main focus of their paper is, however, on spherical dome setups.

The desire to match the domain of the measurements with the natural spherical domain of BRDFs is a major source of physical complexity. The measurements from dome-based basis functions are in the "correct format" and hence relatively easy to interpret, but acquiring them requires the use of custom hardware or delicate calibration and approximations. Publication I takes a different approach: it uses an LCD monitor for emitting basis function patterns, but instead of emulating a spherical domain, it uses a basis that is natural for the planar emitter itself. The physical setup is simple, but the difficulty is shifted towards the measurement interpretation step. Fortunately, the natural basis in a planar domain is the Fourier basis, which enjoys a rich set of mathematical properties — in particular, it turns out to be well suited for modeling the relevant slices of the BRDFs. This in turn enables direct parametric SVBRDF model fitting

by optimization, as opposed to the heuristic (and polarization-dependent) interpretation approaches in previous work.

In this chapter, we will present a brief high-level overview of the key ideas of the method. The reader is referred to the attached publication for the details. In the remaining chapters, we will refer to the sections, appendices, equations and figures in the three publications by suffixing them with the Roman numerals I, II and III.

5.1 Measurements

The physical setup presented in Publication I uses a camera and a flat-panel monitor, obliquely arranged around the flat physical sample. The setup and the notation for the coordinates is illustrated in Figure 5.1. The camera and the monitor are controlled by an attached computer. The idea is to display a sequence of plane wave basis functions on the monitor, and record the reflections off the physical sample using the camera.

The monitor and the camera are in a near-field configuration, and we do not make any distant illumination approximations. Instead, we fully account for the effect of the geometry of the capture setup in the interpretation stage. This allows us to place the monitor close to the sample, which results in a wide coverage of illumination directions for each surface point.

Figure 5.2 shows examples of the photographs captured. The plane wave patterns that were displayed on the monitor can still be observed in the reflections. The BRDF at each surface point affects the reflected pattern in various ways. In particular, the pattern is diminished and colored by albedos, and high-frequency patterns tend to be diminished by low-gloss reflections. Surface normal variations cause spatial distortions in the reflected waves. Intuitively, the goal of the method is to disentangle these effects by finding an SVBRDF that explains them across a variety of observed reflections.

5.1.1 Basis function patterns

Specifically, the displayed patterns are windowed Fourier basis functions. Parameterizing the monitor as a rectangle covering $[-\pi, \pi] \times [-\pi/a, \pi/a]$ of the $\mathbb{R}^2$ plane, where a is the monitor aspect ratio, the pattern correspond-

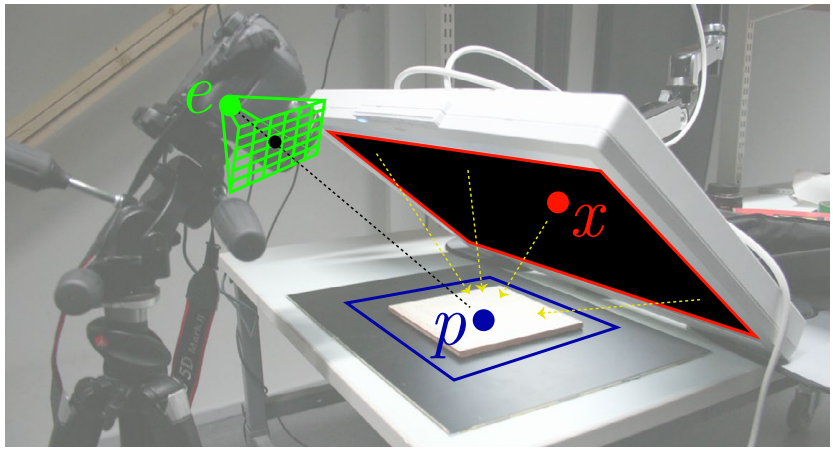


Figure 5.1. Physical capture setup. We denote the position of the camera as $e \in \mathbb{R}^3$, positions on the material sample plane as $p \in \mathbb{R}^2$, and positions on the monitor plane as $x \in \mathbb{R}^2$. Depending on context, we use the same notation for the corresponding three-dimensional world space coordinates. The relevant positions and coordinate systems are established in a separate geometric calibration stage. The general strategy is to emit light from the monitor using a pattern that varies as a function of x, and observe its reflections from each point p on the material towards the camera at e. Each pixel on the image plane receives measurements from a different surface point p.

ing to the frequency $\omega \in \mathbb{R}^2$ is

$$b_\omega(x) = w(x)\exp(-i\omega^{\mathrm{T}}x) = w(x)\cos(\omega^{\mathrm{T}}x) - iw(x)\sin(\omega^{\mathrm{T}}x) \tag{5.1}$$

Windowing Here, $w(x)$ is a windowing function that smoothly fades the edges of the pattern to black towards the edge of the monitor. The reason for using such a function is that the finite spatial extent of the monitor itself necessarily imposes a box windowing function with an abrupt jump at the monitor edges. This distorts the measurements in a manner that is difficult to control. By using a carefully chosen windowing function that approximately fits within the box, we essentially replace the window with something we can more readily model mathematically. Specifically, we use a Gaussian window with a diagonal covariance and a zero mean — the reason for this choice will become obvious later.

Displaying complex-valued images Physical monitors can only display real-valued non-negative images. To simulate the display of the complex-valued patterns, we split them to four parts that are shown separately, as shown in Figure 5.3. Thanks to the linearity (and lack of complex-valued multipliers) of light transport, the same summation of any photographs illuminated by the corresponding patterns results in the correct hypothetical image of the complex-valued radiance reaching the eye. As a welcome

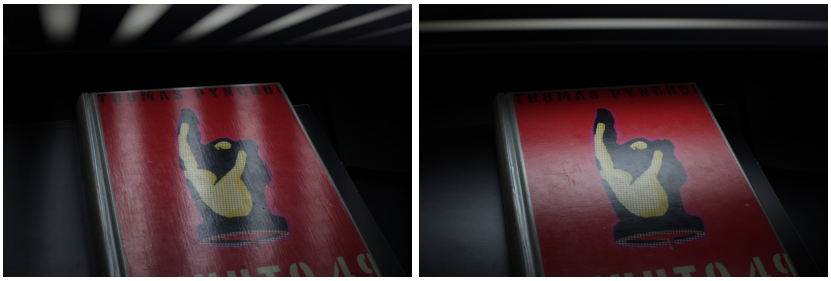


Figure 5.2. Two examples of raw input photographs for the method. A part of the monitor and the emitted patterns is seen at the top of the image. Below, the patterns are seen reflected off the material sample.

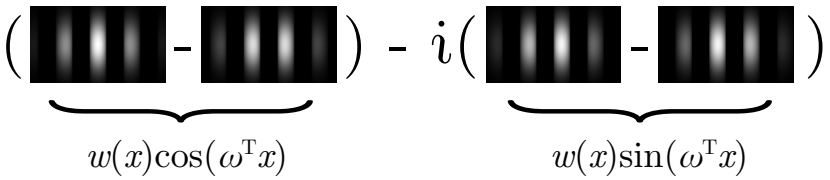


Figure 5.3. Four non-negative real-valued partial patterns are combined into the full complex-valued windowed Fourier basis functions. The complex-valued result is difficult to visualize; we will use real-valued patterns in figures for illustrative purposes in this section. The result is a function with constantly rolling phase, and the windowing function as its magnitude (notice how the patterns fade towards the edge due to the windowing). The 2D frequency of this pattern is $\omega = (4, 0)$: it makes four cycles horizontally, and none vertically.

side effect, the subtractions also cancel out any ambient illumination, as long as it remains constant between the measurements.

Time integration As a practical trick, we display each pattern over a long exposure of a couple of seconds, using only pure black and pure white pixels. Gray values are implemented by turning each pixel on for a precisely controlled amount of time. The reason for this arrangement is that the pixel activation time is easier to control than the non-linear emission from commodity LCD screens. Hence, we may skip an additional delicate photometric calibration stage.

Format of the data We rectify (see Figure 3.1) and crop the region of interest from the input photographs. The input data $z_{i,j}$ is then an $I \times J$ array of complex-valued RGB triplets, where I is the number of pixels in the cropped and rectified images, and J is the number of measured frequencies. Associated with each surface point is also a coordinate p_i in world space. This is obtained from a separate calibration step (Appendix I.A).

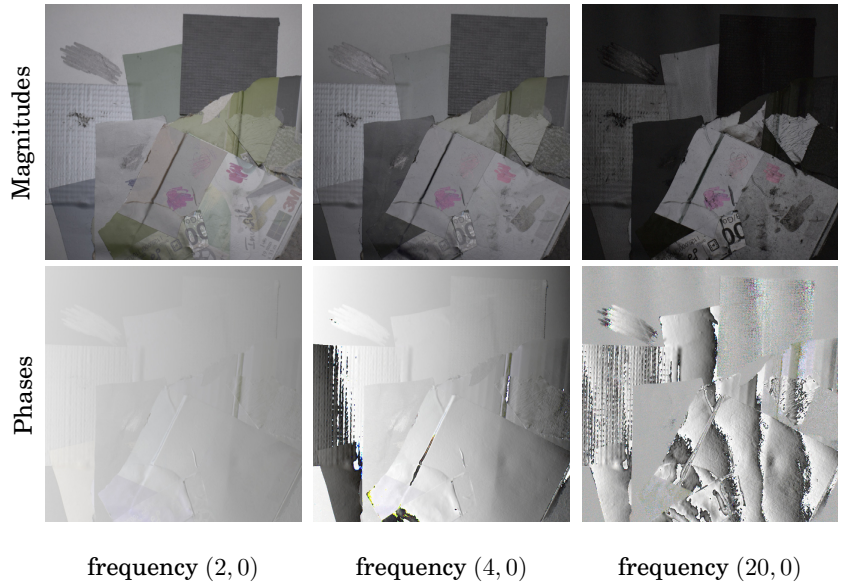


Figure 5.4. The magnitudes and phases of three rectified complex-valued input images shot under different illumination patterns. Roughly speaking, the high frequencies isolate the specular component of the reflections, and the phase absorbs the normal variation of the surface. Notice how the magnitude response of the white paper (top, and left of center in the images) has faded out in the highest-frequency measurement. In contrast, the extremely glossy packaging tape (below center) has a strong response to the pattern. Other materials in the sample (various types of cardboard, tape and pencil markings) fall between these two extremes. The surface normal variations are expressed as distortions in the phases.

We make the measurements at x-axis frequencies $(1, 0)$, $(2, 0)$, $(3, 0)$, $(4, 0)$, $(6, 0)$, $(8, 0)$, $(10, 0)$, $(20, 0)$, and $(40, 0)$, the corresponding sequence on the y-axis, $(0, 1)$, $(0, 2)$, ..., and on the two diagonals, $(1, 1)$, $(2, 2)$, ... and $(1, -1)$, $(2, -2)$, ... Additionally, we make a measurement at the zero (DC) frequency $(0, 0)$, i.e. just the window function without a frequency pattern. The set is heuristically chosen so as to cover a wide range of frequencies and orientations.

Visualizing the input data This complex-valued image data can be visualized by evaluating the magnitude and phase angle of the complex number at each pixel. Figure 5.4 shows such plots for a few different frequencies of a dataset. The magnitudes indicate the strength of the response at a pixel to a given pattern. Intuitively, if the pattern were to be shifted across the monitor over time, the magnitude would reveal how much the intensity of the pixel would oscillate in response. The phases, in turn, express the "position" in the wave that the reflection originated from.

High-frequency patterns tend to isolate the glossy parts of the material. The intuitive reason for this is that diffuse and low-gloss materials "blur" their reflections strongly, and consequently any rapidly varying patterns in the illumination environment become diminished. In contrast, a mirror would replicate the pattern perfectly, no matter what frequency, and hence show strong response also at the high frequencies.

The phases tend to absorb the effect of normal variations, as they indicate the position on the monitor that was seen by the reflection. These effects are discussed in more depth in Section I.4.4 in the publication.

These observations were the original inspiration for research into the method. Unfortunately, these heuristic considerations cannot be used to extract the reflectance information from general surfaces. The main problem is that the simultaneous presence of the diffuse and specular components results in arbitrary distortions in the magnitudes and phases, particularly at low frequencies. Disentangling them properly requires the use of data analysis techniques discussed in Chapter 3. Note that if we did make measurements with readily separated diffuse and specular components, the magnitudes and phases could be used to infer much of the material properties directly. Indeed, the approach used by Ghosh et al. [49] (who use polarization for the separation) is based on somewhat similar reasoning in the context of spherical harmonics.

5.1.2 Image formation model

Let us examine the properties of these measurements, with the goal of building a predictive forward model for use in data fitting. The detailed derivation and formulas can be found in Section I.4.

The use of Fourier basis functions leads one to expect that the measurements are closely related to the Fourier transform of the BRDF. Below, we show that this is indeed the case: at each pixel, they are point samples of the Fourier transform of a slice of the BRDF at the corresponding surface point, projected onto the monitor plane.

The image formation model is in principle simple: the surface points are illuminated by a near-field area light source with a spatially varying emission pattern. A standard change of variables for the reflection equation (Eq. 2.6) predicts the reflected radiance from a surface point $p \in \mathbb{R}^3$ towards the camera at $e \in \mathbb{R}^3$ (i.e. the value of the pixel that sees p) is:

$$L(p \to e) = \int_{\mathbb{R}^2} b_\omega(x)\rho(p, x \to e; u)E(x \to p)G(x)\mathrm{d}x \qquad (5.2)$$

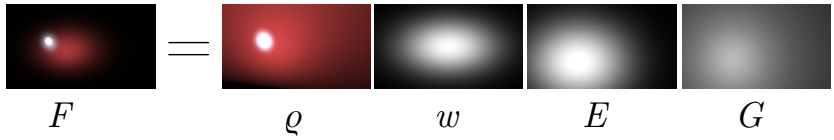


Figure 5.5. The plane-projected BRDF F at some fixed surface point is a product of four terms, expressed in the coordinates of the monitor: a slice of the BRDF itself, the windowing function, the monitor angular emission, and the geometric terms.

The integral is taken over the surface of the monitor plane. Here, $b_\omega(x)$ is the emission pattern described in the previous section, and $\rho(p, x \to e; u)$ is the BRDF at p (to reduce notational clutter, we use x and e to denote unit vectors towards the emitter point and the camera in the BRDF arguments). We assume that it is described by a set of parameters u. For convenience, we also assume that it has absorbed the cosine term from the rendering equation. The term $G(x)$ contains known geometric transformation terms related to the hemisphere-to-plane change of domain, as discussed in Section 2.2.1. We also introduce a spatially invariant but angularly varying emission term E, which models the uneven emission of typical LCD monitors. It is obtained in a separate calibration stage (Appendix I.A).

Substituting the emission pattern of Eq. 5.1 into the above formula, we find

$$L_\omega(p \to e) = \int_{\mathbb{R}^2} \exp(-i\omega^{\mathrm{T}} x) \underbrace{\rho(p, x \to e; u) w(x) E(x \to p) G(x)}_{F(x;p,u)} \,\mathrm{d}x \qquad (5.3)$$

See Figure 5.5 for an illustration of the terms in this integral.

Note that this formula now has a very special form: it is simply the Fourier transform of the product of functions jointly denoted by F:

$$L_\omega(p \to e) = \mathcal{F}\{F(x; p, u)\}(\omega) =: \hat{F}(\omega; p, u) \qquad (5.4)$$

Hence, displaying patterns corresponding to different frequencies $\omega \in \mathbb{R}^2$, we obtain point samples of the Fourier transform of F. Figure 5.6 illustrates this idea. Knowledge of F, in turn, provides information about the slice of the unknown BRDF ρ, as all other functions constituting F are known. This enables us to fit a parametric BRDF model to it.

The image formation process can also be understood via reversibility of light transport. If we were to replace the camera sensor pixel by a "laser" that illuminates a surface point, and the monitor by a diffuse plane, we

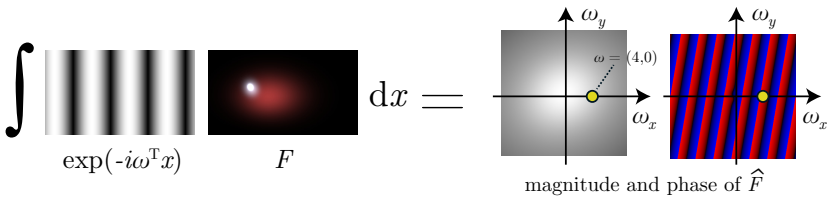


Figure 5.6. Measuring the inner product of the Fourier basis function (here corresponding to the frequency $\omega = (4, 0)$) and the BRDF slice F is equivalent to measuring the Fourier transform of F at position $(4, 0)$ in the frequency plane (shown as the yellow dot). Here, $\hat{F}$ is illustrated using the magnitude and phase.

would find that the light reflected off the surface point forms a similar BRDF slice as predicted by F onto the diffuser. See Figure 5.7 for an illustration of this principle.[1]

5.2 The inverse problem

In terms of recovering the unknown BRDFs, the first idea might be to invert the Fourier transform of the measurements, so as to obtain a primal domain slice of the BRDF, to which a parametric model could be fitted using classical techniques. Unfortunately, this cannot be done: the sampling is highly incomplete, as the value of the FT is only known at a sparse set of frequencies. Assuming a sparse spectrum with zero values outside the sampled locations results in a highly oscillating signal with key features lost (see Fig. I.3 in the publication). Naive interpolation attempts between the sample points are also likely to fail, as the behavior of functions in Fourier domain tends to be complicated. The missing frequency content should be filled with data that is consistent with the frequency content of typical BRDFs — but this behavior is difficult to characterize, other than by explicitly evaluating Fourier transforms of actual BRDF models.

Indeed, at this point a more natural approach would be to keep the data in the frequency domain, and instead compute the Fourier transform for the model predictions. The data fitting can then be performed directly in the Fourier domain. This is the approach we adopt. The main diffi-

[1]These ideas can be made precise by a well known transformation of the rendering equation: instead of light being emitted from the light sources, we may consider so-called *importance* [125, 22] as being emitted from the camera. Importance behaves exactly like light, and we may consider it to represent "vision rays". Conversely, light emitters act as "sensors" for importance.

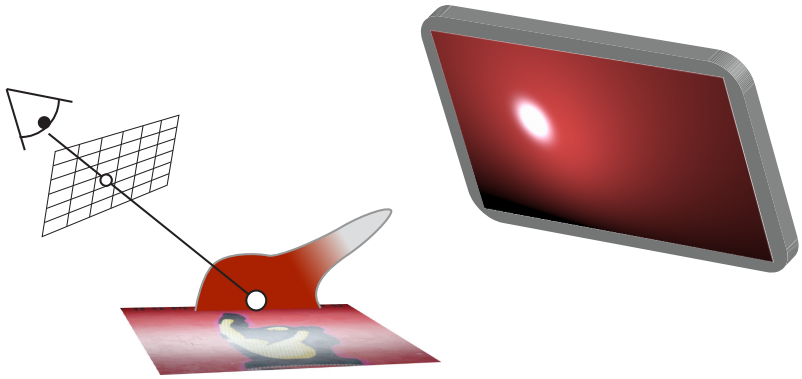


Figure 5.7. The image formation model from the reversed light transport viewpoint. The camera emits "vision rays" through the pixel under consideration. The ray hits the surface and becomes reflected according to the local BRDF. A slice of the BRDF is projected onto the monitor, forming what is essentially the function F. We make measurements of this slice by displaying basis function patterns on the monitor. Notice how the projected BRDF depends on the BRDF on the surface. In particular, its colors, intensity and specular highlight size are determined by the albedos and the glossiness of the BRDF. Its position is determined by the surface normal, along with the relative geometric configuration of the camera, sample and the monitor.

culty of lies in evaluating Fourier transforms of BRDF slices (in the plane-projected form of Eq. 5.3). Before tackling this challenge, let us formulate the optimization problem. We employ the framework of prior-guided data fit optimization developed in Sections 3.1 and 3.2. Given the BRDF ρ, the model developed in the previous section predicts the observed Fourier transform value: it is our *forward model*. We assume that the model also holds in the real world, and hence describes our measurements. Our problem is the *inverse problem* of finding, for each pixel, a ρ that predicts the observed values.

We assume that there exists some underlying set of parameters u_i^* that for each of the I pixels describes the true real-world BRDF at that location. The measurements $z_{i,j}$ are assumed to be randomly corrupted predictions of the forward model, i.e. $z_{i,j} \approx \hat{F}(\omega_j; p_i, u_i^*)$. Following the general reasoning in Section 3.1, we seek to solve the following problem to recover these unknown parameters:

$$\underset{u}{\operatorname{argmin}} \sum_{i=1}^{I} \sum_{j=1}^{J} \|\hat{F}(\omega_j; p_i, u_i) - z_{i,j}\|_2^2 + P(u) \tag{5.5}$$

Here, here the first term corresponds to the data fit, and the second term $P(u)$ corresponds to priors, which we will describe later.

The problem is solved by Levenberg-Marquardt optimization (Section 3.2.2). The required Jacobian matrices are computed by finite differences. The

optimization is initialized based on heuristic estimates about the parameters, which can be read directly off the data (Appendix I.D). For performance reasons, the optimization is performed in a sequence of coarse-to-fine stages, with a special spatial upsampling procedure in between (Section I.7.2).

5.2.1 Approximate Fourier transforms of BRDF models

Fourier transforms of most functions do not admit to a closed form expression. This holds for BRDF models as well. While transforms can be evaluated numerically, this approach would be unwieldy for the kind of optimization problems we are looking to solve. In particular, quadrature-based approaches would require a very fine sample point spacing, and consequently a very large number of BRDF and pattern function evaluations even for just evaluating the Fourier transform value at a single surface point and a single frequency. The requirement of computing derivatives for optimization would further exacerbate the computational difficulty. Furthermore, the use of fixed grid patterns would likely result in structured artifacts in the results.

Fortunately, BRDFs in the plane-projected slice form of Eq. 5.3 can be closely approximated by a class of functions that has a simple analytic Fourier transform formula: Gaussian mixture models. Please refer to Section 3.4 for a review of the relevant properties of these functions. Notice in particular the similarity of the slices in Figure 5.5 and the general shape of Gaussian functions in Figures 3.8 and 3.10

Let us see how the function F in Eq. 5.3 could be approximated as a weighted sum of Gaussians. Note that it is a product of four terms. Because Gaussians are closed under multiplication, approximating each of the terms by Gaussians separately will yield a Gaussian mixture approximation for the entire formula.

Windowing and angular emission terms The windowing function $w(x)$, as described in Section 5.1.1, is already a Gaussian. This is the reason why we chose to use a Gaussian window. The angular emission function E is typically a broad, smoothly falling off function, which is well approximated by a Gaussian. The apparent position and size of the latter varies as a function of the surface point p under consideration; the function scaled and translated accordingly (Section I.4.2).

Specular term The remaining terms — namely, the geometric terms and the BRDF itself — have a somewhat more complex interplay. We split the BRDF into a sum of diffuse and specular components.

We approximate the specular part as a sum of two Gaussians — a wide and a narrow one — centered around the ideal reflection hit point on the monitor, with the surface normal orientation parameter considered. The overall width of the lobe is controlled by a glossiness parameter. The hemisphere-to-plane projection also induces a geometric transformation that stretches the lobe according to the distance and the angle of the incidence to the monitor. We formulate a first-order affine approximation to this transformation, as affine transformations preserve the Gaussianity of a function. The approximation is based on microfacet theory, and hence also reproduces the narrowing of the reflection lobe depending on the surface incidence angle. In particular, it approximates the Blinn-Phong BRDF (Section 2.3.2).

We use a sum of two coinciding Gaussians for the specular term to simulate a commonly observed effect, where the specular lobe is highly peaked around the center, but falls of slowly at the outer edges. This behavior is not well captured by a single Gaussian (nor by many classical parametric models, which very closely resemble Gaussians; recent research has identified this problem and proposed models with heavier tail falloff [84]). The width ratio of the two Gaussians is controlled by a parameter we call *kurtosis*.

The intensity and color of the lobes is controlled by and albedo parameter. The intensity is also modulated based on geometric considerations. For simplicity and computational performance reasons, we drop the geometric terms $G(x)$ from the specular term, as their main intended effect — preservation of measure in the hemisphere-to-plane projection (see Figure 2.6) — is readily achieved by simply using normalized Gaussians as the lobe components.

The details of the specular model are discussed in Section I.5.2 and Appendix I.C.

Diffuse term The diffuse component is simply the cosine term absorbed from the rendering equation, times an albedo multiplier parameter. The orientation of this lobe depends on the surface normal parameter. This component turns out to possess a favorable structure in the plane parameterization: it decomposes into a product of a smooth lobe and an affine function. We absorb the former into the geometric terms $G(x)$, and ap-

proximate the resulting smooth, peaked lobe by a mixture of three predetermined Gaussians. The position, intensity and scale of this lobe depends on the geometric configuration of the point and the monitor. Somewhat surprisingly, the affine function can be handled analytically: multiplication by an affine function corresponds to directional differentiation in the Fourier domain, and Gaussian functions are easily differentiated, as discussed in Section 3.4.

See Section I.5.1 and Appendix I.B in the publication for the details on the diffuse model and the handling of the geometric terms.

5.3 Priors

The main difficulty in the optimization process stems from the ambiguity between the diffuse and specular components. As the camera is stationary, we cannot use the cue that the diffuse component is invariant to the viewpoint. We also choose not to use polarization for the separation, as this would introduce significant complexity to the acquisition setup. Hence, we base the separation on the apparent shapes of the reflectance components: typical specular components are narrow, whereas typical diffuse components conform to the shape of the cosine lobe. Furthermore, we are aided by chromaticity differences, as the lobes often have different colors.

The separation problem becomes ill-posed at points with very dull specular reflectance, as the appearance of the diffuse and specular lobes then becomes very similar. In the worst case, very wide specularity is randomly interpreted as diffuse in some pixels, resulting in extreme noise in the solution.

It is also generally useful to restrict the range of values the parameters should prefer, as sometimes extremely unnatural values can be used to satisfy the data fit term. These solutions are highly unlikely to correspond to the underlying reality.

To discourage this behavior, we use a combination of pointwise priors and smoothness constraints. The former are relatively standard: for each variable, we state a preferred value and a spread around it.

The input data contains significant clues about the positions of abrupt edges in the albedo maps. On the other hand, regions with low variation in the input data are unlikely to contain abrupt jumps in the underlying explanation. Based on these observations, we use data-derived spatially

varying weights for the smoothness priors.

Finally, we enforce the integrability of the normal map by a separate prior that penalizes the curl of the vector field.

The details of the priors are discussed in Section I.7 and Appendix I.E.

5.4 Results and discussion

Implementation We implemented the computations in Matlab, and performed the capture using a relatively high-quality LCD monitor and a high-end Canon SLR. We also tested the method for one dataset using a laptop monitor. The details are discussed in Section I.8.

Test cases and results The method was tested on a variety of surfaces, illustrated in Figure I.6. Figure I.7 in the publication shows the solved parameter maps for each of the captured datasets. Figure I.9 shows a set of novel-angle photographs of the surfaces, with our synthetic renderings made under similar lighting and viewing conditions. The project webpage[2] contains a more complete set of results, including videos of the novel-angle comparisons. An additional result from the method (captured using somewhat different hardware) is also shown in Figure 1.6.

As demonstrated by the results, the captured material descriptors successfully reproduce a wide range of reflectance effects in the datasets. Surface normal variation plays a significant role in all of them. The samples contain a variety of degrees of glossiness and specularity. In particular, the *Crumpled* dataset, which contains a low-gloss white piece of paper overlaid with extremely glossy clear tape, demonstrates two extremes at once. Similar extremes can be seen in the *Mix* dataset, which contains several pieces of paper, cardboard and tape, with wear and tear and pencil markings.

Degrees of glossiness The method is particularly well suited for capturing the extreme gloss in mirror-like reflections. Classical methods based on point sampling struggle with such materials, as the angular space would need to be sampled at an extremely fine resolution. These methods also generally need high dynamic range bracketing [31], as the apparent intensities of reflections from point sources can vary greatly.

The success with highly glossy materials can be seen as a result of the tendency of Fourier transform to reverse the roles of narrow and broad

[2] https://mediatech.aalto.fi/publications/graphics/FourierSVBRDF/

functions. The cost of this is that we sometimes suffer from an opposite problem: very wide specular lobes can be difficult to distinguish from the diffuse component. This is also related to the fact that the monitor only subtends a limited solid angle above each surface point. Some variations in the specular albedo and glossiness in e.g. the *Crumpled* dataset are most likely mathematical artifacts resulting from this ambiguity. In practice, their visual effect in re-renderings is small, but we cannot claim exact photometric accuracy.

The Gaussian approximation framework would in fact readily support certain other types of patterns as well — in particular, localized Gaussians, optionally modulated by plane waves (i.e. Gabor functions [82]). Such hybrid measurements could be used in addition to the frequency measurements to resolve some of these ambiguities.

It should be noted that certain other kinds of variations of the patterns would not be helpful in our problem. In particular, colored patterns (as opposed to the monochrome patterns we use) would merely multiply the entire measurement value by the known pattern color, which would then be trivially divided out during the optimization. Hence, no new information would be revealed.

Angular effects The method gains its efficiency by focusing on some of the most commonly occurring materials: dielectrics that are well modelled by a combination of a diffuse and specular lobe, the characteristics of which can be inferred from a single viewing angle slice. Materials with complex angular behavior may not generalize correctly — the goal with respect to these is merely plausible generalization. In particular, the measurements contain little evidence about the behavior of the Fresnel term. We simply assume a reasonable default behavior for this component.

Failure cases Figure I.11 demonstrates failures resulting from violations of the model assumptions. As we do not support anisotropy, the capture of the brushed metal tray results in a graceful isotropic approximation of the specularity. The fabric is an example of a material that is generally difficult to model using the typical diffuse-specular-normals model. In the case of our algorithm, the strong self-shadowing of the three-dimensional structure of the surface has given rise to spurious high-frequency content, which the optimizer incorrectly interprets as specularity.

Priors The priors, while useful and necessary for reliable capture, are also somewhat heuristic. Their role is also mixed: in addition to enforc-

ing desired qualities of the solution, they guide the convergence of the optimization in a way that prevents the formation of certain types of artifacts related to the non-convexity of the problem. In particular, without smoothness priors, low-gloss surfaces may sometimes become split into distinct regions where a "specular explanation" and a "diffuse explanation" dominate, respectively. Sometimes this gives rise to a "frontline" between the regions that moves as the iteration proceeds. The priors effectively prevent the formation of such artifacts, but this also requires them to be quite strong — possibly exceedingly so for some materials. It would be interesting to explore some alternative ideas — in particular, the preconditioning ideas we discuss in connection with Publication III in Section 6.3.4 might suggest more gentle ways for guiding the convergence (obviously the details would differ significantly).[3]

Design choices The ideas could also be extended to other physical setups. For example, addition of a second or even multiple cameras, shooting measurement simultaneously, could eliminate much of the need for the priors due to the easier diffuse-specular separation problem, and provide clues about the presently missing angular information.

The geometric calibration procedure presented in the paper is relatively simple, but in practice requires the monitor and the sample to fit within the captured photographs simultaneously. This is wasteful in terms of the capture resolution, as the full image area cannot be devoted to the material sample. Alternative calibration procedures could probably be devised. Presently it is not clear how sensitive the method is to errors in both the geometric calibration, and the calibration of the monitor emission pattern. Similarly, the effects of many other configuration parameters have not been explored thoroughly. For example, the set of measured frequencies is somewhat heuristically chosen.

[3] In fact, we have found some success in experimenting (in the context of another algorithm) with the idea of using a different trust region for the Levenberg-Marquardt algorithm. Instead of the usual addition of the λD term in Eq. 3.37, we can use a term $\lambda(D + \kappa\nabla)$, where ∇ is a finite difference matrix and κ is a weight. The idea is to make the *optimization steps themselves* spatially smooth. While we have not studied this procedure in detail, we have tentatively found that it results in a "smooth-to-fine" convergence, where the optimizer first finds a blurry solution and gradually fills in fine detail as the weight λ is driven down. The advantage over smoothness priors is that the expected solution itself is not biased towards smoothness. This might prove useful in context of the algorithm in Publication I, and Publication II as well.

Performance The solver takes roughly two hours per dataset in our unoptimized Matlab implementation. Tentative experiments with a C++-based solver suggest that $10\times$, or even $100\times$ faster solution times might be achievable with reasonable optimization efforts and possible use of GPU acceleration.

6. Stationary materials

A significant portion — arguably the majority — of real-world materials exhibit significant repetition: large swathes of objects are covered by variations of essentially the same piece of material. Man-made objects in particular are often covered "by the metre" by materials such as fabric, wood, stone or leather. These surfaces often look uniform from afar — the only spatial variation occurs below a small characteristic scale.

When the characteristic scale is microscopic, the material can be modeled by a homogeneous microfacet-based BRDF: spatial variation exists but it is invisible at the macroscopic scale. Its aggregate effect is completely summarized in the angular variation encoded in the BRDF, and the underlying spatial features need not be modeled explicitly. Indeed, doing so would be prohibitively expensive.

At the other extreme — implicitly assumed by most spatially varying appearance capture methods — the material varies globally (on *macroscopic* scale), and individual features in the material are unique. An example of this is the wrapping paper in Figure 1.6. For example, even full knowledge of the bottom half of the paper would yield little clue about the specific content of the top half.

In between these two extremes is a very common case, where the characteristic scale is large enough to cause visible spatial variation, but small relative to the size of the object itself. This domain is sometimes referred to as *meso-scale*. Consider a typical leather sofa, as seen in Figure 6.1: ignoring large-scale geometric shape, the surface material consists of tens of thousands of small bumps, few millimeters in size, separated by small ridges. This structure is well modeled by spatially varying BRDF (with surface normal variation included). However, due to the spatial repetition, the full SVBRDF is highly redundant: any few-centimeter example patch (*exemplar*) suffices to describe the material almost entirely, as

Figure 6.1. A typical man-made object covered by a stationary material. While the macroscopic shape of the sofa varies, its surface material is essentially the same everywhere. The only spatial variation in the material occurs within the scale of a few millimeters. A small local neighborhood (e.g. as marked by the rectangle) suffices to describe the key features of the entire material.

the remainder can be seen as random variations of the same piece. In other words, each neighborhood has the same material, *up to some textural permutation* (characterizing which turns out to be non-trivial). From an appearance modeling standpoint, one is rarely interested in the precise placement of the individual bumps, as long as their overall statistical appearance is faithfully reproduced. We call this type of repetition *stationarity*, or *texturedness*. [57, 71, 104]

The above observation is the underlying idea of *texture synthesis*: a variety of powerful synthesis methods exist for amplifying a small exemplar into an extended seamlessly repeating texture. The techniques can be applied to RGB images as well as SVBRDF maps; however, the exemplar must first be authored or captured by other means.

Publications II and III apply these ideas in an unusual fashion to aid appearance *capture*. Their scope is limited to stationary materials. In exchange, they offer a dramatically simplified capturing pipeline, reconstructing detailed SVBRDF representations from only two photographs, and a single photograph, respectively. Both methods use a head-on flash photograph as an input, as shown in Figure 6.2. The underlying idea is simple: the photograph contains hundreds of repetitions of what is essentially the same piece of material. However, due to the near-field flash configuration, each of these pieces is illuminated from a slightly different direction. Recall that this is exactly what is needed in appearance capture: the solution material will be an SVBRDF that reproduces each of these appearances under corresponding illuminations. Figure 6.2 illustrates the various different appearances observed in a single flash image.

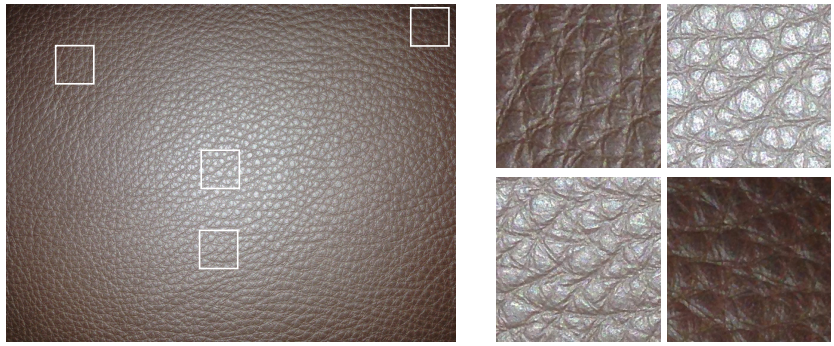


Figure 6.2. A head-on flash image of a stationary leather material. On the right are four blow-ups of local regions of the photograph. Notice how each piece shows the same material under a different lighting condition. However, the pieces cannot be used directly for SVBRDF fitting, because the textural features (i.e. the bumps and the ridges of the leather) do not coincide across the tiles.

The challenge, however, lies in the qualification "*essentially* the same piece of material". The situation is not entirely analogous to having multiple lighting observations of a single, *fixed* piece of material. Were it so, the task of fitting an SVBRDF would be relatively easy: the setup would essentially be equivalent to a spatial gonioreflectometer with a somewhat restricted angular range. However, as described above, the similarity only applies in a textural sense, up to some unknown permutation of the pixels. The two methods take alternative routes in dealing with this problem.

Publication II explicitly undoes the unknown permutation using a separate no-flash *guide image* that reveals the particular spatial arrangement of the textural features in the material. This information is used to rearrange the observed pieces into a synthetic "gonioreflectometer-like" set of observations conductive to direct pointwise BRDF fitting as in Section 3.1.1.

In contrast, Publication III sidesteps the need for explicit undoing of the permutations by adopting a novel approach to SVBRDF fitting itself. Whereas traditional approaches work by essentially reducing the problem to a set of independent BRDF fitting tasks at disjoint pixels (perhaps loosely coupled via priors), our approach drives an entire SVBRDF patch towards *textural similarity* with pieces of the input photograph when rendered. The idea of a textural similarity metric is to compare the appearance of images in a manner that is invariant to textural permutations; in other words, two differently arranged pieces of the same texture should register a high similarity. However, the metrics traditionally used in tex-

ture synthesis have suffered from either low image quality, or unwieldy mathematical formulations. A key component of our approach is the recent neural network based texture descriptor of Gatys et al. [46], which for the first time demonstrated high-quality parametric texture synthesis results via direct gradient-based optimization.

6.1 Texture synthesis

The methods in Publications II and III lean heavily on ideas and methods applied in texture *synthesis*. Let us briefly review some relevant background on the topic.

The typical goal of texture synthesis is to produce new instances of an existing texture exemplar. The most common application is amplification: returning to the leather sofa example above, a texture artist can save a significant amount of time by merely authoring a small, detailed piece of the leather material, and using texture synthesis to seamlessly replicate it to the remaining square meters. Other applications include for example inpainting, where a missing region of a texture is filled in using the information in the surroundings.

On the other hand, quantifying and analyzing texture is an interesting question in itself. Researchers in visual perception have long been interested in texture [57, 71, 6, 88]. Nevertheless, a fully satisfying and practical definition of texture remains elusive.

The underlying idea behind most texture synthesis methods is to extract a feature representation of the input exemplar, and to "forget" the particular spatial arrangement of these features by summarizing them into a set of position-invariant statistics. The synthesis step then consists of generating a novel image with the same feature statistics. The success of this procedure depends on how sensitive the feature representation and its statistical characterization is to visually meaningful patterns in natural images. On the other hand, it also depends on the ability of the synthesis procedure to actually find an image that simultaneously satisfies all of the statistics.

6.1.1 Non-parametric methods

Texture synthesis algorithms can roughly be divided into two classes: *parametric* and *non-parametric* methods.

Non-parametric methods consider the exemplar as a collection of representative pixel neighborhoods, and perform the synthesis by directly copying pixels or continuous patches into the target image. Methods in this category have traditionally produced textures of higher visual quality than the alternatives.

The method of Efros and Leung [40] is a successful non-parametric texture synthesis method based on growing a texture from an initial seed. At each step, a new pixel is copied from the exemplar onto the yet to be filled outer edge of the synthesized image. The pixel is chosen by finding a best match in the exemplar to the already-filled part of the local pixel neighborhood. This greedy procedure is surprisingly successful and produces plausible repetition for a wide range of textures. However, the repetition can sometimes be monotonic: in particular, the method sometimes gets stuck into repeating a tiny part of the exemplar in a regular high-frequency pattern. Efros and Freeman [39] proposed an improvement called Image Quilting, where a larger patch of the exemplar is copied at once. A dynamic programming approach is used to determine a ragged seam that best hides the transition between the old part and the new.

Non-parametric methods can be motivated as Markov Random Field models [77, 47]: an exemplar defines a probability distribution over the occurrence of different pixel neighborhoods. This induces a textural similarity metric, where the distance between two images is the sum of the pixel differences between best-matching neighborhoods.

Non-parametric methods can be seen as heuristics that optimize the value of this type of a metric between the resulting image and the exemplar. Some methods take this viewpoint explicitly [80, 74]. However, given the combinatorial nature of the metric, principled optimization is difficult, and most approaches resort to greedy heuristics that sequentially pick small discrete improvements that yield immediate improvement. Note that, analogous to non-convex continuous optimization, this kind of a procedure is not guaranteed to converge to a global optimum. In fact, finding global optima even for seemingly simple models of this type is difficult, as evidenced by e.g. Ising models. [77, 87] This makes it difficult to adapt these algorithms to other contexts. Indeed, our early attempts at applying these methods to SVBRDF recovery met with little success.

Note however that while these methods are typically formulated in terms of two-dimensional RGB images, they can also be applied on other multi-dimensional multi-channel signals — in particular, SVBRDF maps expressed

as multi-channel 2D images of BRDF parameters.

6.1.2 Parametric methods

Parametric texture synthesis methods are based on summarizing the image features into a set of parametric statistics, as opposed to the direct pixel values.

A very naive example of a parametric method would be to compute the mean and variance of the pixel values in the exemplar, and to output an image of normally distributed noise scaled and shifted according to the measured parameters. Clearly, this model is too weak to reproduce almost any texture of interest.

A more useful textural descriptor is the Fourier modulus. Here, the statistics are the absolute values of the Discrete Fourier Transform of the exemplar. An image with the same statistics can be synthesized by augmenting these values with a random phase, after which an inverse Fast Fourier Transform produces the synthesized texture. [43] The resulting images lack higher-level image features such as edges, but they regardless reproduce some simple textures well.

The method of Heeger and Bergen [61] uses histograms of steerable pyramid coefficients [116] as the statistics. The steerable pyramid is an overcomplete linear basis representation consisting of oriented localized directional derivative -like kernels in a hierarchy of sizes. The statistics are imposed on an initial noise seed image by an iterative procedure. The method often yields a distinct improvement over random phase synthesis. However, the image quality of the results remains clearly inferior to non-parametric methods.

Portilla and Simoncelli [104] take the idea of steerable pyramid coefficient histograms further, by also including various correlation statistics. The synthesis begins from noise and proceeds by iteratively enforcing each kind of statistic in turn. The quality is clearly improved over Heeger and Bergen, but it still lags behind non-parametric methods. Nevertheless, the method can be seen as an interesting experiment regarding the meaning and structure of texture.

The method of Portilla and Simoncelli [104], from the year 2000, was essentially the state of the art in parametric teture synthesis up to 2015, when Gatys et al. [46] introduced a synthesis method based on matching the statistics of convolutional neural network activations. The results of the method rival those of non-parametric methods. Furthermore, in

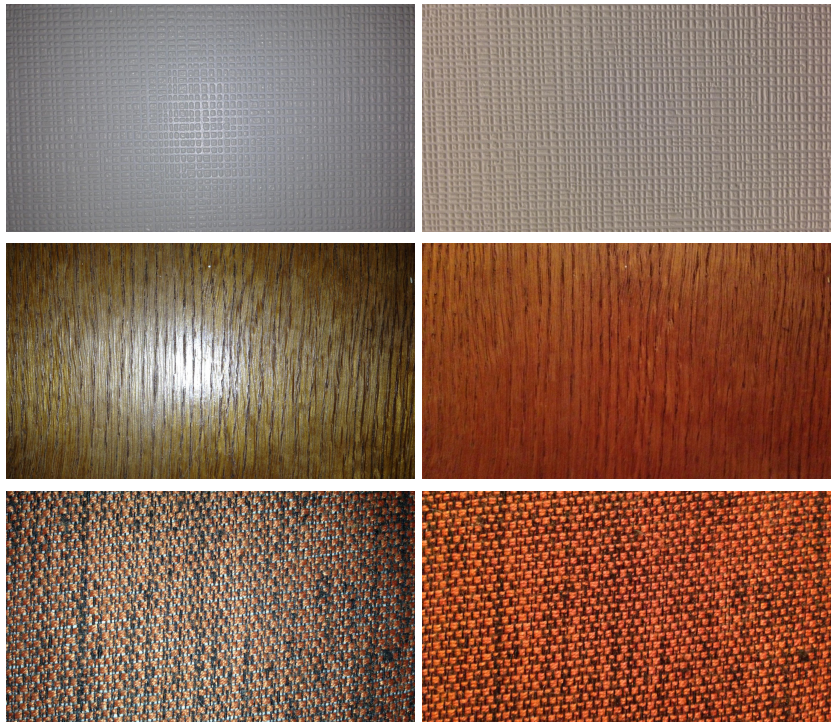


Figure 6.3. Representative input data for the two-shot method.

contrast with most previous work, the synthesis step is based on a well-defined continuous optimization procedure. This opens up possibilities for combining texture synthesis with other data-fitting tasks, which is indeed the approach we take in Publication III. We will review the method in more detail in Section 6.3.1.

6.2 Two-shot method (Publication II)

Publication II presents a method that solves for SVBRDF parameter maps from a pair of photographs: a flash photograph, and a coincident no-flash *guide* photograph. The implementation in the publication uses photographs acquired using a mobile phone camera. Higher-quality cameras could alternatively be used if accuracy is desired at the cost of some convenience. Figure 6.3 shows representative examples of the input photograph pairs.

The input photographs are first split into a regular grid of tiles. The tiles must be large enough as to each contain a representative piece of

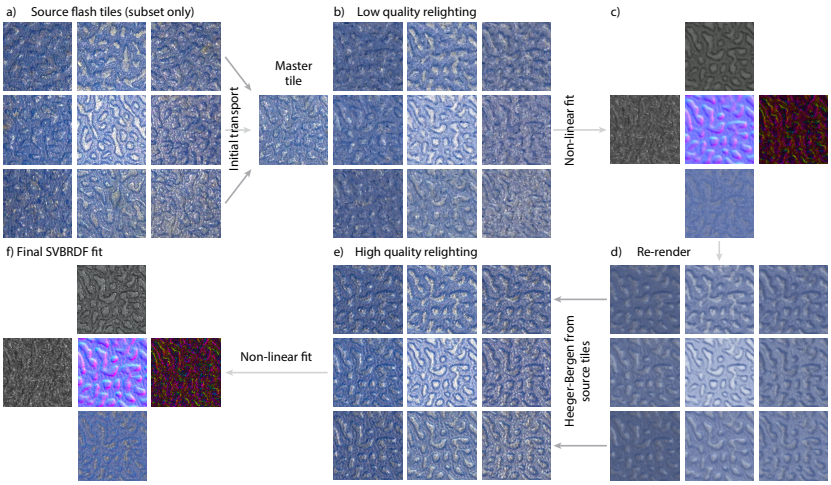


Figure 6.4. Steps of the two-shot algorithm.

the texture, but small enough that the flash lighting is roughly constant within each. The idea is to "summarize" the appearance of all these tiles into a single tile of SVBRDF parameters, in the sense that re-rendering the SVBRDF tile under the same illuminations should reproduce the appearance of the original tiles, up to a textural permutation.

To achieve this, we permute the pixels of each flash photograph tile into a common spatial structure. The guide image is used to find the permutations. The permuted flash tiles are essentially synthetic re-lightings of the same tile under different illuminations (with known illumination directions, due to the simple acquisition geometry). The image stack is analogous to that obtained by a spatial gonioreflectometer, and hence a BRDF can be fitted to each pixel. Finally, the contents of the SVBRDF summary tile are copied back across the entire input image.

6.2.1 Algorithm

Practical implementation of these ideas is less than straightforward. In particular, an additional refinement step is needed due to the noisiness of the permutation estimates.

The algorithm can roughly be divided into five steps. Please refer to the publication for the detailed description of the method; below, we review the key ideas and reasoning behind the steps. The intermediate results corresponding to these steps are illustrated in Figure 6.4. A selection of the input flash photograph tiles is shown in Figure 6.4a.

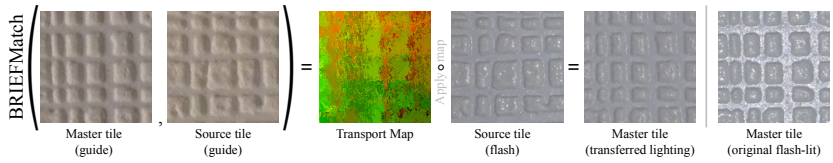


Figure 6.5. The permutation between a source tile and a master tile is found by comparing the corresponding guide images using greedy BRIEF descriptor matches. The result is a "transport map" that describes the closest match for each pixel between the source and the master tile. This map is used to permute the source flash tile pixels into the structure of the master tile. Notice how the result combines the illumination of the source tile with the spatial structure of the master tile.

Initial reflectance transport In the first step, we bring the tiles into a common spatial structure. The key idea is to use the no-flash guide image to find a suitable permutation.

First of all, we need to fix *some* spatial structure to which the flash tiles are permuted. This is easy: because all of the tiles are assumed to represent the texture, we simply choose any one of them as the spatial reference. We call this tile the *master tile*.

The spatial permutations are computed using the guide image. It is taken in a precisely coincident position with the flash photo, and hence any permutation of the guide tiles is also valid for the flash tiles. Furthermore, it is shot under uniform lighting conditions, so that the appearance of the material is uniform across the photo. The key assumption is that local pixel neighborhoods that look identical in the guide photograph also share the same SVBRDF. The permutations are computed by greedily[1] finding the best matching neighborhood between each tile and the master tile, using a feature descriptor [18] that is robust to small discrepancies. Figure 6.5 illustrates a typical pair of tiles and the permutation map between them.

Finally, the pixels of each flash photograph tile are rearranged according to the permutations. The end result is illustrated in Figure 6.4b.

[1]The resulting mapping is not one-to-one, and hence not exactly a permutation; however, we use the term as it describes the general intent of the procedure. We did in fact enforce a convex relaxation of the one-to-one condition at an earlier stage of the project, but found that the benefits did not justify the computational expense of the procedure. Specifically, we solved for a transport map between the tiles using a linear program formulation of the Monge-Kantorovich optimal transport problem [126], with the image feature distances as the pairwise distance matrix. Furthermore, we used an apparently novel spatial continuity prior that favors matches between continuous spatial regions. The problem was solved using the primal-dual proximal method of Chambolle and Pock [19].

Initial reflectance fit Following the permutation, the flash tiles possess the same spatial structure. Hence, the same pixel position in each tile now corresponds to a light sample of the same BRDF. This allows us to fit a parametric BRDF model (the BRDF Model A of Brady et al. [14], extended with anisotropy) to each surface point. The input data and the optimization task are conceptually similar to the example presented in Section 3.1.1.

Plotting the value of the same pixel in different tiles results in a visualization of the estimated BRDF slice at the surface point. See Figure II.6 for an illustration of these slices, and how subsequent processing steps improve them.

The input data to the fitting step is very noisy due to misalignments and false matches in the permutations. Performing the BRDF fit independently at each point leads to a noisy solution with occasional gross outliers. Hence, we use smoothness priors that essentially enable the neighboring pixels to share data between one another: when an individual pixel does not have sufficient data to determine the BRDF, it may choose a plausible solution that appears to be favored by other pixels in the neighborhood. Additionally, we use pointwise priors to specify plausible values of the parameters, and also enforce the normal map integrability by a prior. The data fit problem is solved by Levenberg-Marquardt optimization using finite differences for the derivatives where needed. The process is reminiscent of the optimization in Publication I.

The result of the fitting is a single tile of SVBRDF parameters, shown in Figure 6.4c. Due to the low quality of the input data, the solution tends to be excessively smooth and low-contrast.

Refinement In the third step, we refine the result of the SVBRDF fitting to restore the original crispness of the input photo. First, we re-render the SVBRDF into a flash-illuminated image stack, like the original tiling. The washed-out appearance of the tiles is apparent in Figure 6.4d. Our strategy is to copy high-quality detail onto these tiles from the correspondingly lit input flash photograph tiles, while retaining the consistent spatial structure we achieved in the previous steps.

We achieve this by applying the texture synthesis algorithm of Heeger and Bergen [61] (See Section 6.1.2) in an unusual fashion. Previous work [95, 98] has demonstrated that the algorithm can be used to transfer high-frequency textural detail from one image onto another, by using the latter as a seed image for the synthesis (as opposed to noise). Figure 6.6 illus-

trates this process. We find that applying the same operation on our tiles often almost perfectly restores the appearance of the originals.

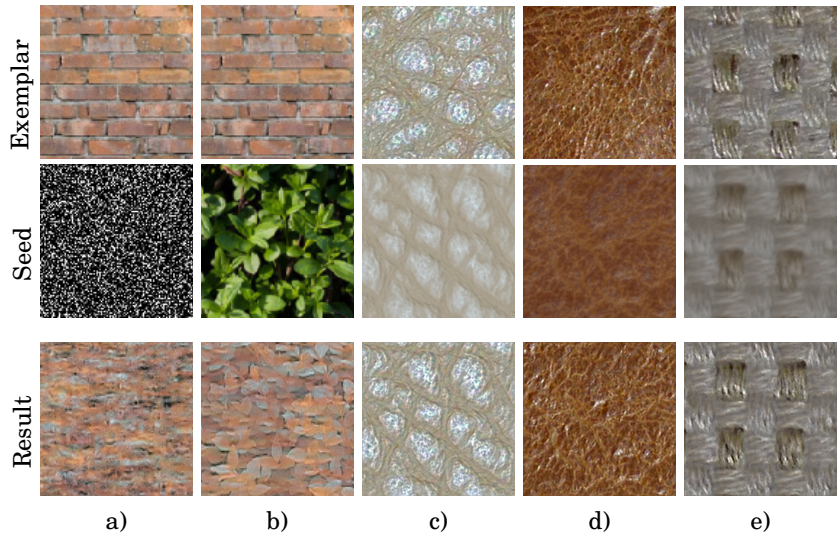


Figure 6.6. Texture synthesis algorithm of Heeger and Bergen [61] applied with different types of seed images. a) Typical use of the algorithm: the steerable pyramid statistics of the exemplar (first row) are iteratively imposed on an initial seed image of noise (second row). The result is a noise with some textural characteristics of the exemplar (third row). b) If an image is used as the seed in place of noise, a hybrid image is produced. The spatial structure of the seed image is combined with the fine details of the exemplar. c-e) In the refinement step, we use the same idea to combine the fine detail from the original flash tiles (first row) with the spatial structure represented by the low-quality relit tiles (second row). Notice that the result image (third row) has inherited the spatial structure of the seed.

The result, shown in Figure 6.4e, is a stack of high-quality images depicting the same piece of material under a variety of lighting conditions.

Final reflectance fit To obtain an SVBRDF corresponding to this stack, we again repeat the fitting procedure of the second step, this time using the refined tiles as the input, and the SVBRDF solution of the second stage as the initial guess. The result is shown in Figure 6.4f.

Reverse reflectance transport The high-quality SVBRDF solution tile from the previous step only covers a small spatial region of the original input photographs. In the final step, we propagate this information to the entire input image. This is done using the same kind of permutations as in the first stage, only computed in the reverse direction. To obtain a smooth solution despite the raggedness of the permutations, we copy not only the SVBRDF values, but their spatial gradients as well. After copying,

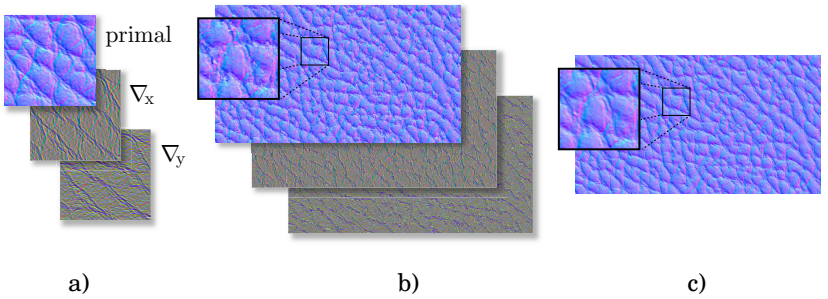


Figure 6.7. The gradient-domain reverse transport procedure visualized for the normal map. The solution SVBRDF, along with its horizontal and vertical finite difference maps (a), is copied to the full-size image using a rough reverse transport map. The result (b) is a spatially extended coarse version of the map and the gradients. Finally, the "primal" map and the gradients are fused together by solving a screened Poisson equation. The result (c) is a high-quality SVBRDF map without visible seams from the transport.

the primal and the derivative values are reconciled by solving a screened Poisson problem [7]. Figure 6.7 illustrates the process. Prior to copying, we also filter the transport map to smoothen the noisy boundaries of the regions, as explained in Section II.4.2.3.

The solution tile could alternatively be used as an exemplar for classical texture synthesis methods such as Image Quilting [39]. In fact, we used this approach before we implemented the reverse transport method described above. It works relatively well, but the texture often becomes somewhat monotonous. The gradient-domain transport idea can be applied for further hiding the seams in Image Quilting as well. To our knowledge, this idea of gradient-domain non-parametric synthesis has not been presented in literature before.

6.2.2 Results and discussion

Implementation We implemented the method in Matlab, and used CUDA to accelerate the brute-force feature matching in the reflectance transport stages. The datasets were captured using the standard camera and flash on an iPhone 5.

Datasets Section II.5 presents the results of the method. We captured a relatively large dataset of 72 flash/no-flash photograph pairs. The full dataset, images of the solved SVBRDF maps, video renderings of the results, and the method source code are available at the project webpage[2].

The captured samples represent a wide range of materials, including

[2] https://mediatech.aalto.fi/publications/graphics/TwoShotSVBRDF/

plastics, paint, leather, wood, paper, metals, fabrics and a few odd ones such as a bowl of seeds. Many of them exhibit significant normal variation, as well as anisotropy. Several datasets violate some of the basic model assumptions, in particular stationarity. We find that the method often tolerates these violations well — see for example Figure II.9, which demonstrates the result from a material with non-stationarities.

We find good results on most of the datasets. Figure II.8 demonstrates a selection of SVBRDF maps solved using the method, along with corresponding pieces of the input data with a re-rendering. Most of the solutions are visually pleasing and appear plausible on an informal examination.

Validation experiments We performed more rigorous evaluations on some of the datasets. In Figure II.10, we compare renderings of the materials side-by-side with photographs of the corresponding physical samples, both in the original as well as novel illumination and viewing conditions. While a good visual agreement is found, the results are not photometrically accurate. This is not surprising, given the fact that the materials were solved from consumer camera phone JPEG images that have gone through an unknown non-linear color processing pipeline.

We conducted two validation expeeriments to examine the effect of the quality of the data on the solutions. In Figure II.13, we demonstrate results from materials captured using a high-quality SLR camera. The general appearance of the solutions is slightly sharper, but qualitatively similar. This experiment also confirms that we have not overfitted our model to the characteristics of the iPhone photographs.

The second experiment was conducted using synthetically rendered data. An SVBRDF was authored in an image editing program, and rendered using our forward model. This rendering was used as an input to the method. Figure II.12 shows that the method recovers an SVBRDF map that matches the known ground truth well. Of course, this experiment is liable to produce overly optimistic results, as the data is free of errors and perfectly conforms to the model assumptions. As a stress test, we introduced various severe distortions (including noise, non-linear color processing, overexposure, stray ambient light, and misalignment of input photographs) into the rendered synthetic data prior to feeding them into the method. The results are robust: while the method obviously has no mechanism for undoing the distortions, it gracefully absorbs them into the solution and preserves the qualitative character of the ground truth.

Finally, to examine how well our solutions correspond to the true physical characteristics of the materials, we used the GelSight scanner [70] to measure sub-micron accurate normal maps for a selection of the samples. We find good qualitative agreement. Figure II.11 shows the results.

Limitations Besides the obvious requirement of stationarity, the limitations of the method are somewhat similar to those of Publication I. We only observe the material from a fixed camera position, and hence measure only a two-dimensional slice of the full four-dimensional angular space. Again, we aim for plausible generalization, and assume reasonable standard behavior for e.g. the Fresnel term. Note however that the Fresnel model is added as a post-process to the optimization. The user is free to specify another Fresnel model, for example if it is known that the material is a metal.

Our observations are made from a limited range of viewing and illumination angles within this slice. These angles are determined by the field of view of the camera (roughly 66 degrees diagonally on the iPhone 5). Specular highlights wider than this opening may be confused with the diffuse component. At the other extreme, we do assume that the appearance of the material is roughly constant within a tile. Hence, specular lobes with an opening of less than a few degrees may not be reliably resolved.

6.3 Neural one-shot method (Publication III)

As discussed, Publication III addresses a similar problem of recovering a representative SVBRDF tile from a flash photo. However, it takes a vastly different approach than Publication II. Instead of using a separate no-flash photograph to undo the permutation of the spatial structures of the tiles, we use a similarity metric that is *invariant* to such permutations.

6.3.1 Neural texture synthesis

The method builds on a recent texture synthesis approach by Gatys et al. [46]. We will review it here. For a review of relevant concepts on convolutional neural networks, see Section 3.5.

As discussed, the methods of Heeger and Bergen [61], and Portilla and Simoncelli [104] use statistics of steerable pyramid decompositions as descriptors of visually salient features in the exemplar. The steerable pyramid essentially decomposes the image into responses of variously sized

and oriented edge-detection filters. This addresses many properties of natural images: their salient features tend to be variously oriented edges, and they exhibit structure at multiple scales. Indeed, mammalian early visual cortices are known to use oriented bandpass filters with similar characteristics [90, 6, 1, 88]. Portilla and Simoncelli in particular explore a variety of statistics related to the decomposition, and identify how they help at e.g. recovering regular structure, or representing directional shading. Despite significant improvements upon Heeger and Bergen's more straightforward approach, the method is only partially successful in terms of image quality typically expected in computer graphics applications.

As noted in Section 3.5, the activations produced by image recognition neural networks are analogous to traditional manually engineered feature descriptors — only, wildly more successful at recognition tasks. This leads one to suspect that they might have wider applicability as well, and this is indeed the case. Razavian et al. [107] demonstrate that the activations beat traditional descriptors by a large margin in tasks such as image retrieval.

Method of Gatys et al. These considerations suggests that neural activations might also be good descriptors for texture synthesis. Indeed, there is some indication that the human visual pathways rely on a hierarchical abstraction of features similar to neural networks [133, 76]. Gatys et al. [46] base their texture synthesis method on this idea.

Their method is based on feeding the exemplar into a pre-trained VGG-19 network, and collecting statistics of the activations from various layers. The statistics are very simple — they are Gram matrices (essentially covariance matrices for non-centered data) of the activations. For l'th layer (having n_l activations) the Gram matrix $G^l \in \mathbb{R}^{n_l \times n_l}$ consists of averages of pointwise products of each pair of activation channels:

$$G^l_{ij} = \mathrm{mean}\{a^l_i \odot a^l_j\}, \tag{6.1}$$

where a^l_k is the k'th activation map of the l'th layer, $\odot$ is the pointwise product, and mean computes the average over the two spatial dimensions of the layer. For example, if a given layer has the spatial resolution of 64×64 and consists of 256 activation channels, the Gram matrix summarizes its activations into 256×256 values (where half the values are redundant due to symmetry). This statistic roughly captures the individual magnitudes and pairwise co-occurrences of the activations, while discarding the information about their particular spatial arrangement.

A full statistical descriptor is obtained by evaluating the Gram matrices for a desired set of convolutional layers and collecting their entries into a vector (optionally with weighting). We denote the full descriptor evaluation function for an image x by $T_G(x)$. Note that the fully-connected layers at the end of the VGG network can be dropped. This also means that the descriptor is not tied to the original resolution (224×224) at which the network was trained, because the activation map size no longer needs to reach 1×1 at the end of the convolutions.

The synthesis procedure calls for finding a novel image that has the same activation statistics. This is implemented by gradient-based continuous optimization of the similarity of the above-defined descriptor:

$$\operatorname*{argmin}_x \|T_G(x) - T_G(y)\| \tag{6.2}$$

where y is the exemplar image and x is the synthesis result. This optimization is performed using the L-BFGS algorithm (see Section 3.2.2).[3] Conveniently, the gradients of this objective function can be evaluated using the same backpropagation machinery that is used for training the networks. Consequently, the method is readily implemented using standard neural network frameworks, and is directly amenable to GPU acceleration.

Figure 6.8 illustrates some results from the method. The results significantly improve upon the state of the art in parametric texture synthesis, reaching a similar level of image quality with non-parametric methods. Interestingly, Gatys et al. also find that an extension to the method can be used to transfer artistic styles from paintings onto photographs [45]. This application is essentially texture synthesis augmented with a content matching term that also uses (non-statistical) neural activations.

6.3.2 Textural data fitting

The fact that the synthesis step of Gatys et al. [46] uses gradient-based optimization is unusual. Almost all other texture synthesis methods are based on heuristic steps; they are essentially hard-coded to perform a given narrow task, and extending them to other contexts is difficult.

This is useful because we are not constrained to merely solving the prob-

[3] Interestingly, practical experiments suggest that basic gradient descent fails to reach the same synthesis quality as L-BFGS, despite of optimizing the same objective function. The method terminates due to failure to make meaningful progress beyond some early stage (i.e. the gradient becomes very small). It is not clear why L-BFGS avoids this problem.

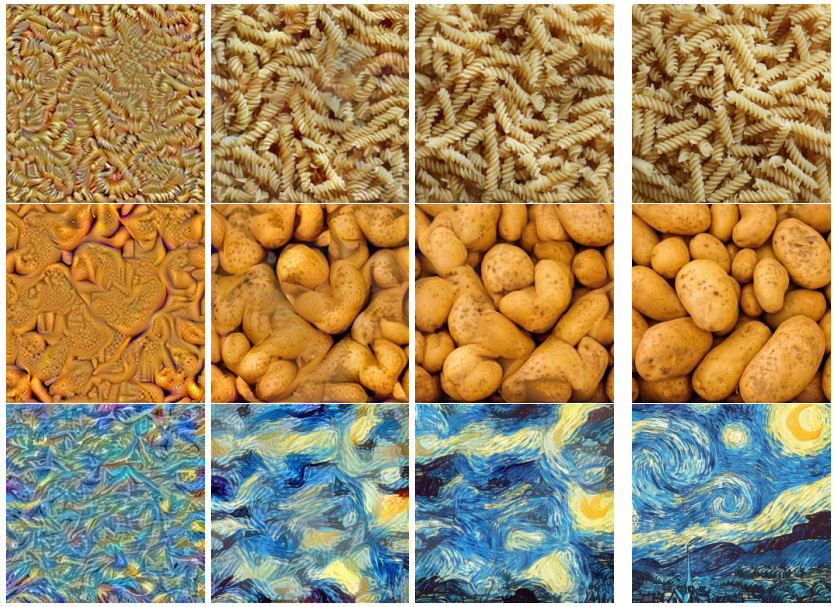


Figure 6.8. Results from the texture synthesis method of Gatys et al. [46] after 20, 100 and 1500 iterations of L-BFGS optimization. For reference, the exemplar is shown on the right. The optimization begins by establishing colors, edges and other low-level features, and gradually builds higher level content out of them. Much of the large-scale content is coarsely established already at around 100 iterations. However, like texture synthesis methods in general, it sometimes struggles with large-scale features. This can be seen in the potatoes, which have not quite found the clean, round shapes present in the exemplar.

lem in Eq. 6.2. The differentiable texture descriptor T_G can be used in a variety of gradient-based optimization tasks where images need to be compared "up to textural permutation".

Recall that this is precisely the capability we were missing in Publication II, which necessitated the manual shuffling of the pixels into a compatible spatial arrangement.[4] Conceptually, Publication II solves the problem

$$\underset{u}{\operatorname{argmin}} \sum_{\mathrm{k}} \|R(u; l_k) - P_{k \to m}(y_k)\| \tag{6.3}$$

where u is the SVBRDF parameter map we are looking to solve for, k loops over the tiles, y_k is a flash photograph tile, and l_k represents its lighting conditions. $R(u; l_k)$ renders the SVBRDF under these lighting

[4]Note that Publication II was submitted prior to the publication of Gatys et al. [46]. In fact, during early stages of development of Publication II, we aimed for a method that would use the flash photograph only, but failed to reach this goal as there existed no high-quality method for comparing textures in a suitable manner. The guide photograph was only then introduced to solve this problem.

conditions. The function $P_{k \to m}$ permutes the data of each tile into some common spatial structure (that of some chosen master tile m).

Data fit term The key idea in Publication III is that we may drop the explicit permutation by considering the permutation-invariant textural difference instead of a pointwise difference:

$$\underset{u}{\operatorname{argmin}} \sum_{\mathrm{k}} \|T_G(R(u; l_k)) - T_G(y_k)\| \tag{6.4}$$

This is essentially the problem we solve in Publication III (aside from some additional terms described in the subsequent sections; see Eq. III.11 for the complete objective). Hence, we are seeking an SVBRDF whose renderings are texturally similar to chosen pieces of the flash photo, as measured by the statistics of neural network activations. Figure III.1 illustrates this principle. Instead of the full regular tiling used in Publication II, we find that matching the appearance at 15 tiles scattered around the specular highlight suffices for solving the problem.

Implementation of the optimization This problem is highly ill-posed and non-convex. It is not clear *a priori* that the optimization procedure should in practice find a solution that satisfies the appearance constraints, nor that the constraints are strict enough as to pinpoint a well-generalizing SVBRDF. We nevertheless find this to be the case — however, certain additional priors and pre-conditioning transformations must be introduced to ensure the success of the process. We will discuss these in subsequent sections.

As noted, the method of Gatys et al. [46] can be implemented using the standard building blocks of neural networks — in particular, the backpropagation algorithm for computing the derivatives of the objective function with respect to the unknown variables. We find that this also applies to our significantly more complex objective function: we can break it down into a directed acyclic graph (DAG) of simple operations, each of which is easy to differentiate in isolation. In particular, the neural network is nothing but a sequence of nodes within this graph. The derivative of the entire objective function can be evaluated by the backpropagation algorithm using the same machinery as in neural network training. Following Gatys et al. [46], we use the L-BFGS algorithm to run the optimization. Figure III.6 shows the computational graph for the full problem. Please refer to Eq. III.12 and the Appendix of Publication III for the details.

Rendering model The rendering operator R is very similar to that in Publication II: we optimize for spatially varying albedos, glossiness and normals, and render the maps under headlight configuration. However, instead of the more advanced anisotropic BRDF model, we restrict the complexity of the problem somewhat by using an isotropic Blinn-Phong BRDF model [9], specified in Eq. III.9.

Pre-processing Gatys et al. [46] pre-process the exemplar textures by subtracting their mean prior to feeding them to the CNN. Once the optimization is complete, the mean is added back onto the result. Our experiments confirm that this significantly improves the synthesis results. The reasoning is that the original VGG-19 was trained using roughly zero-mean images, and hence its activations are tuned to represent features of such images.

For the same reason, we individually subtract the mean off each of the input tiles y_k, and perform the same subtraction on the rendered tiles during the optimization. Furthermore, we normalize the contrast of each tile to the range of a typical training photo. Besides better matching the network's expected input, this normalization also roughly equalizes the contributions of each tile to the data fit term, resulting in better convergence behavior. Note that this does not bias the expected result of the optimization (e.g. by leading to a "whitened" appearance of the solution), because the same *fixed* set of transformations is applied to both the input data and the predictions, and used throughout the optimization. The practical implementation of these operations is described in Section III.4.2.

6.3.3 Stationarity priors

In practice, optimizing the data fit described in the previous section leads to uneven results due to the non-convexity and ill-posedness of the problem.

Generally speaking, the issue of non-convexity in neural networks is subtle. While it has long been known that neural networks can in principle model extremely complicated functions efficiently, until recent years it was thought that finding good parameters for large-scale networks was essentially impossible due to the extreme non-convexity of the training problem. The success of neural networks in recent years has demonstrated this to be, surprisingly, false. The non-convexity is there, but it appears to manifest mostly as symmetries (that is, arbitrary choices of

Non-stationary initial guess | Non-stationary result | Stationary initial guess | Stationary result

Figure 6.9. On the left is a non-stationary initial guess for the method of Gatys et al. [46], and a resulting non-stationary synthesis. On the right we have used a stationary noise as the initial guess. The objective function value is the same for each case, indicating that the textural similarity metric T_G has no means to detect and eliminate the forcibly introduced non-stationarity.

permutation regarding which activation represents which feature). Once the symmetry is randomly broken, the training tends to converge robustly to a near-global minimum. [29, 55, 75]

Gatys et al. [46] appear to inherit this property: their objective function is extremely non-convex, to the extent that one might expect the optimization to often become stuck at a poor solution. Regardless, we observe very robust results in practice. In particular, the synthesis results tend to be stationary, in the sense that the textural features become evenly distributed across the synthesized image, and that the solution quality does not vary spatially. This desirable behavior is not guaranteed: for example, contrived initial guesses can lead to non-stationary results, as in Figure 6.9. Nevertheless, stationarity emerges robustly enough, and no specific mechanism is used to enforce it.

Unfortunately, this state of affairs is broken by the insertion of the rendering operator R (which is non-convex and non-injective) and the presence of multiple simultaneous matching targets in Eq. 6.4. We found that solving the problem directly by optimization leads to poor results. The major problem is that the result is often non-stationary even within the small SVBRDF tile we solve for. It often appears that one region of the map is well-developed (or even "over-developed" as in exhibiting excessive contrast) while others are left washed out or suffer from visual artifacts. The optimizer apparently finds that it can quickly improve the objective value at an early stage by developing individual regions within the tile. It later fails to connect these regions into a coherent and stationary whole. The convergence of Gatys et al. [46], in contrast, tends to be spatially even: similar features emerge concurrently across the image as the optimization proceeds. Indeed, if non-stationarities are introduced during the

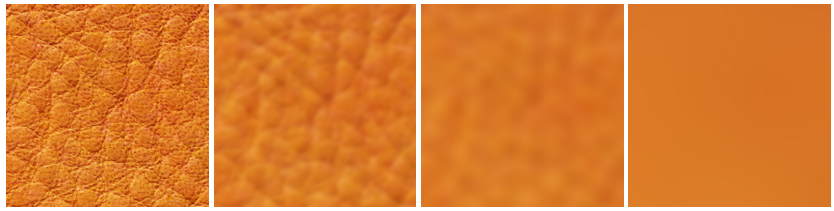


Figure 6.10. A sequence of images of the same tile, blurred with increasingly large blur kernels. Beyond the characteristic scale of the texture, the local averaging destroys all textural detail, and only a constant-colored image remains.

optimization, they may persist; the key to obtaining a stationary solution seems to be to never let non-stationarity appear in the first place.

Based on these considerations, we introduce a *stationarity prior*, which to our knowledge is novel in literature. Additionally, we introduce a complementary preconditioning procedure that enables the optimizer to efficiently explore solutions that are consistent with the prior. Let us first discuss the prior.

Characteristic scales of spatial variation As discussed in the introduction to this chapter, a central notion behind the methods is that of characteristic scale: the size of any spatial features on the surface should not exceed some length scale s. The tiles are chosen to be a few times larger than s, so as to capture some variety.

Another way to formulate the principle is that measuring any image statistic over a neighborhood of size s should produce the same result, regardless of the location of the neighborhood. In particular, measuring the local mean over each neighborhood of size s corresponds to computing the average value of the pixels over each s-sized region in a sliding window manner. Plotting these average values at the region centers, we simply obtain a blurred version of the image. Indeed, the operation is nothing else than the convolution between the original image and a circular filter kernel. If the image is stationary, all features in it vanish beyond some filter size. This is the characteristic scale. Figure 6.10 illustrates this effect.

A practical stationarity prior This suggests a practical way to detect and penalize non-stationarity within the tile: measure a set of key statistics of every neighborhood of some fixed size (by a sliding window), and require that the the statistics agree at every point. In particular, we require the stationarity of the mean, variance, inter-variable correlations, skewness, and kurtosis of the SVBRDF maps.

Recalling the example above, we penalize non-stationarity of the mean by penalizing non-constancy in a blurred version of the SVBRDF maps. To penalize non-stationarity of variance, we first center the parameter maps by subtracting the mean, and raise the result to the second power. Spatial averages over this quantity correspond to local measurements of variance. Again, we penalize deviations from constancy. A similar procedure, for whitened images raised to third and fourth powers, reveals non-stationarities in skewness and kurtosis, respectively. Finally, non-stationarities in correlations between variables are penalized by requiring stationarity of their pointwise products (after whitening).

These filtering operations, and consequently the entire priors, are effectively implemented using the Fast Fourier Transform. Indeed, non-stationarity is directly visible as low-frequency content in the variously processed maps. Section III.4.3 presents the priors from this viewpoint, and describes them in more detail.

6.3.4 Preconditioning

With the addition of the stationarity prior Q (and a weight λ), our optimization problem is now

$$\underset{u}{\operatorname{argmin}} \sum_{\mathrm{k}} \|T_G(R(u; l_k)) - T_G(y_k)\| + \lambda Q(u) \tag{6.5}$$

In practice, the optimizer struggles to find steps that improve the textural data fit objective while satisfying the stationarity priors. These two terms in the objective function often suggest mutually incompatible steps. Recall that in gradient descent the step direction is simply the sum of the (negative) gradients of each term in the objective function:

$$-\sum_{\mathrm{k}} \frac{\partial \|T_G(R(u; l_k)) - T_G(y_k)\|}{\partial u} - \frac{\partial \lambda Q(u)}{\partial u} \tag{6.6}$$

A "step direction" in this context is simply an update to the pixel values in the SVBRDF map, i.e. of the same format as the SVBRDF itself. Often, the data fit term suggests the kind of greedy local updates outlined in the introduction to the previous section. These steps violate the stationarity priors, unless a very short step length is chosen. The progress grinds to a sequence of short back-and-forth steps that very slowly lead towards mutually agreeable solutions. The situation is somewhat analogous to the example in Figure 3.3. The use of the L-BFGS algorithm in place of vanilla gradient descent improves the convergence, but only to an extent.

Note that the converse does not hold. The data fit term would *not* object to stationary steps in general — it merely finds the greedy local steps to be slightly more appealing in the short term. If we had a way to "propose" stationary steps, it would go along. This section presents a principled mechanism for precisely this purpose. By bringing the stationary directions to the forefront via a process of *preconditioning*, we aid the optimizer at discovering them. This leads to a significantly improved convergence behavior without biasing the solution.

Fourier domain preconditioner To facilitate efficient optimization within the space of stationary SVBRDF maps, we introduce a Fourier domain preconditioning procedure. In the standard ("primal") representation of the optimization problem, each optimization variable controls a single BRDF parameter at a single surface point. Instead, we optimize for coefficients of the Discrete Fourier Transforms of the SVBRDF maps. In this parameterization the variables control magnitudes of plane waves, which at high enough frequencies are by definition stationary. Updating such variables is more likely to retain the stationarity of the current SVBRDF estimate, and to satisfy the stationarity prior.

In practice, we transform the variables as follows. Instead of directly optimizing over the SVBRDF parameters u, we optimize for a set of preconditioned parameters $\tilde{u}$. The actual SVBRDF parameters u are computed from $\tilde{u}$ as $u = P(\tilde{u})$. Here, P is a preconditioner that performs an inverse Fast Fourier Transform (with some weightings we describe below). The optimization task of Eq. 6.5 then becomes:

$$\underset{\tilde{u}}{\operatorname{argmin}} \sum_{\mathrm{k}} \|T_G(R(P(\tilde{u}); l_k)) - T_G(y_k)\| + \lambda Q(P(\tilde{u})) \tag{6.7}$$

To discourage the use of the non-stationary low frequencies in optimization steps, we include a downweighting for these frequencies in P prior to evaluating the IFFT. While combinations of high-frequency plane waves may still cause higher-order non-stationarities, the preconditioning eliminates the majority of poor step proposals. What remains is kept in check by the stationarity prior itself.

Notice that this procedure is directly analogous to the toy example of preconditioning discussed in Section 3.2.4. The effect of the per-frequency scaling is a high-dimensional analogue to the simple "squeezing" illustrated in Figure 3.7.

Per-frequency weightings from input data We also include in the preconditioner a heuristic per-frequency weighting by the average frequency

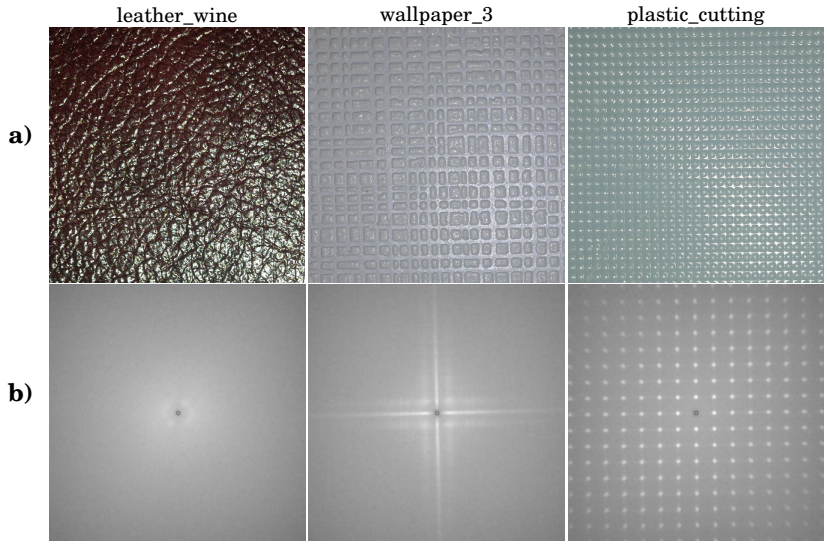


Figure 6.11. Examples of per-frequency weights used for different datasets. Note in particular the strong downweighting of the very lowest frequencies (center of the spectrum). The spectral weightings (b) are computed from the average local spectral magnitude content of the input photographs (a). The regular structure in *wallpaper_3* and *plastic_cutting* is clearly visible as peaks at individual frequencies. This encourages the optimizer to seek steps that reproduce similar regular content in the SVBRDF maps.

content of the input flash photo. See Figure 6.11 for an illustration. While we cannot reliably tell *a priori* which SVBRDF parameters are responsible for the spatial variation in the illuminated appearance of the flash photograph (solving for this information is the problem in the first place), it does nevertheless provide clues about where *some kind of* variation is taking place.

Observe that the underlying idea of heuristically taking advantage of hints in the input data is similar to that applied in spatially varying smoothness prior weights of Publication I (Section 5.3). However, the latter approach is more dangerous, as it more directly attributes effects to individual features of individual variables. In contrast, preconditioning does not *require* the optimizer to converge towards a solution with the same frequency content; at worst, it slows down the convergence. In practice the weighting often significantly accelerates the convergence, and is particularly helpful for materials that exhibit regularly repeating structure.

Discussion We stated above that preconditioning merely changes the convergence behavior, without changing the expected solution. This is

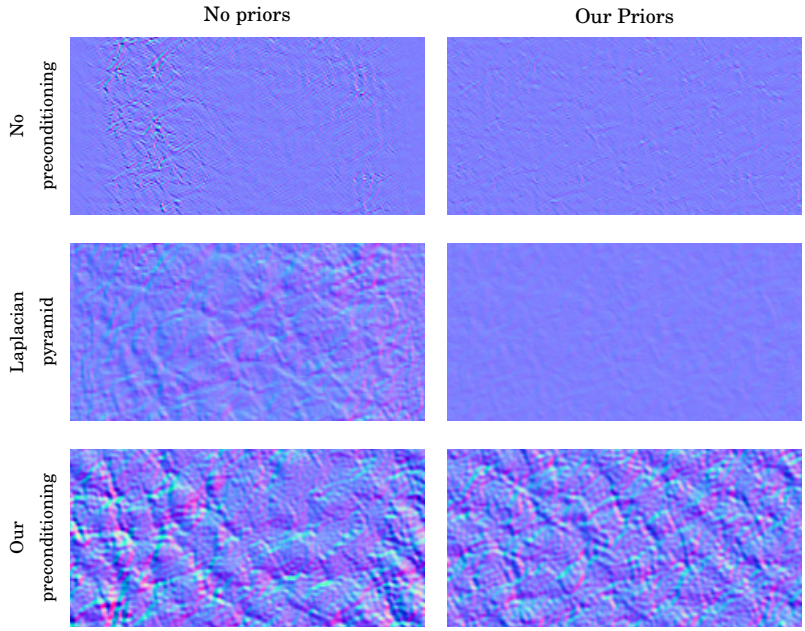


Figure 6.12. The effect of the priors and the preconditioning on the normal map of a leather material, after 100 L-BFGS iterations.

not exactly accurate when dealing with highly non-convex problems such as ours. The path taken may significantly affect the local minimum that the optimizer ultimately converges into. Even the choice of optimization method often has a similar effect. This effect is very difficult to quantify, but we do believe that the preconditioning often significantly improves the quality of the solutions.

Figure 6.12 shows a comparison of results with and without preconditioning, as well as with and without the stationarity priors. As expected, it is the combination of both that reaches a high-quality stationary solution in a timely manner. The figure also contains a comparison to a Laplacian pyramid based preconditioner, which is similar to what Barron and Malik [5] propose; clearly, it is not optimally suited for our problem.

We have omitted some technical details concerning the use of preconditioning. In particular, we also transform some variables in a way that enforces optimization constraints such as non-negativity of albedos, and the integrability of the normal map. For a detailed account, please refer to Section III.4.4.2 in the publication.

6.3.5 Results and discussion

Implementation We implement the method in Matlab using the MatConvNet package that implements the general neural network machinery. In particular, it automatically maps the computations to the GPU. Hence, the time-consuming parts of the computations are all GPU accelerated.

Input data and results We tested the algorithm using the dataset of flash photographs from Publication II. Please refer to Section 6.2.2 for the details. The project webpage[5] contains a complete set of the results and rendered videos.

The results of the method are presented in Section III.5. Full results are available in the supplemental material of the publication. Figure III.10 shows a selection of SVBRDF maps solved by the method, along with a piece of the input data and a corresponding re-rendering. We find good overall agreement between the two.

Figure III.11 shows a selection of images of the input flash photographs, with renderings of the solution tile overlaid at the tiles that were used in the data fitting. The tiles embed well into their surroundings, both in terms of overall color and intensity, as well as the character of the spatial variation. Note that they are not expected to tile *seamlessly* into the surrounding material, as the spatial structures are not expected to match.

Generally, the results are often of a good quality considering the the difficulty of the problem, and the fact that we are only using a single photograph as an input. In fact, to our knowledge our method is the first single-shot method (of which there are very few in general) to apply principled data-fit optimization in solving the problem. However, on closer inspection, the results do not quite reach the quality of the solutions in Publication II.

We perform two different validation experiments. In Figure III.14, we again compare the the normal maps to the ground truth obtained from the GelSight scanner [70] (see Section 6.2.2 for an explanation of this experiment). Although somewhat more rough, the results show a good qualitative agreement.

We also conduct an experiment with synthetic data. To obtain plausible synthetic data, we use re-renderings of the solution SVBRDFs from Publication II. The main objective of this experiment is to explore whether the

[5] https://mediatech.aalto.fi/publications/graphics/NeuralSVBRDF/

shading constraints posed by the flash photograph tiles are strong enough as to guide the solution towards the ground truth, and secondly, whether the algorithm manages to recover it in practice. The results of the experiment in Figure III.15 demonstrate this to be the case: the solutions are generally a good match to the underlying ground truth.

Failure cases and limitations The method shares the same general limitations as the method in Publication II, with the added restriction to isotropic reflectance. The model could, in principle, be extended with anisotropy, but it is possible that the added degrees of freedom would make the optimization problem significantly more difficult. We did, nevertheless, also include the anisotropic datasets in the results. Figure III.11 demonstrates some: while the method often manages to find an SVBRDF that superfically reproduces the appearance from the original angle, close inspection and renderings in novel conditions quickly reveal the solution as incorrect.

Among the datasets that do fulfill the model assumptions, the most common failure the inability to reproduce regular structure. This is clearly visible in e.g. *wallpaper_3* dataset in Figure III.10: the solution has not quite managed to reproduce the regular grid pattern of the input data. While this behavior is common also for classical texture synthesis (see Figure 6.8), the extra complexity in our problem is likely to exacerbate it. Related to this, there are cases where the constraints posed by the flash photograph are probably insufficient for uniquely identifying the surface structure. This is apparent in the result on *plastic_cutting* dataset in Figure III.14. The input data is somewhat degenerate, because in most image regions it mainly consists of evenly spaced dots (the specular highlights).

The method appears to be somewhat more sensitive to the diffuse-specular separation issues than the method of Publication II. This can be seen in the result on the brown leather in the synthetic dataset experiment (Figure III.15): there is some visible cross-talk between the diffuse and the specular albedos due to the very wide specular lobe. It is possible that this problem could be fixed, for example, by simply introducing some basic smoothness priors — note that we have not used any, as we wanted to avoid the added layer of complexity introduced by them.

Discussion An interesting question is to what extent the success of the method depends on the texture descriptor of Gatys et al. [46]. Note that the general idea of solving for SVBRDFs using textural data fitting is novel, even if the texture descriptor were replaced with something else.

We explored this question by simply drop-in replacing the neural descriptor by a simple comparison of power spectra (as used in random phase noise texture synthesis, see Section 6.1). Interestingly, we find that the method still works. However, as expected, the simple descriptor fails to reproduce structured image detail in the inputs, and produces a Perlin noise -like variation in the SVBRDF maps. This is illustrated in Figure III.12.

6.4 Discussion

The two methods presented in this chapter demonstrate that the idea of taking advantage of stationarity can lead to successful high-quality capture of a wide range of SVBRDFs. While the number of input photographs for both methods is very low compared to traditional methods, the results often rival those of significantly more complex approaches.

Perhaps not surprisingly, the quality of the results produced by the two-shot method of Publication II is generally better than that of the single-shot method in Publication III. Nevertheless, both have their place. Aside from the general idea of taking advantage of stationarity, the techniques applied in Publication II are more traditional: pointwise data fitting, traditional texture synthesis, and the use of classical computer vision feature descriptors. While, of the two, Publication II may presently be the method of choice for practical low-cost high-quality SVBRDF capture, the use of neural networks in the manner demonstrated in Publication III has little precedent in previous work, and has the potential for interesting follow-up work. The method is also potentially easier to adapt to different problems and configurations, as it follows the ideal optimization model (Figure 1.8) closely.

Furthermore, the VGG-19 network [117] applied as the texture descriptor in Publication III was the state of the art roughly two years ago at the time of writing — a long time in the rapidly advancing field. It will be interesting to see what kind of improvements might be obtained by simply drop-in replacing the VGG network with more advanced networks to come. We also expect the widely publicized work of Gatys et al. [46, 45] to lead to renewed interest in texture synthesis and related applications. Future work along these lines is likely to have direct relevance for our problems as well.

In terms of practical capture setups, the cell phone camera head-on flash

approach demonstrated in the publications should not be taken as set in stone. The stationarity idea has a much wider applicability — in principle, it could be combined with almost any SVBRDF acquisition setup to potentially significantly reduce the amount of data needed. Whether this is feasible in practice of course depends the particular details of the given method. For example, it would be interesting to combine the idea with the frequency domain measurement setup on Publication I. In particular, because the surface points are seen from different viewpoints due to the near-field camera, this would increase the effective range of viewing angles that contribute to the BRDFs due to the global measurement sharing. Obviously, the cost of this is that the method would then be restricted to stationary materials.

A particularly interesting question is whether the methods could be applied on more general 3D geometry as well. This poses some challenges. In terms of Publication II, obtaining a clean guide image may be difficult due to significantly varying lighting and viewing angles on objects with curvature. Publication III has the potential to side-step this issue, as no guide photograph is needed. Another challenge is the parameterization of the "texture coordinates": what is the local orientation and scale of the texture at any given surface location? While direct application of the methods to this context is difficult due to these challenges, there is clear potential for future work.

One possible extension to the methods would be to incorporate camera motion into the capture procedure: instead of capturing still photos, we might capture a video where the user translates the camera and flash over a short path on the surface. Recent work (e.g. [20, 129]) has demonstrated that even small motions may provide useful additional constraints on the surface shape. This information might be useful in extending the methods to more general geometries, in particular.

The stationarity priors of Publication III are particularly interesting. To our knowledge, they are novel in literature. We can envision many applications and extensions for them, and expect them to lead to interesting future work. For example, could a stronger model be obtained by using the idea to require stationarity of the neural activation statistics themselves, instead of the simple moment statistics we currently use? While the neural texture descriptor cannot directly model texturedness of SVBRDFs (as it assumes RGB input data), this could have some interesting applications in other texture-related problems. It would also be interesting to experi-

ment with direct SVBRDF fitting to the entire input flash photo, without any tiling schemes, instead with a stationarity prior enforcing the global consistency.

Finally, the use of neural networks in Publication III raises some interesting questions about other applications they might have in related problems. We will touch upon this issue again in Section 7.2.

7. Discussion and conclusions

In this thesis, we have presented three successful methods for capturing rich surface material appearance descriptors from real-world surfaces. The trend in our research has been towards ever scarcer input data: going from roughly a hundred photographs and a controlled although simple hardware setup, to two photographs in a loosely controlled mobile phone setup, and finally, to just a single photograph.

We will conclude with some reflection and speculation pertaining to the topics discussed in this thesis.

7.1 Characterization of uncertainty

A central theme in this thesis has been generalization — that is, producing a plausible estimate of the material appearance under novel viewing conditions, given only a partial sampling of the space of reflection directions. The methods presented, as well as the vast majority of the prior work in the field, produce a *point estimate*, *i.e.* an individual SVBRDF (or some other descriptor) that represents the "most plausible" explanation of the data. When the input data is scarce, the choice is sometimes quite arbitrary, as a wide range of solutions might explain the observations almost equally well. In these cases the chosen solution often depends on pre-specified priors and other model parameters. While the user can in principle tweak the priors so as to choose the desired kind of generalization, in practice the parameters are unintuitive, and the procedure is slow due to the need to re-run the solver. The goal of producing a definite point estimate may therefore be questioned: would it not be more principled and useful to instead produce a *probability distribution* over SVBRDFs?

Given such a distribution, the space of the plausible SVBRDFs consistent with the input could perhaps be browsed and edited by interactive

means. The user could flexibly supply domain knowledge that isn't captured by the data or the priors. Similarly, subsequent algorithms (perhaps only tangentially related to materials) could use the distribution as an intermediate representation, without needing to be tightly integrated with the appearance capture algorithm itself (an essentially impossible task in many cases). Finally, it is generally useful to be aware about the degree of certainty of a given solution. Sometimes the data and the priors pinpoint a solution with little ambiguity; at other times, the solution is little more than a guess. Presently, the solution contains no information about this. It is not clear how any of this could be achieved in practice.

Bayesian interpretation of the objective functions In fact, we do possess a sort of probability distribution over the unknowns given the data. Recall that we presented a probabilistic justification for point estimation as Bayesian Maximum A Posteriori (MAP) estimation in Section 3.1.4. Thereafter we largely set the underlying probabilistic interpretation aside, and formulated the optimization tasks from pragmatic considerations. However, now that we have formulated some objective functions for our methods, let us briefly return to this interpretation and work backwards to recover the probability distributions that the objective functions implicitly define.

Recall that we formulated a posterior distribution for the unknown parameters x given the data y in Eq. 3.21 as

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \tag{7.1}$$

The MAP point estimate was obtained as $x^{\text{MAP}} = \underset{x}{\operatorname{argmin}} \; -\log[Cp(x|y)]$. In that context, $C = p(y)$ is an irrelevant normalization constant, as the minimizer is not affected by constant factors.

According to this interpretation, when we minimize an objective function $F(x|y)$ over unknown parameters x given fixed measurements y, such as e.g. in Eq. 6.7 in connection with Publication III, we are in fact minimizing this negative log-posterior. In other words, $F(x|y) = -\log[Cp(x|y)]$. Conversely, this means that by specifying F, we have implicitly defined a posterior

$$p(x|y) = \frac{1}{C}\exp[-F(x|y)] \tag{7.2}$$

The leading constant $\frac{1}{C}$ is problematic. In MAP estimation it can be ignored, but if we wanted to use this function as a probability density (i.e. a non-negative function that integrates to 1), we would need to know the

value. In fact, C can be nothing else than the integral that normalizes the density. That is, $C(y) = \int \exp[-F(x|y)]\mathrm{d}x$. It is a function of the data y. Unfortunately, it is most likely intractable: we have no reasonable means for evaluating this integral for an objective function as complex as ours. Consequently, we have only managed to recover the posterior up to an unknown constant, which limits its usability.

It is not clear if these considerations lead to a dead end, but they do raise some questions. If we *could* normalize the implicit posterior distributions of our methods, would they be sensible in the sense that drawing realizations from the distribution would yield a selection of different plausible solutions to the problem? Or are our objective functions merely "heuristics" designed to drive an optimizer towards good MAP estimates? Would a sensible posterior be beneficial for the quality of the MAP estimates? Could the unnormalized density be sampled e.g. by the Metropolis-Hastings algorithm [60], and would this yield any insight, or perhaps practical ways to characterize the uncertainty around the MAP solution? Could any insights be derived by considering the second-order Taylor expansion of F at the MAP estimate, and building the implicit posterior around this approximation? This results in a Gaussian distribution over all variables and pixels. Is it expressive enough as to be useful?

7.2 Priors and machine learning

Publications II and III are probably close to the limit in terms of how little input data can be used to capture reasonably general SVBRDFs based on *mathematical* constraints. In particular, the single-photograph approach of Publication III clearly could not be simplified much further. Dropping the stationary requirement (and the corresponding algorithmic machinery) would reduce the method to a straightforward but extremely ill-posed per-point data fitting task, with merely one light sample per surface point — a guaranteed failure. On the other hand, the stationarity assumption is only useful if the input data contains a wide range of lighting and/or view directions, such as in the flash photo. Hence, it does not help us to solve SVBRDFs from, say, a single photograph of a directionally lit surface, or a surface with unknown lighting.

While the methods certainly leave room for variations and improvements, it seems unlikely that significant further simplifications in capture setups could come from additional mathematical constraints. In particu-

lar, the problem assigning plausible SVBRDFs on a single photograph of a non-stationary material under loosely controlled illumination is fundamentally ill-posed. The only way to choose among the vast space of valid solutions is the said *plausibility*: a plausible explanation is quite simply a one that a human observer might expect to find upon examining the material further. As briefly discussed in Section 1.3.2, this ability is *learned* by life-long observation of the behavior of surface appearance.

Note that this corresponds precisely to the notion of priors. Indeed, the smoothness constraints and pointwise priors we have applied are attempts at characterizing the space of plausible solutions. While important for the success of the methods in practice, they are nevertheless somewhat unsatisfactory. As discussed in Section 3.1.5, they encode only a loose characterization of the space of SVBRDFs (essentially, "plausible materials have a Perlin noise -like variation"). They are too weak to be of much help for tasks like the single-photograph problem described above.

As already briefly noted, neural networks have proven successful in many learning tasks of this kind. It could be speculated that current neural network architectures are good at solving visual problems that humans can solve "at a glance", i.e. without having to stop and perform logical reasoning. Such examples include recognition [119, 117], depth perception from monocular cues [41], optical flow [38], color assignment to black and white images [68], and many others. It would be reasonable to expect similar success in material-related tasks. Indeed, some work on e.g. material labeling has already been done [23].

Arguably, Publication III takes some steps in this direction. While we do not train a neural network, we do use the features a previously trained network has learned. Hence, it takes advantage of a characterization of the space of natural images in order to find a plausible solution. On the other hand, the reason it finds natural-appearing materials is that it is supplied with natural input photographs. The Gatys et al. [46] texture descriptor merely facilitates this process.

Possible avenues of research What material-related tasks would one train a neural network for, then?

The somewhat obvious approach would be to teach the network to map local patches of images of surfaces to their corresponding SVBRDF maps. Clearly this task is ill-posed, because many different SVBRDFs can plausibly explain such data. Consider for example the local tiles shown in Figure 6.2. Even without any explicit knowledge about the lighting con-

ditions, visual inspection of any of them in isolation suggests a moderate degree of gloss, and a relatively constant brown diffuse color, and surface normal variation as the primary explanation of the apparent variation. In particular, the directional shading is a strong clue. It is not unreasonable to expect that a neural network might learn to identify properties on this level. However, due to the ill-posedness, estimates might vary wildly across an extended piece of the surface, where the shading might vary. Perhaps the stationary priors could be applied in some way? Perhaps the network architecture could consist of two processing paths — one that produces a global set of reflectance features from the entire image, and one for the local detail features are discussed above — the results of which would be joined and reconciled in the final layers of the network. Clearly, there is a large space of possible designs to be explored.

Another interesting application would be methods for modeling the space of naturally occurring spatially varying reflectance. Ideally, we would be interested in replacing the coarse, and generally somewhat unsatisfactory manually specified priors (such as the smoothness constraints) with a prior term that simply penalizes implausible SVBRDFs. Such a prior could, in principle, be combined with any optimization-based SVBRDF fitting task to manage their ill-posedness.

It is not at all clear how such ideas could be implemented in practice. Clearly, image recognition networks such as VGG-19 [117] and GoogLeNet [119] have learned something crucial about the structure of natural images, as evidenced by e.g. the success of the texture synthesis method of Gatys et al. [46] and some interesting applications like “deep dreaming” [97]. The same holds for networks trained explicitly for random generation of natural-looking images [105]. The information about this structure is implicitly encoded in the mappings learned by these networks. Could it be made more explicit? Could some specifically designed training task produce the kind of information we are looking for, perhaps explicitly, or perhaps as a “side effect” like in the recognition networks?

This discussion is, of course, speculative, and only scratches the surface of the topic. We expect to see interesting future research around these questions.

References

[1] Edward H. Adelson and James R. Bergen. The Plenoptic Function and the Elements of Early Vision. In *Computational Models of Visual Processing*, pages 3–20. MIT Press, 1991.

[2] Peter Ahrendt. The Multivariate Gaussian Probability Distribution, January 2005.

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

[4] Michael Ashikhmin and Simon Premoze. Distribution-based BRDFs. Technical report, 2007.

[5] J. Barron and J. Malik. Shape, Illumination, and Reflectance from Shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (to appear), 2015.

[6] James R. Bergen and Edward H. Adelson. Early vision and texture perception. *Nature*, 333(6171):363–364, 05 1988.

[7] Pravin Bhat, Brian Curless, Michael Cohen, and C. Lawrence Zitnick. Fourier Analysis of the 2D Screened Poisson Equation for Gradient Domain Problems. In *Proceedings of the 10th European Conference on Computer Vision: Part II*, ECCV '08, pages 114–128, Berlin, Heidelberg, 2008. Springer-Verlag.

[8] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[9] James F. Blinn. Models of light reflection for computer synthesized pictures. *Computer Graphics (Proc. SIGGRAPH)*, 11(2):192–198, July 1977.

[10] James F. Blinn. Simulation of Wrinkled Surfaces. *SIGGRAPH Comput. Graph.*, 12(3):286–292, August 1978.

[11] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[12] Yuri Boykov, Hossam N. Isack, Carl Olsson, and Ismail Ben Ayed. Volumetric Bias in Segmentation and Reconstruction: Secrets and Solutions. *CoRR*, abs/1505.00218, 2015.

[13] R.N. Bracewell. *The Fourier Transform and its Applications*. McGraw-Hill Kogakusha, Ltd., Tokyo, second edition, 1978.

[14] Adam Brady, Jason Lawrence, Pieter Peers, and Westley Weimer. genBRDF: Discovering New Analytic BRDFs with Genetic Programming. *ACM Trans. Graph.*, 33(4):114:1–114:11, July 2014.

[15] Brent Burley. Physically-Based Shading at Disney, August 2012.

[16] Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A Limited Memory Algorithm for Bound Constrained Optimization. *SIAM J. Sci. Comput.*, 16(5):1190–1208, September 1995.

[17] Paul H. Calamai and Jorge J. More:9A. Projected Gradient Methods for Linearly Constrained Problems. *Math. Program.*, 39(1):93–116, October 1987.

[18] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF: Binary Robust Independent Elementary Features. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, pages 778–792, Berlin, Heidelberg, 2010. Springer-Verlag.

[19] Antonin Chambolle and Thomas Pock. A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging. *J. Math. Imaging Vis.*, 40(1):120–145, May 2011.

[20] M. Chandraker. The Information Available to a Moving Observer on Shape with Unknown, Isotropic BRDFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(7):1283–1297, July 2016.

[21] Guojun Chen, Yue Dong, Pieter Peers, Jiawan Zhang, and Xin Tong. Reflectance Scanning: Estimating Shading Frame and BRDF with Generalized Linear Light Sources. *ACM Transactions on Graphics*, 33(4), August 2014.

[22] P. H. Christensen. Adjoints and Importance in Rendering: An Overview. *IEEE Transactions on Visualization and Computer Graphics*, 9(3):329–340, July 2003.

[23] M. Cimpoi, S. Maji, I. Kokkinos, and A. Vedaldi. Deep Filter Banks for Texture Recognition, Description, and Segmentation. *International Journal of Computer Vision (IJCV)*, 2016.

[24] Ryan Clark. CrazyBump, www.crazybump.com, 2010.

[25] Robert L. Cook and Kenneth E. Torrance. A Reflection Model for Computer Graphics. *ACM Transactions on Graphics*, 1(1):7–24, January 1982.

[26] K. J. Dana. BRDF/BTF measurement device. *International Conference on Computer Vision ICCV*, 2:460–6, July 2001.

[27] Kristin J. Dana, Bram van Ginneken, Shree K. Nayar, and Jan J. Koenderink. Reflectance and Texture of Real-world Surfaces. *ACM Trans. Graph.*, 18(1):1–34, January 1999.

[28] Kristin J. Dana and Jing Wang. Device for convenient measurement of spatially varying bidirectional reflectance. *Journal of the Optical Society of America A*, 21(1):1–12, 2004.

[29] Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2933–2941. Curran Associates, Inc., 2014.

[30] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. In *Proc. SIGGRAPH*, pages 145–156, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[31] Paul E. Debevec and Jitendra Malik. Recovering High Dynamic Range Radiance Maps from Photographs. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, pages 369–378, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[32] 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.

[33] 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*, 33(6), December 2014.

[34] 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, December 2011.

[35] Yue Dong, Jiaping Wang, Xin Tong, John Snyder, Yanxiang Lan, Moshe Ben-Ezra, and Baining Guo. Manifold bootstrapping for SVBRDF capture. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 29(4):98:1–98:10, July 2010.

[36] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, April 2006.

[37] David L. Donoho. For Most Large Underdetermined Systems of Linear Equations the Minimal ℓ_1-norm Solution is also the Sparsest Solution. *Comm. Pure Appl. Math*, 59:797–829, 2004.

[38] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. FlowNet: Learning Optical Flow with Convolutional Networks. In *IEEE International Conference on Computer Vision (ICCV)*, Dec 2015.

[39] Alexei A. Efros and William T. Freeman. Image Quilting for Texture Synthesis and Transfer. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 341–346, New York, NY, USA, 2001. ACM.

[40] Alexei A. Efros and Thomas K. Leung. Texture Synthesis by Non-Parametric Sampling. In *Proc. International Conference on Computer Vision (ICCV '99)*, volume 2, pages 1033–1038, 1999.

[41] David Eigen, Christian Puhrsch, and Rob Fergus. Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. In Z. Ghahramani, M. Welling, C. Cortes, N.d. Lawrence, and K.q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2366–2374. Curran Associates, Inc., 2014.

[42] Sing Choong Foo. A Gonioreflectometer For Measuring The Bidirectional Reflectance Of Material For Use In Illumination Computation, 1997.

[43] B. Galerne, Y. Gousseau, and J. M. Morel. Random Phase Textures: Theory and Synthesis. *IEEE Transactions on Image Processing*, 20(1):257–267, Jan 2011.

[44] Andrew Gardner, Chris Tchou, Tim Hawkins, and Paul Debevec. Linear light source reflectometry. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 22(3):749–758, July 2003.

[45] L. A. Gatys, A. S. Ecker, and M. Bethge. A Neural Algorithm of Artistic Style. *CoRR*, abs/1508.06576, 2015.

[46] L. A. Gatys, A. S. Ecker, and M. Bethge. Texture Synthesis Using Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 28*, 2015.

[47] Stuart Geman and Christine Graffigne. Markov random field image models and their applications to computer vision. In *Proceedings of the International congress of mathematicians 1986 Ed.*, pages 1496–1517, Berkeley, California, 1987. American Mathematical Society.

[48] Abhijeet Ghosh, Shruthi Achutha, Wolfgang Heidrich, and Matthew O'Toole. BRDF Acquisition with Basis Illumination. In *Proc. IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[49] Abhijeet Ghosh, Tongbo Chen, Pieter Peers, Cyrus A. Wilson, and Paul Debevec. Circularly polarized spherical illumination reflectometry. *ACM Transactions on Graphics (Proc. SIGGRAPH ASIA)*, 29(6):162:1–162:12, December 2010.

[50] Abhijeet Ghosh, Tongbo Chen, Pieter Peers, Cyrus A. Wilson, and Paul E. Debevec. Estimating Specular Roughness and Anisotropy from Second Order Spherical Gradient Illumination. *Computer Graphics Forum (Proc. Eurographics Symposium on Rendering)*, 28(4):1161–1170, 2009.

[51] Mashhuda Glencross, Greg Ward, C. Jay, J. Liu, F. Melendez, and R. Hubbold. A Perceptually Validated Model for Surface Depth Hallucination. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 27(3):59:1–59:8, August 2008.

[52] D.B. Goldman, B. Curless, A. Hertzmann, and S.M. Seitz. Shape and spatially-varying BRDFs from photometric stereo. In *Proc. IEEE International Conference on Computer Vision*, volume 1, pages 341–348, October 2005.

[53] Gene H. Golub and Charles F. Van Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[54] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. Book in preparation for MIT Press, 2016.

[55] Ian J. Goodfellow and Oriol Vinyals. Qualitatively characterizing neural network optimization problems. *CoRR*, abs/1412.6544, 2014.

[56] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. Modeling the Interaction of Light Between Diffuse Surfaces. *SIGGRAPH Comput. Graph.*, 18(3):213–222, January 1984.

[57] R. M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, May 1979.

[58] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[59] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[60] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[61] David J. Heeger and James R. Bergen. Pyramid-based Texture Analysis/Synthesis. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 229–238, New York, NY, USA, 1995. ACM.

[62] Stephen Hill and Stephen McAuley. Siggraph 2015 Course: Physically Based Shading in Theory and Practice, August 2015.

[63] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury. Deep Neural Networks for Acoustic Modeling in Speech Recognition. *Signal Processing Magazine*, 2012.

[64] Michael Holroyd, Jason Lawrence, Greg Humphreys, and Todd Zickler. A Photometric Approach for Estimating Normals and Tangents. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 27(3):133:1–133:9, December 2008.

[65] Michael Holroyd, Jason Lawrence, and Todd Zickler. A Coaxial Optical Scanner for Synchronous Acquisition of 3D Geometry and Surface Reflectance. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 29(4):99:1–99:12, July 2010.

[66] David H. Hubel and Torsten N. Wiesel. Receptive Fields and Functional Architecture of Monkey Striate Cortex. *Journal of Physiology (London)*, 195:215–243, 1968.

[67] Peter J. Huber. Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35(1):73–101, March 1964.

[68] 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. of SIGGRAPH 2016)*, 35(4), 2016.

[69] Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan. A Practical Model for Subsurface Light Transport. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 511–518, New York, NY, USA, 2001. ACM.

[70] Kimo Johnson, Forrester Cole, Alvin Raj, and Edward Adelson. Microgeometry capture using an elastomeric sensor. *ACM Trans. Graph.*, 30(4):Article 40, 2011.

[71] Bela Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 290(5802):91–97, March 1981.

[72] J. T. Kajiya and T. L. Kay. Rendering Fur with Three Dimensional Textures. *SIGGRAPH Comput. Graph.*, 23(3):271–280, July 1989.

[73] James T. Kajiya. The Rendering Equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '86, pages 143–150, New York, NY, USA, 1986. ACM.

[74] Alexandre Kaspar, Boris Neubert, Dani Lischinski, Mark Pauly, and Johannes Kopf. Self Tuning Texture Optimization. *Computer Graphics Forum*, 34(2), 2015.

[75] K. Kawaguchi. Deep Learning without Poor Local Minima. *ArXiv e-prints*, May 2016.

[76] Seyed-Mahdi Khaligh-Razavi and Nikolaus Kriegeskorte. Deep Supervised, but Not Unsupervised, Models May Explain IT Cortical Representation. *PLoS Comput Biol*, 10(11):1–29, 11 2014.

[77] R. Kinderman and S.L. Snell. *Markov random fields and their applications*. American mathematical society, 1980.

[78] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598):671–680, 1983.

[79] Philipp Krahenbuhl and Vladlen Koltun. *Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials*. 2011.

[80] Vivek Kwatra, Irfan Essa, Aaron Bobick, and Nipun Kwatra. Texture Optimization for Example-based Synthesis. *ACM Trans. Graph.*, 24(3):795–802, 2005.

[81] Jason Lawrence, Szymon Rusinkiewicz, and Ravi Ramamoorthi. Efficient BRDF Importance Sampling Using a Factored Representation. *ACM Trans. Graph.*, 23(3):496–505, August 2004.

[82] Tai Sing Lee. Image Representation Using 2D Gabor Wavelets. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(10):959–971, October 1996.

[83] Hendrik P. A. Lensch, Jan Kautz, Michael Goesele, Wolfgang Heidrich, and Hans-Peter Seidel. Image-based Reconstruction of Spatial Appearance and Geometric Detail. *ACM Trans. Graph.*, 22(2):234–257, April 2003.

[84] J. Löw, J. Kronander, A. Ynnerman, and J. Unger. BRDF Models for Accurate and Efficient Rendering of Glossy Surfaces. *ACM Transactions on Graphics*, 31(1):9:1–9:14, January 2012.

[85] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

[86] Wan-Chun Ma, Tim Hawkins, Pieter Peers, Charles-Felix Chabert, Malte Weiss, and Paul Debevec. Rapid Acquisition of Specular and Diffuse Normal Maps from Polarized Spherical Gradient Illumination. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, EGSR'07, pages 183–194, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.

[87] David J. C. MacKay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, New York, NY, USA, 2002.

[88] Jitendra Malik and Pietro Perona. Preattentive texture discrimination with early vision mechanisms. *Journal of the Optical Society of America A*, 7:923–932, 1990.

[89] Tom Malzbender, Dan Gelb, and Hans Wolters. Polynomial Texture Maps. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 519–528, New York, NY, USA, 2001. ACM.

[90] D. Marr. Early Processing of Visual Information. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 275(942):483–519, 1976.

[91] S. R. Marschner, S. H. Westin, E. P. F. Lafortune, and K. E. Torrance. Image-Based Bidirectional Reflectance Distribution Function Measurement. 39:2592–2600, June 2000.

[92] Stephen R. Marschner, Stephen H. Westin, Adam Arbree, and Jonathan T. Moon. Measuring and Modeling the Appearance of Finished Wood. *ACM Trans. Graph.*, 24(3):727–734, July 2005.

[93] Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan. A Data-driven Reflectance Model. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 22(3):759–769, July 2003.

[94] Wojciech Matusik, Hanspeter Pfister, Matthew Brand, and Leonard McMillan. Efficient Isotropic BRDF Measurement. In *Proceedings of the 14th Eurographics Workshop on Rendering*, EGRW '03, pages 241–247, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[95] Wojciech Matusik, Matthias Zwicker, and Frédo Durand. Texture Design Using a Simplicial Complex of Morphable Textures. *ACM Trans. Graph.*, 24(3):787–794, July 2005.

[96] David Kirk Mcallister. *A Generalized Surface Appearance Representation for Computer Graphics*. PhD thesis, 2002. AAI3061704.

[97] Alexander Mordvintsev, Christopher Olah, and Mike Tyka. Inceptionism: Going Deeper into Neural Networks. https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html, last accessed Jun. 8, 2016.

[98] Addy Ngan and Frédo Durand. Statistical Acquisition of Texture Appearance. In *Proceedings of the 17th Eurographics Conference on Rendering Techniques*, EGSR '06, pages 31–40, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.

[99] Addy Ngan, Frédo Durand, and Wojciech Matusik. Experimental Analysis of BRDF Models. In *Proc. Eurographics Symposium on Rendering*, pages 117–226, 2005.

[100] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis. Radiometry. chapter Geometrical Considerations and Nomenclature for Reflectance, pages 94–145. Jones and Bartlett Publishers, Inc., USA, 1992.

[101] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 2nd edition, 2006.

[102] Jorge Nocedal. Updating quasi-Newton matrices with limited storage. *Mathematics of computation*, 35(151):773–782, 1980.

[103] Bui Tuong Phong. Illumination for Computer Generated Pictures. *Commun. ACM*, 18(6):311–317, June 1975.

[104] Javier Portilla and Eero P. Simoncelli. A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients. *Int. J. Comput. Vision*, 40(1):49–70, October 2000.

[105] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *CoRR*, abs/1511.06434, 2015.

[106] Ravi Ramamoorthi and Pat Hanrahan. A Signal Processing Framework for Inverse Rendering. In *Proc. SIGGRAPH*, pages 117–128, 2001.

[107] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[108] Peiran Ren, Jiaping Wang, John Snyder, Xin Tong, and Baining Guo. Pocket reflectometry. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 30(4):45:1–45:10, July 2011.

[109] Fabiano Romeiro and Todd Zickler. Blind Reflectometry. In *Proceedings of the 11th European Conference on Computer Vision: Part I*, ECCV'10, pages 45–58, Berlin, Heidelberg, 2010. Springer-Verlag.

[110] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear Total Variation Based Noise Removal Algorithms. *Phys. D*, 60(1-4):259–268, November 1992.

[111] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Neurocomputing: Foundations of Research. chapter Learning Representations by Back-propagating Errors, pages 696–699. MIT Press, Cambridge, MA, USA, 1988.

[112] Szymon Rusinkiewicz. A New Change of Variables for Efficient BRDF Representation. In George Drettakis and Nelson L. Max, editors, *Rendering Techniques*, Eurographics, pages 11–22. Springer, 1998.

[113] Iman Sadeghi, Oleg Bisker, Joachim De Deken, and Henrik Wann Jensen. A Practical Microcylinder Appearance Model for Cloth Rendering. *ACM Trans. Graph.*, 32(2):14:1–14:12, April 2013.

[114] Christophe Schlick. An Inexpensive BRDF Model for Physically-based Rendering. *Computer Graphics Forum*, 13:233–246, 1994.

[115] Steven A. Shafer. Color. chapter Using Color to Separate Reflection Components, pages 43–51. Jones and Bartlett Publishers, Inc., USA, 1992.

[116] E P Simoncelli and W T Freeman. The Steerable Pyramid: A Flexible Architecture for Multi-Scale Derivative Computation. In *Proc 2nd IEEE Int'l Conf on Image Proc*, volume III, pages 444–447, Washington, DC, Oct 23-26 1995. IEEE Sig Proc Society.

[117] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, abs/1409.1556, 2014.

[118] Peter-Pike Sloan. Stupid Spherical Harmonics (SH) Tricks, February 2008.

[119] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going Deeper with Convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[120] Meng Tang, Ismail Ben Ayed, Dmitrii Marin, and Yuri Boykov. Secrets of GrabCut and Kernel K-means. *CoRR*, abs/1506.07439, 2015.

[121] Albert Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2004.

[122] Natalya Tatarchuk, Sebastien Hillaire, Tomasz Stachowiak, Andrew Schneider, Huw Bowles, Daniel Zimmerman, Beibei Wang, Ari Silvennoinen, Ville Timonen, Matt Pettineo, Alex Evans, Alex Evans, Ulrich Haar, and Sebastian Aaltonen. Siggraph 2015 Course: Advances in Real-Time Rendering, August 2015.

[123] James T. Todd, J. Farley Norman, and Ennio Mignolla. Lightness Constancy in the Presence of Specular Highlights, 2004.

[124] Borom Tunwattanapong, Graham Fyffe, Paul Graham, Jay Busch, Xueming Yu, Abhijeet Ghosh, and Paul Debevec. Acquiring Reflectance and Shape from Continuous Spherical Harmonic Illumination. *ACM Trans. Graph.*, 32(4):109:1–109:12, July 2013.

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

[126] Cedric Villani. *Optimal transport : old and new*. Grundlehren der mathematischen Wissenschaften. Springer, Berlin, 2009.

[127] 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):172:1–172:12, December 2011.

[128] Jiaping Wang, Shuang Zhao, Xin Tong, John Snyder, and Baining Guo. Modeling Anisotropic Surface Reflectance with Example-based Microfacet Synthesis. *ACM Trans. Graph.*, 27(3):41:1–41:9, August 2008.

[129] Ting-Chun Wang, Manmohan Chandraker, Alexei Efros, and Ravi Ramamoorthi. SVBRDF-invariant shape and reflectance estimation from light-field cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[130] Gregory J. Ward. Measuring and Modeling Anisotropic Reflection. *SIGGRAPH Comput. Graph.*, 26(2):265–272, July 1992.

[131] Tim Weyrich, Jason Lawrence, Hendrik Lensch, Szymon Rusinkiewicz, and Todd Zickler. Principles of appearance acquisition and representation. *Foundations and Trends in Computer Graphics and Vision*, 4(2):75–191, 2008.

[132] D. Rod White, Peter Saunders, Stuart J. Bonsey, John van de Ven, and Hamish Edgar. Reflectometer for measuring the bidirectional reflectance of rough surfaces. *Appl. Opt.*, 37(16):3450–3454, Jun 1998.

[133] Daniel L. K. Yamins, Ha Hong, Charles F. Cadieu, Ethan A. Solomon, Darren Seibert, and James J. DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, 111(23):8619–8624, 2014.

[134] Shuang Zhao, Wenzel Jakob, Steve Marschner, and Kavita Bala. Structure-aware Synthesis for Predictive Woven Fabric Appearance. *ACM Trans. Graph.*, 31(4):75:1–75:10, July 2012.

[135] Todd Zickler, Peter N. Belhumeur, and David J. Kriegman. Helmholtz Stereopsis: Exploiting Reciprocity for Surface Reconstruction. *International Journal of Computer Vision*, 49(2/3):215–227, 2002.

[136] Todd Zickler, Ravi Ramamoorthi, Sabastian Enrique, and Peter 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):1287–1302, 2006.

Errata

Publication III

- Eq. 3 is missing the addition of a per-layer bias term.

- Fig. 6 is missing a branch for the variance stationarity prior, corresponding to a sequence subtract mean -> pow 2 -> FFT -> magnitude -> weight -> norm

Aalto-DD 199/2016

ISBN 978-952-60-7048-3 (printed)
ISBN 978-952-60-7047-6 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)

Aalto University
School of Science
Department of Computer Science
www.aalto.fi

BUSINESS +
ECONOMY

ART +
DESIGN +
ARCHITECTURE

SCIENCE +
TECHNOLOGY

CROSSOVER

**DOCTORAL
DISSERTATIONS**