Acquisition, Transmission and Rendering of Objects with Optically Complicated Material Appearance

Dissertation

zur
Erlangung des Doktorgrades (Dr. rer. nat.)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Dipl.-Inf. Jens Christopher Schwartz

aus Köln

Bonn, November 2014

Universität Bonn
Institut für Informatik II
Friedrich-Ebert-Allee 144, D-53113 Bonn

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms Universität Bonn

1. Referent: Prof. Dr. Reinhard Klein
2. Referent: Prof. Dr. Matthias B. Hullin

Tag der Promotion: 28. August 2015
Erscheinungsjahr: 2015

CONTENTS

ZUSAMMENFASSUNG

Ein wichtiges Ziel der Computergrafik ist die Generierung fotorealistischer Bilder. Dabei hängt der Grad des Realismus heutzutage nicht mehr so sehr von den verwendeten Renderingverfahren, sondern vielmehr von der Qualität der Modellierung der virtuellen Szenen ab. Neben zeitaufwändiger manueller Erstellung durch einen Künstler können die Modelle und Parameter dieser Szenen auch durch Digitalisierung direkt von realen Vorbildern erfasst werden.

In dieser Dissertation beschäftigen wir uns mit der Digitalisierung und fotorealistischen Darstellung von ganzen Objekten inklusive ihrer optischen Materialeigenschaften. Einsatzzwecke für realitätsgetreu darstellbare Objekte gibt es viele. Als Beispiele seien hier die Erstellung von virtuellen Requisiten für Spezialeffekte in Filmen und Computerspielen, die Katalogisierung und Präsentation von kulturellem Erbe im Internet oder Produktdarstellungen in Onlineshops genannt.

In unserem Verfahren spielt vor allen Dingen die von uns gewählte Form der digitalen Materialrepräsentation durch eine *bidirektionale Textur Funktion* (BTF) eine Schlüsselrolle. Die BTF stellt den Anteil von reflektiertem Licht – die Reflektanz – positions- sowie blick- und lichtrichtungsabhängig dar. Dies ermöglicht einen sehr hohen Grad an Realismus sowie die akkurate Reproduktion von selbst kleinsten Oberflächendetails. Dabei ist die BTF eine sog. „datengetriebene“ Repräsentation. Im Gegensatz zu „modellgetriebenen“ Verfahren, welche die Reflektanz über analytische Funktionen mit einer Hand voll Parameter beschreiben, wird bei der Auswertung der BTF direkt zwischen gemessenen Datenpunkten interpoliert.

Um die BTF einer Oberfläche in ausreichend hohem Detailgrad zu bestimmen, sind allerdings viele Milliarden Messpunkte notwendig. Deswegen präsentieren wir zunächst Aufbauten zur schnellen, automatisierten Aufnahme der notwendigen Reflektanzdaten sowie der 3D Geometrie der Objekte. Beides erfordert eine präzise Kalibrierung der Messgeräte. Die eingesetzten Algorithmen stellen wir im Detail vor.

Aus den so ermittelten Messpunkten erstellen wir in einem Nachverarbeitungsschritt das eigentliche digitale Objekt. Aufgrund der praktischen Beschränkungen der Messaufbauten sind die ermittelten Reflektanzdaten an vielen Stellen unvoll-

ständig. Wir schlagen deswegen einen auf Matrixfaktorisierung basierenden Ansatz vor, um die Lücken datengetrieben aufzufüllen.

Eine Evaluation des Ansatzes an insgesamt 27 digitalisierten Objekten mit unterschiedlichsten Formen und Materialien zeigt, dass die so gewonnenen Resultate das Erscheinungsbild der realen Objekte im Allgemeinen sehr gut reproduzieren. Wir zeigen aber auch Grenzen des Verfahrens auf.

Ein großer Nachteil der aufwändigen BTF Repräsentation gegenüber modellbasierten Ansätzen ist der sehr hohe Speicherverbrauch – selbst nach Einsatz von aktuellen Kompressionsverfahren. Wir zeigen daher im Weiteren zwei Ansätze auf, mit denen man die großen Datenmengen für die Übertragung im Internet sowie für die Echtzeitdarstellung in den Griff bekommen kann. Unsere Experimente belegen, dass es möglich ist, die digitalisierten Materialien stärker zu komprimieren und progressiv zu übertragen, so dass innerhalb von wenigen Sekunden bereits eine sehr hohe Qualität der Reproduktion erreicht wird. Weiterhin gelingt es uns, durch geschicktes Level of Detail Rendering den tatsächlichen Speicherbedarf auf der GPU um bis zu 97% zu reduzieren und weitestgehend gering zu halten. Dies ergmöglicht auch die Echtzeitdarstellung von Szenen mit mehreren digitalisierten Objekten.

Die zu den verschiedenen Teilaspekten vorgeschlagenen Verfahren in dieser Dissertation bauen aufeinander auf und ergänzen sich. Zusammengenommen bilden sie ein in vielen Situationen praktisch anwendbares Ökosystem rund um fotorealistische digitale Objekte.

ABSTRACT

A major goal in computer graphics is the generation of photorealistic images. Nowadays, the degree of realism is often not restricted by the rendering algorithm but instead mainly depends on the quality of the virtual scene description. Besides manual modeling by artists, the parameters of the virtual scene's objects can also be determined from measurements of real-world exemplars.

In this thesis, we will explore the acquisition and faithful representation of whole objects, including their optical material properties. The applications for realistic virtual objects are manifold. They can be used as digital props in special effects in movies and computer games, for the documentation and public dissemination of cultural heritage over the Internet or as product previews in online shops, just to name a few examples.

The key in our proposed digitization method is the choice of the *bidirectional texture function* (BTF) to convey the digital material appearance. The BTF defines the amount of reflected light – the reflectance – in dependence on view and light directions and spatial position. This provides a high degree of realism and allows to faithfully reproduce even tiny details. In contrast to so called "model-driven" methods, which derive the reflectance values from an analytical mathematical function with a few parameters, the BTF is a "data-driven" representation. Here, the reflectance is the result of direct interpolation between densely measured values.

However, to describe the BTF with sufficient detail, it is necessary to capture billions of datapoints. For this reason, we first propose setups for the fast automated acquisition of these reflectance samples as well as the objects' 3D geometry. In both cases a precise calibration is mandatory. Hence, we explain the employed calibration algorithms in detail.

The final digitized object is the result of a consecutive postprocessing on the measured data. Due to practical limitations of the setups, the sampling of the reflectance data is often incomplete. Hence, we propose to employ a data-driven hole filling approach based on matrix factorization.

Our evaluation on 27 different objects with variations in shape and material demonstrates that the proposed digitization approach in general results in a very faithful reproductions of the original appearance. However, we also show the limitations of our method.

Even after applying state-of-the-art compression algorithms, one major disadvantage of BTFs with respect to model-based techniques is the tremendous memory requirement. We thus propose two approaches for the transmission of BTF materials over the Internet and real-time rendering on the GPU that cope with the large amounts of data. Our experiments show that by using an additional compression as well as progressive transmission, digital materials can be streamed over the Internet and display a high-quality appearance after just a few seconds. Furthermore, we manage to reduce the GPU memory footprint by up to 97% using a clever level of detail strategy. This way, the GPU's memory bottleneck is mostly avoided and the real-time rendering of virtual scenes containing several digitized objects becomes possible.

The different aspects tackled in this thesis complement each other. Together the proposed techniques form an ecosystem for digital object appearance that is practically applicable in many scenarios.

ACKNOWLEDGEMENTS

Above all, I would like to show my gratitude towards my supervisor Prof. Dr. Reinhard Klein who inspired and encouraged me to write this dissertation. Thanks to his support as well as the freedom he gave me, I was able to pursue my own direction of research.

After spending several years of my life working at the University of Bonn's Institute of Computer Science II, I also need to thank all of my wonderful colleagues. I always enjoyed working with you and spending some evenings together in the various bars in Bonn. I especially thank my co-authors Roland Ruiters, Michael Weinmann, Martin Rump, Ralf Sarlette, Dr. Arno Zinke, Dr. Patrick Degener, Dr. Ruwen Schnabel, Tim Golla, Aljosa Osep, Burkhard Güssefeld and Dr. Daniel Kondermann. Roland and Martin also helped me by proofreading this thesis and provided me with many valuable remarks and comments, for which I am sincerely grateful.

I further would like to acknowledge Nils Jenniche, who implemented preliminary prototypes for both rendering algorithms presented in Part III of this thesis and Dr. Gero Müller, James Stevenson, Prof. Dr. David Arnold, Béatrice Bieber, Prof. Dr. Thomas Bieber and Kai Jäger for providing me with challenging objects to test the approach.

Of course I also thank the two research projects of the the European Community's Seventh Framework Programme (FP7/2007-2013) I worked for: *3D-COFORM* (grant agreement n° 231809; 2008-2012) and *Harvest4D* (grant agreement n° 323567; 2013-2016). They provided me with funding, interesting and practical research to pursue and allowed me to work with nice and excellent colleagues from all over Europe.

My appreciations also go to my family, especially my parents Doris and Bernd Schwartz, who encouraged me to pursue my academic carrier to this point and supported me in many ways. And last but definitely not least I thank Katharina Suhren from the bottom of my heart for bearing with me and supporting me throughout the many deadlines and the often stressful process of writing papers and this thesis.

List of Abbreviations

3G	third generation (of mobile telecommunications technology)
ABRDF	apparent BRDF
ABF	angle-based flattening
BPP	bits per pixel
BRDF	bidirectional reflectance distribution function
BSSRDF	bidirectional scattering-surface reflectance distribution function
BSVTF	bidirectional sparse virtual texture function
BTF	bidirectional texture function
CCD	charge-coupled device (sensor)
CFA	color filter array
CMOS	complementary metal-oxide-semiconductor (sensor)
CNC	computer numerical control (mill)
CPU	central processing unit
DCT	discrete cosine transformation
DFMF	decorrelated FMF
DLP	digital light processing (projector)
DPI	dots per inch
DRC	dynamic range compression
DSL	digital subscriber line
DSLR	digital single-lens reflex (camera)
EM-PCA	expectation maximization PCA
EV	exposure value
FMF	full matrix factorization
FPS	frames per second
GDL	gas discharge lamp
GPU	graphics processing unit
HDMI	high-definition multimedia interface
HDR	high dynamic range
HDR-VDP-2	HDR visual difference predictor 2
HMI	hydrargyrum medium arc length iodide (lamp)
HSDPA	high-speed downlink packet access (network)
HTML	hypertext markup language

IBR	image-based rendering
IR	infrared
JIT	just-in-time (compiler)
JSON	JavaScript object notation
LDR	low dynamic range
LED	light-emitting diode (lamp)
LPCA	local PCA
MSE	mean squared error
MS-SSIM	multiscale SSIM
NMF	non-negative matrix factorization
NNLS	non-negative least squares
OpenGL	open graphics library
PC	personal computer
PCA	principal component analysis
PCI	peripheral component interconnect
PCIe	PCI express
PTM	polynomial texture map
P&S	point-and-shoot (camera)
QTH	quartz tungsten halogen (lamp)
RAM	random access memory
RANSAC	random sample consensus
RBF	radial basis function
RMSE	root mean squared error
ROI	region of interest
RTI	reflectance transformation imaging
SBA	sparse bundle adjustment
SDK	software development kit
SfM	structure from motion
SLAM	simultaneous localization and mapping
SSD	sum of squared distances
SSE	sum of squared errors
SSIM	structural similarity
SVBRDF	spatially varying BRDF
SVD	singular value decomposition
SVT	sparse virtual texturing
USB	universal serial bus
UV	ultraviolet
VGA	video graphics array
WebGL	web graphics library

Part I

Introduction

CHAPTER 1

INTRODUCTION

In the past decades, rendering methods for computer generated images reached a level of realism that make the results almost indistinguishable from photographic pictures. Due to the increasing computational power and ubiquity of *graphic processing units* (GPUs), this is not only true for sophisticated offline renderings but can be observed in real-time applications as well. The major bottleneck for generating convincing photorealistic images is the creation of digital 3D content, including 3D models and reflectance properties.

Recent trends towards stereoscopy (e.g. in television and movies), augmented reality (e.g. Google Glass[1]) and virtual reality (e.g. the Occulus Rift[2]) raise the bar for the quality of content even higher, as inaccuracies and errors are becoming harder to hide from the user. Thus, the creation of digital content has become one of the most pressing issues for realism in 3D computer graphics.

Simply put, there are two main ways to obtain such digital content. First, there is modeling, i.e. a skilled artist creates the digital model from scratch. Naturally, higher degrees of realism impose an increased effort and thoroughness on the modeler. For many classes of objects, this creation process is far from being trivial and usually requires a tremendous amount of manual work by expert 3D artists. The second option is digitization. Here, aspects from a real-world exemplar are captured and brought into a suitable format. If necessary, the digitized item can serve as the foundation for further artistic modifications by a skilled user. However, such interaction is not a prerequisite any more. In contrast to modeling, digitization directly provides a plausible and complete digital asset.

For these reasons, the practice of using real-world samples as assets in computer graphics has been introduced for almost all types of content. Geometric models are obtained from real-world counterparts using 3D scanners, surface textures are

[1] `http://www.google.com/glass/start`
[2] `http://www.oculusvr.com`

often taken from digital photographs and even the lighting is reproduced using captured high dynamic range light probes.

Yet, the digitization of complete real-world objects remains a challenging issue for the fields of computer vision and computer graphics. First approaches often described objects only by their geometry, possibly in conjunction with texture-mapped pictures. Although this form of representation is still current practice in many applications, in recent years the research goal has shifted towards obtaining more faithful digital reproductions. This requires not only a high-quality reconstruction of the geometry of the acquired object, but also of its optical material appearance. For this, it is important to note that the appearance of an object, i.e. its visual impression to an observer, comes from the interaction of light with the object's surface and interior. Light coming from all different directions is absorbed, scattered or reflected by the object and, eventually, some of it reaches the eye. For a faithful digital reproduction, these interactions of light with the object's surface have to be captured, digitally represented and later appropriately simulated during rendering.

In this work, we will investigate the digitization of the full appearance of an object and the involved tasks of its transmission and rendering.
As argued above, the resulting *digital replicas* can be useful for games, special effects or advertisement. In addition, faithful object digitization can be used for the creation of *virtual surrogates* of real-world objects, e.g. for display of fragile or precious cultural heritage artifacts or products in online shops. The captured accurate digital material models also greatly benefit computer-based product design and virtual prototyping. They can even help to aid real-world production processes by providing a well-defined specification of the desired appearance.

1.1 Digital Material Appearance

The appearance of materials and objects is an important stimulus for the human perception. It influences the overall impression of an object and even invokes emotions. For instance, casings made from brushed metals appear more valuable than casings made from plastics, furniture made from wood is perceived as warm and cozy, and cloth that has a silky appearance is perceived to be cooler and smoother than cloth made from wool fabrics. These effects are well known (see the study by Giesel *et al.* [GZ11b]) and are for instance utilized in industrial product design. The ability to have these associations is deeply rooted in our nature. In the case of foods, we are used to gauge the freshness based on the appearance of the surface. For human skin, we are even able to see subtlest differences and assess attributes such as healthiness or mood of our fellow men.

Figure 1.1: *The digital replica of a Ganesha figurine made from labradorite, a mineral showing a play of colors. This object exhibits drastic changes in appearance depending on the viewpoint and direction of illumination, changing from a greasy impression of soapstone to a bright blue gleam.*

We consider optical *appearance* to be the impression of an object when perceived by a human observer. Unfortunately, this impression can depend on psychological factors, such as mood or expectations. For the sake of simplicity we will largely ignore these subjective influences in our line of argument. Instead we aim to capture and reproduce appearance at the level of the stimulus in the human visual system.

Still, as demonstrated in Figure 1.1, the impression can change drastically, depending on various inherent and external factors. Among these factors are the geometry and spatial variation of optical properties of the object itself as well as aspects related to the observation, such as illumination and point of view. For example, different angles of illumination will lead to changes in the distribution of highlights and shadows. In addition, the nature and magnitude of this depend on geometry and optical properties of the object. In Figure 1.1, not only the intensity but also the hue of parts of the object's surface is subject to change.

Furthermore, human perception is trained to assess appearance of materials in combination with the given environmental factors. For example, the perception of hues is constantly adapted to the white point of the surrounding light sources. It is therefore necessary to explicitly consider this dependency. Here, photorealistic rendering techniques can be employed to generate images of complete virtual scenes, correctly conveying the visual impression.

To achieve this, the realistic reproduction of surface materials plays an important role. In the real world, the visual impression of materials is the result of complex light scattering within small geometric structures on and under the surface. For a restricted set of materials, such as perfect mirrors, some metals or plastics, a visually pleasing rendering can be achieved by employing physically motivated, analytical reflection models. However, the majority of the rich variety of material classes encountered in everyday life cannot as easily be represented by simple analytical models.

Therefore, a viable alternative is the usage of data-driven light scattering models. Here, the result of the light interactions on the surface is represented by tabulated data. Depending on the density or resolution of the data, this correctly conveys even very complicated effects that would be hard to describe with a mathematical model having only few parameters. This can be seen in analogy to vector graphics versus raster graphics for the representation of digital images. Vector graphics have beautiful properties, such as exact precision, small file sizes, lossless editing and transformation and often also intuitive forms of description, e.g. construction from geometric shapes. However, raster graphics are the predominant format to represent the full richness of natural images. This is mainly because here the simpler model-driven descriptions of vector graphics do not suffice. In addition,

Figure 1.2: *Example of virtual surrogates acquired and presented with the proposed approach. The grid-lines on the background show a centimeter raster for comparison. Rendering was performed with the Mitsuba path tracer [Jak10]. Figure 7.1 shows the same scene rendering in real-time on the GPU.*

data-driven representations lend themselves for the purpose of digitization: An acquisition device simply needs to measure all the entries. In case of the image example, digital cameras directly capture raster images.

Yet, in contrast to 2D images, full material appearance is much harder to capture. The amount of light that is reflected towards the observer from the object's surface is dependent on a large number of variables, such as position of entry and emission, directions of entrance and of observation, wavelength or time (see Section 3.2.4). This large dimensionality makes a full acquisition infeasible at reasonably high resolutions, due to the necessary measurement times and the amount of data. We therefore concentrate only on a subclass of material effects to enable the practical capture and representation of objects. We neglect fluorescence, phosphorescence and polarization as well as transparency or translucency. The remaining effects, which are still sufficient to represent a large number of materials encountered in everyday life, can faithfully be expressed using the *bidirectional texture function* (BTF) [DVGNK97]. Often these materials are still not perfectly opaque. That is, parts of the incident light is not reflected immediately at the point of entrance but is transported below the surface and can be observed at another spot on the object. If this subsurface light scattering is restricted to an area close to the point of entrance, we denote it "locally subsurface scattering". In contrast to alternative material representations, the BTF is capable of reproducing this appearance. We discuss this in more detail in sections 3.2 and 3.3.

As a result, we are able to build a working pipeline to create faithful virtual surrogates of 3D objects made from opaque as well as locally subsurface scattering materials. Figure 1.2 demonstrates the variety of objects that fulfill these criteria and can be represented via BTFs.

1.2 Organization and Main Contributions

In this thesis, we propose an integrated, automated, high-quality acquisition for obtaining both, a highly detailed 3D geometry of an object and its faithful material appearance. We furthermore present solutions to transmit and interactively inspect the obtained digital replicas via GPU-based real-time rendering on ordinary PC hardware and over the Internet. We will in detail explain the individual steps and components involved in this acquisition and reproduction pipeline and roughly follow the logical workflow from measurement over processing to transmission and rendering of virtual surrogates of a physical object. We present several results that have successfully been accomplished and discuss strengths and weaknesses of the overall approach.

The thesis is organized in four thematic parts. A part is divided into multiple chapters, each containing an independent set of contributions and evaluations. Together the developments in the single chapters constitute a complete pipeline. Later chapters utilize the devices, algorithms or data introduced in the prior ones.

Part I gives an **introduction** to the tasks approached in this thesis, providing a problem description and the necessary background. Chapter 2 will present challenging examples that should be handled by the presented digitization approach. These examples serve as an informal problem description and provide a vivid impression, making it easier to understand and follow our reasoning in later parts. In Chapter 3, we will provide the reader with the necessary background on the representation of material appearance with the BTF, recapitulate some necessary preliminaries and introduce our notation. Finally, we discuss related work that is also concerned with digitization of full object appearance.

The technical contributions can be found in Part II and Part III.
Part II deals with the **integrated acquisition** of the geometry and material appearance. In this part, we first present an in-depth discussion of suitable measurement devices in Chapter 4. Here, our main contributions beyond the state of the art are the detailed description of design and implementation of three hardware setups, a thorough comparison of the proposed setups and a thorough review and comparison of other approaches found in the literature. Chapter 5 investigates the necessary processing steps to derive an efficient and compact digital representation from the raw measurement data. In this chapter, we contribute to the state of the art by describing a novel resampling and hole filling algorithm for BTF data and performing a broad evaluation of the approach.

Part III describes the solutions found for **transmission and rendering** of the digitized objects. This part is comprised of two mostly orthogonal techniques. Chapter 6 presents a novel streaming approach and an implementation of a browser-based renderer that allows the quick rendering of the virtual surrogates over the Internet. This chapter contributes to the state of the art by the application of a wavelet-based image compression scheme for BTF transmission. A key factor is the efficient implementation of the decompression in a browser. The chapter furthermore introduces a novel precomputed ambient term to real-time BTF rendering. A different focus is given in Chapter 7. Here we explore a new hierarchical level of detail rendering scheme for the memory efficient real-time rendering of BTFs. The main contribution to the state of the art is the novel representation that unifies two independent level of detail hierarchies found in BTFs as well as a streaming approach based on the level of detail.

Finally, Part IV provides **closure**. This last part summarizes the presented details and gives an outlook on prospects of future research.

1.3 Publications

Most of the content presented in this thesis has already been published:

- Christopher Schwartz, Michael Weinmann, Roland Ruiters, and Reinhard Klein. Integrated high-quality acquisition of geometry and appearance for cultural heritage. In *International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage (VAST)*, pages 25–32, Prato, Italy, October 2011. Eurographics Association.
- Christopher Schwartz, Roland Ruiters, Michael Weinmann, and Reinhard Klein. WebGL-based streaming and presentation framework for bidirectional texture functions. In *International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage (VAST)*, pages 113–120, Prato, Italy, October 2011. Eurographics Association. Awarded as best paper.
- Christopher Schwartz, Michael Weinmann, Roland Ruiters, Arno Zinke, Ralf Sarlette, and Reinhard Klein. Capturing shape and reflectance of food. In *SIGGRAPH Asia 2011 Sketches*, pages 28:1–28:2, Hong Kong, China, December 2011. ACM.
- Christopher Schwartz and Reinhard Klein. Acquisition and presentation of virtual surrogates for cultural heritage artefacts. In *EVA 2012 Berlin Conference on Electronic Media and Visual Arts*, pages 50–57, Berlin, Germany, November 2012. Gesellschaft zur Förderung angewandter Informatik e.V.
- Christopher Schwartz, Ralf Sarlette, Michael Weinmann, and Reinhard Klein. DOME II: A parallelized BTF acquisition system. In *Eurographics Workshop on Material Appearance Modeling: Issues and Acquisition*, pages 25–31, Zaragoza, Spain, June 2013. Eurographics Association.
- Christopher Schwartz, Roland Ruiters, Michael Weinmann, and Reinhard Klein. WebGL-based streaming and presentation of objects with bidirectional texture functions. *ACM Journal on Computing and Cultural Heritage (JOCCH)*, 6(3):11:1–11:21, August 2013.
- Christopher Schwartz, Roland Ruiters, and Reinhard Klein. Level-of-detail streaming and rendering using bidirectional sparse virtual texture functions. *Computer Graphics Forum (Proceedings of Pacific Graphics)*, 32(7):345–354, October 2013.
- Christopher Schwartz, Ralf Sarlette, Michael Weinmann, Martin Rump, and Reinhard Klein. Design and implementation of practical bidirectional texture function measurement devices focusing on the developments at the University of Bonn. *Sensors*, 14(5):7753–7819, May 2014.

CHAPTER 2

APPLICATION SCENARIOS

2.1 3D Digitization for Cultural Heritage

In the process of writing this thesis, one focus has been especially on the application of the digitization technique in the field of *cultural heritage*. Here, the digitization of 3D objects has recently gained importance and is on the verge of becoming a standard tool for the practitioners. There are many reasons for this development. For instance, digitizing cultural heritage collections allows the safe and instant access to all items, be it for researchers, curators or the general public. 3D digitization may also serve a documentation purpose, capturing the state of an object prior and after restoration, before lending it to some other institution or for monitoring its decay.

Objects digitized with high quality can be used as *virtual surrogates* of their physical counterpart. This does not only take risks from fragile or expensive objects, but it even allows for forms of dissemination that would be impossible with the physical object. The virtual surrogate can be showcased at arbitrary many locations simultaneously without any significant additional costs. It can easily be put into any digital scene, e.g. to illustrate its possible location in a virtual reconstructed historical site or in a measured 3D model of the excavation site. Here, it can be combined with other virtual objects, e.g. to illustrate a hypothesis how different objects could have been used together. And last but not least, virtual surrogates can even be used for the public dissemination over the Internet, providing the capability to reach an enormous audience.

2.1.1 Requirements on Appearance Reproduction

Especially in the field of cultural heritage, where even subtle details can drastically change the interpretation of an object, it is important not to rely on simplified or exaggerated representations, but to capture and convey the full appearance. As mentioned in Section 1.1, this is the visual impression of the object on a human observer and depends on the object itself as well as the observation conditions. In this application, the difference of appearance under different conditions might matter. An object can make a different impression in a gloomy room lit by candles, under a cloudy sky, in a showcase with neon bulbs or in the bright sunlight.

Therefore, in order to fully understand and experience a 3D object, the observer should be able to inspect it from all sides and to put it under different illumination conditions. Providing such an interactive inspection experience is one of the key aspects of the technology explored in this thesis. This is for instance comparable to holding the physical object itself in the hand, turning it around, holding it closer or further away or keeping the physical object fixed while moving a light source around the object. Even walking around an object that is standing in a showcase is a bit of the same experience, but definitely less immersive and satisfactory.

In both cases, the inspection provides not only insight into the 3D shape of the object but also in the materials it is composed of. Depending on the orientation towards the observer or the light source, wear and scratches are revealed or patina or dirt becomes apparent. It is therefore important to faithfully capture and reproduce even such seemingly minor details of the appearance as well.

In the course of this thesis, we demonstrate that our proposed approach is capable of capturing the appearance at the necessary level of detail. This process requires an elaborate measurement procedure and produces large amount of measurement and temporary processing data. Yet, almost all steps are fully automatic and the proposed resulting representation – a medium resolution triangle mesh and a compressed high resolution BTF file – requires only about 1.8 GB disk space. Thus, the technology could enable the digitization of entire museum collections, bringing hidden treasures back to the surface.

2.1.2 Public Dissemination of Virtual Surrogates

The steep increase in bandwidth and ubiquitous availability of digital networks provides tremendous capabilities for public dissemination to cultural heritage institutions. For many media types such as historic documents, books, pictures, audio and video recordings, the Internet already offers compelling means of distribution and presentation. Here, the public dissemination of cultural heritage content on the Internet is a quasi-standard nowadays (e.g. [BHKD09]).

However, the presentation of 3D objects, ranging from the plastic arts to archeological artifacts, was not very well supported until lately. So far, photographs have been in use as an unsatisfying substitution. Unfortunately, images will never be able to capture the full experience of three-dimensional exhibits.

As argued above, the ability to discover the objects from every angle offers a considerably higher degree of insight, providing essential information about the shape, material and surface structure. The labradorite Ganesha figurine in Figure 1.1 is an excellent example why an interaction is superior over still images. The object shows an amazing play of colors when observed from just the right combinations of viewing angle and light direction. The intricate details of this effect can be best understood by intuitively turning the object back and forth or changing the lighting directions and observing the change of colors directly on the screen.

More recently, techniques for rendering of 3D geometries or 3D point clouds in the browser and their transmission over the Internet have emerged (e.g. [BEJZ09, DBPGS10, JBG11, MSB12]). However, many of the current solutions display the object with static colors that remain the same for all points of view. Such an approach is not sufficient for optically more complex and interesting objects like the aforementioned labradorite figurine.

By considering the usage of 3D objects together with BTFs in this domain, we explore a viable alternative in this thesis. Despite the comparably large file size of the digital master files for the virtual surrogates, we demonstrate in Chapter 6 that an effective and efficient progressive transmission of BTFs over the Internet is in fact possible. The interactive exploration of virtual surrogates is enabled from directly within the website in the browser and can start after less than one megabyte of the digital material appearance is transmitted, which takes only a few seconds with widely available DSL or 3G connections.

2.2 Virtual Surrogates of Dermatological Moulages

An interesting application for the techniques proposed in this thesis can be found at the cross section of cultural heritage, medical education and research.

From the 17th to the early 20th century, moulages, anatomic wax models depicting diseases, were used as the state-of-the-art visualization technique for medical education purposes. These wax models were cast using a mold that was directly taken from the respective body parts of hospital patients. The prepared models were then hand-painted and touched up to mimic the disease as realistically as possible. Different finishes have been used to depict pustules, scurf, tumors or weeping rashes. Carefully, even small details such, e.g. implanted body hair, have been applied to the moulages to increase the level of realism.

Figure 2.1: *Dermatological moulages rendered in our interactive WebGL viewer. Top row: a hand displaying the symptoms of psoriasis. Bottom row: a nose with basal-cell carcinoma. The user can view the objects from freely chosen vantage points and illumination conditions to highlight important details such as the plaques on the hand. In both cases, the whole piece, including the base plate and handwritten labels, is shown, also presenting the moulages in their context as cultural heritage artifacts. The images are adapted with kind permission by Prof. Thomas Bieber, Deptartment of Dermatology, University of Bonn.*

The moulages have been used at universities and hospitals in the education of the medical staff up to the 1950s. Then they eventually have been replaced by color photography, which prior to this point had been considered not to be of sufficient quality. Especially in the field of dermatology the exact reproduction of the appearance of the diseases is of great importance, as diagnoses are mostly made based upon visual cues. Here, the moulages provide a much richer impression of the disease's appearance than for example photographic images. Since they are plastic 3D objects with realistically crafted surfaces, change of perspective can reveal additional details or provide a better understanding of the condition of the diseased skin, e.g. how moistened or dry it is. Therefore, medical moulages have even been reintroduced in today's lectures at some universities, e.g. at the University of Zurich in Switzerland, who maintain a large collection in the museum of wax moulages together with the University Hospital Zurich [Gei12].

There is no doubt that the historic pieces themselves are an important part of our cultural heritage. First, they are manually crafted intricate pieces of art, made by masters of a trade that is rarely practiced today. They are thus of special interest for current practitioners and conservators as well as art historians. Second, they reflect the medical knowledge and the diseases of the time they have been made. In fact, many of them show diseases that are rare or considered eradicated today. Thus, they are also interesting from the point of view of a medical historian [Gei09].

Yet, even for diseases that are still encountered today, the moulages often provide better examples than even real patients would. Due to the better and earlier treatment in the modern health systems, the condition is usually cured before exhibiting all visual symptoms.

In their original purpose as a visualization for the education of medical practitioners, medical moulages have been used in the lecture room and even passed along through the rows by the students. Furthermore, they were on display for self-study, e.g. in a library, and still are today. However, moulages are expensive and very fragile. Being made of wax, they require constant temperature, air humidity and as little light exposure and agitation as possible to prevent damage. This severely limits the direct use of those pieces in current lectures.

The possibility of having virtual surrogates for the multiple purposes of these delicate pieces therefore holds a huge benefit for everyone: medical practitioners, researchers, teachers and students, art historians, conservators, current moulage makers and eventually the general public who can enjoy the display of medical heritage. Our proposed digitization, transmission and rendering techniques are especially useful in this case. First, the accurate reproduction of the appearance of the moulages is of great importance. The objects can show faint and subtle details. Visual apprehension of the disease pattern can rely on the ability to discover these details by taking different points of view or varying the light conditions. Our technique allows exactly these operations and reproduces the view- and light-dependent appearance of the original moulage. Second, the accessibility of the objects via the Internet is very helpful for students, e.g. for self-study, as well as for researchers, allowing concurrent access to objects in moulage collections all over the world, for which currently, due to logistical reasons, only pictures can be made available. Rendering and streaming of BTF content in a web browser as an enabling technology further facilitates this scenario: The virtual surrogates of the moulages can this way easily be linked from or even embedded into existing hypertext resources, such as online medical textbooks or museum collection websites, much like pictures are today.

The suggested usage of a BTF for the representation of the appearance instead of other possible forms, e.g. SVBRDFs (see Section 3.3), is clearly advantageous in the presented scenario. Although the subject of displaying human skin realistically using other reflectance models is well studied (see [PB11] for a recent example), to the best of the authors' knowledge, so far no efforts have been undertaken to find a solution for diseased skin. On the other hand, the data-driven BTF is well known for handling fine details like pores, scars or hairs rather well.

To evaluate the applicability of our approach, we acquired two exemplary moulages from the collection of the Department of Dermatology of the University of Bonn.

The details of the procedure will be described in the later chapters. The measured objects depict the disease of psoriasis of the nails and hand and basal-cell carcinoma of the nose. Figure 2.1 demonstrates how our web browser-based viewer is used to explore a virtual moulage. The dermatologists considered the virtual surrogate a useful tool and as a result of the first experiments expressed their interest in digitizing larger parts of the collection.

2.3 Capturing Shape and Reflectance of Food

Another possible application of the proposed acquisition and rendering techniques is the utilization of faithful digitized models in advertisement. Great endeavors are undertaken to obtain convincing photographic pictures of food. A good impression on the necessary effort is given in [BB08]. A whole art team might be working on a still shot, carefully optimizing every aspect, such as the arrangement and preparation of items on a table, the choice of perspective or the lighting. Furthermore, the depicted food items often need to be prepared extensively by professional food stylists in order to look “fresh” and “tasty”. It is even customary to create facsimiles from substitute foods or apply paint on the food.

Thus, food photography is both, a time-consuming and yet also a time critical procedure. Here, digital models could improve and speed up the creative process and avoid a waste of food, time and money. Digital assets need to be acquired only once and could be captured separately. An eye-pleasing composition can be put together without a hurry. The artist can experiment with different arrangements without fearing degradation, wear or deformations.

However, the 3D modeling of fresh food is an especially challenging example. Humans have developed a high degree of sensitivity to this subject matter, and, hence, even slightest errors may yield visually unconvincing results. Even subtle visual cues in surface reflectance can make the difference between a tasty and an unsavory or artificial impression. Therefore, huge efforts need to be undertaken for obtaining a realistic digital reproduction. In many cases, it is desirable that the generated content can be used in arbitrary synthetic scenes with possibly multiple objects, novel viewpoints and novel lighting. This requires a full geometric representation as well as spatially and directionally varying reflectance properties.

Therefore, we explore the use of our proposed 3D digitization technique for reproducing the appearance of real-world food items. This imposes serious restrictions on the acquisition pipeline: most fresh food degrades quickly. As a consequence, the acquisition process needs to be fast enough to avoid changes in appearance during measurement. In addition, food tends to deform over time, which means

(a) separate rendering

(b) composite scene

Figure 2.2: *Renderings of the digitized foods under global illumination. We employ standard path tracing and use the "kitchen" HDR light probe of Paul Debevec [Deb98] for lighting. The rendering in (b) shows a virtual composition of the separately captured foods (a). The objects exhibit a realistic depiction of highlights and fine surface details, complex occlusions (e.g. leafs of the strawberries), shadowing (e.g. apples) and indirect illumination. Please note that the different fruits in (b) are not additionally captured items but rather multiple instances of the same digital objects, arranged in different poses.*

that both shape and reflectance have to be captured within a short time frame, ideally without moving the object.

In this thesis, we approach the problem using the highly parallelized Dome 1 acquisition system described Section 4.6. It is sufficiently fast to capture the appearance of fresh food in the limited amount of time. To reduce degradation effects, the room containing the setup was actively cooled to about 16° C. In all other aspects, the acquisition process follows the description in Section 4.6 and Chapter 5. The resulting proposed representation is compact enough to be of practical use, even for real-time applications.

Once a model has been acquired, it can be used to create photorealistic images, showing accurate shadowing and global illumination for almost arbitrary lighting conditions. Our results indicate that the proposed pipeline is able to handle even optically complex cases that normally require extremely costly manual modeling. With our BTF-based representation we manage to reproduce complex appearance details, such as crumbs in baked dough, little cracks in sugar coating, the seductive gloss on molten chocolate, the lightness of subsurface scattering apple pieces and strawberries and the savory appearance of baked crust on a crispy roast pork (see Figure 2.2). Table 5.1 contains a description of the captured food items.

2.4 Summary

To provide a good motivation as well as illustrate the requirements and challenges, this chapter outlined two specific example applications that benefit from techniques developed in this thesis.

The considerations on digitization and presentation for cultural heritage (Section 2.1) have been discussed as part of the following publications:

- "Integrated High-Quality Acquisition of Geometry and Appearance for Cultural Heritage" [SWRK11] as a paper at the VAST 2011 conference.
- "WebGL-based Streaming and Presentation Framework for Bidirectional Texture Functions" [SRWK11] as a paper at the VAST 2011 conference.
- "Acquisition and Presentation of Virtual Surrogates for Cultural Heritage Artefacts" [SK12] as an invited talk at the EVA 2012 Berlin conference.
- "WebGL-based Streaming and Presentation of Objects with Bidirectional Texture Functions" [SRWK13] as an extended journal article in ACM JOCCH.

The application of the proposed techniques for the digitization of food items (Section 2.3) has been published as a dedicated technical sketch "Capturing Shape and Reflectance of Food" [SWR*11] at the SIGGRAPH Asia 2011 conference.

CHAPTER 3

PRELIMINARIES

The contents discussed in parts II and III of this thesis mostly rely on fundamentals of computer graphics. However, especially the acquisition and processing described in Part II also require a basic knowledge of radiometry as well as computer vision. In this chapter, we aim to provide a brief summary of those aspects of all three fields of research that are most relevant to this thesis and thereby also introduce our notation and terminology. Finally, we provide a basic description of the data formats and error metrics that are employed throughout this thesis.

3.1 Basic Notation

Although every rule has its exception, for the sake of easy readability we attempt to follow a fixed notation in all equations found in this thesis:

- The set of integer numbers and set of real valued numbers are denoted $\mathbb{N}$ and $\mathbb{R}$ respectively.
- Scalar variables are written in lower case italic font, e.g. $i \in \mathbb{N}$ or $\alpha \in \mathbb{R}$. Scalar constants are written in upper case italic font, e.g. N or M.
- Vector valued variables are written in lower case bold font, such as $\mathbf{v} \in \mathbb{R}^N$. A single element of a vector $\mathbf{v} \in \mathbb{R}^N$ with index $i \in \{1, 2, \ldots, N\}$ will be denoted by the same letter in lower case italic font, i.e. v_i.
- A vector with homogeneous coordinates (see Section 3.1.1) of a point $\mathbf{x} \in \mathbb{R}^N$ in Euclidean space will be denoted by the same letter in the same script with an upperscript h in front of it: ${}^{\mathrm{h}}\mathbf{x} \in \mathbb{R}^{N+1} = \mathbb{P}^N$. Here, $\mathbb{P}^N$ denotes the projective space for $\mathbb{R}^N$. Similarly, single elements of this vector are indicated the same way as before, i.e. ${}^{\mathrm{h}}x_i$.
- Matrices are written in upper case bold font, e.g. $\mathbf{M} \in \mathbb{R}^{M \times N}$. The element in the i-th row and j-th column of a Matrix $\mathbf{M} \in \mathbb{R}^{M \times N}$,

$i \in \{1, 2, \ldots, M\}$, $j \in \{1, 2, \ldots, N\}$, is indicated by the same letter in lower case italic font, i.e. $m_{i,j}$.
The j-th column-vector of a matrix $\mathbf{M}$ will be written as the same letter in lower case bold font, i.e. $\mathbf{m}_j$.

- If no more specific notation is given, the norm $\|\mathbf{v}\|$ of a vector denotes the 2-norm $\|\mathbf{v}\|_2 = \sqrt{\sum_i v_i^2}$.
  Similarly, if no more specific notation is given, the norm $\|\mathbf{M}\|$ of a matrix denotes the Frobenius-norm $\|\mathbf{M}\|_F = \sqrt{\sum_i \sum_j m_{i,j}^2}$.
- Sets are often written in fraktur letters, such as $\mathfrak{S} \subset \mathbb{R}$ or $\mathfrak{T} \subset \mathbb{R}^3$. For special sets (e.g. the set of directions on the hemisphere, see Section 3.1.3) we employ the usual symbol found in the literature (in this case Ω).
  Elements of a set are represented by the same letter in a font that indicates whether they are vectors or scalars, e.g. $s \in \mathfrak{S}$, $\mathbf{t} \in \mathfrak{T}$ or $\omega \in \Omega$.
- In cases in which functions have to be made clearly distinguishable from other elements, they will be written in calligraphic letters, such as $\mathcal{F}$.
- We use the customary SI prefixes *nano* (n), *micro* (μ), *milli* (m), *centi* (c), *kilo* (k), *mega* (M), *giga* (G) and *tera* (T) to indicate 10-based multiples or fractions of units, e.g. 1 millimeter (mm) = 10^{-3} m or 1 megapixel = 10^6 pixel. For numbers referring to computer memory, e.g. the size of a file, we also use the prefixes mega, kilo, giga and tera but follow the convention of using powers of 1,024, i.e. $1{,}024^3$ bytes= 1 GB.
- We generally write time in 24-hour notation, e.g. 1:30 instead of 1.5 hours.

3.1.1 Homogeneous Coordinates

Homogeneous coordinates are employed in computer graphics, computer vision and robotics as a common tool to express affine and perspective transformations via matrix multiplication [HZ04]. For this, coordinates of a point in Euclidean space (mostly in $\mathbb{R}^2$ and $\mathbb{R}^3$) are expressed via homogeneous coordinates in projective space (e.g. $\mathbb{P}^2$ and $\mathbb{P}^3$). Let $\mathbf{x} = (x_1, x_2, \ldots, x_N)^T \in \mathbb{R}^N$ be a point in N dimensional Euclidean space. This point is expressed in projective space $\mathbb{P}^N$ as a set of $N+1$ dimensional vectors ${}^{\mathrm{h}}\mathbf{x} = \left\{\alpha\,(x_1, x_2, \ldots, x_N, 1)^T | \alpha \in \mathbb{R} \setminus 0\right\}$, i.e. a line through the origin in $\mathbb{R}^{N+1}$. Note that all points on the line correspond to a single coordinate in $\mathbb{R}^N$. Vice versa, a vector in homogeneous coordinates ${}^{\mathrm{h}}\mathbf{y} \in \mathbb{P}^N$ can be converted back to it's corresponding point in Euclidean space as $\mathbb{R}^N \ni \mathbf{y} = \frac{({}^{\mathrm{h}}y_1, {}^{\mathrm{h}}y_2, \ldots, {}^{\mathrm{h}}y_N)^T}{{}^{\mathrm{h}}y_{N+1}}$. Projective coordinates with ${}^{\mathrm{h}}y_{N+1} = 0$ do not have a corresponding Euclidean point. However, $({}^{\mathrm{h}}y_1, {}^{\mathrm{h}}y_2, \ldots, {}^{\mathrm{h}}y_N)^T$ can then be interpreted as a direction, pointing towards a coordinate at infinity.

3.1.2 Representation of Digital Images

Throughout this thesis, we often deal with images. In our case, images are always defined on a rectangular region and have one or more color channels. We assume the color channels to be independent and can hence describe every image with multiple color channels as multiple monochromatic images. Mathematically, a monochromatic image with dimension $M \times N$ (in arbitrary units) can be defined as a function assigning each point in the image a grayscale value g:

$$\begin{aligned} \mathcal{I} : [0, M) \times [0, N) \subset \mathbb{R}^2 &\rightarrow \mathbb{R} \\ (r, s) &\mapsto g. \end{aligned} \tag{3.1}$$

However, digital images, e.g. taken by a digital camera or generated by a rendering algorithm, are usually represented as raster images. This means the image domain is discretized into a lattice with $W \times H$ pixels. Each pixel then holds a single grayscale value. We still assume this value to be a real valued number (see Section 3.7 for a discussion of digital representations for elements in $\mathbb{R}$). Hence, we can also think of a digital image as a matrix $\mathbf{I} \in \mathbb{R}^{W \times H}$.

In this thesis we make use of both forms. Depending on their context we consider images either as a matrix $\mathbf{I}$ or as a function $\mathcal{I}$. We use the same letter with the respective form of notation to indicate that both terms represent the same image. For the sake of simplicity, we assume that the domain of the function is given in pixels, i.e. $\mathcal{I} : [0, W) \times [0, H) \rightarrow \mathbb{R}$ if $\mathbf{I} \in \mathbb{R}^{W \times H}$. The values of the continuous image are derived via bilinear interpolation of the respective entries in the matrix.

Following convention, we also use the term *texture* to refer to images that represent texture, i.e. spatially varying appearance of a surface. Here, we use the term *texel* instead of pixel.

Note that it is also possible to understand a discretized image $\mathbf{I} \in \mathbb{R}^{W \times H}$ as a vector of pixels $\mathbf{i} \in \mathbb{R}^{W \cdot H} = (i_{1,1}, i_{2,1}, \ldots, i_{W,1}, i_{1,2}, \ldots, i_{W,H})^T$. Vice versa, values in a high-dimensional vector can also be considered as an image. We will make use of this idea in chapters 6 and 7 to store column vectors of a matrix as separate 2D textures on the GPU.

3.1.3 Directions and Solid Angles

An excellent overview on this topic can be found in the textbook "Principles of Digital Image Synthesis" [Gla95] by Andrew Glassner. In the following paragraphs, we provide a brief summary that introduces the utilized terms, concepts and notations.

A solid angle ω [sr] is the projected area of a cone of directions on the unit sphere. A single direction in 3D space has a differential solid angle, written as $\mathrm{d}\omega$. Confusingly, in computer graphics literature, it is customary to use the letter ω to refer to solid angles and directions alike. Often, ω is used to denote a direction and $\mathrm{d}\omega$ its corresponding differential solid angle. In this thesis, we follow this convention. In case it is not directly clear from the context, we explicitly mention whether the symbol refers to a direction or solid angle.

$\mathcal{S}^2 \subset \mathbb{R}^3$, i.e. the surface of a unit sphere, is the set of all directions in $\mathbb{R}^3$. Consider a point $\mathbf{s}$ on a surface. The orientation of $\mathbf{s}$ is given by the direction $\mathbf{n} \in \mathcal{S}^2$ perpendicular to the surface. We refer to the vector $\mathbf{n}$ as *normal direction* or – in short – *normal*. $\Omega \subset \mathcal{S}^2$ denotes the set of directions on the upper hemisphere with respect to the surface orientation. If hemispheres on multiple points are considered simultaneously, we use $\Omega(\mathbf{s})$ as an explicit notation. Similarly, we use $\Omega(\mathbf{n})$ in the case of multiple orientations. In contrast to [Gla95], we do not consider different signs to distinguish between incoming and outgoing directions, but regard all directions vectors, i.e. incoming directions $\omega_i \in \Omega$ and outgoing directions $\omega_o \in \Omega$, to be pointing outward from the position on the surface.

In the course of this thesis, we often deal with bidirectional functions that depend on both, incoming directions ω_i and outgoing directions ω_o. We thus introduce the additional set of bi-hemispheres $\Omega^2 := \Omega \times \Omega$. We denote its elements as $\omega_{io} \in \Omega^2$ and define them as a tuple $\omega_{io} := (\omega_i, \omega_o)$, first listing the incoming and then the outgoing directions. Additionally, we often use finite sets of directions as a discretized representation of the full hemisphere. We denote these sets $\mathfrak{L}$ for *light directions* and $\mathfrak{V}$ for *view directions*, which correspond to the incoming and outgoing direction hemispheres. Similar to Ω^2, we write $\omega_{io} \in \mathfrak{L} \times \mathfrak{V}$.

3.1.3.1 Direction Parameterization

In order to perform computations based on directions ω_i and ω_o, we need a suitable representation of the domain of directions from the upper hemisphere Ω. Given the normal direction $\mathbf{n}$, the set of all directions in the hemisphere can thus be expressed as Cartesian coordinates $\Omega_{\text{Cartesian}} = \{\mathbf{x} \in \mathbb{R}^3 : \|\mathbf{x}\| = 1 \wedge \mathbf{x} \cdot \mathbf{n} \geq 0\}$.

However, although the points on the hemisphere lie on the 2D surface $\mathcal{S}^2$, this representation requires 3D coordinates. This poses an unnecessary overhead.

A commonly used alternative are the *spherical coordinates* or *polar coordinates* θ, ϕ. Then, the domain of hemispherical directions can be expressed as $\Omega_{\text{polar}} = \{(\theta, \phi) \in [0, \frac{\pi}{2}] \times [0, 2\pi)\}$. Here, θ is the angle of inclination with respect to the normal $\mathbf{n}$. The angle ϕ is the azimuth angle and is measured on the plane orthogonal to the zenith direction with respect to a given reference direction. In our case, we choose one of the surface tangents $\mathbf{t}$ to be the azimuthal reference direction. Given directions $(x_1, x_2, x_3)^T \in \Omega_{\text{Cartesian}}$ in Cartesian coordinates, their corresponding polar coordinates can computed as

$$x_1 = \mathbf{t}\sin\theta\,\cos\phi, \qquad x_2 = (\mathbf{t}\times\mathbf{n})\sin\theta\,\sin\phi, \qquad x_3 = \mathbf{n}\cos\theta. \tag{3.2}$$

Polar coordinates have the advantage that the angles θ and ϕ have direct geometrical interpretations. Unless stated otherwise, we will consider directions to be given in polar coordinates throughout the remainder of this thesis. Intuitively, $\omega_{io} = (\theta_i, \phi_i, \theta_o, \phi_o)$, as illustrated in Figure 3.1a.

However, polar coordinates also have some disadvantages. First, they have a singularity at the zenith. Here, all coordinates with $\theta = 0$ and arbitrary azimuth ϕ refer to the same direction. This results in an ambiguity of the coordinates that needs to be handled consistently. Furthermore, an equidistant sampling in the domain of polar coordinates results in a distorted sampling of directions with a strong oversampling towards the zenith. The second disadvantage lies in the fact that the azimuthal angle is periodic and hence has a "wrap-around" at 2π. Depending on the application, this needs to be considered. Finally, due to azimuth flips at the zenith, the singularity and the periodicity, we cannot use the Euclidean distance in $\mathbb{R}^2$ to measure or even approximate the distance between directions.

Therefore, we occasionally employ *parabolic coordinates* $\mathbf{p} \in \Omega_{\text{parabolic}} \subset [-1, 1]^2$ as a third form of direction parameterization:

$$x_1 = 2\mathbf{t}\frac{p_1}{\|\mathbf{p}\|^2+1}, \qquad x_2 = 2\,(\mathbf{t}\times\mathbf{n})\,\frac{p_2}{\|\mathbf{p}\|^2+1}, \qquad x_3 = \mathbf{n}\frac{1-\|\mathbf{p}\|^2}{\|\mathbf{p}\|^2+1}, \tag{3.3}$$

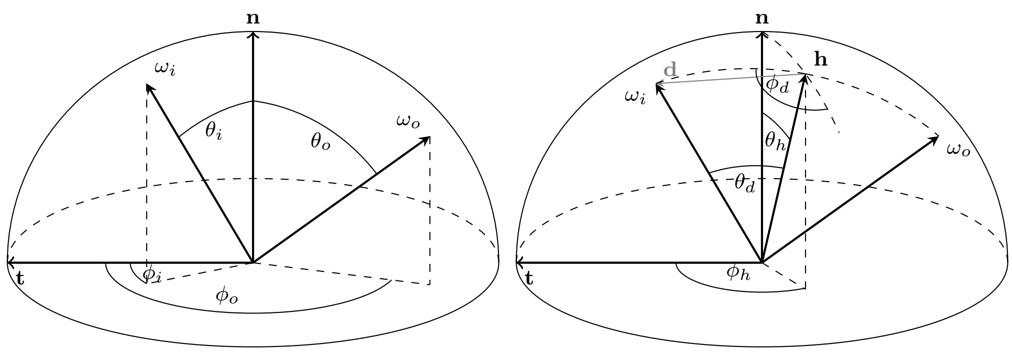
$$\theta = \arccos\frac{1-\|\mathbf{p}\|^2}{\|\mathbf{p}\|^2+1}, \qquad \phi = \arccos\frac{p_1}{\|\mathbf{p}\|}, \tag{3.4}$$

with $(x_1, x_2, x_3)^T \in \Omega_{\text{Cartesian}}$ and $(\theta, \phi)^T \in \Omega_{\text{polar}}$.

Parabolic coordinates have the zenith located at the origin. Points of equal inclination (i.e. equal angle θ in polar coordinates) are arranged on a circle with radius dependent on θ:

$$(\theta, \phi)^T \to \left(\frac{\sin(\theta)\cos(\phi)}{1+\cos(\theta)}, \frac{\sin(\theta)\sin(\phi)}{1+\cos(\theta)}\right)^T. \tag{3.5}$$

This avoids the ambiguity at the zenith (the zenith falls onto a single point) and any wrap-around issues. The circles in $\mathbb{R}^3$, described by the azimuth, are in $\mathbb{R}^2$ again mapped to circles. Furthermore, within local neighborhoods, the Euclidean metric on the parabolic coordinates offers a good approximation of the relative distances between points on the hemisphere.



(a) "in/out" parameterization **(b)** Rusinkiewicz parameterization

Figure 3.1: *Common parameterizations of pairs of light and view directions. In the case of the "in/out" parameterization in (a), the bidirectional configuration is directly characterized by the light and view directions* (ω_i, ω_o)*, e.g. by their polar coordinates* $(\theta_i, \phi_i, \theta_o, \phi_o)$*. The Rusinkiewicz parameterization in (b) instead employs the derived halfway and difference vectors* $\mathbf{h}$ *and* $\mathbf{d}$*, e.g.* $(\theta_h, \phi_h, \theta_d, \phi_d)$

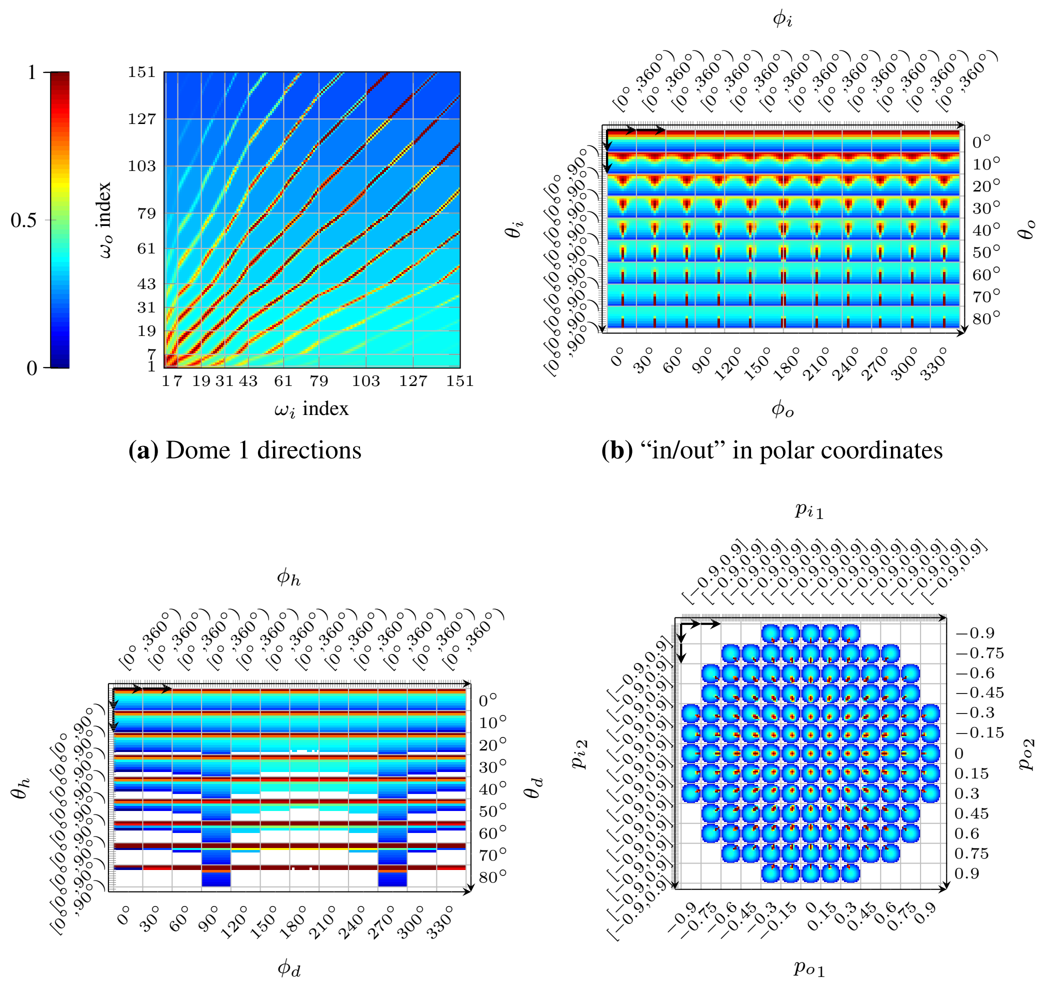
The disadvantage of this parameterization is the fact that the hemisphere is not mapped to a rectangular domain but only a disc (see Figure 3.2d). Thus, when mapping the pixel coordinates of an image or texture into directions given as parabolic coordinates, about 21% of the pixels cannot be used. This issue is also discussed by Guthe *et al.* [GMSK09]. They propose to introduce two modifications to the mapping, rectifying the occupancy problem and achieving a better area preservation. However, we did not attempt to follow these optimizations, as they are not critical to our application.

Finally, we also need the *Rusinkiewicz parameterization* [Rus98], which was developed in the context of bidirectional reflectance and explicitly considers pairs of directions. Here, the direction pair (ω_i, ω_o) is expressed as a *halfway* $\mathbf{h} \in \Omega_{\text{Cartesian}}$ and a *difference* direction $\mathbf{d} \in \Omega_{\text{Cartesian}}$. Consider $\omega_i, \omega_o \in \Omega_{\text{Cartesian}}$ to be given as Cartesian coordinates, then the halfway vector, which is simply the bisector of the arc between them, can be computed as

$$\mathbf{h} = \frac{\omega_i + \omega_o}{\|\omega_i + \omega_o\|}. \tag{3.6}$$

The difference vector describes the direction from the halfway vector $\mathbf{h}$ to ω_i. It can be obtained by applying a rotation transformation that aligns the halfway vector with the pole on ω_i, i.e.

$$\mathbf{d} = \mathbf{R_b}(-\theta_h)\, \mathbf{R_n}(-\phi_h)\, \omega_i. \tag{3.7}$$



(a) Dome 1 directions

(b) "in/out" in polar coordinates

(c) Rusinkiewicz in polar coordinates

(d) "in/out" in parabolic coordinates

Figure 3.2: *Values of a BRDF (Cook-Torrance [CT82] with* k_d=0.5*,* k_s=0.5*,* m=0.2*,* F=0.2*), regularly sampled in different direction parameterizations. (a) uses the light and view directions of the Dome 1 setup, described in Section 4.6, ordered by their indices (see Table 4.2). (b) and (c) regularly sample four dimensional polar coordinates. In (b), the light and view directions* ω_i *and* ω_o *are sampled directly via* $(\theta_i, \phi_i, \theta_o, \phi_o)$*. Instead, in (c) the halfway and difference vectors* $\mathbf{h}$ *and* $\mathbf{d}$ *of the Rusinkiewicz parameterization were sampled as* $(\theta_h, \phi_h, \theta_d, \phi_d)$ *and light and view directions were derived. Finally, (d) illustrates the distribution of samples when employing a regular grid of parabolic coordinates* $\mathbf{p}_i, \mathbf{p}_o \in [-1, 1]^2$*. In (c) and (d), samples that do not fall on the upper hemisphere are left white.*

Here the halfway vector $\mathbf{h}$ is for convenience given in spherical coordinates (θ_h, ϕ_h). The matrix $\mathbf{R_n}(-\phi_h) \in \mathbb{R}^{3\times 3}$ describes a rotation θ_h around normal direction $\mathbf{n}$, aligning the azimuthal angle of the halfway vector with the reference tangent $\mathbf{t}$. Similarly, the matrix $\mathbf{R_b}(-\theta_h)$ rotates around the bitangent $\mathbf{b} = \frac{\mathbf{n}\times\mathbf{t}}{\|\mathbf{n}\times\mathbf{t}\|}$, bringing the halfway vector onto the normal direction $\mathbf{n}$. Please see Figure 3.1b for an illustration. Often, the resulting halfway and difference vectors are themselves then expressed in spherical coordinates: $(\theta_h, \phi_h, \theta_d, \phi_d)$. In this form, the Rusinkiewicz parameterization has the property that important BRDF features, i.e. the highlight and the direction of anisotropy, are aligned with the coordinate axes (see Figure 3.2c).

3.2 Physical Background

Trying to understand the scattering and distribution of light has a long standing tradition in science. It would be completely out of the scope of this thesis to provide a comprehensive summary. However, to be self contained, we will briefly touch those facts and findings that are most important to our application. We consider the presented facts and arguments common knowledge and will not provide citations for every single one. However, we recommend the already mentioned textbook by Andrew Glassner [Gla95] as a source of reference.

3.2.1 Nature of Light

The interaction of light with matter is a well studied phenomenon in the field of physics. Such interactions are categorized by the scale in which they occur and studied in separate disciplines.

On the smallest scale, *quantum mechanics* denotes a quantum of light as *photon*. The photon is the boson that carries the electromagnetic force and interacts with other elementary particles. Although important for understanding effects such as phosphorescence and fluorescence, simulating large systems on this level of granularity is computationally infeasible. Fortunately, it is also not necessary, since a single quantum mechanic interaction has basically no influence on the perceived image. We will therefore not consider quantum effects but instead fall back to higher levels of abstraction found on larger scales.

The second scale, called *physical optics* or "wave optics", considers light as an *electromagnetic wave*. This subdiscipline studies the interaction of electromagnetic waves with structures at approximately the size of the wavelength, i.e. approximately $380\ \mathrm{nm}$ to $780\ \mathrm{nm}$ for visible light. Considerations at this scale explain phenomena such as interference, diffraction and polarization. Yet, since the involved structures are so small, an accurate simulation of these phenomena is not practical for the envisioned application.

Hence, we build upon the models that developed in the considerably older field of *geometric optics* or “ray optics”. Here, light propagation is modeled via *rays*. Geometric optics and wave optics are well connected, as the “light ray” describes the direction of propagation of the corresponding electromagnetic wave.

In computer graphics, it is common to adopt the ray model from geometric optics to simulate light propagation in virtual worlds. Besides direction and origin, a light ray holds additional properties, such as wavelength, polarization and power of radiation. These properties are accounted for when computing reflections and refractions and hence provide a simplified model for the effects found in wave-optics. Although diffraction and interference cannot be correctly simulated at this level of abstraction, the model is usually a very good approximation if the geometric structures the light interacts with are considerably larger than the wavelength of light. This is usually the case for structures that are perceivable by the naked eye or conventional photographic cameras. Therefore, this approximation suits our purpose of appearance reproduction.

3.2.2 Light Ray Attributes

As mentioned in above, a light ray holds several attributes. In accordance with the mathematical definition of a ray, a light ray has a **point of origin** $\mathbf{o} \in \mathbb{R}^3$ and a (signed) **direction** $\mathbf{d} \in \mathcal{S}^2$. Using the Cartesian coordinates of the direction, any point on the ray is hence described by

$$\begin{aligned} r : \mathbb{R}^+ &\rightarrow \mathbb{R}^3 \\ t &\mapsto \mathbf{o} + t\,\mathbf{d}. \end{aligned} \tag{3.8}$$

Then, a light ray has a spectral power distribution $\lambda \in \mathbb{R}^+ \mapsto L \in \mathbb{R}^+$, mapping **wavelength** λ of the related electromagnetic wave to its amount of **radiance** L. For our purposes, we usually consider λ between $380\,\mathrm{nm}$ (violet) and $780\,\mathrm{nm}$ (red) – a range signified as the *visible spectrum*. The adjacent wavelengths below $380\,\mathrm{nm}$ are denoted *ultraviolet* (UV). Those above $780\ \mathrm{nm}$ are called *infrared* (IR). Both are not visible for a human observer.

In Chapter 4, we briefly discuss multispectral measurement devices. Here, the continuous spectrum is discretized into a set of spectral bands $\Lambda = \{[\lambda_i - \frac{x}{2}, \lambda_i + \frac{x}{2}]\}_i$ with bandwidth x. However, in most parts of this thesis we instead only consider tristimulus values: red, green and blue (RGB). These three values can be obtained from a spectral power distribution via inner products with suitable color matching functions (e.g. CIE RGB [SG31]).

Radiance L measures the radiant power per unit solid angle per unit projected area [W sr^{-1} m^{-2}], a basic unit of radiometry. Some other quantities of radiometry are relevant to this thesis as well. We will therefore briefly recapitulate their properties and relationships. A comprehensive overview can for example be found in [PG10].

- Q [J]: The *radiant energy*.
- Φ [J s^{-1}] or [W]: The *radiant power* or radiant flux. This is the radiant energy per unit time: $\Phi = \frac{\mathrm{d}Q}{\mathrm{d}t}$.
- I [W sr^{-1}]: The *radiant intensity*. The differential amount of radiant power that is emitted into a differential solid angle $\mathrm{d}\omega$: $I = \frac{\mathrm{d}\Phi}{\mathrm{d}\omega}$.
- L [W sr^{-1} m^{-2}]: The *radiance*. The radiant power per unit solid angle passing through a surface patch with differential area $\mathrm{d}A$: $L = \frac{\mathrm{d}I}{\mathrm{d}A} = \frac{\mathrm{d}\Phi}{\mathrm{d}\omega\,\mathrm{d}A\cos(\theta)}$. Here, θ is the inclination angle of the the direction ω with respect to the orientation of the surface patch. With its single direction ω and infinitesimal small cross-section $\mathrm{d}A\cos(\theta)$, the radiant power of a hypothetical single light ray exactly corresponds to this measure.
- E [W m^{-2}]: The *irradiance*. The total radiant power incident from the upper hemisphere on a surface with unit area: $E = \frac{\mathrm{d}\Phi}{\mathrm{d}A} = \int_\Omega L\cos(\theta)\mathrm{d}\omega$. Here, Ω is the set of directions over the hemisphere above the infinitesimal small surface patch.

Assuming vacuum, the radiance remains constant along a ray [Gla95]. Hence, it is viable to use it as an attribute for light rays in the simplified model of geometric optics. The presence of a medium, e.g. air or water, can be accounted for during propagation simulation by introducing corresponding interaction events that change the power of the light ray.

Another possible attribute of a light ray is the polarization, i.e. the orientation of the electromagnetic wave perpendicular to the propagation direction in space. Although the human visual system can faintly detect the polarization of light and polarization of light can be utilized with the help of tools, e.g. watching stereoscopic movies with 3D glasses, employing polarization filters, it is in most practical situations not perceivable. Please refer to [WW11] for a short list of cases in which simulating polarization in computer graphics might be necessary. We therefore neglect this attribute in the scope of this thesis.

3.2.3 Propagation Simulation

Instead of explicitly simulating the interplay of waves or even quantum effects, in geometric optics the propagation of light in a scene is described by mathematical rules that influence a ray's attributes. For instance at the interface between two media, the direction and wavelength of a light ray might be changed. Several approaches exist that allow the efficient computation of the light propagation following this description. They mainly vary in complexity and the optical effects that can efficiently be described.

In its most general form, the properties of an emitted light ray at a point in space may be computed by the *full radiance equation* [Gla95]. The equation describes the radiance $L(\mathbf{x}, \omega_o, \lambda, \mathbf{e}, t)$ in dependence of all attributes of a light ray, i.e. origin $\mathbf{x}$, direction ω_o, wavelength λ, polarization $\mathbf{e}$, and even time t to model delayed reemission as found in phosphorescence. The term is formulated as a recursive function that simulates the propagation of radiance in a scene, considering all mentioned variables at interaction events. However, in his textbook, Glassner also states that there is no hope of solving this equation analytically and for efficient numerical solutions, reasonable approximations have to be found.

Common approximations are to neglect the polarization, as already discussed in Section 3.2.2, dispose of the time dependency and to assume that there is no energy transfer between different wavelength bands. The latter allows to simulate the light propagation for each wavelength band independently. Hence, in literature, many equations related to the propagation or scattering of light do not explicitly include wavelength as a parameter. We adopt this shorter notation whenever possible.

Furthermore, it is often assumed that the light rays travel through vacuum. Then, the full radiance equation is simplified to the *rendering equation*:

$$L_o(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \int_\Omega f(\mathbf{x}, \omega_i, \omega_o)\, L_i(\mathbf{x}, \omega_i) \cos\theta_i \, \mathrm{d}\omega_i. \tag{3.9}$$

The rendering equation was formulated by James Kajiya in [Kaj86] as a unifying mathematical framework for prior attempts to model light propagation. It describes the radiance L_o of light emitted at a point $\mathbf{x} \in \mathbb{R}^3$ into the direction ω_o. The domain Ω denotes the set of directions on the hemisphere above the surface at point $\mathbf{x}$. ω_o and ω_i denote the incoming and outgoing directions. θ_i is the angle of inclination of the direction ω_i with respect to the surface normal $\mathbf{n}$ at point $\mathbf{x}$.

The function f is called scattering distribution function and models the rules of light interaction with the surface in a probabilistic way. In the case of the rendering equation, the scattering function only depends on the point on a surface $\mathbf{x}$ and the incoming and outgoing directions ω_i and ω_o.

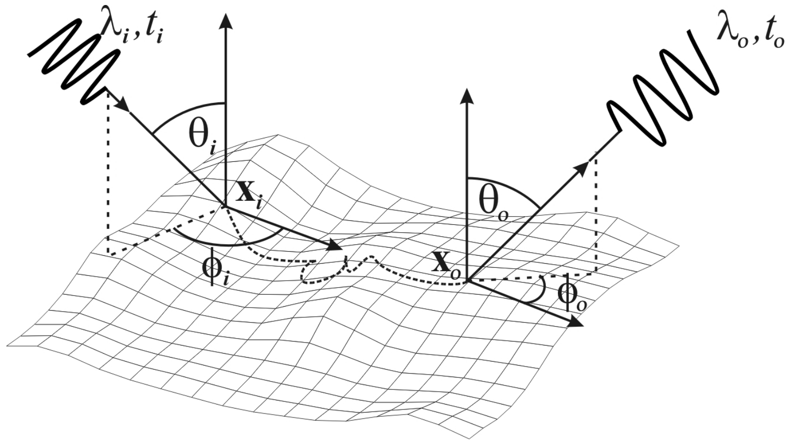


Figure 3.3: *The parameters of the 12D scattering function, enabling the description of light interaction with an object's surface. Image taken from [MMS*04].*

Popular examples in computer graphics that aim to simulate the light propagation in a scene under the given approximations and thus numerically solve Equation 3.9 are path tracing [Kaj86], bidirectional path tracing [LW93] and metropolis light transport [VG97]. Although later approaches of bidirectional path tracing also include simulation of participating media, e.g. [LW96], for our case we focus on light interaction that occurs at the interface between matter and air and neglect scattering in participating media outside the object and non-local light scattering within the object.

Synthetic images that are generated by such techniques contain almost all optical effects a human can perceive in everyday situations. However, in order to provide convincing results, the virtual scene that is the basis of the simulation has to be realistic as well. Therefore, instead of improving or developing a synthesis algorithm, we concern ourselves exactly with the acquisition and representation of the scattering distribution function f and the surfaces on which the points $\mathbf{x}$ can be found. The advocated bidirectional texture function representation $\rho(\mathbf{x}, \omega_i, \omega_\cdot)$, described in Section 3.2.6, can directly be employed in the rendering equation instead of f.

3.2.4 The Scattering Function

In order to provide a better understanding of the advantages of the BTF for the applications outlined in Chapter 2, we compare its properties to other models for light scattering. We do so by employing a top-down approach. We start with the most complete model for scattering and gradually derive simpler models.

In the most general case, the light transport within an object can be described using the 12D *scattering function* [VH74]

$$S(\omega_i, \mathbf{x}_i, \lambda_i, t_i; \omega_o, \mathbf{x}_o, \lambda_o, t_o). \tag{3.10}$$

Then, $p := S\ \cos\theta_o\ \mathrm{d}a_o\ \mathrm{d}\omega_o\ \mathrm{d}\lambda_o\ \mathrm{d}t_o$ denotes the probability of a photon that hits the object to be scattered and observed exiting the object. The scattering can occur directly at the surface or inside the object.

The probability density p depends on the photon entering

- with wavelength λ_i
- coming from direction ω_i
- at surface position $\mathbf{x}_i$
- at the time t_i

and leaving the object

- through a surface patch of size $\mathrm{d}a_o$ at position $\mathbf{x}_o$
- in a time interval $\mathrm{d}t_o$ at time t_o
- into cone with solid angle $\mathrm{d}\omega_o$ around direction ω_o
- within wavelength range $\mathrm{d}\lambda_o$ containing λ_o.

The direction ω_i is also often referred to as the illumination or *incoming* direction. Similarly, ω_o is denoted as the viewing or *outgoing* direction. Both are given as local directions in $\Omega(\mathbf{x}_i)$ and $\Omega(\mathbf{x}_o)$, respectively. A schematic diagram of the parameters can be found in Figure 3.3. Since the scattering function is defined for a surface, $\mathbf{x}_i$ and $\mathbf{x}_o$ are 2D vectors, describing the point on the surface rather than a point in 3D space. In combination with the usage of polar coordinates for the directions, this results in the mentioned twelve parameter dimensions.

In order to be completely general, all of the twelve parameter dimensions would need to be sampled during capture and considered during image synthesis, using the above mentioned full radiation equation. In addition to the problem of efficiently solving the full radiation equation (discussed in Section 3.2.3), sampling the full parameter domain is a hard problem as well. Independently, measurements of subsets of the parameter domain have been performed. Often with several restrictions with respect to to the materials and geometries that can be acquired. For example, in [JMLH01], Jensen *et al.* measure the scattering of light below the surface for a multitude of materials. However, they impose several restrictions on the materials. First, the materials have to be “optically thick”. That means no light from behind will be scattered through the material sample. Marble and jade would be two examples of materials fulfilling this condition. Second, the probes need to be perfectly planar as well as spatially homogeneous. Finally they assume that the light propagation below the surface is isotropic and independent of the angle of incidence. Hullin *et al.* [HHA*10] captured the angular dependency as well as the fluorescence of several materials. Yet, they only consider opaque and spatially homogeneous materials that are applied on a sphere geometry. The reason is that a complete measurement, accounting for all kinds of material effects, is prohibitively costly and would be extremely time-consuming. To the best of our knowledge a full 12D acquisition cannot yet be performed on arbitrary 3D objects with any single device.

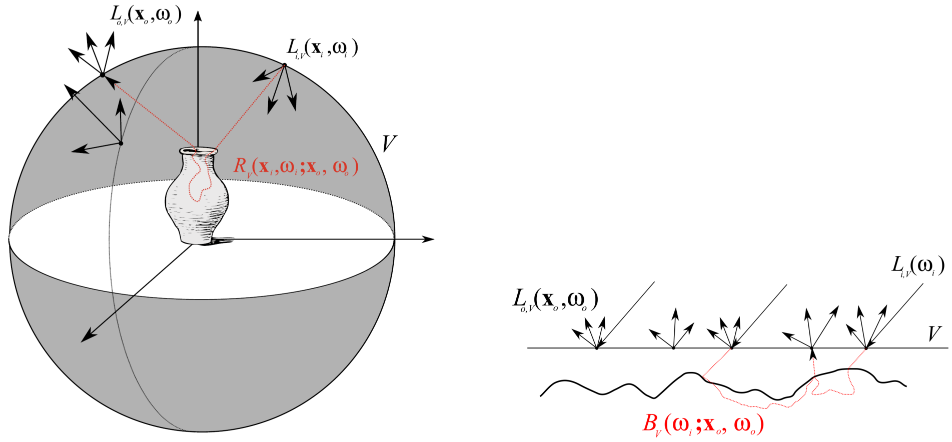
When neglecting the distribution of radiance over wavelengths and time, the scattering can be modeled using the 8D *bidirectional scattering-surface reflectance distribution function* (BSSRDF) [NRH*77]

$$S(\omega_i, \mathbf{x}_i; \omega_o, \mathbf{x}_o). \tag{3.11}$$

For this reduced set of parameters, measurement setups are within grasp. However, in practical applications it may often not be desirable or even possible to describe the true surface of an object. In the case of clothing, for example, the real surface, i.e. the boundary between the object-matter and air, would be the surface of the individual fibers that are spun to yarn, woven to fabric and sowed to the item of clothing (see 3.5). Describing the appearance of clothes by the scattering of light at the level of individual fibers requires a simulation with an enormous complexity and a complete model of the surface, including every single fiber. Although for highly specialized domains such as cloth this approach is actually followed [SZZ12], in more generic applications, often an approximation of the surface must suffice. The fine details of the true surface that make up the material appearance should then be captured by an appropriate reflectance model such as the BTF.

3.2.5 Reflectance Fields

When considering a single moment in time and a fixed wavelength, the radiance for all light rays originating at a point in space $\mathbf{p} \in \mathbb{R}^3$ and heading into direction $\omega \in \mathcal{S}^2$ is described by the *plenoptic function* $P(\mathbf{p}, \omega)$ [AB91]. Taking a photographic picture, observing a scene with the naked eye or synthesizing a computer generated image all boil down to sampling a 2D slice of the respective plenoptic function. This insight can be used for the generation of novel images without computing a full blown optical simulation of the light scattering on the true scene surface. Instead, the plenoptic function can be sampled from a real-world exemplar using photographs and the appearance from new vantage points can be reconstructed from these samples.



(a) reflectance field **(b)** BTF

Figure 3.4: *Illustration of the observation that the outgoing light field $L_{o,V}(\mathbf{x}_o, \omega_o)$ depends on the incident light field $L_{i,V}(\mathbf{x}_i, \omega_i)$, both parameterized over the surface of the bounding volume V. This dependency is fully described as the reflectance field $R_V(\mathbf{x}_i, \omega_i; \mathbf{x}_o, \omega_o)$, shown in (a). When considering a planar bounding surface and far field illumination one can derive the BTF $B_V(\omega_i; \mathbf{x}_o, \omega_o)$ depicted in (b). Images taken from [Mül09].*

In the absence of a participating medium or solid occluders, the radiance and thus the plenoptic function is constant along rays. Consider an arbitrarily complex surface that is encapsulated in a virtual bounding volume V, such that the observer is always located outside V. It is then sufficient to sample the radiance originating from points $\mathbf{x} \in \partial V \subset \mathbb{R}^2$ on the surface of the bounding volume into the outbound directions $\omega_o \in \Omega(\mathbf{x})$ to faithfully reconstruct the appearance for a given, static illumination [GGSC96, LH96]. The 5D plenoptic function $P(\mathbf{p}, \omega)$ is reduced to a 4D *light field* $L_{o,V}(\mathbf{x}, \omega_o)$ parameterized over the bounding surface of V. Similarly, if the observer is always inside the bounding volume, the appearance of a completely static scene on the outside is fully described by the light field $L_{i,V}(\mathbf{x}, \omega_i)$ with inbound directions $\omega_i \in \Omega(\mathbf{x})$ [LH96]. We deliberately choose the letter L here, as the light field describes radiance.

However, light fields only sample static scenes, i.e. fixed lighting, objects and materials. Varying the illumination will lead to completely different plenoptic functions. Debevec *et al.* [DHT*00] make the observation that for a given bounding volume V the outgoing light field $L_{o,V}$ is directly dependent on the incident light field $L_{i,V}$ (see Figure 3.4a). The authors use this to describe the exitant radiance from V under every possible incident illumination as an 8D *reflectance field*

$$R_V(L_{i,V}; L_{o,V}) = R_V(\mathbf{x}_i, \omega_i; \mathbf{x}_o, \omega_o). \tag{3.12}$$

Given that the observer and illumination are outside of V, the reflectance field can be used for reconstructing appearance under arbitrary new viewpoints and illumination conditions. For this purpose, the outgoing light fields are sampled under a set of basis incident light fields. New light variants are reconstructed as a linear combination of the illumination basis by exploiting the principle of superposition.

Note that the reflectance field is closely related to the BSSRDF S in Equation 3.11. When using the true surface as ∂V, both functions are identical. Yet, the approximation using a boundary surface makes the reflectance field easier to sample and reconstruct, as was attempted in [DHT*00, LCV*04, GTLL06, CNR08, TMY12] (the setup used in [TMY12] has previously been published in [MTK*10]). In turn, this means that reflectance fields, and thus also BTFs (see Section 3.2.6), are completely eligible for use in optical light scattering simulations as if they were BSSRDFs. The only restriction is that all scattering events occur outside and on the bounding volume. The light scattering within the bounding volume is already encoded in the reflectance field and does not need to be simulated.

3.2.6 Bidirectional Texture Functions

When assuming far field illumination, i.e. the sources of the illumination are always infinitely far away, the incident radiance for a given direction ω_i is the same at all points $\mathbf{x}_i$: $L_{i,V}(\mathbf{x}_i, \omega_i) = L^d_{i,V}(\omega_i)$ (see Figure 3.4b). This reduces the dimensionality to a 6D reflectance field that is called the *bidirectional texture function* (BTF) [DVGNK97]:

$$B_V(\omega_i; \mathbf{x}_o, \omega_o). \tag{3.13}$$

For the purpose of representing material appearance, the proxy surface ∂V is usually considered to be planar, as depicted in Figure 3.4b, but can in principle still be an arbitrary surface bounding the sample.

Much like the reflectance field is related to the BSSRDF, the BTF closely resembles the *spatially varying BRDF* (SVBRDF) $\rho_{\mathbf{x}}(\omega_i; \omega_o)$. The SVBRDF basically assigns a separate *bidirectional reflectance distribution function* (BRDF) to each point on the surface. Yet, by definition a BRDF $\rho(\omega_i; \omega_o)$ [Nic65] implies physical properties which are only satisfied if scattering is a completely local phenomenon, i.e. $\forall \mathbf{x}_i \neq \mathbf{x}_o : S = 0$ in Equation 3.11. Therefore, SVBRDFs cannot account for non-local light transport, such as subsurface scattering.

Even for objects that are completely opaque and therefore could fulfill the locality property, this is only the case if $\mathbf{x}$ is parameterized on the true surface. As argued at the end of Section 3.2.4, the notion of a true surface is often ill posed or extremely complex. When assuming only an approximate geometry V, materials often exhibit even more non-local appearance effects. The reflectance behavior at one point on a surface can be influenced by neighboring material inherent structures, e.g. small fibers in wool yarn of woven fabrics, which are not considered in the proxy geometry. These structures may cast shadows or interreflections, occlude the point from certain views or transport light via subsurface scattering, thereby breaking the assumption of locality. Furthermore, these structures may very well be smaller than the spatial resolution that is used in the digitized representation of the material. Therefore, the orientation of the surface itself, and with it the direction-dependent reflectivity, could vary within one spatial sampling point.

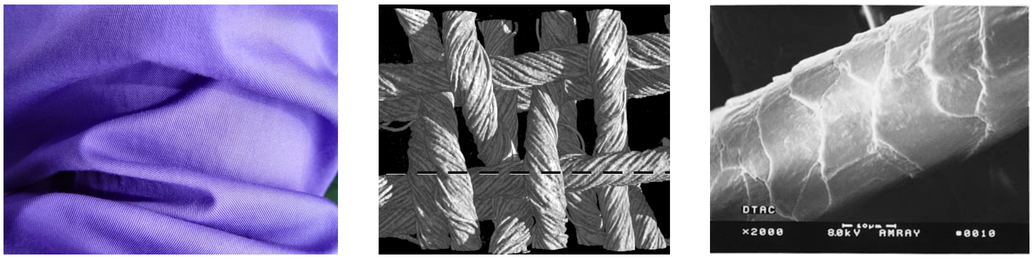
The outgoing light fields in the BTF by their nature include all of these subtleties. Thus, the BTF is applicable to faithfully capture the appearance of many kinds of optically dense materials. That is, materials that at most exhibit localized subsurface scattering. In theory, the BTF also captures light fields of materials that contain subsurface scattering over larger distances. However in practical rendering situations, the underlying assumption of distant illumination is easily violated. This would then result in an incorrect appearance reproduction.

On small patches of the object surface, however, the variation in illumination is usually so low that the distant illumination assumption is still valid. One important and frequently encountered exception is the sudden change in illumination due to shadow boundaries. Thus, unless the object is perfectly opaque and the true surface geometry is used instead of a proxy surface, the appearance cannot be reproduced completely correctly by this method in the vicinity of hard shadow boundaries. Nonetheless, high-quality results have been reported using BTFs for reproducing the appearance of many material samples [DVGNK97, HP03, SSK03, KMBK03] as well as complete objects [FKIS02, MBK05]. We show several of our high-quality results of objects with BTFs in Chapter 5.

3.3 Digital Appearance Representation

The remaining open question is how the different ingredients that are necessary for computing the light propagation simulation should be represented. On the one hand, optical effects should be faithfully reproduced. On the other hand the simulation should still be efficient and practical.

Without participating media, the only light scattering interactions happen at the boundary surfaces of objects. These interactions can roughly be grouped by the size of the geometric structures they are influenced by. The literature distinguishes



(a) macroscopic scale[1] **(b)** mesoscopic scale[2] **(c)** microscopic scale[3]

[1]CC-BY 2.0 Scott Robinson. [2]Taken from [STOK05]. [3]Taken from [Rob02].

Figure 3.5: *A piece of woven fabric illustrating the different scales considered for light interaction. In (a) only the clearly visible large wrinkles are apparent as 3D geometry. An X-Ray CT scan (b) reveals the geometry of the yarns and their numerous fibers. The micrograph (c) taken with a scanning electron microscope finally even shows the microscopic scales found on an individual wool fiber.*

between features on three different scales. Westin *et al.* [WAT92] denote them *object scale*, the *milliscale* and the *microscale*. In this thesis, we follow the notation proposed by Fournier [Fou92b] and use the terms *macroscopic*, *mesocsopic* and *microscopic* scale, respectively.

Structures on the **macroscopic scale** scale can be regarded to define the shape and geometry of a model, independent of what would be considered material. For instance, in a piece of clothing this would be the general shape (e.g. the "T" shape of a T-shirt) and possibly also wrinkles in the fabric, shown in Figure 3.5a.

At the opposite end, there is the **microscopic scale**. On this level, the light interacts with microscopically small facets on the surface of a material, such as scales on wool fibers (see micrograph in Figure 3.5c). These *microfacets* are orders of magnitudes below the resolution of the human eye. However, microfacets have a major influence on the view and light direction-dependent appearance of the surface: The amount of reflected light can be influenced by the orientations and the facets can occlude or cast shadows or interreflections on one another. Although not directly visible to the human eye, the microfacets influence the probabilities for photons from an incoming direction ω_i to be reflected towards an outgoing direction ω_o, and by this determine the view- and light-dependent intensity and shape of highlights.

In between the two extremes lies the **mesoscopic scale**. Features in this scale are just still visible with the naked eye but are usually not considered to be part of the defining shape of an object, such as the small fibers in wool yarn in Figure 3.5c. Other examples might be scratches and patina on worn objects or holes in the surface of an eggshell, which lead to its cavernous appearance.

3.3.1 Mesh, SVBRDF and Bump Map

In most actual rendering implementations, each of the three scales has a different form of representation so it can be most efficiently captured, represented and employed during rendering.

At the macroscale, explicit 3D geometry is used. Structures at this scale can be efficiently captured by existing 3D acquisition methods, which are discussed in Section 3.4. Eventually, the data is often stored as polygon meshes. During simulation, points on the surface are approximated by the surface of the polygons. Alternative approaches based on volumetric or implicit surface representations exist but are not well supported in prevalent real-time renderers or path tracers.

At the microscale, this approach would be infeasible. Devices that capture geometric structures at this granularity in 3D are rare, extremely expensive and have a rather limited working volume. Similarly, storing and simulating this incomprehensible amount of tiny structures directly for a complete object exceeds practical computational and memory capabilities. Here, it is instead common to model the effects of the microscale structures via statistical distributions, i.e. using a BSSRDF or a BRDF. This approach is a very good choice for homogeneous materials and several measurement techniques exist. The distribution of a BRDF can for instance be measured with gonioreflectometers (see Section 4.3.1) or even with more light weight setups [MPBM03]. The measured data or a compressed version can directly be used for rendering. Alternatively, one of many available analytical BRDF models can be fitted to the data. However, the BRDF or BSSRDF becomes problematic if the material composition varies over the surface of the object. Then, many different distributions might have to be employed, presenting challenges for both, acquisition and memory efficient representation. The distributions have to be fitted to the measured data, accounting for the local geometry. However, while the effort to capture and store the data will in the worst case be comparable to that for BTFs, the fitting is the true challenge in case more complex models are employed. A comprehensive overview for acquiring and representing BSSRDFs and BRDFs can be found in [WLL*09].

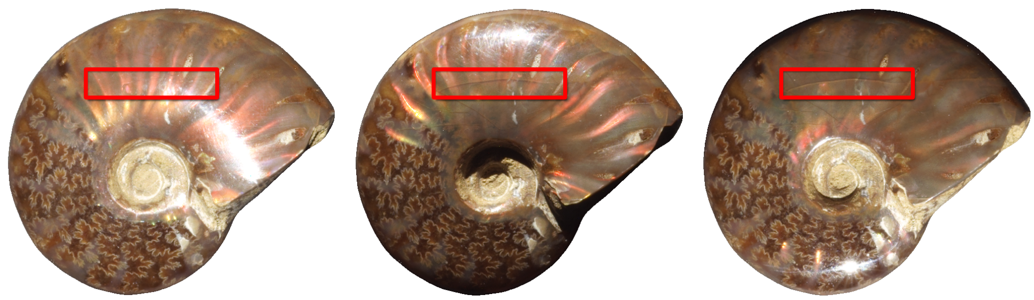
The mesoscale calls for a third solution, as it contains structures that do not fit into either of the categories. They are not defining the overall shape but are still visible by the naked eye. Representing them directly with a polygonal mesh requires an enormous effort during scanning, a lot of storage space for the representation and vast computational resources during rendering to really perform a fine-grained light simulation on that many polygons. Westin *et al.* [WAT92] propose to use a statistical distribution approach to model the influence of mesoscale structures on the surface, similar to the microscale. However, this is often not feasible as well: Mesoscopic details can be clearly and individually identified by a human observer.

Thus, an exact appearance reproduction, which might be necessary for certain applications (e.g. cultural heritage, see Section 2.1), would not be possible.

These features are therefore often encoded in a texture, which is then used to perform bump mapping [Bli78], normal mapping [Fou92b, Fou92a, COM98], parallax mapping [KTI*01] or displacement mapping [Coo84, Max88]. The first three techniques aim to add the effect of additional mesoscale structures by modifying ω_i, ω_o – and in case of parallax mapping also $\mathbf{x}$ – prior to evaluating the scattering function f in Equation 3.9. While bump mapping and normal mapping influence the shading, they cannon convey non-local effects. Later extensions to parallax mapping [Tat05] account for self-occlusion (masking) but can still not reproduce interreflections or self-shadowing. Displacement mapping adds the mesoscale structures by refining the macroscale polygon mesh. This form of representation saves space on disk and in theory allows non-local effects, simulating the propagation with the refined geometry. However, this again raises the issue of the inefficient computation due to the large number of polygons.
Either way, all four approaches require to somehow capture these fine structures to create the representative textures. This either necessitates an extremely precise 3D scanner or image space methods, such as photometric stereo (see Section 3.4). Both do not work well for materials that are not completely opaque. Finally, the texture is limited in resolution. The influence of structures that are slightly smaller will not be accounted for. None of these techniques can reproduce the effect of mesoscopic structures that lie within or slightly below the surface. See Figure 3.6 for an example of such a structure.

Nonetheless, there are several acquisition approaches that digitize the complete appearance of objects in this form [SWI97, LKG*03, GLL*04, HLZ10, PCDS12, NJRS13]. All of these approaches first acquire the macroscale geometry, either with a 3D laser scanner [SWI97, GLL*04], structured light [LKG*03, HLZ10, PCDS12, NJRS13] or computer tomography [LKG*03]. Then a set of images from different points of view and different light positions is captured. With the exception of Holroyd *et al.* [HLZ10], all approaches additionally employ a normal map to represent mesoscale geometry. The normal map is either estimated from the captured data [SWI97, LKG*03] or preserved from a captured high resolution mesh prior to a simplification [PCDS12, NJRS13]. Finally, analytical models, describing the microscale effects, are estimated. Several approaches [SWI97, HLZ10, PCDS12] first estimate the diffuse albedo color and then perform a second fitting round to find the direction-dependent specular properties.

In [LKG*03, HLZ10, PCDS12], the surface is clustered and a set of representative analytical basis BRDFs is fitted. Then, the spatially varying distribution of the basis BRDFs is estimated and stored in a texture map [LKG*03, PCDS12] or as per-vertex attributes of the mesh [HLZ10]. Sato *et al.* [SWI97] fit parameters for



(a) light from front **(b)** light from above **(c)** light from below

Figure 3.6: *An iridescent ammonite fossil, showing subsurface meso-structure effects. Notice the horizontal fissure in the material (highlighted by a red box) that is only visible when lit from above (b) and below (c). This structure is not a bump or crack that protrudes or intrudes the surface, but is rather located within the material. Yet, it changes the appearance in dependence of the incoming light direction. Dismissing this effect in a virtual surrogate would in the worst case prohibit discovery of the fissure at all.*

an analytical BRDF model at selected points on the surface and interpolate them linearly in a texture map. Nöll *et al.* [NJRS13] estimate separate parameters of a single analytical BRDF model for each texel in a texture map instead of a mixture of a few analytical basis BRDFs.

A notable exception is the approach of Goesele *et al.* [GLL*04]. Instead of an analytical BRDF, they fit a model for diffuse subsurface scattering. Their model is a simplification of the full BSSRDF from Equation 3.11 that neglects the angular dependency and only depends on the incoming and outgoing positions on the surface: $S(\mathbf{x}_i, \mathbf{x}_o)$. Due to abandoning angular dependency, a normal map is not necessary as well.

The setups vary in the amount of automation and integration. In [SWI97, HLZ10, NJRS13] an integrated approach is employed, capturing geometry and the images with varying view and light directions with the same setup. In [LKG*03, GLL*04, PCDS12], the geometry is digitized by a separate device, requiring a registration to the images. While [SWI97, HLZ10, NJRS13] capture all data automatically, in [LKG*03, GLL*04, PCDS12] the acquisition is performed manually. Lensch *et al.* [LKG*03] place the camera and light source at selected positions in a lab environment. Mirror-spheres are employed to calibrate the light source position. Goesele *et al.* [GLL*04] also manually place the camera at different positions. Palma *et al.* [PCDS12] perform their procedure under uncontrolled illumination conditions. A circular sequence of images from around the object is captured with a hand-held video camera. The position of the most dominant light sources is then estimated using a heuristic based on highlight detection and surface normals of the 3D geometry.

In this thesis, we propose the use of an integrated and fully automatic setup (see Chapter 4). However, due to the shortcoming of the representation with normal map and BRDF/BSSRDF, we employ a different form of representation that we discuss in Section 3.3.2.

The topics of transmission and rendering of meshes with BRDFs and bump maps (or normal maps, etc.) are not explicitly addressed by any of the publications listed in this section. This is probably for two reasons. First, the employed analytical BRDF models have only a few terms and can be evaluated very efficiently and in a straight-forward manner on the GPU. Second, the resulting data in the form of a simplified polygon mesh and several texture layers, which can be compressed with lossless image compression, is already very compact. Hence, the authors probably saw no practical need to investigate streaming or progressive transmission of the data.

3.3.2 Image-based Rendering

A different avenue is followed in *image-based rendering* (IBR). Here, the full appearance of a scene or object is reconstructed from images. Image-based rendering techniques have the advantage that no restrictive model assumptions have to be made. The synthesized image is obtained via interpolation of samples that are captured by a photographic camera. Hence, the result includes all optical effects that have been visible to the camera, regardless of the geometric scale of the structures. However, image-based representations usually require a large amount of data, making efficient compression and rendering an important consideration for each of the discussed techniques.

In its most simple form, the renowned `Quicktime VR`, proposed by Shenchang Chen [Che95], already constitutes an image-based rendering method. The author presents techniques to provide (very restricted) rotating and orbiting camera movements in virtual environments. The rotation movement is achieved by reprojecting 360° panoramic images. To render viewpoints in an orbiting camera movement, a corresponding image closest to the current camera view is displayed. Each image is compressed separately. The potentially large amount of redundancy in the data is not exploited.

Another very simple approach to image-based rendering was proposed by McMillan and Bishop [MB95]. Here, images from new viewpoints are computed by “warping” the pixels of an image to a new position. In order to do so, the depth information at each pixel is required. However, this actually already constitutes a geometric representation of the scene. The method introduces major visual artifacts, resulting from exposure of parts of the scene that were not visible in the original view. Furthermore, view-dependent changes in reflectance are not reproduced.

Light field rendering, first proposed in [LH96, GGSC96], is a more elaborate example that allows free camera movement without explicit knowledge of the scene geometry. Levoy *et al.* [LH96] take numerous images of a scene from a regular grid of camera positions and assign each pixel a ray in a 4D ray space. In their work, the space is parameterized by two planes: The plane of the camera positions (camera plane) and a parallel plane with some offset (focal plane). Radiance of arbitrary light rays within this space is obtained via quadrilinear interpolation. The large amount of data requires the use of more agressive lossy compression techniques. The authors propose to use vector quantization. However, this approach has the limitation that structures that do not lie on the focal plane become blurred by the interpolation.

A solution to this problem can be found by employing a hybrid approach, using a polygon mesh geometry for macroscopic scale structures and image-based rendering for the remainder of the surface. In the case of light field rendering, Gortler *et al.* [GGSC96] follow an approach similar to Levoy *et al.*, but resample the rays onto a coarse geometry of the scene, improving the sharpness of interpolation. Miller *et al.* [MRP98] and Wood *et al.* [WAA*00] extend this idea to the notion of *surface light fields*. Here, the light field is directly parameterized via points $\mathbf{x} \in \mathbb{R}^3$ on the surface of a coarse proxy geometry and directions $\omega \in \Omega(\mathbf{x})$ on the hemisphere. $\Omega(\mathbf{x})$ is defined with respect to to the normal of the proxy geometry at $\mathbf{x}$. This reparameterization to the surface results in improved sharpness during interpolation and increased compression efficiency. Recently, Palma *et al.* [PDCS13] presented a solution to estimate surface light fields from images shot at irregular positions, captured by a hand-held video camera.

However, surface light field techniques all consider outgoing light fields L_o. While this allows them the efficient and faithful reproduction of captured scenes or objects, the resulting virtual scenes need to be static and do not allow for a change in illumination. Yet, as the example of the fissure in the ammonite in Figure 3.6 demonstrates, the possibility to change the illumination could be crucial to convey all appearance details of an object. It is furthermore not possible to coherently combine separately acquired surface light fields in a novel virtual scene: The shading of the objects would not match, no shadows or interreflections could be cast, etc. However, for a number of applications, reaching from inspection to presentation, it would be desirable to have this option. Consider for example presenting digitized cultural heritage artifacts in different contexts and arrangements such as an exhibition site, the excavation site or a virtually reconstructed historical environment. In these examples, the objects need to be composed with their environment.

To overcome this limitation, several approaches have been proposed. Wong *et al.* [WHON97] extend the concept of light fields by additionally capturing variations in lighting for every view direction they take. Then, they fit spherical harmonic functions to each pixel. During rendering the spherical harmonic coefficients are first evaluated to obtain relit images. These images are then interpolated similar to ordinary light fields as in [GGSC96]. In their paper, Wong *et al.* make the important observation that the spherical harmonics at each pixel approximate the slice of an *apparent BRDF* (ABRDF). Unlike the BRDF definition by Nicodemus *et al.* [Nic65], the ABRDF may violate assumptions such as reciprocity or energy conservation and represents a more general distribution of the reflected light. This way, it is able to encode position-dependent non-local effects, such as interreflections, shadows, masking or subsurface scattering, cast from neighboring geometry onto the material surface. In [MGW01], Malzbender *et al.* present a similar approach, but propose to use polynomials instead of spherical harmonics, denoting the result *polynomial texture map* (PTM). Consecutive studies such as [EMM10, PCC*10] focus on the utilization of the different illuminations captured in a PTM for the enhancement of surface details. As a result, the technique is also often referred to as *reflectance transformation imaging* (RTI). Although Malzbender *et al.* only consider a single view, later publications, e.g. [GWS*09], use this technique with multiple views as well, following the rendering scheme of [WHON97] by first computing relit images which are then interpolated. Note, that the ABRDF-slices are fitted and stored per pixel in images of each view. This results in comparably large amounts of data. Apart from the fit of spherical harmonic or polynomial coefficients to a larger number of light directions, no compression is performed.

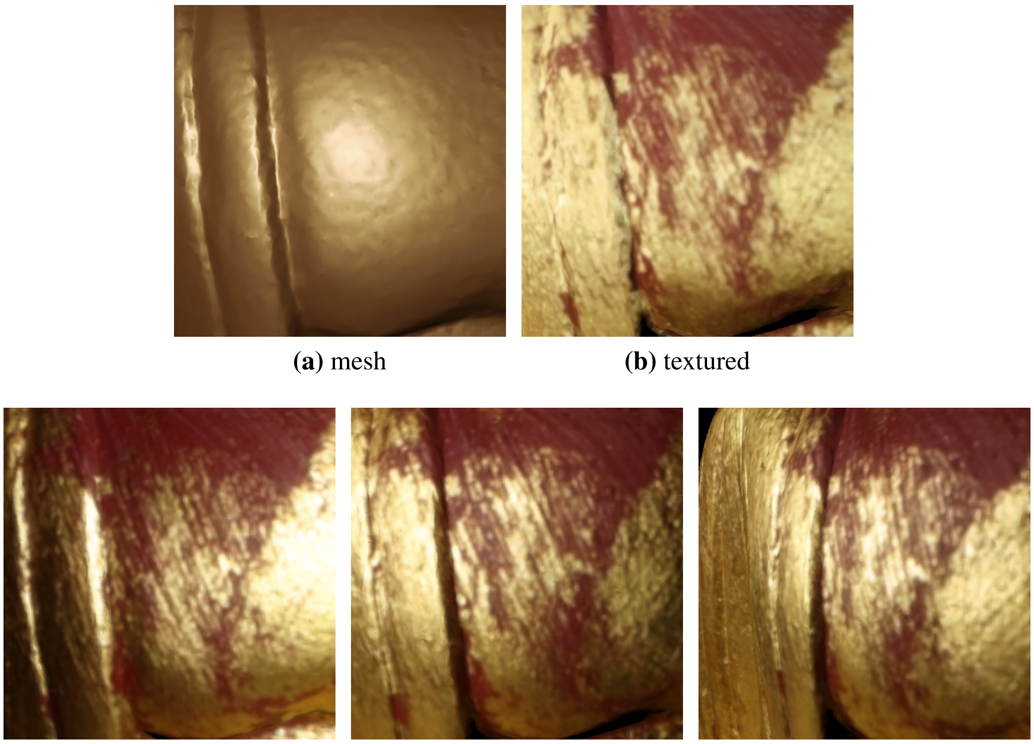
Debevec *et al.* introduced reflectance fields [DHT*00], combining incoming and outgoing light fields. The authors directly propose the use of surface geometry to parameterize the reflectance field. In analogy to the surface light field, they use the term *surface reflectance field*. The motivation behind this is again the increased coherence of the captured data, which improves sharpness and compression efficiency. However, in their practical implementation, Debevec *et al.* introduce several simplifications. First, far field illumination is considered, effectively making the reflectance field a BTF (see Section 3.2.6). Furthermore, the reflectance field/BTF is neither fully captured nor directly used for image-based rendering. Instead, a parametric model for the reflectance of human skin is fitted to the rather sparse set of captured data (see Section 4.8.2.3).

In contrast, Furukawa *et al.* [FKIS02] and Müller *et al.* [MBK05] present approaches that directly use the BTF to capture and reproduce the appearance of an object. As no further parametric model is fitted, the correct appearance reproduction is in a strict sense only possible under far field illumination as well. However, the authors make a convincing argument that it is a reasonable approximation to simply assume far field illumination for the application of material appearance representation – even under other illumination conditions. Compared to the size of the geometric details found in the true material surface, e.g. small bumps, cracks or fibers, the source of illumination is usually located several orders of magnitude away from the surface. Therefore, incident rays from one source are locally almost parallel and the spatial variation of illumination will in most cases be much lower than the spatial variation within the material. The only notable exception is the case of hard shadow boundaries, as mentioned in Section 3.2.6. Similar to surface reflectance fields, the BTFs are parameterized over the macroscale geometry of the object. This geometry is obtained from the objects silhouettes in the reflectance images using a visual hull approach [FKIS02, MBK05] or by a separate laser scanner [FKIS02]. Here, the view and illumination reflectance samples are completely represented in texels on the surface instead of slice-wise via pixels in image space [WHON97, MGW01]. This allows a far more efficient compression, as the additional redundancy in the view domain can be exploited.

Most recently, Ihrke *et al.* [IRM*12] proposed the use of a kaleidoscopic setup to capture surface reflectance fields of objects under far field illumination. They then estimate a set of basis ABRDFs and their spatial distribution from the measured data. The resulting representation is very closely related to factorized BTFs (see Section 5.5.4). However, instead of fully tabulated basis ABRDFs, Ihrke *et al.* use a combination of radial basis functions, implementing the suggested approach in [RPWL07]. Unfortunately, this approach is restricted to isotropic reflectance only.

Investigations in using high-resolution photographs with 3D geometry to provide a consistent texture for the surface (e.g. Dellepiane *et al.* [DCC*10]) may also be considered an attempt to acquire the surface reflectance. However, mesoscopic and microscopic information is only expressed by a single color, which (except for perfect Lambertian objects) is not enough for a faithful reproduction of different view and illumination directions (see for example Figure 3.7).

In this thesis, we therefore follow the approach proposed in [FKIS02, MBK05] to represent an object's appearance. We employ a polygon mesh to describe the *macroscopic* 3D geometry. Macroscale self-occlusions and self-shadowing are simulated using this geometry. For all optical effects that originate from structures at smaller scales, we use a BTF representation that is parameterized over the macroscopic surface. In contrast to the related SVBRDFs, the BTF is characterized by a unique ABRDF at each point of the surface. This way, it is able to account for optical effects that originate from complex or intricate surface structures, such as fur, fabric, cracks, bumps or fissions without the need to explicitly model them. Subsurface light transport is captured as well, and can in many cases be reproduced,



(a) mesh **(b)** textured

(c) BTF **(d)** BTF **(e)** BTF

Figure 3.7: *Impact of image-based representations of mesoscopic details on the appearance. The 3D mesh (a) (shaded with a uniform BRDF) is missing several fine details. The texture (b) can reproduce some details. However, texture is limited to a single viewpoint and a single light direction and hence details not seen in this particular combination will not be captured. In contrast, the BTF-based representation allows variation in both light ((c) versus (d)) and view direction ((d) versus (e)). Note for example the changes in shading and highlights on the bumpy diagonal grooves in the gold leaf, that occur in (c), (d) and (e). Please refer to figures 5.16 and 5.18, for comparisons between BTFs, SVBRDFs and PTMs.*

including complex instances of iridescence (e.g. figures 1.1 and 3.6). Faithful replication of a large variety of materials encountered in everyday life can be achieved.

Although raw BTFs exhibit rather unhandy file sizes of several tens to hundreds of gigabytes – in this thesis we work with high-resolution BTFs with up to two terabytes (see Table 5.2) – several compression techniques are available to cope with this problem. The compression, transmission, editing and rendering of BTFs have been active areas of research in the fields of computer graphics and computer vision for more than a decade. In recent years, BTFs have started to find application in industrial settings and in the domain of cultural heritage. Several recent surveys of existing techniques acknowledge that the BTF is a high-quality and very general model for describing digital material appearance [WLL*09, FH09, HF11, HF13].

3.4 3D Scanning

While the acquisition of the BTF to represent the mesoscale and microscale effects is an integral part in this thesis and will be discussed in detail in Part II, capturing the macroscale geometry of an object is an intensively studied subject as well with numerous approaches known in literature. Good overviews over existing techniques for 3D scanning can be found in [D'A06, STD09, IKL*10, HW11]. In this section, we will therefore only give a short overview on the taxonomy of methods for 3D geometry acquisition (see also Table 3.1) and discuss their applicability to our scenario. We refrain from giving detailed references and instead refer the interested reader to the mentioned surveys.

3.4.1 Non-optical 3D Scanning Methods

Early reliable devices to capture the 3D shape of objects worked **contact-based**. A feeler that is attached to a manipulable arm with multiple joints is manually placed on a point of the object by an operator. The 3D position is inferred by the angles of the joints. Although this method provides a robust way to acquire 3D points of almost all solid surfaces, regardless of the optical complexity, it has mostly been discontinued because of its two severe drawbacks: First, the shape is manually acquired point by point, making this a very cumbersome and time consuming approach with extremely limited resolution. Second, not all objects can be touched. This is easily conceivable for the application examples of digitizing sensitive and precious cultural heritage artifacts (Section 2.1). In addition, objects might deform

non-optical	**contact**	feelers on manipulable arms	
	tomography	magnet resonance imaging, X-ray computed tomography, ultrasonic sonography	
	time-of-flight	ultrasonic rangefinder, laser rangefinder	
optical		**active**	**passive**
	integration	photometric stereo, shape from specularity, shape from reflection, Helmholtz stereopsis	shape from shading
	triangulation	laser scanning, structured light	(multiview) stereo
	volumetric	structured light consistency, multiview normal field integration	photo consistency, shape from silhouette, shape from defocus

Table 3.1: *A taxonomy of 3D acquisition methods. This table makes no claim to completeness.*

under pressure of contact (e.g. fabrics), making it pointless to capture their shape this way. Therefore, the majority of employed scanners today work contactless. Still, there is a large variety of possible fundamentally different approaches (see Table 3.1).

Tomography captures not only the surface but a complete volumetric data set of an object. Depending on the employed tomography method, different sets of materials, including for instance glass, can be captured robustly. X-ray microtomography captures even very fine details, such as individual fibers in fabric. However, computed tomography scanners are extremely expensive and bulky. They furthermore need to enclose the object in full and are thus not suitable for an integrated acquisition approach that requires to capture additional images with varying view and light directions for the material appearance.

Time-of-flight distance measurement would not have this drawback. The employed sensors are sufficiently small to find application in several space restricted scenarios, such as robotics or as autofocus mechanism of some hand-held cameras. Similar to contact-based methods, time-of-flight rangefinders sample the distance pointwise. They are based on measuring the time difference between emission and detection of the reflection of either a laser beam or ultrasonic sound. Ultrasonic rangefinders are unaffected by the optical reflection and refraction properties of the material. This would make them applicable even for mirroring objects or glass. However, they have a poor spot accuracy in comparison to the laser-based method. Otherwise, both techniques have similar general restrictions. They achieve a mediocre precision and do not allow to capture multiple points at once. Hence, we do not consider them for our purpose.

3.4.2 Optical 3D Scanning Methods

To the best of our knowledge, the majority of other 3D scanning methods are based on optically measuring the reflection of light (visible or near infrared) from the surface to obtain a 3D model.

They can be classified in two dimensions: type of illumination (active versus passive) and method of depth inference (triangulation versus integration versus volumetric). Passive approaches only record the light reflected off the object from natural ambient illumination, whereas active approaches influence the illumination in a controlled manner. Integration methods first capture the surface normals. Based on the normals, the gradient vectors, i.e. the first derivatives of the surface, are computed and numerically integrated to reconstruct the surface. In contrast, triangulation methods directly infer the position of points on the surface via triangulation. Thirdly, volumetric methods represent the surface as an implicit function based on occupancy or consistency values. These values are computed for a complete volume in space. Eventually, a surface, e.g. an isosurface or minimal cut, is extracted.

Several appearance acquisition methods presented in the previous section rely on the volumetric **shape from silhouette** approach to obtain a proxy geometry, e.g. [GGSC96, FKIS02, MBK05]. This technique segments images from different sides of the object into foreground (the object) and background (the rest of the image). Then, a binary occupancy volume is created. If the projection of a voxel is inside the object's silhouette in all images, i.e. falls in the foreground region, it is marked as occupied. Finally, the surface of the object is obtained as the isosurface between occupied and unoccupied parts of the volume.

Shape from silhouette has the advantage that it does not require any additional scanning pass or scanner hardware. It can instead directly be used with reflectance measurement images. It is furthermore theoretically invariant to the surface material: If the object can be segmented from the background, it will work. In practice, however, finding the silhouette is a hard problem on its own. Fully automatic methods mostly rely on matting. This requires to introduce a matting background to the acquisition setup. In case of camera array setups or light domes (see Chapter 4), this would block opposing cameras or lights. Additionally, the quality of the reconstruction does not only rely on the quality of the silhouette but also on the number of available views. Reconstruction of smooth surfaces is impossible with a limited number of views. Yet, even with an infinite number, this technique can only reconstruct the *visual hull* of an object. The visual hull, however, cannot represent all concavities. While the visual hull would for instance correctly reproduce the shape of a handle on a coffee mug, it would not include the hollow inside of the mug itself. For these reasons, we refrain from using this approach.

Another possible passive volumetric approach would be **shape from defocus**. However, while the method is passive in the sense that it does not require a controlled illumination, it necessitates to change the distance of the plane of focus. The camera is set to a shallow depth of field and takes a sequence of pictures while the plane of focus is swept through the scene. Each picture only provides a sharp depiction of those points of the surface that lie close to the plane of focus. Based on this observation, a volumetric measure can be defined by evaluating the sharpness in images based on the contrast in pixel neighborhoods. However, this requires the surface to be textured. In weakly textured parts, correctly detecting the sharpest image might fail. Furthermore, the amount of images to accurately capture a full 3D shape is immense. We also have practical reasons not to consider this technique for our application: Shape from defocus requires to reliably set the point of focus, either by changing the focal length or by moving the camera. Both is not automatically possible with our setups.

Methods based on integration have several problems as well. Small errors in the estimated normals accumulate during integration and lead to significant deviations from the correct overall surface shape. Furthermore, estimation of normals usually works on images from a single viewpoint. From a single viewpoint, however, the absolute depth of the reconstructed surface cannot be determined. So, while a full object could be reconstructed from multiple integrated geometries from points of views on different sides, this is not trivially possible. To overcome the first two drawbacks, several publications therefore investigate **multiview normal field integration**, an approach that combines normals from different views using a consistency measure. This is either achieved via volumetric methods or surface evolution. Still, the computation of the normals themselves is a challenging task on its own. Most methods are specific to certain types of materials: **Photometric stereo** or **shape from shading** only work for surfaces with Lambertian reflectance. **Shape from specularity** requires a highly specular and **shape from reflection** a perfectly specular material. **Helmholtz stereopsis** should work for all materials that obey reciprocity. While this property is fulfilled for all opaque materials, it might be violated in the case of subsurface scattering.

Still, we found Helmholtz stereopsis to be a promising technique for digitization of objects with more complex optical material properties and explored this topic together with Weinmann *et al.* in [WRO*12]. The results of our experiments suggest that Helmholtz stereopsis could be used together with **structured light consistency** to achieve accurate 3D reconstructions using the Dome 2 setup presented in Section 4.7. In a consecutive publication [WORK13], Weinmann *et al.* furthermore propose a multiview normal field integration approach, obtaining the normals either via photometric stereo or from reflections of coded displays. Neither the Dome 2 setup nor the proposed techniques were available at the time this thesis started. Therefore, we employ a different technique for the 3D reconstruction step in Chapter 5. Nonetheless, it could easily be substituted by one of the more recently published methods. This would be a very fruitful way of future work.

3.4.2.1 Triangulation Methods

The term triangulation has multiple meanings. In the context of computer graphics, it often refers to the division of a surface into triangles. Here, we instead refer to triangulation in the context of computer vision and surveying: determining the position of a point, given a known baseline by measuring angles from both endpoints. In computer vision, these measurements are derived from images. If the employed camera is calibrated (see Section 3.5.1.1), each point $\mathbf{p} \in \mathbb{R}^2$ in the image is exactly one such angular measurement. Assuming a pinhole camera model, a point $\mathbf{p}$ in the 2D image can be mapped to a direction $\mathbf{d}$ as

$$\mathbf{d} = \frac{(\mathbf{K}\,\mathbf{R})^{-1}\begin{pmatrix} p_1 \\ p_2 \\ 1 \end{pmatrix}}{\left\|(\mathbf{K}\,\mathbf{R})^{-1}\begin{pmatrix} p_1 \\ p_2 \\ 1 \end{pmatrix}\right\|}, \tag{3.14}$$

with $\mathbf{K}$ denoting the intrinsic calibration matrix and $\mathbf{R}$ the rotation matrix, i.e. the orientation of the camera. This direction $\mathbf{d}$ and the camera's origin $\mathbf{o}$ form the ray through the physical location of the point on the camera sensor or film and the camera's center of projection. The 3D position of the surface point depicted in $\mathbf{p}$ can be computed by the intersection of at least two rays. In our practical implementation, we extend the model of the pinhole camera by radial and tangential lens distortions, following the distortion model of `OpenCV` [Ope].

In passive methods, triangulation is performed by **(multiview) stereo** algorithms. The input are at least two images of an object taken from different points of view. First, correspondences between points in the images are established. Two points are corresponding, if they depict the same point on a 3D surface. Then, the 3D position is computed by intersecting the respective rays. Finding correspondences between images is the crux of this method. If the full calibration for all cameras is known, the search space in the images can be significantly reduced. However, finding sufficiently many matches for a dense 3D reconstruction is still a hard problem. Corresponding points are usually identified based on the assumption that the depiction of the same position on a 3D surface is identical in different images. The principle is often referred to as **photo consistency** and also finds application as a consistency measure for volumetric reconstruction methods. This poses two major problems: First, uniform regions with low texture variation lead to ambiguous correspondences. Further, the assumption that the surface will look the same from different points of view is only fulfilled in the case of Lambertian reflectance. Although the method can still be applied for a wider range of mostly diffuse materials, almost all digitized objects presented in Section 5.6 severely violate this assumption.

Active triangulation methods solve the correspondence problem by actively illuminating the scene with sequences of patterns, e.g. dots or stripes from a laser or fringe projections from an ordinary (digital) image projector. The first case is usually referred to as **laser scanning**, the second as **structured light**. The projected patterns on the object surface are captured by a camera and provide a correspondence between points in the camera image and the projection device. The projector is calibrated as well, allowing a triangulation of 3D positions from these correspondences. While laser scanners sequentially sweep the object surface with a single dot or stripe, structured light approaches utilize a complete 2D pattern. This significantly speeds up the acquisition, as a low number of patterns usually suffices to provide reasonable correspondences. More details on 3D scanning with structured light can be found in Section 5.4.1.

Active light triangulation methods are very robust and precise. They work on almost all opaque objects. Only bright interreflections, subsurface scattering and very specular or mirroring materials can produce false matches and hence disturb the 3D geometry. Using more than one camera helps to identify and eliminate most of these cases. Transparent or translucent objects, however, cannot be captured. Due to the lack of other practical alternatives, we still resort to structured light scanning as the most appropriate and general solution to obtain accurate macroscale geometry for a wide variety of objects.

3.5 Camera Model

In order to impute the geometry and reflectance of 3D objects from images, we need to refer to a mathematical model to describe the geometry of the image formation process. Analogously, a model for the virtual camera is required for rendering the captured objects in a virtual scene. We employ models derived from the principle of a pinhole camera for both cases. This is a very popular choice in both, computer graphics and computer vision. It allows to formulate the image formation as a simple matrix multiplication in homogeneous coordinates and describes the behavior of real rectilinear perspective cameras sufficiently well. Please refer to Section 3.1.1 for a short recap of homogeneous coordinates.

3.5.1 Real Camera

A very good treatment of camera models used in computer vision can be found in the textbook "Multiple View Geometry in Computer Vision" by Hartley and Zisserman [HZ04]. For our application, we consider the camera to behave like a *finite projective camera*. We will now briefly recite the most basic properties. Note that our notation partially differs from that of Hartley and Zisserman.

Consider a point $\mathbf{x} \in \mathbb{R}^3$ on a surface of an object. When a camera following this model takes a picture of the surface, it depicts $\mathbf{x}$ at the point $\mathbf{x}' \in \mathbb{R}^2$ in the image. The process is called *projection* and $\mathbf{x}'$ the projected point or projection of $\mathbf{x}$ in the camera image. Using homogeneous coordinates, the projection $\mathbf{x}'$ can be computed with the camera's *projection matrix* $\mathbf{P} \in \mathbb{R}^{3\times 4}$:

$$^{\mathrm{h}}\mathbf{x}' \quad = \quad \mathbf{P}\, {}^{\mathrm{h}}\mathbf{x} = \mathbf{K}\,[\mathbf{R}\,\mathbf{t}]\; {}^{\mathrm{h}}\mathbf{x}. \tag{3.15}$$

Here, $\mathbf{R} \in \mathbb{R}^{3\times 3}$ is a *rotation* matrix describing the orientation of the camera and $\mathbf{t} \in \mathbb{R}^3$ is the *translation* of the camera's center of projection. Together, rotation and translation are denoted as the six *extrinsic parameters* of the camera. Note that $\mathbf{t}$ does not describe the position of the camera's center of projection in world space. However, it can be easily expressed as $\mathbf{o} = -\mathbf{R}^T\mathbf{t}$.

Matrix $\mathbf{K} \in \mathbb{R}^{3\times 3}$ is the *calibration matrix* of the camera and holds the five *intrinsic parameters* c, a, s, h_1 and h_2:

$$\mathbf{K} = \begin{pmatrix} c & s & h_1 \\ 0 & a \cdot c & h_2 \\ 0 & 0 & 1 \end{pmatrix}. \tag{3.16}$$

The parameters a and s describe the *aspect ratio* and *skew* of the image coordinate system's axes. The parameter c denotes the *focal length* and $(h_1, h_2)^T$ gives the position of the *principal point*, i.e. the point where the optical axis intersects the image plane.

3.5.1.1 Camera Resectioning

The projection matrix $\mathbf{P}$ has a total of eleven degrees of freedom. If the 3D position of at least six points $\mathbf{x}_i \in \mathbb{R}^3$ (in general positions) and their projection into the camera image $\mathbf{x}'_i \in \mathbb{R}^2$ are known, they can be used to obtain the camera parameters [HZ04]. This process is called *camera resectioning* or camera calibration. We also refer to it as *geometric camera calibration*, to distinguish it from a radiometric calibration. We utilize this to calibrate the cameras of our two dome setups, as described in sections 4.6.2.1 and 4.7.2.1.

Of course, the finite projective camera model, which is derived from the functional principle of a pinhole camera, is only an approximation. Real cameras actually employ elaborate optics with several lenses to focus the light. These lenses can introduce nonlinear distortions, such that $\mathbf{x}$ is depicted at point $\mathbf{x}''$ instead of $\mathbf{x}'$. We therefore additionally use the `OpenCV` distortion model [Ope] with four parameters k_1, k_2, p_1 and p_2 to model the projection $\mathcal{P} : \mathbf{x} \mapsto \mathbf{x}''$:

$$\begin{aligned} {}^{\mathrm{h}}\mathbf{y}' &= \mathbf{R}\mathbf{x} + \mathbf{t} \\ \mathbf{y}'' &= (1 + k_1 r^2 + k_2 r^4)\mathbf{y}' + \begin{pmatrix} 2p_1 y'_1 y'_2 + p_2(r^2 + 2y'^2_1) \\ p_1(r^2 + 2y'^2_1) + 2p_2 y'_1 y'_2 \end{pmatrix} \\ {}^{\mathrm{h}}\mathbf{x}'' &= \mathbf{K}\,{}^{\mathrm{h}}\mathbf{y}'', \end{aligned} \tag{3.17}$$

with $r = \|\mathbf{y}'\|$. Given a sufficiently large number of observations of known 3D positions, these parameters can be estimated as well.

A precise knowledge of the 3D point positions and their depictions is necessary to obtain reasonably accurate camera parameters. If the 3D coordinates are not well determined or not known at all, as is the case for the Dome 1 setup described in Section 4.6, a calibration can be obtained by utilizing observations from multiple cameras. Depending on the field of research this is achieved by different but closely related techniques, such as *bundle adjustment* in photogrammetry, *structure from motion* (SfM) in computer vision and *simultaneous localization and mapping* (SLAM) in robotics. They all have in common that the 3D position of points and the camera parameters are jointly estimated from corresponding observations of the same points in multiple cameras (or multiple poses of the same camera).

In this thesis, we employ *sparse bundle adjustment* (SBA) [LA09]. The approach operates on a set of observations $\mathbf{x}''_{i,j}$ of unknown 3D points $\mathbf{x}_i$ by cameras with unknown projection matrices $\mathbf{P}_j$. In a nutshell, the method minimizes the sum of reprojection errors:

$$\sum_i \sum_j \left\| \mathbf{x}''_{i,j} - \mathcal{P}_j\left(\mathbf{x}_i\right) \right\|. \tag{3.18}$$

A (locally) minimal solution is found using a nonlinear Levenberg-Marquard optimization [Lev44].

3.5.2 Virtual Camera

Depending on the purpose and rendering technique, we employ slightly different virtual camera models. For high-quality offline light scattering simulations that include global illumination effects between object, we employ the physically-based path tracer `Mitsuba` [Jak10]. Mitsuba provides several camera models to choose from. We use the "perspective pinhole camera" model. The projection is the same as for the projective finite camera without any lens distortion, described in Equation 3.15. Note that in Mitsuba's "perspective pinhole camera" the camera parameters $s = 0$ and $a = 1$ are fixed and the principal point $(h_1, h_2)^T$ is always at the center of the rendered image. For evaluation purposes, we also would like to create synthetic images that can be compared with the taken measurement images. In these cases, we thus additionally employ a self-written camera plug-in for Mitsuba that accounts for all parameters available in our model of a real camera, described in Equation 3.17.

For the generation of synthetic images in real-time, we employ the popular *Open Graphics Library* (OpenGL). Here, we use the perspective camera model that is based on a view transformation, a frustum-based perspective projection and finally a viewport transformation, as described in the OpenGL "Redbook" [SSKLK13]. All operations are carried out as a series of matrix multiplications on homogeneous coordinates. The product of the three matrices corresponds to the projection matrix $\mathbf{P}$ in Equation 3.15 with fixed parameters $s = 0$, $a = 1$ and the principal point $(h_1, h_2)^T$ located at the center of the image. The only difference is an additional fourth row that enables the computation of a depth value.

3.6 High Dynamic Range Imaging

Capturing and conveying the dynamic range of natural scenes has been the subject of intense studies for almost two decades. In the following, we use the textbook "High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting" by Reinhard *et al.* [RWPD05] as a source of reference.

In general, the *dynamic range* of a given signal denotes the ratio between the lowest value l and highest value h: $\frac{h}{l}$. Common notations for dynamic range are directly giving the ratio, e.g. as 1000:1, or expressing it as a single value on a logarithmic scale. In the context of images, two such scales are often employed: *stops* or *f-stops*, which express the value with respect to the second logarithm $\log_2\left(\frac{h}{l}\right)$ and *decibels* (dB), which uses the tenth logarithm times ten: $10\log_{10}\left(\frac{h}{l}\right)$. F-stops have an intuitive relation to photography, as opening the aperture by one additional "stop" doubles the amount of light that will reach the sensor. Exposure values of a camera are for instance counted in stops. Nonetheless, in this thesis we will use the more widespread decibel scale to express dynamic ranges.

Images can simply be considered 2D signals. This is true regardless whether the image is formed by light falling onto the retina of an eye (or the sensor of a camera) or is the result of a light scattering simulation in a computer. The strength of the amplitude corresponds to the radiance of the light rays.

Natural scenes, e.g. an indoor scene with a window showing a sunlit exterior, reach dynamic ranges in the order of 30 dB. The human visual system evolved to cope with this situation. Within a single scene, it can simultaneously grasp visual information with a dynamic range of 50 dB. Unfortunately, the films or sensors of photographic cameras usually exhibit a much lower sensitivity of about 20 dB to 25 dB (see for example Table 4.5). If radiance values exceed this range, they will be clipped: The pixels become oversaturated or underexposed and cannot any more be distinguished from other over or underexposed pixel, which in reality might have been exposed to a different radiance. This results in a loss of information. Here, techniques to increase the available dynamic range need to be employed to recover the full radiance information.

This is especially important for our application of digitizing reflectance. In controlled capture conditions, the ratio between the brightest part of a highlight and the darkest parts in shaded areas are rather large. For example, some specular materials in the MERL database of measured BRDFs[1] exhibit a dynamic range of more than 75 dB.

Furthermore, this dynamic range must also be reproduced by the light transport simulation to allow the accurate display of the digitized reflectance. We take this into account and utilize the full available dynamic range during rendering. However, an ordinary computer screen and the paper this thesis is printed upon only provide low dynamic ranges of about 20 dB. Therefore, the dynamic range of the rendered images (and our depicted reference photographs with increased dynamic range) needs to be reduced to convey most information without clipping.

In literature, the (kind of fuzzy) distinction between *low dynamic range* (LDR) images and *high dynamic range* (HDR) images is made. Usually, the term LDR is used to refer to images which do not exceed the dynamic range of ordinary cameras and output media. Images that portray the full dynamic range of a scene in sufficient detail are instead referred to as HDR or *radiance maps*.

3.6.1 HDR Combination

HDR images can be obtained with ordinary LDR cameras by employing a technique called *exposure bracketing* [MP95, DM97]: A scene is captured multiple times with different *exposure values* (EV). Consider 0 EV to correspond to the ideal exposure of the scene, i.e. it leads to the LDR image that covers most of the

[1] `http://merl.com/brdf/`

radiance values correctly. A value of -1 EV will capture only half the radiance at every point in the image. While most of the image is now underexposed, parts that were previously overexposed now become distinguishable. Similarly, a value of +1 EV will overexpose most of the image but recover information in previously underexposed regions, such as shadows. The final HDR image is obtained via a combination of correctly exposed parts from all available LDR images. Most often, the exposure values are modified by varying the time of exposure, e.g. +1 EV is achieved by doubling the exposure time. Nonetheless, in Section 4.6 we also present a way to create HDR images by varying the exposure values via flash light intensity and ISO speed of the camera.

3.6.2 Tone Mapping

Vice versa, LDR information necessary for output on a screen or print can be computed from HDR images. This general process is called *tone mapping* and several different approaches exist. It is for instance desirable to take the reaction of the human visual system into account to create LDR images that approximately convey the impression of directly observing the radiance values captured in the HDR image. This is usually achieved by adjusting the contrast locally, simulating the local adaption of rods and cones in the eye. However, due to our familiarity to LDR images, e.g. printed photographic pictures or movies and TV, images tone-mapped in such a way might sometimes look artificial to us. In this thesis, we therefore utilize global tone mapping operators $t : L \to g$, mapping radiance L to LDR grayscale values g. For most HDR images we imitate the tone reproduction of the Canon PowerShot A75 camera to create a "photorealistic" impression. From radiometric calibration (see Section 4.6.2.2), we obtain the response function of the camera, mapping recorded radiance to LDR values. We simply apply this function to the values in the rendered HDR radiance maps. In some occasions, we instead employ gamma correction $g = \sqrt[\gamma]{e \cdot x}$, with e being an adjustable exposure compensation, as it better reproduced the visual features that we considered important for the figure. The exact tone mapping technique is not that important for most images in this thesis – except of course for aesthetic purposes. We therefore refrain from mentioning the tone mapping in every single figure caption. It is just important to keep in mind that all shown renderings are calculated in HDR and merely tone-mapped for their final output. Similarly, all captured data is acquired as HDR data.

3.7 Binary Data Formats

Throughout this thesis we will present file sizes, compression ratios, transmission speeds and memory footprints. We will thus briefly touch the topic of the actual binary representation of the digitized appearance data, so all these values can be put into the right context. The formats are discussed in the order of their occurrence in the digitization and presentation procedures.

We mostly work on raw measurement data that was captured by two devices: the Dome 1 and the Dome 2, presented in sections 4.6 and 4.7, respectively. In both cases the raw measurement data consists of a set of separate image files. The Dome 1 stores standard `JPEG` images [ITU94] with 8 bit per color channel. For the data of the Dome 2 we employ a proprietary image format, developed by Martin Rump, we call "Bitpacked Raw". The format stores the raw 12 bit values of the cameras tightly packed, requiring 1.5 bytes per pixel. It additionally employs a very simple and lightweight lossless compression. Rows of the image with no pixel exceeding a value of 1023, i.e. 10 bit, are stored with 1.25 byte per pixel. Similarly, rows with at most 8 bit values are stored with one byte per pixel.

The digitized macroscale 3D geometries are first stored as triangle meshes in `Wavefront OBJ` file format [Red]. The OBJ files contain 3D position coordinates, 2D texture coordinates and normals per vertex. They further contain a list of faces defined their vertex indices. In chapter 6, the geometry is instead stored in *JavaScript object notation* (JSON) [Bra14] with the same list of attributes and an additional tangent vector per vertex. In both cases, all values are stored in plain text instead of a binary representation. The geometry could certainly be stored more efficiently. However, as this is not the scope of this thesis, we simply report all values for the two ASCII formats.

All processed reflectance values used in this thesis are represented in floating-point format according to the IEEE 754-2008 standard [IEE08]. The 2008 standard knows several formats with different levels of precision. For the sake of readability, we employ the old naming convention from IEEE 754-1985 [IEE85] and `OpenEXR`[2], using the terms *half*, *single* and *double* precision for the respective 16, 32 and 64 bit binary floating-point formats (called *binary16*, *binary32* and *binary64* by the 2008 standard).

We use single- and half-precision values for input and output data on disk or in RAM. For numerical operations, we temporarily convert to higher double precision. Especially the uncompressed BTF and the factorized representations are given in half precision to save disk space and (GPU) memory. Unless explicitly stated otherwise, all file sizes in this thesis are thus given for 16 bit per stored value.

[2] `http://www.openexr.com/`

Since all objects presented in this thesis employ a texture map containing unoccupied space, the uncompressed BTF files are not stored with their full spatial texture resolution. Instead, a bitmap indicating the occupancy is given in the header and then only the occupied texels are stored.

In contrast, the compressed representation on disk, in RAM and on the GPU, preserves the original full texture dimensions and contains values for unoccupied texels as well. The rationale is to avoid costly indirections in real-time applications and facilitate bilinear interpolation on the GPU. Two more efficient formats to store the compressed BTF data for the purpose of fast transmission and lower GPU memory utilization are discussed in detail in chapters 6 and 7, respectively.

3.8 Error Metrics

In order to evaluate the faithfulness of the appearance reproduction from the digitized data, we need to first define suitable error measures. This is also necessary to formalize optimization problems that aim to find a solution under given constraints with the goal to uphold the maximal possible visual quality, e.g. the factorization-based compression in Section 5.5.4, the transmission order for streaming in Section 6.4 or the priority of texture tiles in Section 7.5.2.

Several possible error measures could be employed. Among the most common are metrics based on the L^2 norm. For instance the *sum of squared errors* (SSE), also referred to as *sum of squared distances* (SSD), or the *root mean squared error* (RMSE). Given two data matrices of the same size $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{M \times N}$, e.g. an uncompressed and a compressed BTF or a photographic and a synthesized image, the SSE is expressed by $\sum_{i=1}^{M} \sum_{j=1}^{N} (a_{i,j} - b_{i,j})^2$. This corresponds to the square of the Frobenius norm $\|\mathbf{A} - \mathbf{B}\|_F^2$. The RMSE is a very similar measure:

$$\sqrt{\frac{1}{M\,N} \sum_{i=1}^{M} \sum_{j=1}^{N} (a_{i,j} - b_{i,j})^2} = \sqrt{\frac{1}{M\,N}} \, \|A - B\|_F \,.$$

Because of their favorable mathematical properties and fast evaluation, L^2 metrics are popular in minimization problem formulation. We therefore employ this quality measure as well for our optimization problems.

However, Wang and Bovik [WB09] very convincingly illustrate that the *mean squared error* (MSE) (and thus also the RMSE or SSE) is not a good indicator for the perceived error when it comes to images. The human visual system is more sensitive to certain distortions in the original signal than others. As a consequence, the authors propose to use metrics that are motivated by the principles of human visual perception, e.g. the *Structural SIMilarity* (SSIM) index [WBS*04].

The SSIM index between two images expresses their similarity by values in $[-1, 1]$, with 1 denoting perfect identity. For this, small patches in the images are compared to each other with respect to luminance, contrast and structural similarity. The latter is evaluated via cross correlation of the patches. The final score is computed as the average over the single patches. Wang *et al.* also proposed a multiscale version denoted *MS-SSIM* [WSB03]. Sheikh *et al.* performed an extensive user study and found that MS-SSIM was among the best performing algorithms in predicting the similarity of images for a human observer [SSB06].

Alternatively, the *HDR Visual Difference Predictor 2* (HDR-VDP-2) [MKRH11], which is especially tailored to work on HDR images, would lend itself in our scenario. HDR-VDP-2 simulates various physiological and psychological responses of the human visual system to incident radiance to provide a good approximation of the impression on an actual human observer. However, the authors state that HDR-VDP-2 is designed to predict the visibility of (rendering) artifacts rather than judging the overall image quality. In a comparison with MS-SSIM, both methods achieve a similar prediction rate for responses of human participants.

However, in [GMSK09], Guthe *et al.* demonstrate that image-based metrics cannot directly applied for BTF data. They therefore develop an extension to the S-CIELab measure, called *BTF-CIELab*, taking into account angular consistency, and employ it for perception driven compression of BTFs. Although the presented result images show a (subjectively) improved image quality, we will not employ the compression approach in this thesis for two reasons. First, it introduces a computational overhead in comparison to the proposed factorization techniques. Second, in Part III of this thesis, we will exploit several properties of the L^2-norm-based SVD for our purposes, which would not directly be transferable to BTF-CIELab. Still, integrating a better error-approximation in the presented methods would be an interesting avenue of future research.

We would still like to take human perception into account for judging the quality of the achieved results. We can do this by using image-based measures on renderings under different conditions. The best approach would be to perform an extensive user study, letting participants judge the quality of appearance reproduction (ideally in explorative interaction with the virtual objects). However, this is far outside the scope of this thesis. We therefore use the MS-SSIM index to assess the renderings created with resampled and compressed BTF data. We use the MATLAB implementation of the original authors, which is freely available online [3]. Because the algorithm only operates on monochrome images, we separately apply it to each of the RGB color channels and report the average result.

[3] `http://www.cns.nyu.edu/~lcv/ssim/`

Part II

Integrated Acquisition

CHAPTER 4

CAMERA ARRAY SETUPS FOR THE RAPID ACQUISITION OF APPEARANCE

In this chapter, we provide an in-depth discussion of the attributes that are required in a BTF measurement setup. We survey the literature and compare different general approaches with respect to these properties. Because many of the insights in this thesis also apply for BTFs of flat material samples, we do not limit our investigation to setups for object digitization.

We further present three setups that have been developed at the Institute of Computer Science II of the University of Bonn in detail, juxtaposing the strengths and weaknesses of different design decisions. Here, we find convincing arguments that especially camera array setups promise a very high degree of efficiency and lend themselves for the digitization of complete 3D objects. Consequently, we employ two camera array setups to capture the digitized objects presented in this thesis. The description of their exact hardware design as well as their calibration and measurement procedure can be found in this chapter in sections 4.6 and 4.7.

In our comparison, we consider the acquisition pipeline up to the point that a full BTF tensor $\mathcal{B} \in \mathbb{R}^{|\mathfrak{L}|\times|\mathfrak{V}|\times|\Lambda|\times|\mathfrak{X}|}$ is recorded. Here, $\mathfrak{L}$, $\mathfrak{V}$ and Λ denote the sets of basis illuminations, view directions and wavelength bands. $\mathfrak{X}$ is the set of sampled spatial positions on the bounding surface ∂V. We measure the spatial resolution of the surface details that can be achieved with the given $\mathfrak{X}$ in *dots per inch* (DPI).

Note that in both cases $\mathfrak{L}$ and $\mathfrak{V}$ refer to the direction sampling of the measurement device. In this chapter, we will not tackle the problem of resampling the captured data onto the surface ∂V. A technique to achieve this will be the contribution in Chapter 5.

4.1 Introduction

The key to the realistic impression of the digital replicas is the data-driven BTF representation of the surface materials. BTFs can be obtained by systematically tabulating the reflectance of real-world samples. For being truly general, brute force approaches, densely sampling all dimensions of the parameter domain, are necessary. This way, the captured data can directly be used in light simulations to enable the faithful reproduction of the material appearance. The accurate capture of BTFs requires the thorough acquisition of billions of data points. Since the introduction of BTFs in [DVGNK97], several acquisition setups have been proposed to capture BTF data. In almost all devices, the principle of acquisition boils down to taking a large number of pictures of a surface under different illuminations and from different points of view. To achieve a high visual quality during rendering, this requires several tens of thousands of pictures.

The use of the captured data is not limited to the mere reproduction of appearance. Densely sampled reflectance data can serve as a basis to gain deeper knowledge about the optical effects found in certain materials. It may aid the development and evaluation of specialized lightweight acquisition devices and elaborate mathematical material descriptions. Therefore, the insights presented in this chapter are also of great use for future research outside our scope of image-based rendering.

Yet, in this thesis we focus on the brute force acquisition of reflectance data for the purpose of rendering. One of the major goals of the proposed setups is the acquisition of the appearance of objects on all three scales. As argued in Section 3.3, this also includes a 3D geometry to represent the macroscopic scale. Passive 3D reconstruction methods, such as multiview stereo, shape from silhouette or integration-based approaches could be used directly on the images that are taken when capturing the BTF. Yet, the discussion in Section 3.4 revealed that these methods are not well suited for the envisioned application scenarios. Several of the solutions proposed in the literature therefore capture the geometry and reflectance in two separate steps, using off-the-shelf 3D scanners to capture the macroscale geometry with higher quality. However, to form a single digital replica, the two captured data sets need to be registered. Even for an expert user, aligning pictures with a geometry is a non-trivial and tedious task. Taking into account the tens of thousands of pictures that are necessary for the reconstruction of a BTF, this cannot be performed manually. Instead, potentially error prone automatic registration algorithms need to be employed. To make things worse, even slight misalignment reduces the final visual quality. Therefore, an integrated setup that is capable of capturing high-quality geometry as well as view- and light-dependent reflectance properties of arbitrary 3D objects is desirable. We acknowledge this insight by explicitly reporting the 3D reconstruction possibilities of the discussed setups.

Further, two of our three setups implement such an integrated acquisition approach, based on structured light.

In summary, our main contributions are

- a survey and comparison of existing BTF measurement setups,
- the introduction of the "Dome 2", a new practical BTF measurement setup with integrated geometry acquisition,
- an in-depth discussion of design, implementation, calibration and measurement for three different BTF measurement setups,
- a series of experiments to quantitatively compare important attributes of the measurement devices.

In order to illustrate the possible design choices that can be followed for the acquisition of BTFs, we will first establish design requirements for the measurement of reflectance in Section 4.2. We will then skim through the literature in Section 4.3 and sort existing devices into three distinct design categories. In sections 4.4 to 4.7 we will provide an in-depth description of three reflectance measurement devices. The detailed discussion is intended to aid the interested reader and explain the reasoning behind the respective design choices. While the description of the employed hardware is rather particular, other information, e.g. the abstract design or employed calibration methods, will also provide valuable insight beyond the individual setups. Eventually, we compare our setups and setups found in literature in Section 4.8 and draw first conclusions in Section 4.9.

4.2 Design Requirements of a BTF Measurement Apparatus

To enable the acquisition of BTFs, several basic attributes and abilities should be considered by the design of a measurement setup: light field capture, controlled illumination, high dynamic range imaging, radiometric calibration and spectral sampling. To digitize complete objects, the setups should also provide integrated 3D scanning. Last but not least, although not strictly necessary from a theoretical standpoint, practical requirements should of course be considered as well. In the following, we will give a detailed explanation of each one.

Light Field Capture: Any setup that measures BTFs has to capture outgoing light fields $L_{o,V}$ from real-world material exemplars. As argued in [LH96], outgoing light fields are best sampled taking a set of photographic images of the exemplar from different camera positions on a sphere around the bounding volume, always facing the sample. For a planar bounding surface, a hemisphere of positions is sufficient. Positions below the plane of reference would not provide observations from outside the volume any more and hence invalid samples.

Controlled Illumination: To account for varying illumination, it is necessary to consider arbitrary far field incident light fields $L^d_{i,V}$. As in [DHT*00], the principle of superposition can be exploited. The setup has to be capable of controlling the lighting and alternate through a set of basis illuminations $\{L_j\}_j$, from which any far field incident light field can be reconstructed as a linear combination $L^d_{i,V} \approx \sum_j l_j L_j$ with corresponding weights $l_j \in \mathbb{R}$. It is furthermore important that the basis illumination is homogeneous over the complete sample surface in order to fulfill the far field assumption of the BTF. In practice, most setups choose a set of approximately directional light sources, such that each basis illumination sheds light from a single direction ω_j.

High Dynamic Range Imaging: Material reflectance usually exhibits rather high dynamic ranges. On the one hand, high radiance values are observed when the light that is reflected comes from the perfect mirroring direction. On the other hand, light from grazing angles in combination with a view direction outside the specular lobe of the material leads to very low radiance values. In spatially varying materials, there can also be a considerable difference in albedo. Rough materials also introduce self-shadowing. This further increases the ratio between largest and lowest observable values. However, the dynamic range that can be captured by CMOS or CCD sensors of digital cameras is limited and easily exceeded by the reflected radiance (see Section 3.6). If used directly, this either results in faulty measurements due to oversaturation of sensor pixels or – if exposure time is minimized to compensate for this effect – in extremely high noise levels in all other pixels. Thus, it is good practice to employ exposure bracketing to generate an HDR image from multiple differently exposed LDR images to be capable of capturing the full range of reflectance values.

Radiometric Calibration: For the BTF to be applicable in predictive light transport simulations, any measurement setup should be carefully radiometrically calibrated. The sampled entries then give the ratio of reflected radiance to incoming irradiance in sr^{-1} for the given combination of directions ω_i and ω_o, wavelength λ and surface position $\mathbf{x}$.

Spectral Sampling: Surface appearance is dependent on the spectrum of the light. A BTF measurement setup should at least be able to capture tristimulus images and provide for a basis illumination such that the perception of the material for a human observer under natural illumination (e.g. daylight) is captured. However, to facilitate the predictive simulation of different light sources, a dense spectral imaging of the material reflectance would be preferable. Often different terms are used, depending on the number of considered spectral bands. However, the distinction is rather fuzzy. The term *multispectral* is used for "a few" spectral bands. *Hyperspectral* refers to denser configurations. A very dense sampling, resulting in a seemingly continuous spectral curve, is sometimes called *ultraspectral* or full spectral imaging. In this thesis, we just use the term multispectral and specify the exact number of bands.

3D Scanning: In the case that flat real-world material samples cannot be employed for acquisition, e.g. for stiff naturally curved materials such as egg-shell, or if the reflectance behavior of objects should be digitized, it is advantageous to additionally capture the 3D shape of the sample. In principle, reflectance fields could also suffice with a coarse proxy geometry, e.g. a bounding sphere. Yet, having a more precise geometric shape model of the surface is advantageous for compression as well as rendering (see discussion in Section 3.3.2). Furthermore, light simulation based on reflectance fields is only correct if the proxy geometries do not intersect each other. Too expansive bounding surfaces therefore unnecessarily limit the possible arrangements of digitized objects. BTFs introduce the additional issue of disocclusion by the proxy geometry, as light transport through transparent parts of the proxy geometry is modeled insufficiently.

There exists a whole bunch of different approaches for the acquisition of 3D geometry(see Section 3.4 for a brief overview). Many off-the-shelf solutions are available, covering the full range of options in terms of accuracy as well as price. However, including a 3D scanning solution into the process of reflectance capture is not a trivial task. For an automated acquisition, the scanner should better be integrated into the reflectance measurement device, which restricts the possible 3D acquisition approaches. If an external device is employed, the issue of registration of the 3D measurement with the reflectance samples has to be tackled.

Practical Requirements: It is in general not sufficient to capture only a few radiance values. For a faithful reconstruction, the sampling rate in all six dimensions of the parameter space should be adequately high (consider the Nyquist-Shannon sampling theorem [Sha49]). For this, millions and billions of data points have to be recorded. That makes a computer-controlled setup mandatory, as manual sampling or extensive user-interaction would make the process completely infeasible. Here, the measurement time should be as short as possible and the sampling as dense as necessary. The reflectance samples should be of high quality: All spatially varying effects that are not part of the reflectance (e.g. sensor noise, inhomogeneity of illumination, etc.) should be eliminated. The actual sampling directions should show as little variation from the ideal directions as possible. For industrial application, the setup also has to function reliably without supervision and show a high durability as well as the capability to measure in rapid succession. Finally, the measurement volume should be large enough for the application, i.e. capturing all the spatial variations in material samples or even complete objects.

4.3 Classification of Device Designs

By far not all reflectance acquisition setups found in literature aim to fulfil all of the above design requirements. The measurement, representation and reproduction of optical phenomena is an interdisciplinary and very active field of scientific research with a lot of specialized solutions. Excellent surveys on techniques for surface reflectance acquisition and representation are given in [MMS*04, WLL*09, FH09, HF11, HF13]. In the following, we will consider only those publications on setup designs that are the most relevant to our application, i.e. that are in principle capable to meet the requirements established in Section 4.2.

Among those, we have identified three general categories of BTF measurement devices: gonioreflectometers, mirror-based setups and camera array setups. Still, the individual designs often follow additional application specific approaches and differ with respect to speed, flexibility, resolution or complexity. In the next paragraphs, we provide a brief summary of the categories and the covered publications. A more detailed discussion and comparisons of the setups can be found later in this chapter in Section 4.8.

4.3.1 Gonioreflectometer Setups

Classically, a gonioreflectometer is a device consisting of a single light source and a photodetector. A bidirectional reflectance measurement is performed by moving the employed light source and the detector to several different locations around the sample. Gonioreflectometers have been employed for the measurement of BRDFs for a long time [NRH*77, PB96, Foo97, WSB*98, SI99, LFTW06, LFD*08]. In [WSB*98], for instance, a fully automated BRDF acquisition setup is presented that can take up all angular configurations on the hemisphere above the exemplar. The light source and the detector are mounted on movable mechanical arms and the material sample and the light source arm are additionally mounted on a turntable and a ring bearing respectively. Recent publications additionally focus on multispectral BRDF acquisition [PB96, SI99, LFTW06, LFD*08].

Gonioreflectometers can be used to acquire spatially varying reflectance by employing a spatial camera-sensor (CMOS or CCD) instead of a single photoresistor. Different ways to achieve the bidirectional measurement have been explored. Several setups propose to place the light source or the detector at a fixed position and achieve the necessary angular configurations by changing the orientation of the material sample [DVGNK97, SSK03, McA02, KMBK03, TAN*05, TSA*05, KTT06]. The setups proposed in [HLZ10, FVH*13] instead move both, sensor and light source, around the sample.

We exemplary present our own gonioreflectometer setup [HEE*02, SSK03, MMS*04, RSK10] in detail in Section 4.5.

4.3.2 Mirror and Kaleidoscope Setups

For taking multiple BRDF measurements of the same material sample in parallel, Ward *et al.* [War92] proposed a setup with a curved mirror in combination with a CCD fish-eye camera. Using the mirror they were capable of capturing the full hemisphere of view directions ω_o simultaneously. This idea was followed in several subsequent publications, such as [GAHO07] or [MSY07]. A projector is used to illuminate a specific point on the mirror. The ray is reflected and illuminates the sample surface from a direction ω_i. The scattering of the incident light by the material sample is observed through the same mirror by a camera that has an identical optical axis as the projector by using a beam splitter.

The same principle can be applied for measuring spatially varying reflectance by moving the material sample on a translation stage to capture reflectance at different points on the surface [Dan01, DW04, WD06]. Alternatively, a piecewise planar mirror geometry can be employed in order to allow a spatially extended illumination and observation of the sample under constant directions. This can either be a few mirrors arranged as a kaleidoscope [HP03, IRM*12], utilizing interreflections to form more directions, or an elliptical arrangement of severalpiecewise planar mirrors [LCV*04, GTLL06, MTK*10], showing only the direct reflection.

We do not provide an exemplary implementation for a mirror-based setup. As we will argue in more detail in Section 4.8.2.2, this class of devices can have some considerable drawbacks with respect to accuracy, possible sample size and resolution. We therefore direct our focus on camera array setups as a more practical alternative with similar advantages.

4.3.3 Camera and Light Array Setups

Similar to kaleidoscope setups, camera arrays feature a parallel acquisition of the spatial dimensions $\mathbf{x}$ and (parts of) the outgoing directions ω_o. Yet, in contrast to the mirror-based setups, multiple cameras are employed for the simultaneous direction acquisition, so that the full sensor resolution can be utilized for the spatial domain. Often, camera arrays are combined with light arrays, avoiding time-consuming mechanical repositioning steps of a light source.

Existing camera array setups either consist of a few fixed cameras [DHT*00, HCD01, DWT*02, WGT*05, WMP*05, WMP*06, WLDW11, HQS10, HWQ13] that sample only a slice or sparse set of the possible view directions – sometimes complemented with a turntable [FKIS02, MPZ*02, MPN*02, NZG05, TWL*05, NZG06, KNRS13, NJRS13, SSWK13] to cover a larger set of directions – or employ a dense hemispherical camera arrangement [MMS*04, SWRK11].

We will present two camera array setups that we implemented ourselves in Sections 4.6 and 4.7. We denote the setups as *Dome 1* and *Dome 2*. The first one [MMS*04, MBK05, RMS*08, SWRK11] consequently follows the approach of simultaneous view direction acquisition, capturing the full outgoing light field at once with a large number of cameras. The second one [WRO*12, SK12, SSWK13] implements a semi-parallel acquisition, using fewer cameras in combination with a turntable.

4.3.4 Integrated 3D Shape Acquisition

Examples of setups that perform an integrated geometry acquisition to facilitate reflectance capture on objects can be found in all three categories.

In [FKIS02, MPZ*02, MPN*02, MBK05], a coarse shape is reconstructed from object silhouettes. [FKIS02, WMP*05, WMP*06] employ additional auxiliary 3D scanners and register the geometry to the reflectance measurement. Several other devices [HLZ10, SWRK11, SK12, IRM*12, SSWK13, KNRS13, NJRS13] instead rely on an integrated structured light approach. This holds the advantage that the geometry is already registered with the reflectance measurements.

4.4 Common Provisions of Our Setups

Before going into the specific details of our BTF measurement setups, we want to list some basic provisions and procedures that are common to all three of them. These provisions apply in similar form to most other BTF setups as well.

As the acquisition of BTFs requires full control over the illumination conditions, all of the discussed setups operate in a controlled lab environment. In our case, the provisions resemble those of a photographic studio, similar to the measurement laboratory reported by Goesele *et al.* [GHLS00]. We sealed all windows with opaque black foil to avoid any outside illumination. We further blackened the ceiling, and laid a dark carpet and black curtains to minimize the effect of stray light. Finally, all parts of the employed equipment that potentially face a camera or the sample have been painted with a diffuse black coating. Status LEDs of the nearby control computers have been disconnected or blinded with black tape.

Obtaining high-quality reconstructions of surface reflectance behavior imposes the precise geometric and radiometric calibration of the involved components. The geometric calibration consists of the intrinsic and extrinsic parameters of cameras, projectors and light sources with respect to the sample. The radiometric calibration of all components establishes the radiometry of light source, sensor and lens system. This includes spatially varying effects of the optics, e.g. vignetting and light falloff, as well as colorimetry of the sensors, i.e. color profile and white balance.

We illustrate that all three devices show fundamentally different requirements and approaches to achieve an accurate calibration, which we describe in sections 4.5.2, 4.6.2 and 4.7.2.

For flat material samples, all three setups have in common that they use additional black-and-white border markers to further improve the spatial registration of the measured data. Similar additional markers can be found in almost all other BTF measurement setups for that purpose as well.

In our case, the borders are automatically detected in the captured images using contour finding and line fitting. Then we determine the corners of the corresponding quadrilateral with subpixel precision using the active contour model proposed by Chan and Vese [CV01]. The rectangular material sample, $\mathfrak{X} = \{0, 1, \ldots, W-1\} \times \{0, 1, \ldots, H-1\}$ is simply a full lattice with spatial resolution W times H. The reflectance samples are transformed to the respective rectified $W \times H$ pixel image by computing the homography to the common planar proxy. An example is shown in Figure 4.16. This technique allows a precise spatial alignment of the captured samples, even if the underlying calibration of the setup is less accurate.

In the case of 3D objects, however, this simple strategy is not sufficient. It is necessary to describe the mapping from the surface manifold to the spatial dimensions of the BTF tensor. We assume a planar embedding of the bounding surface into a finite rectangle exists. The spatial positions $\mathfrak{X} \subset \{0, 1, \ldots, W-1\} \times \{0, 1, \ldots, H-1\}$ are given by a bijection $\mathcal{P} \cdot \Pi$ onto that rectangle, i.e. the projection into a texture map of resolution $W \times H$ (see Section 5.5.2).

For general shapes, this mapping cannot be deduced from auxiliary markers. Instead, the calibration procedure has to provide an ample precision. Therefore, the achievable accuracy of the devices' calibrations and the registration of 3D geometry to camera images is given special attention in this thesis.

4.5 Gonioreflectometer

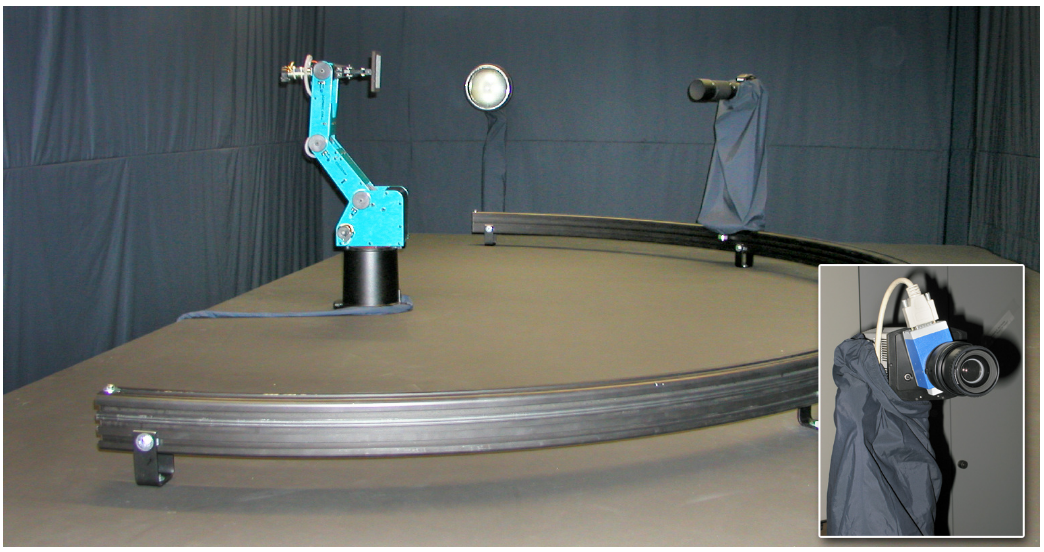


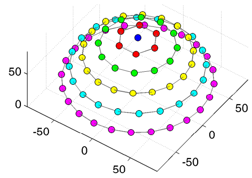
Figure 4.1: *Our gonioreflectometer setup with the original equipment from 2002. Image taken from [HEE*02]. The inset on the lower right side shows the camera that is employed for multispectral measurements since 2009. Note the tunable spectral filter behind the lens.*

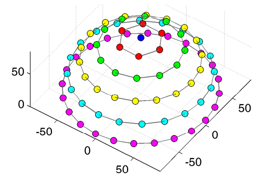
Our gonioreflectometer setup (see Figure 4.1), published in [HEE*02, SSK03, MMS*04, RSK10], was constructed between 2001 and 2002 to allow spatially varying and bidirectional measurement of material appearance from flat samples. In 2009 it was extended to perform multispectral measurements.

4.5.1 Hardware

The design of the device was intended to follow and improve upon the original BTF measurement approach proposed by Dana *et al.* [DVGNK97]. In contrast to [DVGNK97], the camera changes the position automatically in our setup, avoiding cumbersome manual placement and orientation. This is achieved via a computer-controlled rail system. The rail is bent such that the orientation of the camera

sampling 1 θ	sampling 1 $\Delta\phi$	sampling 2 θ	sampling 2 $\Delta\phi$	# dir
$0°$	$-^{\star}$	$0°$	$-^{\star}$	1
$15°$	$60°$	$17°$	$60°$	6
$30°$	$30°$	$34°$	$30°$	12
$45°$	$20°$	$51°$	$20°$	18
$60°$	$18°$	$68°$	$18°$	20
$75°$	$15°$	$85°$	$15°$	24





*Only one direction at $\phi = 0°$.

Table 4.1: *The sampling of the hemisphere used during a measurement with our gonioreflectometer setup. View and illumination hemispheres are sampled identically, but two different sets of directions have been employed, depending on the material.*

towards the sample is maintained on every position. Furthermore, the robot employed by Dana *et al.* could not rotate the sample around its normal direction, i.e. sample the angle ϕ_o. For measuring anisotropic materials, they proposed to manually change the orientation by moving the sample and performed a second measurement. This procedure poses considerable effort but still yields only a very coarse sampling of ϕ_o with two directions. In contrast, our setup employs a robot that is capable of assuming all necessary poses for an automated and dense sampling of the full angular domain. In their setup, Dana *et al.* employed a professional 3 CCD video camera with analog output together with a VGA-resolution frame-grabber (640×480 pixels). This particular combination showed a lot of color noise and only captured a single, fixed exposure in LDR with 8 *bits per pixel* (BPP). We instead utilize high-resolution digital still cameras with favorable noise characteristics and higher bit depth of 12 BPP, yielding a higher dynamic range.

4.5.1.1 Robot & Rail

As in [DVGNK97], the employed light source is placed at a fixed position. The camera, however, can be moved into different azimuthal angles ϕ_o via a custom built semicircle rail system. An Intellitek SCORBOT-ER 4u robot arm is placed in the center of the semicircle. It is used to present the mounted material sample to the camera in such a way that, in combination with the rail system, every angular configuration $(\theta_i, \phi_i, \theta_o, \phi_o)$ on the view and illumination hemispheres above the sample can be reached. Table 4.1 shows the measurement directions on the hemisphere above the material sample that are used. For this, the robot arm tilts and turns the sample – even in headlong positions. Unfortunately, the necessity to move the sample into slant positions makes the acquisition of 3D objects or delicate and granular materials infeasible. Rail, lamp and robot are affixed on a solid laboratory bench. The distance of the camera to the material sample is 170 cm, the distance of the light source is 240 cm.

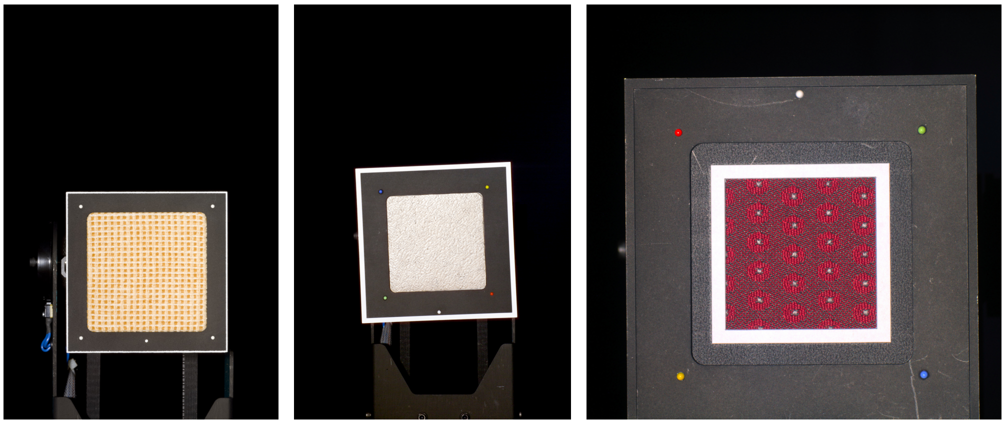
Due to constraints in the working envelope of the robot, not all azimuthal configurations for $\theta_o > 80°$ can be reached reliably. This has to be considered when measuring with sampling set 2. To still capture direction samples for views above 80° inclination, the measurement is paused in-between and the lamp is manually repositioned at the opposite side of the rail, avoiding borderline robot poses.

4.5.1.2 Camera

In its original configuration, reported in [HEE*02, SSK03], a Kodak DCS760 *digital single-lens reflex* (DSLR) camera with a six megapixel CCD was employed. The camera captures raw images at 12 BPP, yielding a dynamic range of 35 dB, with a Bayer-patterned *color filter array* (CFA) to measure RGB color. The camera was replaced in 2004 [MMS*04] by a Kodak DCS Pro 14n with a 14 megapixel full-frame CMOS sensor to achieve higher spatial resolutions. The DCS Pro 14n also captures Bayer-patterned raw images with 12 BPP, but has a lower dynamic range of 31 dB. The choice of camera was also influenced by the fact that Kodak provided a *software development kit* (SDK) that supported changing the camera settings as well as directly transmitting raw images to a PC.

For performing multispectral measurements [RSK10], the setup is now equipped with a four megapixel Photometric CoolSNAP K4 camera. The camera has a Peltier cooled monochrome CCD chip with 12 BPP, which is sensitive to electromagnetic radiation from 350 nm to 1000 nm. As the sensor is operated at approximately -25° C, it exhibits a very low noise level despite the prolonged exposure times necessary to capture the low amount of radiance passing the narrow spectral bandfilters. Thus, the cameras achieves approximately 32 dB dynamic range in a single shot. 32 different wavelength bands between 410 nm and 720 nm are sampled with a bandwidth of ten nanometer via a CRi VariSpec multispectral tunable liquid crystal filter (see inset in Figure 4.1).

On the two Kodak DSLRs, a Nikon AF 28-200mm / 3.5-5.6G IF-ED lens was used at 180 mm focal length. Note that the CCD sensors of the cameras have different extents. The 35 mm equivalent focal length is therefore 240 mm for the Kodak DCS760 and 180 mm for the Kodak DCS Pro 14n. The Photometric CoolSNAP K4 is used with a Schneider-Kreuznach Componon-S 5.6/135 lens with 135 mm (35 mm equivalent of 270 mm). Figure 4.2 shows the field of view of the respective cameras. The maximum spatial resolution of the material sample is 280 DPI, 330 DPI and 290 DPI for the camera models.



(a) 2002 **(b)** 2004 **(c)** 2010

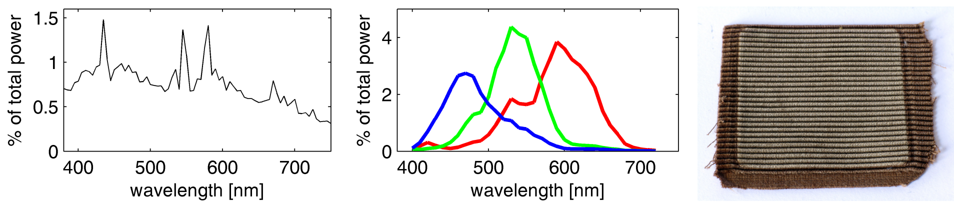
Figure 4.2: *Original measurement images for $\theta_o = 0°$, taken with the Kodak DCS760 (a), the Kodak DCS Pro 14n (b) and the Photometric CoolSNAP K4 (c). The pictures also illustrate the progression in the design of the sample holder. Notice the increase in size of the registration border and the utilization of an additional inset (c).*

4.5.1.3 Light Source

As light source, we employ a full-spectrum Broncolor F575 lamp with a 575 W Osram *hydrargyrum medium arc length iodide* (HMI) bulb. We use a parabolic reflector to achieve a directional light characteristic, so that the incoming directions ω_i are approximately the same at every point on the sample surface. After initial experiments, a UV filter was added to prevent damage of the material sample from the prolonged exposure of several hours necessary for full BTF measurements (see Figure 4.3c). Still, the lamp shows an even distribution of energy across all wavelengths considered by the RGB Bayer pattern CFAs or the spectral filter (see Figure 4.3) and has a color-temperature of 6,000 K. This facilitates to capture a natural impression of the reflectance with color characteristics comparable to daylight illumination when employing the RGB sensors of the Kodak DSLRs.

We also tested an Oriel *quartz tungsten halogen* (QTH) lamp with 1,000 W and a very smooth spectrum at a color temperature of 3,200 K. However, the lamp was disregarded because it showed a very low energy in the blue spectral bands and has an expected lifetime of 150 hours, in combination allowing only two multispectral measurements in a row.

To initially determine the accurate placement of the lamp, the robot presents a planar white target with increasing inclination angles θ_i. The brightness of the



(a) HMI bulb **(b)** DSLR Camera **(c)** UV damage

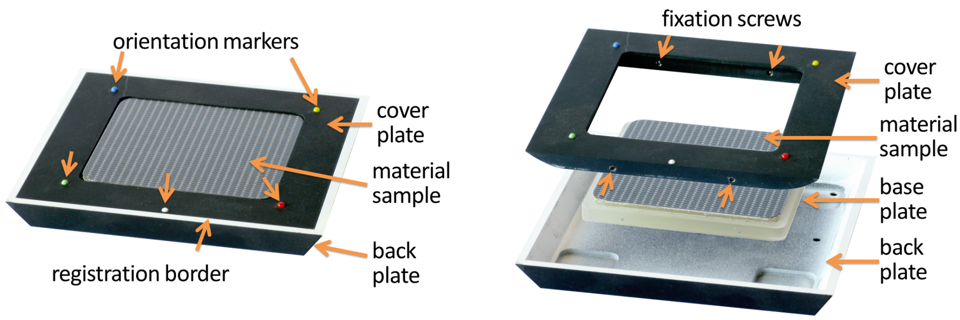
Figure 4.3: *Spectral power distribution of the employed HMI bulb (a) and sensitivity of the DSLR camera (Kodak DCS Pro 14n) (b). The red, green and blue curves correspond to the respective primaries in the Bayer pattern. (c) shows the damage caused prior to installing a UV filter in front of the lamp: The uncovered area of the material sample is bleached due to prolonged UV exposure during measurement.*

target is observed through the camera, which is arranged perpendicular to the light direction (i.e. in the center of the rail). For $\theta_i < 90°$, the white target should still be illuminated by the lamp, whereas for $\theta_i \geq 90°$ this should no longer be the case. The position and orientation of the lamp is adjusted manually until an approximate placement at $\theta_i \approx 90°$ is achieved. The procedure for adjusting ϕ_i is similar.

4.5.1.4 Sample Holder

The material sample that is presented to the camera is held tightly in place by a separate bespoke sample holder that is grasped by the robot. Thus, the material can be prepared without a hustle prior to acquisition. The sample holder has to fulfill multiple requirements: First, the sample has to be held tight enough, not too move or change shape even in a headlong position. Second, the maximum size is restricted by the robot's working envelope but has to be large enough to contain the spatial variations of the captured material. Thirdly, it should facilitate automatic registration and postprocessing of all captured images.

For this, our sample holder consists of three distinct parts, a back plate, a base plate and a cover plate, depicted in Figure 4.4b. The cover plate and back plate are made from aluminium that was milled by a CNC mill. The black coating is achieved by airbrushing the parts with matte black blackboard paint. In contrast to other black spray paint, we found the blackboard paint to show virtually no problematic direction depending highlights. A rectangular patch of the material is applied to the base plate made out of acrylic glass. The base plate is embedded into the cover plate, so that the surface of the material is on the same level as the cover. This is then fixated by four screws that penetrate the acrylic glass. Depending on the



(a) sample holder with sample **(b)** individual parts

Figure 4.4: *The sampleholder employed in the gonioreflectometer setup.*

material, the sample is either held in place by mechanical pressure from the cover plate or it is glued onto the base plate.

The cover plate exhibits several markers, aimed to facilitate automatic registration (see Figure 4.4a). First, the white registration border at the outside that is used to rectify the captured images (see Section 4.1). Furthermore, five differently colored orientation markers are used to verify the orientation and rectification.

Over time, several changes have been made to this design (see for example Figure 4.4). Most notably, the width of the cover plate and registration border has increased to avoid recognition problems during the automatic registration. The sides of the back plate were chamfered to avoid misdetection due to low contrasts under some light directions. Furthermore, the registration border, which originally were white colored stripes on the black cover plate, was separated to the back plate to show a more distinct edge. Eventually, for the spectral measurements in [RSK10], an additional inset with registration borders was added.

From 2004 onwards, all constructed sample holders have a size of 13 cm$\times$13 cm and a height of approximately 1 cm to 1.5 cm. In all cases, the cover plate gives room for a 8 cm$\times$8 cm region of the material sample. With the additional inset (see Figure 4.2c), the effective sample size is reduced to 6.5 cm$\times$6.5 cm.

4.5.2 Calibration

Due to the bad repeatability, a full a priori calibration of the cameras is not feasible. However, we calibrate correction factors for the lens distortion by capturing a checkerboard pattern. We do not attempt to recover any other camera parameters or light source positions for single measurements. Since a telephoto lens with a long focal length of 180 mm is employed, we instead assume the camera to be orthographic. Similarly, we consider the light source to be perfectly directional. Note that this is merely a crude approximation. In later setups, described in sections 4.6 and 4.7, we employ the more sophisticated models of finite projective cameras with lens distortions (see Section 3.5.1) and light sources with spot light characteristics. Still, at the given distances of 170 cm to the camera and 240 cm to the light source, the deviation of the direction across the sample is at most 1.9° and 1.4° respectively. This deviation is in the same order of magnitude as the error introduced by the robot arm and rail system (see row "geometric repeatability" in Table 4.5 in Section 4.8.1). Putting more effort into a different camera and light model would therefore not really improve the precision.

We assume that the hardware correctly takes up the ideal directions set by the measurement program. In principle, more accurate directions could be deduced for each measured image from calibration markers on the sample holder. We first-time implemented such a calibration refinement procedure for the successor setup described in Section 4.6. Nonetheless, in order to bring the spatial positions of different images into subpixel precise alignment, we additionally employ the registration border found on the target. This registration step is performed as part of the postprocessing after the measurement.

To facilitate the measurement of reflectance values, a radiometric calibration of the setup is performed. In the following, we assume the captured radiance to be directly proportional to the employed exposure time. Although digital camera sensors can exhibit nonlinear exposure time dependent effects, such as blooming and leakage currents, this assumption is approximately correct and finds widespread application in HDR imaging.

First *dark frames* have to be subtracted from all images to correct for hot pixels and sensor bias. Thus, an image $\mathcal{D}_\lambda$ of the completely unlit room is captured for every wavelength band $\lambda \in \Lambda$, using the same exposure time as the BTF measurement. Moreover, the response function χ of the camera needs to be inverted to obtain energy values proportional to radiance from the pixel values of the raw images. For this, the inverse response function χ_λ^{-1} is computed for every wavelength band from shots of a white standard with varying exposure times [RBS03].

This way, for a given direction combination ω_i, ω_o and spectral band λ, radiance values up to a unknown but constant factor $\alpha_{\mathbf{x},\omega_i,\lambda}$ can be derived by taking an image $\mathcal{I}_{\omega_i,\omega_o,\lambda}$:

$$\alpha_{\mathbf{x},\omega_i,\lambda} L_o\left(\mathbf{x}, \omega_o, \lambda\right) = \frac{\chi_\lambda^{-1}\left(\mathcal{I}_{\omega_i,\omega_o,\lambda}\left(\mathbf{x}'\right) - \mathcal{D}_\lambda\left(\mathbf{x}'\right)\right)}{T}. \tag{4.1}$$

Here, $\mathbf{x}$ denotes a position on the sample's surface and $\mathbf{x}'$ its corresponding pixel in the camera image. The correspondence is determined by the mapping explained in Section 4.4. $L_o(\mathbf{x}, \omega_o, \lambda)$ is the reflected radiance in $\mathrm{Wm}^{-2}\mathrm{sr}^{-1}$ observed by the camera at pixel $\mathbf{x}'$. The term T refers to the exposure time of the shot.

Note that $\alpha_{\mathbf{x},\omega_i,\lambda}$ is dependent on wavelength, illumination direction and spatial position in the image. The factor accounts for the mixture of the irradiance of the light source (including spatially varying attenuation and vignetting), the (spatially varying) opacity of the different spectral filters and vignetting by the camera lens.

To correct for all of these effects at the same time, we capture a set of *white images* $\mathcal{W}_{\omega_i,\omega_o,\lambda}$ of a white standard instead of a material sample, using the same wavelengths and direction combinations as the actual measurement. We employ SphereOptics Zenith UltrawWhite [Sph], P/N SG3110, which – according to the datasheet – exhibits an almost perfectly Lambertian reflection with about 99% albedo across the visible spectrum. Our experiments indicate that the SphereOptics material is in fact not completely diffuse at grazing angles. We therefore employ a fitted Cook-Torrance BRDF model [CT82] to describe the reflectance of the white standards. However, for the sake of readability, we will assume a Lambertian reflectance in the equations presented in this thesis. Using a_λ to denote the known albedo of the white standard for wavelength λ, we can therefore approximate the reflectance with the constant factor $\frac{a_\lambda}{\pi}$.

Under the reasonable assumption that the considered materials do not actively emit light but only reflect incoming light, the rendering equation (Equation 3.9) defines the amount of radiance captured by the camera for a point $\mathbf{x}$ in direction ω_o as

$$L_o(\mathbf{x}, \omega_o, \lambda) = \int_\Omega \rho(\mathbf{x}, \omega_i, \omega_o, \lambda)\, L_i(\mathbf{x}, \omega_i, \lambda) \cos\theta_i \,\mathrm{d}\omega_i \tag{4.2}$$

for some arbitrary surface reflectance function ρ and incoming radiance $L_i(\mathbf{x}, \omega_i, \lambda)$.

From Equation 4.1, we know that we can obtain the radiance reflected off the white standard $L_o^w(\mathbf{x}, \omega_o, \lambda)$ up to a proportionality factor $\alpha_{\mathbf{x},\omega_i,\lambda}$ from the captured pixel value. Under the mentioned approximation that the white target is perfectly Lambertian with known albedo a_λ, ρ is simply $\frac{a_\lambda}{\pi}$ and the captured radiance $\alpha_{\mathbf{x},\omega_i,\lambda} L_o^w(\mathbf{x}, \omega_o, \lambda)$ is explained by Equation 4.2 as

$$\begin{aligned}
\alpha_{\mathbf{x},\omega_i,\lambda} L_o^w(\mathbf{x}, \omega_o, \lambda) &= \alpha_{\mathbf{x},\omega_i,\lambda} \int_\Omega \frac{a_\lambda}{\pi} L_i(\mathbf{x}, \omega_i, \lambda) \cos\theta_i \,\mathrm{d}\omega_i \\
&= \alpha_{\mathbf{x},\omega_i,\lambda} \frac{a_\lambda}{\pi} \int_\Omega L_i(\mathbf{x}, \omega_i, \lambda) \cos\theta_i \,\mathrm{d}\omega_i. \qquad (4.3)
\end{aligned}$$

Here, L_i denotes the (unknown) radiance coming from a light source on the considered wavelength band.

Furthermore, we know from radiometry (see Section 3.2.2) that the irradiance $E^w_{\mathbf{x},\omega_i,\lambda}$ at the observed point on the surface must be

$$\begin{aligned} E^w_{\mathbf{x},\omega_i,\lambda} &= \int_\Omega L_i\left(\mathbf{x},\omega_i,\lambda\right)\cos\theta_i \,\mathrm{d}\omega_i \\ &= \frac{\pi}{a_\lambda} L^w_o\left(\mathbf{x},\omega_o,\lambda\right). \end{aligned} \tag{4.4}$$

Thus, the irradiance $E^w_{\mathbf{x},\omega_i,\lambda}$ can be determined from the pixel value using equations 4.1 and 4.4 up to a factor of $\alpha_{\mathbf{x},\omega_i,\lambda}$ as

$$\alpha_{\mathbf{x},\omega_i,\lambda} E^w_{\mathbf{x},\omega_i,\lambda} = \frac{\pi}{a_\lambda} \frac{\chi^{-1}_\lambda\left(\mathcal{W}_{\omega_i,\omega_o,\lambda}\left(\mathbf{x}'\right) - \mathcal{D}_\lambda\left(\mathbf{x}'\right)\right)}{T}. \tag{4.5}$$

Now, consider the measurement of a material with an arbitrary unknown reflectance function $\rho_m\left(\mathbf{x},\omega_i,\omega_o,\lambda\right)$. Let us further assume that the light illuminating the observed surface point comes from a cone of directions covering only a small solid angle ω. In our scenario, this assumption is reasonable, as the only source of illumination is a single light source with point light characteristics. For a single pixel in a single measurement image we can thus consider ω_i and ω_o to be fixed directions with negligible solid angles and the depicted point $\mathbf{x}$ to be a fixed single position with negligible area. Then, ρ_m becomes a constant and the measured radiance $\alpha_{\mathbf{x},\omega_i,\lambda} L^m_o\left(\mathbf{x},\omega_o,\lambda\right)$ is given via Equation 4.2 as

$$\begin{aligned} \alpha_{\mathbf{x},\omega_i,\lambda} L^m_o\left(\mathbf{x},\omega_o,\lambda\right) &= \alpha_{\mathbf{x},\omega_i,\lambda} \int_\Omega \rho_m L_i\left(\mathbf{x},\omega_i,\lambda\right)\cos\theta_i \,\mathrm{d}\omega_i \\ &= \alpha_{\mathbf{x},\omega_i,\lambda}\, \rho_m \int_\Omega L_i\left(\mathbf{x},\omega_i,\lambda\right)\cos\theta_i \,\mathrm{d}\omega_i \end{aligned} \tag{4.6}$$

Using Equation 4.4 to substitute the integral with $E^w_{\mathbf{x},\omega_i,\lambda}$, we can thus determine the value of ρ_m for the fixed sample $\left(\mathbf{x},\omega_i,\omega_o,\lambda\right)$ via

$$\begin{aligned} \alpha_{\mathbf{x},\omega_i,\lambda} L^m_o\left(\mathbf{x},\omega_o,\lambda\right) &= \alpha_{\mathbf{x},\omega_i,\lambda}\, \rho_m\left(\mathbf{x},\omega_i,\omega_o,\lambda\right) E^w_{\mathbf{x},\omega_i,\lambda}. \\ \Leftrightarrow \rho_m\left(\mathbf{x},\omega_i,\omega_o,\lambda\right) &= \frac{\alpha_{\mathbf{x},\omega_i,\lambda} L^m_o\left(\mathbf{x},\omega_o,\lambda\right)}{\alpha_{\mathbf{x},\omega_i,\lambda} E^w_{\mathbf{x},\omega_i,\lambda}} \\ &= \frac{a_\lambda}{\pi} \frac{\alpha_{\mathbf{x},\omega_i,\lambda} L^m_o\left(\mathbf{x},\omega_o,\lambda\right)}{\alpha_{\mathbf{x},\omega_i,\lambda} L^w_o\left(\mathbf{x},\omega_o,\lambda\right)}, \end{aligned} \tag{4.7}$$

with $\alpha_{\mathbf{x},\omega_i,\lambda} L^m_o\left(\mathbf{x},\omega_o,\lambda\right)$ and $\alpha_{\mathbf{x},\omega_i,\lambda} L^w_o\left(\mathbf{x},\omega_o,\lambda\right)$ being the measured values from the image $\mathcal{I}_{\omega_i,\omega_o,\lambda}$ taken during measurement and its corresponding white image $\mathcal{W}_{\omega_i,\omega_o,\lambda}$, respectively.

Finally, using equations 4.1, 4.5 and 4.7, the spatially varying reflectance samples $\rho_m(\mathbf{x}, \omega_i, \omega_o, \lambda)$ for an image $\mathcal{I}_{\omega_i,\omega_o,\lambda}$ captured under a given angular configuration ω_i, ω_o can directly be computed from the pixel values as

$$\rho_m(\mathbf{x}, \omega_i, \omega_o, \lambda) = \frac{L_o^m(\mathbf{x}, \omega_o, \lambda)}{E^w_{\mathbf{x},\omega_i,\lambda}} = \frac{\chi_\lambda^{-1}\left(\mathcal{I}_{\omega_i,\omega_o,\lambda}(\mathbf{x}') - \mathcal{D}_\lambda(\mathbf{x}')\right) a_\lambda}{\chi_\lambda^{-1}\left(\mathcal{W}_{\omega_i,\omega_o,\lambda}(\mathbf{x}') - \mathcal{D}_\lambda(\mathbf{x}')\right) \pi}. \tag{4.8}$$

As corresponding measurement image $\mathcal{I}_{\omega_i,\omega_o,\lambda}$ and white image $\mathcal{W}_{\omega_i,\omega_o,\lambda}$ are used, the factor $\alpha_{\mathbf{x},\omega_i,\lambda}$ is simply canceled out.

Since the correction with a full set of white images requires an enormous amount of calibration data and the poor repeatability of the setup complicates a precise spatial alignment, a simplification is proposed in [RSK10]. Instead of all angular combinations, the white target is only captured under the perpendicular view and light direction and a single average value over a region of interest in the resulting image is used. The correction with this reduced set of factors neglects any spatial variation in α but still accounts for the dependency on wavelength. Nonetheless, for setups with a better repeatability one would like to avoid such a crude approximation. For the Dome 2 setup, we use a more accurate but still compact white image representation, explained in Section 4.7.2.2.

4.5.3 Measurement Process

The sample holder with the prepared sample is mounted on the robot. Before beginning the automated data acquisition, the desired ISO speed, aperture and a fixed exposure time per wavelength are chosen manually. Although exposure bracketing could be employed, this has never been implemented. Still, different exposure times are used for the different wavelength bands. All other camera settings remain fixed throughout the measurement. A single personal computer executes the automatic measurement program, controlling the robot, rail system, tunable spectral filter and camera. Currently, this is an Intel Core 2 Quad with 2.67 GHz and 2 GB RAM.

The measurement of different angular configurations is completely sequential. Thus, the time necessary for measurement increases linear with the number of angular combinations and quadratic with the number of samples per hemisphere. Hence, we limit ourselves to an angular sampling of 81 directions, i.e. 6,561 combinations. The angular samples are distributed in six rings at varying inclination angles θ. Each ring is divided into a different amount of azimuthal angles with distance $\Delta\phi$ to achieve an even distribution of samples across the hemisphere. The average minimal distance between two sampled directions on the hemisphere is $14.7° \pm 0.4°$ for sampling set 1 or $16° \pm 0.8°$ for sampling set 2 (see Table 4.1). The

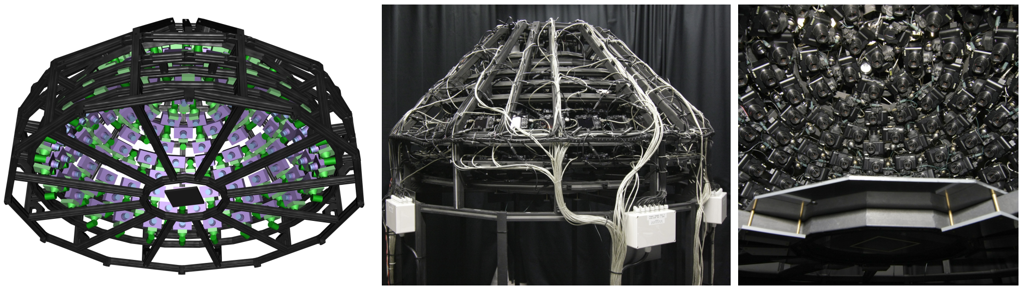
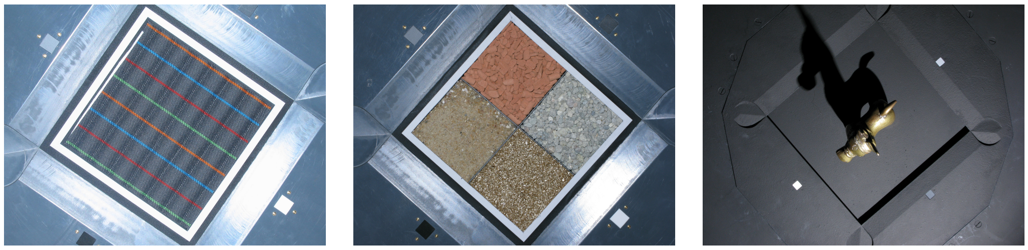


Figure 4.5: *The Dome 1 setup as a schematic illustration (left) and in photographs from the outside (center) and inside (right).*

directions are not distributed completely uniformly, but the low standard deviations (i.e. $\pm 0.8°$ and $\pm 0.4°$) indicate a good approximation. The selected azimuthal distribution of the samples ensures that, for a planar piece of material, the ideal reflection direction is captured. For the 81 configurations with identical light and view directions $\omega_i = \omega_o$, an offset of $10°$ was added to the light direction, so that the camera would not occlude the light source.

For reaching the different angular combinations, the robot arm and camera automatically reposition. Movements of the robot arm take between 1 s and 5 s. Moving the camera on the rail takes longer, also because the mechanical movements induce vibrations, requiring a waiting period before taking a picture. To minimize delays, the sample points are ordered in that way that in most cases only the wrist of the robot arm needs to be turned. Moreover, the ordering minimizes the movement of the camera, since this is the most time-consuming operation.

When capturing RGB data, the camera takes a single picture with the predefined exposure time for each angular configuration. The raw images of the Kodak DCS Pro 14n are about 13 MB in size, adding up to 83 GB per measurement. Thus, the images need to be directly downloaded to the control PC and stored on the hard disk. Typical measurement times are about 14 hours. For a multispectral measurement, it is additionally necessary to tune the spectral filter to the different bands. The camera is triggered after each filter change. In order not to waste any time, changing of filters runs in parallel to the data transmission. Still, multispectral measurements with 32 narrow bands take 60 hours. The images of the Photometric CoolSNAP K4 are 6 MB in size, thus requiring a total of 1.2 TB per spectral measurement.



(a) fabric material, Canon PowerShot A75 **(b)** granule materials, Canon PowerShot G9 **(c)** Donkey object, Canon PowerShot G9

Figure 4.6: *Pictures taken by the topmost camera of the Dome 1. (a) depicts a fabric material sample taken with a Canon PowerShot A75 camera. (b) shows four granule material samples captured simultaneously and (c) a 3D object, both captured with the Canon PowerShot G9 camera. The material in (a) is used with courtesy of Volkswagen AG.*

4.6 Dome 1

The Dome 1 setup (see Figure 4.5), constructed in 2004 and published in [MMS*04, MBK05, RMS*08, SWRK11], is a completely view-parallelized BTF acquisition device. To the best of our knowledge, it is the only camera array setup that provides a dense angular sampling without relying on moving cameras or moving the sample. It mounts 151 compact cameras. Between 2008 and 2009 it was completely re-equipped with a new set of cameras. In 2011 [SWRK11], it was furthermore extended to support an automated, integrated 3D geometry acquisition based on structured light. More details on the utilized structured light approach can be found in Section 5.4.

Figure 4.6b demonstrates the capability to capture delicate materials samples, in this case granules and sands, due to the horizontal alignment and rigidity of the sample holder. Figure 4.6c shows the acquisition of a complex 3D object.

4.6.1 Hardware

Due to the practical experience with our gonioreflectometer setup, we identified the long measurement time is a major hurdle for BTF measurements that needs to be overcome. Thus, the design goal was a maximal parallelization of the acquisition and complete avoidance of any mechanical movement. At the time of construction, this was already approached by Han and Perlin [HP03], using a kaleidoscope setup. Yet, spatial resolution and possible sample sizes were dissatisfactory. In order to allow for practice-oriented sample sizes and resolution in the spatial domain, a hemisphere of cameras was implemented instead. The parallel acquisition with a large number of cameras also holds the advantage that the workload during a measurement is equally distributed over many components. This is favorable in terms of durability.

Since a setup without moving parts necessarily only allows for a single, fixed angular sampling, the number of directions on the hemisphere was provisionally increased to 151 rather than the 81 of the gonioreflectometer measurements.

4.6.1.1 Gantry

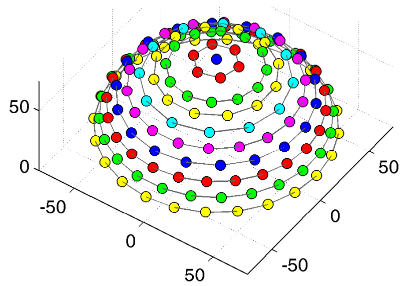
The 151 cameras are held by a hemispherical gantry structure with an outer diameter of approximately 190 cm, holding the cameras at a distance of 65 cm to the sample. It is organized in ten camera rings with different inclination angles from 0° to 75°. Each ring holds a different amount of cameras, distributed across the azimuthal angles with distance $\Delta\phi$. The resulting sampling, shown in Table 4.2, covers the hemisphere with an almost uniform distribution of directions, having an average minimal distance of $9.4° \pm 1°$. In azimuthal direction, the dome's rings are split into twelve segments. The rings are held by 18 vertical struts: one per segment and six additional struts in between each pair. The sample holder mount is held by rods, protruding from each of the segments at an inclination of 90° and meeting in the center. The hemispherical gantry with the cameras is standing on twelve legs which are strutted as well for additional stability. In contrast to the original design depicted in Figure 4.5 (left), two pairs of rods that hold the sample holder have been removed to allow an operator access to the inside of the dome. To enter, e.g. for placing a material sample or performing maintenance, the operator has to step in through the openings from below. Due to the legs, it is possible to stand upright while working inside.

The frame is made completely from Bosch Rexroth Profiles out of aluminium. It has sufficient strength for holding all cameras and auxiliary components, such as cables, power couplings or projectors. Due to the many struts, the gantry is perfectly rigid.

4.6.1.2 Cameras

To keep costs and proportions manageable, we decided to employ compact *point-and-shoot* (P&S) cameras instead of bulky DSLRs. This has the additional advantage that the built-in flashes found in these cameras can serve as light sources. In its first configuration from 2004 [MMS*04], the Dome 1 setup was equipped

θ	$\Delta\phi$	# dir's
$0°$	$-^{\star}$	1
$11°$	$60°$	6
$23.5°$	$30°$	12
$30°$	$30°$	12
$37.5°$	$30°$	12
$45°$	$20°$	18
$52.5°$	$20°$	18
$60°$	$15°$	24
$67.5°$	$15°$	24
$75°$	$15°$	24



*Only one direction at $\phi = 0°$.

Table 4.2: *The fixed hemispherical direction sampling in our Dome 1 setup.*

with Canon PowerShot A75 cameras. The CCD sensor has a resolution of 3.2 megapixel with 10 BPP and a Bayer pattern CFA for RGB color. Unfortunately, the camera does not give access to the raw 10 bit data, but only stores and transmits color-processed and JPEG compressed 8 BPP images. Canon provides an SDK to remotely control the camera via USB, which allows to change focal length, ISO speed, aperture, exposure time and flash intensity (minimum, medium, maximum). It also allows to perform an autofocus and toggling flash exposure.

Setting a focal length of 16.22 mm (35 mm equivalent focal length: 116 mm) for the built-in lens allows to capture a material sample with 235 DPI (see Figure 4.6a to get an impression of the field of view).

However, after about a hundred measurements, the CCD chip of the low-end PowerShot A75 cameras started to fail. Shifted colors, overexposed image regions and clearly visible horizontal stripe patterns appeared. Eventually, the cameras did not produce image content at all. This turned out to be a systematic defect of the camera model [Can05], being caused by loosening internal wiring of the CCD chip's electronics. Thus, between 2008 and 2009 the Dome 1 was re-equipped with the medium segment Canon PowerShot G9. The new cameras have a higher sensor resolution of twelve megapixel with 12 BPP. Although this camera supports to store raw images on the internal memory card, the SDK foresees no way of raw image transmission. We again obtain color-processed and JPEG compressed 8 BPP images. Thus, all radiometric correction steps described in Section 4.6.2.2 apply for both camera types.

With the PowerShot G9, we capture material samples at a spatial resolution of 450 DPI (see Figure 4.6b) using a focal length of 22 mm (35 mm equivalent focal length: 104 mm). For capturing the appearance of complete objects, we adjust the focal length to cover the necessary working volume. Figure 4.6c depicts an object captured with a focal length of 11 mm (35 mm equivalent focal length: 52 mm), yielding a maximum spatial resolution of 225 DPI.

Figure 4.7: *Modified Canon PowerShot G9 camera.*

As the Canon SDK does not give access to the raw sensor data but instead transmits a color-processed and JPEG-compressed 8 BPP image, the camera's response function is not linear. To provide pleasing results close to human perception, the resolution is higher for low energies. When considering the most favorable resolution, the PowerShot A75 captures incident radiance with a dynamic range of 28 dB (at ISO 50) to 21 dB (at ISO 400), but exhibits gross quantization errors of up to 2.4% for the highlights. The situation is almost the same for the PowerShot G9, showing 26 dB (at ISO 80) to 24 dB (at ISO 400) with quantization errors of up to 1%. We thus use exposure bracketing with sufficient overlap for capturing high dynamic range values with almost equal resolution. We employ the built-in flash as a light source. The flash emits a single, strong pulse of light in a fraction of a second. Hence, it is not possible to control the overall exposure using different exposure times. However, we also have to use a fixed narrow aperture of $f/8$ in order to have a sufficiently high depth of field. Thus, we instead vary the flash intensity and the ISO speed of the sensors to obtain a multiexposure image series, yielding a dynamic range of about 33 dB for the PowerShot A75 and 44 dB for the PowerShot G9.

We slightly modified the hardware of the cameras in a few aspects. First, as mentioned in Section 4.1, we painted reflective surfaces on the front of the camera black and blinded the cameras' autofocus LED lights with black tape. The latter measure also prevents the cameras from confusing each other during the focus procedure. Second, the PowerShot G9 does not provide a jack for an auxiliary power supply. At first we employed self-made battery dummies but eventually found the mechanical contacts to be too unreliable and soldered the power cable directly to the cameras. Finally, we make the power button of the cameras remotely operable by soldering an additional cable to the button as well. The resulting modifications on a PowerShot G9 camera are shown in Figure 4.7.

4.6.1.3 Light Sources

We use the built-in flash lamps of the cameras as light sources. This has several advantages: First, it saves space, wiring and controller logic. Secondly, the camera manufacturer took care that the flash illumination has a well-chosen spectrum to produce natural colors in the images. Thirdly, the flashes have sufficient power for short exposure times, even for materials with low albedo. Finally, in contrast to a strong continuous light source, multiple but very short pulses do expose the material sample to exactly the amount of light necessary for the imaging process, avoiding prolonged UV exposure. Using 151×3 flash strobes emits roughly as much UV light (287 nm – 400 nm) as a few seconds of an off-the-shelf 100 watt tungsten halogen lamp.

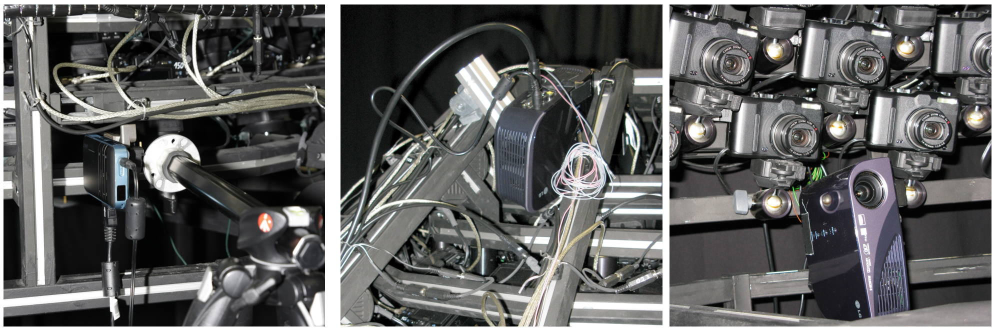
The built-in flashes are affixed on the camera and close to the lens. Hence, the direction sampling of view and light hemisphere is almost identical. However, we still account for the small difference by separately calibrating the point of origin of the flash illumination. We assume the flash illumination to have a quadratic falloff behavior with respect to distance and a conic distribution with a cosine falloff. Unfortunately, the flashes show a low repeatability regarding color and intensity. We therefore measure a correction factor for every single flash exposure and account for it during the radiometric correction of the captured images. More details on this can be found in Section 4.6.2.2.

We additionally employ a continuous light source that is installed directly above the material sample in the tip of the dome. The off-the-shelf lamp-socket with a tungsten halogen bulb is remotely toggled using a radio plug. The lamp is used as a light source for camera autofocus and to verify the correct placement of the sample and focal length of the cameras.

4.6.1.4 Projectors

For acquiring 3D geometry of objects or material samples, we perform an integrated structured light acquisition. We use projectors to impose coded patterns onto the object. Then the 151 cameras capture the illuminated surface. We decode the imaged patterns and triangulate their 3D positions. In order to acquire the complete shape without requiring to reposition the object, we use multiple projectors to provide structured light illumination from all sides.

For first experiments with this approach, we simulated the arrangement of multiple projectors by placing a single Acer C20 LED-Projector (848×480 pixels, LED-DLP, 20 lm), mounted on a tripod (see Figure 4.8a), at five to eight different positions: five 72° azimuthal steps on the height of the lowest camera ring, which is placed at an inclination of $\theta \approx 75°$, and two or three positions at $\theta \approx 20°$.



(a) Acer C20 on tripod **(b)** LG HS200G on gantry

Figure 4.8: *Projectors in the Dome 1 setup. At first, different projection directions were achieved by placing an Acer C20 projector (a) on a tripod at multiple positions. Later, we installed nine LG HS200G projectors (b) at fixed positions on the gantry.*

These initial experiments were considered to be a success, as the first four captured objects depicted in Figure 5.9 demonstrate. However, manually placing the projector is cumbersome and contradicts the automatic acquisition. Furthermore, the imaging quality and the brightness of the Acer C20 projector were unsatisfactory.

As a consequence, we installed nine LG HS200G projectors (800×600 pixels, LED-DLP, 200 lm); six at $\theta \approx 82.5°$ inclination with an even spacing of $\Delta\phi = 60°$ and three at $\theta \approx 17°$ with $\Delta\phi = 120°$. These particular projector models were chosen for several reasons: First of all, they are compact enough to find a place in the tightly arranged gantry structure (see Figure 4.8b). Then, they have a sufficiently near projection distance and large depth of field. The LED light source does not produce too much heat. They can almost instantly be switched on and off without long warm up or cool down times. And finally, they are also reasonably priced consumer products and therefore blend nicely with the rest of the Dome 1 hardware selection philosophy. Although the resolution of the projectors is rather low, this is compensated by a multiprojector-based superresolution approach, we developed together with Weinmann *et al.* [WSRK11].

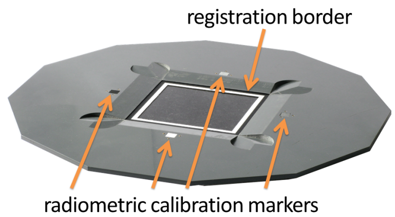
We use Gray code [Gra53] to uniquely identify points on an object surface. Here, the number of patterns depends on the resolution of the projector. To be more robust, we employ vertical as well as horizontal codes, an additional fully lit pattern and a second pass through the sequence with the inverse of the former signal. More details about the code can be found in Section 5.4. We eventually project a total of $2\,(1 + \lceil \log_2 800 \rceil + \lceil \log_2 600 \rceil) = 42$ patterns, which are displayed on the projector by a PC via HDMI. Since, by principle, there is always only one projector switched on at a time, we can use a single computer and distribute the signal with a cascade of Aten VS184 4x HDMI splitters. We toggle the projectors by simulating the appendant remote control using computer-controlled infrared LEDs.

Unfortunately, using off-the-shelf consumer projectors also has some pitfalls. We observed that after turning on, the projection drifts and takes up to 15 minutes to stabilize. Additionally, the colors and intensities alternate periodically with a slightly irregular pattern. Often such a behavior comes from the usage of a color wheel and can be avoided (for black-and-white projection) by removing the component. However, our chosen projectors use LEDs with different spectra instead of a color-wheel. This makes it necessary to synchronize exposure with projector frequency in order to avoid intensity shifts. Note that the slowest frequency of the projector irregularities might still be faster than the projectors refresh rate of $60\,\mathrm{Hz}$. We measured the irregularities by deflecting the projection onto a screen with a mirror rotating at $60\,\mathrm{Hz}$. We found that, although the three primary colors seem to cycle at a higher frequency, the elements of the digital micromirror device produce an irregular pattern that repeats after exactly $\frac{1}{60}s$. Thus, we achieve synchronization by choosing exposure times in multiples of this fraction. This way, we ensure that the cameras always integrates over at least one full irregularity period, effectively avoiding flickering.

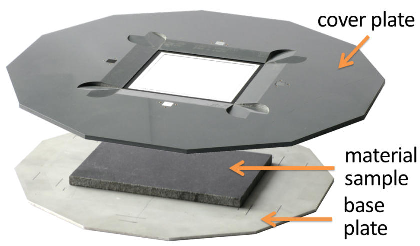
4.6.1.5 Sample Holder

Since the Dome setup does not require the material sample to be moved, there is no risk that the sample will change shape or get out of place. Thus, it is not necessary to completely mechanically restrain the sample in a sample holder, as it has been done for the gonioreflectometer. However, for planar material samples we still employ a sample holder design that combines a base plate with a cover plate, depicted in figures 4.9a and 4.9b. To prevent curling or wrinkling (e.g. in fabrics or wallpaper), the sample is either fixated on the base plate using double-sided tape or held in place by the weight of the cover plate. The back plate is made from aluminium while the cover plate is made of a hard PVC material. Both are milled with a CNC mill.

The cover plate also contains several markers to facilitate the automatic registration and radiometric correction of the flash illumination. A black-and-white registration border is framing the visible $10.5 \times 10.5\,\mathrm{cm}$ region of the sample. Four radiometric calibration markers are distributed around the sample. The markers are made from SphereOptics Zenith UltraWhite [Sph], showing almost perfectly Lambertian reflectance, except at grazing angles (see brief discussion in Section 4.5.2). We employ a set with four different albedos, P/Ns SG3053, SG3059, SG3080 and SG3102, diffusely reflecting 2%, 10%, 30% and 99% of the visible light.



(a) for materials



(b) individual parts

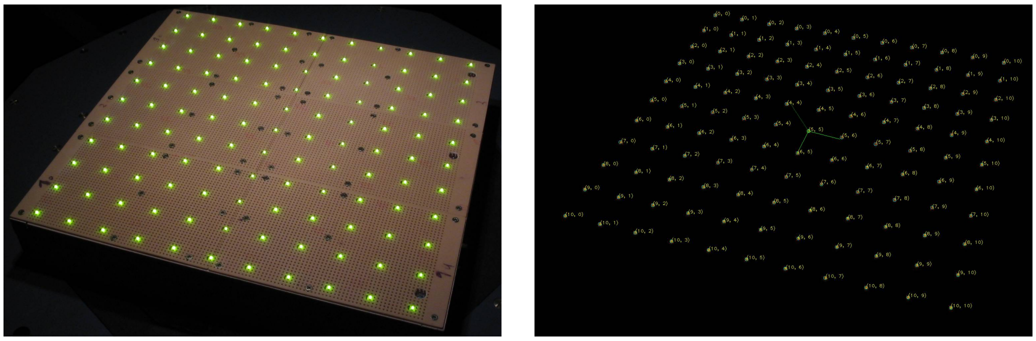
(c) for objects

Figure 4.9: *The sample holder design employed in the Dome 1 setup.*

For the acquisition of 3D objects, we utilize a variation of the sample holder design, which is demonstrated in Figure 4.9c. The sample holder is blackened in order not to cast any caustics or indirect light onto the object. Similar to the sample holder of the gonioreflectometer, we employ airbrushed blackboard paint. There is also no registration border, since the spatial domain of the BTF will be parameterized over the object surface and not a quadrilateral. Yet, the sample holder still has four radiometric calibration markers, since they are necessary to calibrate the flash illumination. The inner frame has an extent of 20.5 cm×20.5 cm, providing a larger acquisition volume in the case of objects.

4.6.2 Calibration

Since the Dome 1 setup is by construction completely rigid and does not require movement of cameras, light sources or the sample, we aim to have a more precise calibration than in the gonioreflectometer setup. The position of the cameras, and thus also of the light sources, can be determined a priori and remain fixed for multiple measurements. The same applies for the radiometric properties of the CCD sensors.



(a) **(b)**

Figure 4.10: *Geometric calibration target for the Dome 1 device: (a) shows the 11×11 LED calibration target under room light (for illustration purposes). (b) shows a picture taken under calibration conditions with annotation of automatically detected features. The LEDs are aligned on a stripboard in a regular grid with a horizontal and vertical distance of 27.94 mm, spanning a total square of 28 cm×28 cm. We assume the stripboard to be manufactured sufficiently accurate for our purpose. Pictures taken from [RMS*08].*

Unfortunately, the deviations of the cameras' flashes require a radiometric correction for each exposure. Furthermore, the poor repeatability of the cameras' zoom-lenses and autofocus as well as some mechanical play in the sample holder design require an additional fine calibration. This – as well as a registration for a precise alignment – is obtained using the registration borders, similar to the method employed for the gonioreflectometer. For 3D objects, a self-calibration of the camera parameters is instead performed using the structured light features. We dismiss the calibration of the projectors entirely, because of the mentioned problems with the initial shift of the projection.

Eventually, the precise geometric calibration, described in detail in the following section, allows us to employ the model of a finite projective camera (see Section 3.5.1) and the model of a spotlight for determining accurate sample directions for every spatial position.

4.6.2.1 Geometric Calibration

The procedure to establish an a priori camera calibration consists of an initial coarse calibration of the extrinsic parameters, which is followed by a subsequent nonlinear estimation of the intrinsic parameters. Further, for any given measurement, an additional fine calibration step is performed.

For the initial calibration, we use a planar calibration target with 11×11 LEDs (see Figure 4.10). The target is placed in the center of the dome instead of the sample holder. The emitters of the LEDs can be accurately detected in each camera image. We apply prior knowledge about the ideal direction ω_o to resolve the symmetry of the target. The advantage of using LEDs in comparison to typical checkerboard targets is that the emitters can even be robustly detected under grazing viewing angles. Using a fixed planar calibration target, however, is not sufficient for estimating both extrinsic and intrinsic parameters of the cameras.

As a consequence, we employ a two-tiered approach. First, the intrinsic parameters of the 151 cameras are assumed to be identical. This is a reasonable approximation, since all the cameras are of the same product model. Under this assumption, the calibration method proposed by Zhang [Zha00] can be used to estimate the common intrinsic parameters and individual extrinsic parameters for each of the cameras. In a subsequent step, the extrinsic parameters are assumed to be fixed and an optimization of the individual intrinsic camera parameters is performed. For this, the calibration target is captured with the different focal length settings accessible via the Canon SDK. The optimization is initialized with a linear extrapolation of the detected LED emitter positions using the idealized focal length.

While this calibration procedure yields good and stable results, the intrinsic parameters of the cameras are unfortunately not constant throughout multiple measurements. Although the SDK offers to set the camera to a given focal length, the repetition accuracy of the mechanical zoom for the built-in lens is not precise enough. Furthermore, it is necessary to perform an autofocus at the beginning of each measurement, also with low repeatability. In practice, the field-of-view differs by a significant amount of pixels. Therefore, a subsequent fine calibration of the camera parameters is performed for every single measurement. This step is performed as postprocessing after the measurement, but we will still discuss it as part of the geometric calibration.

In the case of flat material samples, we use the registration borders found on the cover plate (see Figure 4.9a) for registration and calibration. In contrast to the gonioreflectometer, the subpixel precise detection of the material sample region has to be performed only once per view direction instead for every image. As in all our setups, we employ the detected quadrilaterals to rectify the spatial samples. Furthermore, we use the corners as a set of accurate and reliable correspondences between the cameras to find their respective 3D positions and refine the camera parameters. In principle, four points that lie on a plane present a degenerate configuration for estimating all camera parameters. We perform a nonlinear optimization [Lev44] procedure and assume that it acts well-behaved when given a good initialization.

When capturing geometry and reflectance of objects, we deliberately do not employ registration borders (see Figure 4.9c). Instead, we make use of the structured light patterns for simultaneously reconstructing the 3D geometry and performing a self-calibration of the setup [WSRK11]. By decoding the structured light patterns in the 151 cameras, we obtain a large set of reliable and sufficiently accurate correspondences between the views. Given a set of correspondences, it is possible to obtain the depicted 3D geometry and the camera calibration simultaneously using *sparse bundle adjustment* (SBA) [LA09]. SBA performs a global nonlinear optimization, minimizing the reprojection error of the 3D points to the decoded labels in the camera images. However, it requires a good initialization and is susceptible to outliers, i.e. false correspondences due to decoding errors. We therefore follow an iterative approach, alternating between two steps: Fist, we triangulate the correspondences to obtain a 3D point cloud using the given camera calibration. Then, we update the camera calibration and the point cloud via SBA. In the first step, we employ a *random sample consensus* (RANSAC) [FB81] approach to eliminate outliers. A random subset of three cameras is used to triangulate a point and the other correspondences are used to accept or reject the obtained 3D point, based on its reprojection error. The procedure is initialized with the calibration obtained from the LED grid.

As the utilized flash light sources are affixed to the cameras, their positions are given by a fixed offset to the lens. We determined the offset using a ruler. After calibrating the cameras, we apply this offset to the computed center of projection to obtain the light's position. Furthermore, we assume that the light cone of the flash has the same direction as the camera's optical axis.

4.6.2.2 Radiometric Calibration

The radiometric calibration of the Dome 1 device is rather complicated. This is because of the sheer number of employed CCDs, the cameras' inability to transmit their raw data and most importantly the utilization of flash illumination. The cameras' flashes do not show a constant behavior. Instead, color and intensity vary for every discharge. Furthermore, we are forced to use different ISO speed settings and flash intensities for obtaining a multiexposure series. Each ISO speed again implies a different response function of the CCD. In total, the radiometric calibration of each image depends on the tuple describing a single flash discharge event $\mathbf{r} = (f, q, i)$, with f denoting the flashing camera, q the flash intensity quantity and i the ISO speed, as well as the response function $\chi_{c,i,\lambda}$ of the camera c taking the picture for color channel λ. Please refer to Table 4.3 for a comprehensive overview of all terms and symbols used to describe the radiometric calibration.

c	shooting camera index in $\{1, 2, \ldots, 151\}$
f	flashing camera index in $\{1, 2, \ldots, 151\}$
i	ISO speed setting in $\{50, 100, 400\}$ for PowerShot A75 and $\{80, 100, 400\}$ for PowerShot G9
q	flash intensity quantity in {minimum, medium, maximum}
$\mathbf{r}$	a specific combination of employed flashing camera, flash quantity and ISO speed, i.e. a tuple $\mathbf{r} = (f, q, i)$
$\mathcal{I}^{c,\mathbf{r}}$	image of camera c taken under radiometric conditions $\mathbf{r}$
λ	color channel index in {red, green, blue}
$\mathcal{I}^{c,\mathbf{r}}_{\lambda}(\mathbf{x})$	8 bit pixel value of image $I^{c,r}$ for spatial position $\mathbf{x}$ and color channel λ
$\chi_{c,i,\lambda}$	response function of camera c for ISO speed i and color channel λ
$L^c_{\mathbf{x},\lambda}$	radiance (in $\mathrm{Wm}^{-2}\mathrm{sr}^{-1}$) for spatial position $\mathbf{x}$ and corresponding direction towards camera c for color channel λ
k	index of radiometric calibration marker in $\{1, 2, 3, 4\}$
a_k	albedo of radiometric calibration marker k
$\bar{m}_{\mathbf{r},k}$	average pixel value for marker k in the image of camera 1 under radiometric condition $\mathbf{r}$
$\hat{E}_{\mathbf{r},k}$	predicted irradiance (in Wm^{-2}) at marker k for flash discharge with attributes $\mathbf{r}$
$\mathbf{v}_{f,k}$	vector in $\mathbb{R}^3$ from marker k to flash f
$\Vert\mathbf{v}_{f,k}\Vert_2$	distance between marker k and flash f
$\mathbf{n}_k$	surface normal in $\mathcal{S}^2$ of marker k.
$\mathbf{n}_f$	central direction in $\mathcal{S}^2$ of the flash cone for flash f (identical to the orientation of the camera with index f).
w	continuous weighting function with $w(p) = 0$ for underexposed or saturated and $w(p) = 1$ for well exposed values p
$\hat{E}_{\mathbf{r},\mathbf{x}}$	predicted irradiance (in Wm^{-2}) at spatial position $\mathbf{x}$ for flash discharge with attributes $\mathbf{r}$
$E_{\mathbf{r},\mathbf{x}}$	true irradiance (in Wm^{-2}) at spatial position $\mathbf{x}$ for flash discharge with attributes $\mathbf{r}$
$\kappa_{\mathbf{r}}$	radiant power (in W) of the flash discharge with attributes $\mathbf{r}$

Table 4.3: *Overview of symbols and terms used in the radiometric calibration of the Dome 1.*

We employ a two stage approach for radiometric calibration: In a nonrecurring first step, we calibrate the response functions $\chi_{c,i,\lambda}$ for each camera c, ISO speed i and color channel λ from pictures with varying exposure times [RBS03]. For this purpose, a white standard is lit with a continuous illumination. Similar to the considerations for the gonioreflectometer, the radiance can be computed from a given image $\mathcal{I}^{c,\mathbf{r}}$ that was made with camera c at radiometric attributes $\mathbf{r} = (f, q, i)$ as

$$\alpha_\lambda L^c_{\mathbf{x},\lambda} = \chi^{-1}_{c,i,\lambda}\left(\mathcal{I}^{c,\mathbf{r}}_{\lambda}(\mathbf{x})\right), \tag{4.9}$$

with $\mathcal{I}^{c,\mathbf{r}}_{\lambda}(\mathbf{x})$ denoting the grayscale value for spatial position $\mathbf{x}$ and color channel λ. However, here we refrain from subtracting a dark frame, because our experiments indicate that the cameras' internal image processing already performs this step. As in Equation 4.1, the radiance is only known up to a constant factor α. In our practical implementation, we consider the factor to be spatially uniform because we cannot easily measure its spatial variation in the Dome 1 setup.

Note that we do not divide by the exposure time. This is because we do not employ a continuous light source and hence do not record an integral of radiance over time. Instead, the flash discharge can be considered a Dirac delta function, rendering the exposure time irrelevant.

The second step of the radiometric calibration requires to establish the irradiance on the material sample. It is performed for every single flash discharge and is therefore part of the postprocessing of a measurement. As all cameras simultaneously capture images of one particular flash discharge, using the image of just one camera is sufficient for radiometrically calibrating the light source. For this, the pixel intensity values of four radiometric calibration markers attached to the sample holder (see Figure 4.9) are recorded in the image of the topmost camera $c = 1$ (see Figure 4.6 for examples). We employ multiple markers with different albedos a_k to ensure that at least one marker can reliably be used in a given raw image, whereas the others might be underexposed or oversaturated.

For a particular recorded flash discharge $\mathbf{r} = (f, q, i)$, the idealized irradiance at marker k can be predicted using

$$\hat{E}_{\mathbf{r},k} = \kappa_{\mathbf{r}} \left(\frac{\mathbf{v}_{f,k}}{\|\mathbf{v}_{f,k}\|_2} \cdot \mathbf{n}_k \right) \left(\frac{d}{\|\mathbf{v}_{f,k}\|_2} \right)^2 \left(\frac{\mathbf{v}_{f,k}}{\|\mathbf{v}_{f,k}\|_2} \cdot \mathbf{n}_f \right). \tag{4.10}$$

Here, the first dot product models the foreshortening of the light according to Lambert's cosine law. The next term models the quadratic light falloff. The last term models the falloff due to the conic shape of the flash. In our implementation, the quadratic light falloff is normalized to a distance of $d = 65\ \mathrm{cm}$, which approximately corresponds to the inner radius of the dome. The term $\kappa_{\mathbf{r}}$ denotes the true radiant power of the flash in watt. We include it in our theoretical considerations to end up with the correct radiometric units. However, the radiant power $\kappa_{\mathbf{r}}$ can eventually be modeled as part of the correction factor $\beta_{\mathbf{r},\lambda}$. Hence, we do not explicitly determine it in our implementation of the calibration procedure.

A correction factor $\beta_{\mathbf{r},\lambda}$, describing the variance of a particular flash discharge, can be obtained by taking the weighted average over all four markers

$$\beta_{\mathbf{r},\lambda} = \frac{1}{\sum_k w(\bar{m}_{\mathbf{r},k})} \sum_k w(\bar{m}_{\mathbf{r},k}) \frac{a_k}{\pi} \frac{\hat{E}_{\mathbf{r},k}}{\chi^{-1}_{1,i,\lambda}(\bar{m}_{\mathbf{r},k})}, \tag{4.11}$$

where $\bar{m}_{\mathbf{r},k}$ denotes the average pixel value for marker k. The term $\frac{a}{\pi}$ models the Lambertian reflectance of the respective marker. w is a weighting function to omit over- and underexposed markers from the computation of the factor. In the event of capturing 3D objects, a marker might be in shadow for some light directions (see Figure 4.6c). We account for that by applying a weight of zero in these cases.

Using these factors, the true irradiance at position $\mathbf{x}$ can be described as:

$$\alpha_\lambda E_{\mathbf{r},\mathbf{x}} := \beta_{\mathbf{r},\lambda} \hat{E}_{\mathbf{r},\mathbf{x}}. \tag{4.12}$$

The constant factor α_λ is contained because β is normalized by $\chi^{-1}_{1,i,\lambda}(\bar{m}_{\mathbf{r},k}) =: \alpha_\lambda L^1_{k,\lambda}$, i.e. the radiance-proportional energy from marker k in camera 1.

Figure 4.11: *The X-Rite ColorChecker Passport. The color rendition chart is used to calibrate the cameras' color profiles in the Dome 1 and Dome 2 as well as the cameras' response functions of the Dome 2 (see also Section 4.7.2.2).*

Finally, the high dynamic range reflectance values for a given combination of capturing camera c and flashing camera f can be obtained by combining the multiple differently exposed pictures $\mathcal{I}^{c,\mathbf{r}}$ in a weighted sum (similar to [RBS03])

$$\begin{aligned}\rho_{\mathbf{x},\lambda} &= \frac{1}{\sum_{q,i} w\left(\mathcal{I}_\lambda^{c,\mathbf{r}}\left(\mathbf{x}\right)\right)} \sum_{q,i} w\left(\mathcal{I}_\lambda^{c,\mathbf{r}}\left(\mathbf{x}\right)\right) \frac{\chi_{c,i,\lambda}^{-1}\left(\mathcal{I}_\lambda^{c,\mathbf{r}}\left(\mathbf{x}\right)\right)}{\beta_{\mathbf{r},\lambda}\hat{E}_{\mathbf{r},\mathbf{x}}} \\ &= \frac{1}{\sum_{q,i} w\left(\mathcal{I}_\lambda^{c,\mathbf{r}}\left(\mathbf{x}\right)\right)} \sum_{q,i} w\left(\mathcal{I}_\lambda^{c,\mathbf{r}}\left(\mathbf{x}\right)\right) \frac{\alpha_\lambda L_{\mathbf{x},\lambda}^c}{\alpha_\lambda E_{\mathbf{r},\mathbf{x}}}. \end{aligned} \tag{4.13}$$

As can be seen, the factor α_λ is canceled out.

Although the camera images' metadata specifies that the colors are given in sRGB, a comparison with multispectral measurement data indicates that Canon applies additional color processing, such as intensifying the saturation for some colors. We therefore perform an additional color calibration. We use an X-Rite ColorChecker Passport color rendition chart (see Figure 4.11) to establish the CIEXYZ color profile for each camera.

4.6.3 Measurement Process

Despite the construction with twelve segments, the Dome 1 is logically divided into eight azimuthal parts. Each octet consists of 19 or 18 cameras and has a separate power supply and control PC. The current control PCs are each equipped with an Intel Core 2 Quad CPU with 2.33 GHz, 1.75 GB RAM, a NVIDIA GeForce 9300 GPU and a 1 TB hard-drive. The camera and flash settings are controlled via USB and the captured images are directly transmitted to the respective computer.

Since the setup is rigid, the homography for spatial registration can be computed as soon as the focal length and autofocus are set. Thus, in case of flat materials, it is possible to directly perform the rectification during measurement. The control PCs have a sufficient computational capacity to process the 19 incoming images on the fly using the GPU. Nonetheless, the raw measurement data is stored on disk as well. The first of the client computers is furthermore used to show the patterns for structured light via HDMI.

The overall acquisition process is controlled via a ninth host computer that is connected to the clients via 100 Mbit/s Ethernet. The master computer is also responsible for the continuous autofocus light source, the remote control of the projectors and switching the camera power on and off.

After all cameras have been turned on and detected by their respective control computers, some basic camera settings, such as white balance, shutter-speed and aperture, are applied. The focal length is adjusted for the measurement task: 16.22 mm for the PowerShot A75 or 22 mm for the PowerShot G9 for materials samples, a flexible focal length for objects. Then, the autofocus procedure is performed and locked. For this, we shortly activate the continuous light source. In case the material sample shows a low contrast for some of the cameras, we place a printed black-and-white focus target on the material and remove it after the successful autofocus.

For capturing the HDR reflectance, a set of LDR sequences with different ISO speeds i and flash intensities q is shot. We employ the lowest ISO speeds whenever possible, since they provide a better signal-to-noise ratio. Only if the dynamic range of the material reflectance exceeds the dynamic range of the flash intensities, we switch to higher ISO settings as well. We implement the measurement program for a single LDR step (i, q) by first setting the ISO speed i and flash intensity q for all cameras. All flashes are precharged for a fast response, but set not to discharge with the exposure. Then, we loop through each flash $f \in \{1, 2, \ldots, 151\}$. Camera f activates the flash. All cameras are triggered to simultaneously take a picture; during this, camera f will flash, since it has been activated. Finally, the flash for camera f is deactivated and the loop continues with the next camera. Note that this procedure requires only 151 flash discharges per 22,801 images.

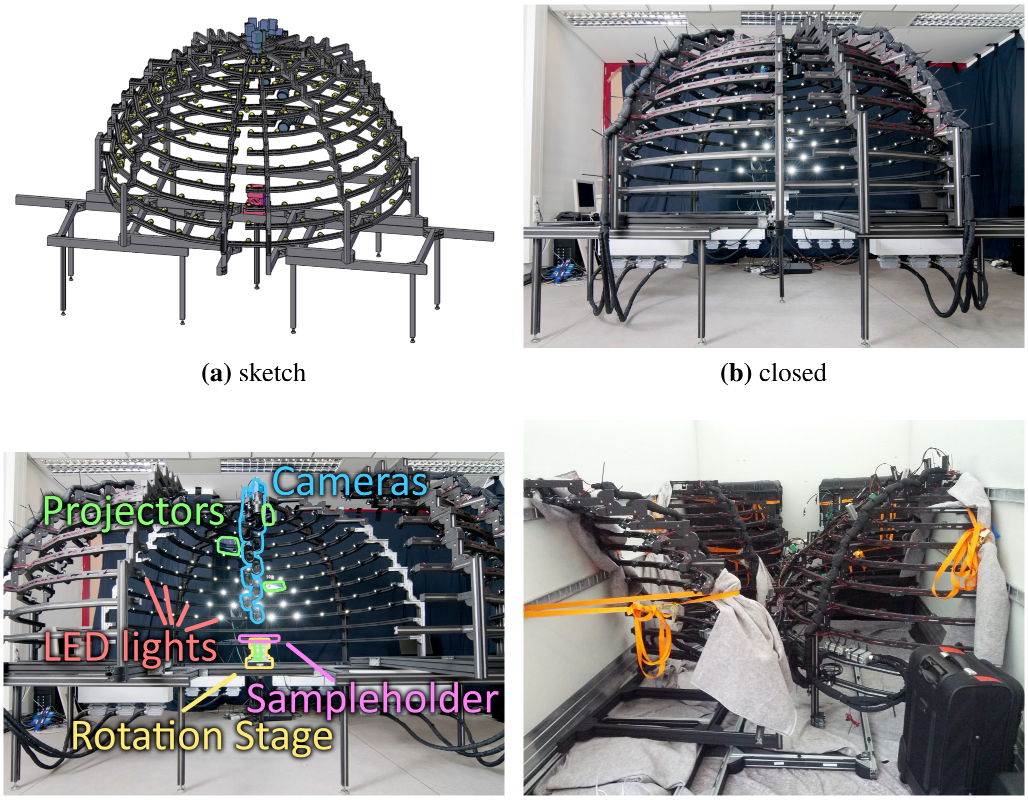
Since the cameras are controlled using different computers and via a USB connection, it is not trivial to synchronize the exposure of all other 150 cameras with the one camera that will flash. To tackle this, we use a rather long exposure time of 1 second for the PowerShot A75 and 2.5 seconds for the PowerShot G9. Camera f is triggered 0.5 seconds after the others, so the flash definitely falls within the exposure interval. The cameras directly transfer the image data to the control PCs in JPEG format. The images of the PowerShot A75 camera have an average file size of 251.2 KB and can be transmitted in about ten seconds. The higher-resolution PowerShot G9 requires an average of 3.16 MB per image and the transmission takes about nine seconds. Interestingly, in both cases, the total time amounts to about eleven seconds per light direction. Capturing one full LDR sequence takes 27 – 28 minutes. We typically utilize four combinations of ISO speed and flash quantity. Thus, a total of 91,204 raw images are captured in 1:50 hours. The raw data sizes are 21.85 GB for the PowerShot A75 and 281.15 GB for the PowerShot G9, respectively.

As described earlier, we optionally perform a structured light acquisition to obtain an accurate 3D geometry of the sample. The details of the 3D reconstruction can be found in Section 5.4.1. We capture the structured light patterns using exposure bracketing. Since the projectors are continuous light sources, varying exposure times can be applied for this. Therefore, we set the cameras to the lowest ISO speed and take multiple sequences of Gray code patterns $g \in \{1, 2, \ldots, 42\}$ from all projectors $p \in \{1, 2, \ldots, 9\}$ using different exposure times t.

Switching the projectors on and off takes the longest time and is thus performed least frequent. First, the current projector p is powered on and all cameras are set to exposure time t. Then we project each pattern g and take pictures of the pattern illuminated object with all cameras simultaneously. After all patterns have been displayed, we proceed with the next exposure time. When all exposure sequences are captured, we power off the current projector and repeat the procedure with the next one.

Here, we do not need to be too careful with the synchronization. Instead, we wait 100 ms after each pattern-change and then directly capture the image with all cameras simultaneously. We proceed with the next pattern as soon as all cameras have finished transmitting their images. The shorter exposure times make the process faster. The transmission of the images is faster as well: Due to the Gray code illumination, more than half of the image content is black, resulting in 1.7 MB for the JPEG images – nearly half of the average file size. The time for the transmission is about four seconds. In most of our structured light measurements, we employ three different-exposure times $t \in \{50\text{ ms}, 125\text{ ms}, 500\text{ ms}\}$. This results in a total of 171,234 images, which are captured in about 1:25 hours.

Note, however, that the captured data is stored fragmented over eight control PCs and needs to be copied to its permanent storage destination after the acquisition. We refrain from a transmission during measurement to avoid synchronization issues due to lags in the network communication with the master computer.



(a) sketch **(b)** closed **(c)** opened **(d)** disassembled

Figure 4.12: *The Dome 2 setup as a schematic illustration (a), photographed in the closed (b) and fully opened configuration (c) as well as disassembled and packed into a light commercial vehicle (d).*

4.7 Dome 2

The Dome 2 setup (see Figure 4.12), published in [WRO*12, SK12, SSWK13], is a camera array setup that combines a fixed light dome, comparable to the Dome 1 setup or the setups in [DWT*02, WGT*05, WMP*05, WMP*06, WLDW11, KNRS13, NJRS13], with a multicamera arc and a turntable, similar to [FKIS02, MPZ*02, MPN*02]. It was built between 2011 and 2012 with the goal to combine the strengths and overcome the shortcomings of the two previous setups. The Dome 2 was designed to facilitate integrated 3D acquisition from the start. Thus, the sample is always leveled, as in the Dome 1 setup. However, similar to the goni-oreflectometer, we now employ high-end cameras and well-behaved continuous light sources to avoid the calibration issues of the first dome.

We also included the experience gained with user requirements in our design. The Dome 2 setup is capable of reliable nonstop measurement operation. The design foresees the possibility of an automatic feed for material samples and the easy and fast deployment of the setup on-site.

4.7.1 Hardware

After more than five years in use, the weaknesses of the consumer grade point-and-shoot cameras in the Dome 1 device became very apparent. Thus, the new design consequently employs high-end industrial parts. However, this decision would make the construction of another complete camera hemisphere prohibitively costly. We therefore employ a hybrid approach that still features some parallelism in the view direction sampling: We equip a quarter circle above the material sample with eleven cameras, which observe the sample from different inclination angles θ_o in parallel. A turntable is used to achieve a sampling of different azimuthal angles ϕ_o. To keep time-consuming mechanical movement to a minimum, we employ a full hemisphere of 198 rigidly positioned light sources, avoiding any movement when sampling the light directions. As with the Dome 1, most of the angular resolution is thereby again predetermined by the hardware. However, the proposed arrangement shows yet another increase in angular resolution and the turntable provides additional flexibility for balancing azimuthal resolution and measurement speed. We eventually use 198×264 directions (see Table 4.4), instead of 151×151 or 81×81. Note however, that, due to the rigid arrangement of lights and cameras, the azimuthal sampling of the view direction and of the light direction are coupled. As a consequence, for different azimuthal angles ϕ_o and ϕ'_o the directions of the light sources can be different, i.e. in general $\mathfrak{L}_{\phi_o} \neq \mathfrak{L}_{\phi'_o}$.

4.7.1.1 Gantry

The construction consists of four basic parts: A base, standing on nine legs, on which two quarters and one half of the dome are mounted. For transportation, the frame can be quickly disassembled into these parts and packed into a light commercial vehicle. All of them fit through standard doorframes. When assembled, the quarters of the hemisphere can be slid open, giving access to the inside. Figure 4.12 shows the Dome 2 setup in all three configurations. There is also enough space to let an automatic feed pass through for continuously measuring several material samples in sequence.

Similar to the Dome 1 setup, all components are held by a hemispherical gantry. The gantry is again made from Bosch Rexroth profiles and organized in rings that

264 view directions

θ_o	$\Delta\phi_o$	$+\varphi_o$	#dir's
$0°$	$15°$	$0°$	24
$7.5°$	$15°$	$7.5°$	24
$15°$	$15°$	$0°$	24
$22.5°$	$15°$	$7.5°$	24
$30°$	$15°$	$0°$	24
$37.5°$	$15°$	$7.5°$	24
$45°$	$15°$	$0°$	24
$52.5°$	$15°$	$7.5°$	24
$60°$	$15°$	$0°$	24
$67.5°$	$15°$	$7.5°$	24
$75°$	$15°$	$0°$	24

198 light directions

θ_i	$\Delta\phi_i$	$+\varphi_i$	#dir's
$2.5°$	$180°$	$90°$	2
$7.5°$	$60°$	$37.5°$	$6+1$
$15°$	$30°$	$15°$	$12+1$
$22.5°$	$30°$	$22.5°$	$12+1$
$30°$	$30°$	$15°$	$12+1$
$37.5°$	$15°$	$0°$	$24+1$
$45°$	$15°$	$7.5°$	$24+1$
$52.5°$	$15°$	$0°$	$24+1$
$60°$	$15°$	$7.5°$	$24+1$
$67.5°$	$15°$	$0°$	$24+1$
$75°$	$15°$	$7.5°$	$24+1$

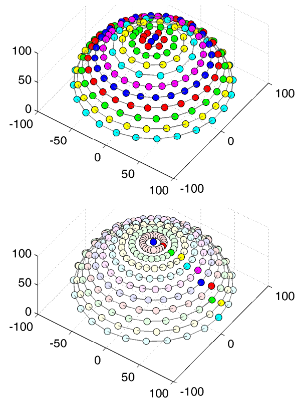


Table 4.4: *The hemispherical direction samplings in our Dome 2 setup. On every second ring, the azimuthal view angle ϕ_o is displaced by $\varphi_o = 7.5°$. On each ring, the azimuthal light angle ϕ_i is displaced by φ_i to be arranged symmetrically around the cameras. Furthermore, there is one additional lamp in every ring except the first at $\phi_i = \phi_o + 180°$, i.e. the perfect mirror direction of the respective camera.*

are held rigidly by nine vertical struts. However, the Dome 2 has a larger inner diameter of 2 m. Due to this and because the cameras are only arranged on an arc, the rings can now be spaced evenly at the inclination angles $\theta = 0°, 7.5°, \ldots, 90°$. The eleven cameras are installed on the rings from $\theta = 0°, 7.5°, \ldots, 75°$ right below each other. On every second ring the cameras are displaced by $\varphi_o = 7.5°$ to have enough space. The different azimuthal angles are reached using a turntable in $\Delta\phi_o = 15°$ steps. The resulting view directions, listed in Table 4.4, are distributed slightly denser than in the Dome 1 setup and have an average minimal distance of 7.6° ±2.6°. Note that the higher standard deviation indicates a less uniform distribution of the directions.

198 LED lamps are installed on the rings as well. 188 of them are placed in equidistant azimuthal angles $\Delta\phi_i$ to achieve an even sampling over the hemisphere with an average distance of 9° ±1.2°. They are aligned symmetrically around the camera on the respective ring by applying an azimuthal displacement of $\varphi_i = \frac{1}{2}\Delta\phi_i + \varphi_o$. This arrangement was chosen, because it facilitates the acquisition of reciprocal image pairs for turntable rotations of $n \cdot \Delta\phi_i + \varphi_i$. In turn, this allows to use the Helmholtz reciprocity principle in 3D reconstruction [WRO*12]. Note that the spacing of the light sources with $\Delta\phi_i$ as multiples of 15° causes the light direction samples to be mostly identical for the different turntable rotations with $\Delta\phi_o = 15°$. Another ten lights are placed at the perfect mirror direction of the cameras, i.e. $\phi_i = \phi_o + 180°$. See Table 4.4 for a detailed listing.

4.7.1.2 Cameras

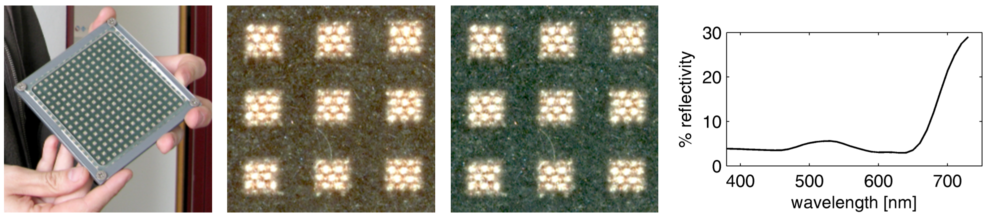
We employ SVS Vistek SVCam CF 4022COGE industrial video cameras. The CCD sensor has a resolution of four megapixel with 14 BPP. It has a quadratic shape of 16×16 mm, which reduces the amount of pixels that do not display the material sample in an image. Like the DSLR cameras of the gonioreflectometer, the Vistek cameras have a Bayer-patterned CFA to measure RGB color. The large pixels show a high light sensitivity, i.e. low noise levels, providing a high dynamic range of about 32 dB per image. We additionally use exposure bracketing to account for higher dynamic ranges. For this, the electronic shutter has customizable exposure times from 50 microseconds to infinity. The cameras are connected via gigabit Ethernet and are capable of transmitting up to eight images per second with 12 BPP. All eleven cameras are operated by a single computer, avoiding any synchronization issues or the fragmented storage of the captured data.

The cameras are equipped with high-quality ZEISS Makro Planar T*2 ZF-I prime lenses. Aperture and focus can be fixated using locating screws. Therefore, all lens-dependent intrinsic parameters are constant, vastly improving the stability of camera calibration and its validity throughout multiple measurements. For measuring flat material samples, we employ a focal length of 100 mm (35 mm equivalent focal length: 190 mm), offering approximately 380 DPI spatial resolution. For the acquisition of larger 3D objects, we exchange the lenses with a second set of 50 mm focal length (35 mm equivalent focal length: 95 mm), providing 190 DPI. In both cases, we use a fixed aperture of $f/19$ on all lenses to have a sufficiently large depth of field and focus on the center of the Dome 2 setup.

In contrast to the consumer photo cameras employed on the gonioreflectometer and Dome 1, the CCD sensor of the Vistek cameras does not have an infrared cut-off filter. However, as illustrated in Figure 4.13, blocking the near-infrared is important to preserve the natural color impression to a human observer. We therefore use additional B+W 486 UV/IR cut-off filters on our lenses. Figure 4.14 demonstrates the spectral sensitivity of the employed camera with and without the filter.

4.7.1.3 Light Sources

We decided to employ LED lamps in the Dome 2 setup for two main reasons: First, they provide continuous illumination. Thus, the problems encountered with flash light sources in the Dome 1, such as the complex radiometric calibration and the inconvenient exposure bracketing, can be avoided. Second, they are inexpensive and reliable, allowing us to use the high amount of fixed 198 light directions. The decision for LED lamps inferred two additional considerations: We wanted to use single emitter LED lamps to be able to presume an ideal point-light illumination



(a) photograph **(b)** without IR filter **(c)** with IR filter **(d)** reflectance spectrum

Figure 4.13: *Color reproduction of a green fabric material captured in the Dome 2 setup. (a) shows a picture of the material taken with a P&S camera under natural lighting. (b) and (c) are images taken with a Vistek camera without and with the IR cut-off filter, respectively. Note how in (b) the material appears to have a red tint. The measured reflection spectrum (d) of the green part shows a significant peak in the infrared.*

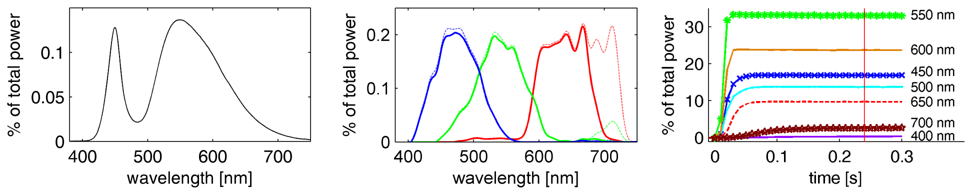
for computing the light directions. Further, the LEDs should be phosphor-coated to exhibit a continuous spectrum rather than three narrow peaks, facilitating a natural image impression. We selected Barthelme Bari DC 2.5 watt showcase LED lamps (215 lm). Their LED emitter was amongst the most powerful available at the time. In addition, the lamps come with optics to achieve a spotlight characteristic, concentrating most of the emitted radiance on the material sample. We account for the strong spatial variance of the illumination in our radiometric calibration procedure.

Although the LEDs have an uneven spectral distribution, there are no holes in the spectrum (see Figure 4.14a) and most of the power is actually concentrated in the spectral bands to which the cameras are sensitive. All LEDs are from one batch to avoid differences in brightness and spectra. Similar to the HMI lamp of the gonioreflectometer, their color temperature is 6,000 K. After switching on an LED we wait for 250 milliseconds for it to reach stable operating conditions and spectral characteristics (see Figure 4.14c).

4.7.1.4 Projectors

Similar to the Dome 1, the Dome 2 setup is equipped with four digital projectors for an integrated 3D reconstruction via structured light [WSRK11, WRO*12]. The projectors are installed next to the camera arc at different inclination angles $\theta \in \{0°, 30°, 60°, 90°\}$.

At the moment, the setup is equipped with LG HS200G LED projectors. However, due to the projectors shortcomings, discussed in Section 4.6.1.4, we propose



(a) LED **(b)** camera **(c)** LED time series

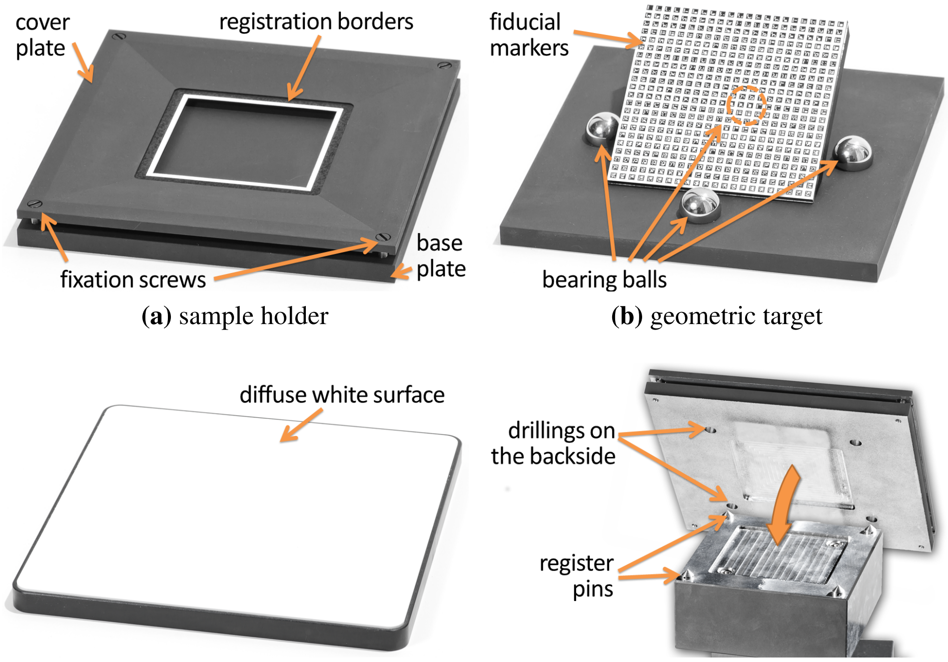
Figure 4.14: *Spectral power distribution of the employed LEDs (a) and sensitivity of the cameras (b). The red, green and blue curves correspond to the respective primaries in the Bayer pattern. Dashed curves indicate the response without the IR cut-off filters. In (c) the change in spectral power distribution of a LED after activation (at 0 s) is shown. Higher wavelengths take longer to reach their final power output. The vertical red line at 240 ms marks the time at which the 99$^{\text{th}}$ percentile of the final power is reached. After this point we consider the spectral characteristics to be stable.*

to replace them with CASIO XJ-A141 (1024×768 pixels, LED-DLP, 2,500 lm) models. In our first experiments, the Casio projectors do not exhibit a drift and also support shorter synchronization times.

4.7.1.5 Turntable & Sample Holder

To achieve the different azimuthal view angles, we utilize a Newport URS-150BCC computer-controlled precision rotation stage with a guaranteed unidirectional repeatability of 0.002°. This is in agreement with the results of our experiment, sketched in Section 4.8.1. Here, we obtained an average pose deviation of 0.0035° for repeatedly capturing the same sequence of rotations. We limit the maximum rotation speed, acceleration and deceleration to avoid shifting or deforming the sample. Rigidly attached to the turntable, we installed a vertical-stage dummy with four conic register pins on top (see Figure 4.15d). The register pins interlock with drilled holes on the backside of our sample holders and calibration targets. This way, the sample holders and targets can be exchanged and put back into exactly the same position. The fixation is rigid and has virtually no mechanical play, ensuring a high precision and repeatability.

Flat material samples are fixated on a blackened sample holder (see Figure 4.15a). If necessary, the sample can be glued to the base plate with double sided tape as in the previous setups. A cover plate is put on top of the material sample and provides an adjustable clamping pressure via four screws. All parts of the sample holder and the fixation mechanism are made from aluminium with a CNC mill. Following



(a) sample holder **(b)** geometric target

(c) radiometric target **(d)** fixation on turntable

Figure 4.15: *The Dome 2 sample holder (a) and custom-tailored geometric (b) and radiometric (c) calibration targets. All three are fixated using the mechanism shown in (d).*

our experience from the previous two setups, we again employ airbrushed matte blackboard paint as black coating. In contrast to the cover plate of the Dome 1 setup, the adjustable clamping pressure avoids squeezing of soft materials, which could otherwise change the appearance. The visible area of the material sample is 7.5 cm$\times$7.5 cm. The black-and-white registration borders framing the material are used for the automatic registration. For capturing 3D objects, we simply place them on the base plate without applying the cover plate. This is possible because, in contrast to the Dome 1, no radiometric calibration markers are required.

4.7.2 Calibration

One huge advantage of the Dome 2 setup is the fact that all components show a high repeatability. First, most of the hardware is rigidly affixed, even the focal length, aperture and focus distance of the camera lenses. Second, the only movable components, i.e. the rotation stage and detachable sample holders, show a high precision and good repeatability. Radiometrically, the situation is the same with the LED lights reaching a stable and repeatable state very quickly. This allows to perform a single accurate calibration that remains valid as long as the hardware components are not disrupted, eliminating the need for an additional refinement of the calibration per measurement. We currently do not calibrate the projectors, due to the problems with the initial shift of the projection described in Section 4.6.1.4.

4.7.2.1 Geometric Calibration

In contrast to the Dome 1, the cameras capture the sample at multiple turntable positions. Hence, a registration of the different rotated acquisitions is required. For this purpose, it is necessary to calibrate the rotation axis and center of the turntable in addition to the parameters for the cameras and light sources. Further complexity of calibration is introduced by the fact that the positions of light sources and cameras are decoupled in this setup. This prevents the solution found for the flashes of the Dome 1 of adding a known offset. Instead, the light sources are calibrated independently.

The geometric calibration of all parts is performed utilizing a custom-tailored target (see Figure 4.15b) that consists of a plate with fiducial markers [MS13] and four polished bearing balls. The calibration target is designed to fill most of the cameras' field of view. For the 50 mm lenses, we employ a target with an extent of 25 cm$\times$25 cm, whereas the target for the 100 mm lenses is 18 cm$\times$18 cm in size. Markers and balls have a known size and position. We employ bearing balls with a diameter of 50 mm and 20 mm respectively. The target is rotated by the turntable to capture it in various different poses.

We first calibrate the cameras and the turntable using the fiducial markers. The markers are uniquely identifiable and orientable and have a Hamming distance of three, avoiding accidental misclassification. By subpixel accurately detecting the corners of the markers, we obtain a large set of highly reliable homologous points (four per marker) between the different poses of the target as well as the different cameras. Using these correspondences, we first employ Zhang's algorithm [Zha00] to obtain an initial guess for performing a consecutive bundle adjustment [LA09]. The resulting reprojection errors are 0.16 pixels on average, which corresponds to a spatial error of 11 $\mu\mathrm{m}$ (for the 100 mm lenses) and an angular error of 0.001° in the view direction. The turntable's axis and center of rotation are obtained from the triangulated 3D locations of the markers' corner points. After calibration, different poses can be brought into alignment with an average deviation of 0.003°. This is at the level of the repetition accuracy of the turntable and therefore sufficiently accurate.

For calibrating the light positions, we identify for each light its reflection point in all four bearing balls. Let $\mathbf{l}_l \in \mathbb{R}^3$ denote the true position of the LED l (in our setup, $l \in \{1, 2, \ldots, 198\}$), then $\mathbf{l}'_{l,c,b} \in \mathbb{R}^2$ is the detected position of its reflection in ball b in the image taken by camera c. We describe the bearing balls via their center $\mathbf{c}_b$ and their fixed diameter. Because the marker plate can cause shadowing and occlusion for some combinations of light sources and cameras, we capture the balls under a sufficient number of rotated poses. These can be thought of as additional "virtual" bearing balls where the center $\mathbf{c}_b$ is described by the rotation $\mathbf{c}_b = \mathbf{R}(\alpha)\,\mathbf{c}'_b$ of an unrotated bearing ball b' for an angle α. Without loss of generality, we use $b \in \{1, 2, 3, 4\}$ to refer to the unrotated bearing balls. Due to the fixed arrangement of the on the calibration target and the precision of the already calibrated turntable, we can predict the position of all balls very precisely.

Using the detected highlights and the good initial estimate for the bearing balls' positions, we compute the reflection rays via ray tracing and triangulate the LED position from them. Afterwards, we perform a nonlinear optimization on all LED locations and the unrotated sphere positions simultaneously to reduce the reprojection error of the observed reflections:

$$\underset{\tilde{\mathbf{c}}_1,\ldots,\tilde{\mathbf{c}}_4,\tilde{\mathbf{l}}_1,\ldots,\tilde{\mathbf{l}}_{198}}{\arg\min} \sum_{l,c,b} \left\| \mathbf{l}'_{l,c,b} - \tilde{\mathbf{l}}'_{l,c,b} \right\|^2 . \tag{4.14}$$

Here, $\tilde{\mathbf{l}}'_{l,c,b}$ is the projection of the reflection of LED l with estimated position $\tilde{\mathbf{l}}_l$ in bearing ball b with estimated center $\tilde{\mathbf{c}}_b$ into the camera image c. Note that we only need to consider the centers of the unrotated bearing balls during optimization as those for other rotations are directly derived from them. We employ the Levenberg–Marquardt algorithm [Lev44] to find a solution. The optimization terminates after about 50 iterations, taking a total of about 20 minutes on our processing computer (see Section 4.7.3).

A similar approach was recently published by Ackermann *et al.* [AFG13]. Here, the authors used a single camera and did not include the position of the balls in their optimization. With our global optimization of all parameters, we can report an average error of 0.4 pixels, corresponding to an angular error of about 0.08° for the light directions.

4.7.2.2 Radiometric Calibration

The radiometric calibration of the Dome 2 is closely related to the radiometric calibration performed in our gonioreflectometer setup, described in Section 4.5.2. The situation for a single direction combination is very similar: The Vistek cameras provide us with a 12 BPP raw data from the CCD and the employed LED light sources are continuous light sources with a constant illumination. Please note that we are currently not considering longtime degradation effects on the light yield, which could be antagonized by periodic radiometric recalibration.

Following the procedure described of the gonioreflectometer setup, we first take *dark frames* $\mathcal{D}_{c,T}$ for every camera c to correct for hot pixels and sensor bias. Although the Vistek cameras capture RGB color, this is achieved using a color filter array in front of the sensor. Thus, the obtained raw images are monochromatic prior to demosaicking, similar to the situation of the multispectral gonioreflectometer. However, while the gonioreflectometer employs different exposure times for each wavelength band, this is not the case in the Dome 2 setup. Hence, in contrast to the procedure described in Section 4.5.2, we do not need to account for a wavelength band or color channel λ. Instead, we take a multiexposure series with different exposure times T. We use linear interpolation between the dark frames $\mathcal{D}_{c,T}$ to perform dark frame subtraction for arbitrarily exposed measurement images. We further compute the response function χ_c for every camera by employing the method of Robertson [RBS03], taking an exposure series of an X-Rite ColorChecker Passport color rendition chart (see Figure 4.11) that is placed on the sample holder's base plate. The color chart is chosen as a target because it provides favorable variations in intensity and hue. Similar to the Dome 1 calibration, we also use the color chart to establish the CIEXYZ color profile for each camera.

For a given camera c, LED lamp l and turntable rotation r, this allows us to obtain radiance values up to a unknown but constant factor $\alpha_{\mathbf{x},l,c}$. The radiance $L_o(\mathbf{x}, \omega_o)$ is computed from pixel $\mathbf{x}'$ in image $\mathcal{I}_{l,c,r}$. The mapping of $\mathbf{x}$ to $\mathbf{x}'$ is given from the geometric calibraiton of the camera (see Section 3.5.1) and the turntable. The directions ω_i and ω_o are derived from the position on the surface $\mathbf{x}$, the turntable rotation r and the position of light l and camera c, respectively. Similar to the Dome 1 setup, we use multiple exposures to increase the available dynamic range. For a given series of images $\{\mathcal{I}^T_{l,c,r}\}_T$ with exposure times $T \in \mathbb{R}^+$, we use the weighted sum (similar to [RBS03])

$$\alpha_{\mathbf{x},l,c} L_o(\mathbf{x}, \omega_o) = \frac{1}{\sum_T w\left(\mathcal{I}^T_{l,c,r}(\mathbf{x}')\right)} \sum_T w\left(\mathcal{I}^T_{l,c,r}(\mathbf{x}')\right) \frac{\chi^{-1}\left(\mathcal{I}^T_{l,c,r}(\mathbf{x}') - \mathcal{D}^T_c(\mathbf{x}')\right)}{T}. \tag{4.15}$$

Here, $\mathcal{I}^T_{l,c,r}(\mathbf{x}')$ denotes the value of the pixel $\mathbf{x}'$ in the captured image. $\mathcal{D}^T_c(\mathbf{x}')$ denotes the value of the same pixel in the linearly interpolated dark frame of camera c.

We obtain the radiance for different color channels $\lambda \in \{\text{red}, \text{green}, \text{blue}\}$ by demosaicking L_o according to the Bayer pattern of the Vistek's CFA, using the method of Lu and Tan [LT03].

Similar to the radiometric calibration of the gonioreflectometer, we capture a set of camera- and LED-dependent *white images* $\mathcal{W}_{l,c}$ of a white standard (see Figure 4.15c). The white standard is made of Labsphere Spectralon [lab], P/N SRT-99-100, which is supposed to be almost perfectly Lambertian with an albedo a of 99% in the visible spectrum. Again, this is not entirely the case. In practice, we use a fitted Cook-Torrance BRDF model to describe the reflectance behavior, similar to the procedure for the SphereOptics Zenith UltraWhite targets. However, as this is not important for the following theoretical considerations, we still employ a diffuse BRDF model in the following equations for the sake of readability.

With the white standard, the irradiance of the light source can be determined up to the factor of $\alpha_{\mathbf{x},l,c}$ by

$$\alpha_{\mathbf{x},l,c} E_{\mathbf{x},l} = \frac{\pi}{a} \frac{\chi^{-1}\left(\mathcal{W}_{l,c}\left(\mathbf{x}'\right) - \mathcal{D}_c^T\left(\mathbf{x}'\right)\right)}{T}, \tag{4.16}$$

following the same considerations that are explained in detail in Section 4.5.2. Due to the Lambertian reflectance of the white standard, using a single exposure time T is sufficient to capture its full dynamic range.

When placed on the turntable, the surface of the white standard is at the same level as a material surface would be during measurement. Given the precise repeatability of the Dome 2, the factor $\alpha_{\mathbf{x},l,c}$, which implicitly contains spatially varying illumination effects such as vignetting, chromatic aberrations or distance falloff, is therefore exactly the same in the white images and the measurement images. By using the corresponding pair of measurement image $\mathcal{I}_{l,c,r}^T$ and white image $\mathcal{W}_{l,c}$, the factors $\alpha_{\mathbf{x},l,c}$ are canceled out. All spatially varying illumination effects are therefore corrected without the need for an explicit model and the spatially varying reflectance sample ρ_m at position $\mathbf{x}$ and for directions ω_i and ω_o is given as

$$\rho_m\left(\mathbf{x}, \omega_i, \omega_o,\right) = \frac{L_o\left(\mathbf{x}, \omega_o\right)}{E_{\mathbf{x},l}} = \frac{\alpha_{\mathbf{x},l,c} L_o\left(\mathbf{x}, \omega_o\right)}{\alpha_{\mathbf{x},l,c} E_{\mathbf{x},l}}. \tag{4.17}$$

Please refer to Section 4.5.2 for a more detailed derivation.

We observed that the variation in illumination over the surface is low frequent. To save memory, we therefore store the white images $\mathcal{W}$ rectified to the quadratic 3D surface of the white standard in a low resolution.

Note that the implicit correction is not possible for points that significantly protrude from the surface of the white target. Plans are in place to capture the white target at different heights and use a trilinear interpolation for the radiometric correction on 3D objects. However, currently we instead assume a conic shape of the light distribution and hence extrapolate the value of $\mathcal{W}\left(\mathbf{x}\right)$ along the ray to the light source position to obtain a volumetric correction factor. Here, we also account for the quadratic falloff with respect to distance to the light source.

4.7.3 Measurement Process

The measurement is controlled by a single computer that is equipped with two Intel Xeon E5620 CPUs with 2.4 GHz, 24 GB RAM, an NVIDIA GeForce GTX 460 GPU and 14 gigabit Ethernet ports. Similar to the Dome 1, the computer is capable of performing rectification, HDR combination and radiometric correction on-the-fly during the measurement. Nonetheless, all raw images are written to disk as well.

The measurement data is captured directly onto a freshly formatted two or three terabyte hard disk to avoid loosing write speed due to file system fragmentation. Still, depending on the exposure time the data rate can reach 528 MB/s and the disk's write speed becomes a limiting factor for the measurement performance. Therefore, we employ a write queue in RAM, which is worked off during more time-consuming operations. After the measurement, the hard disk, which is mounted in a hot-swap drive bay, can be swiftly exchanged to enable further measurements without delay. This improves upon the Dome 1, where the measurement data is fragmented over eight PCs and has to be copied to a permanent storage destination over the network.

For flat material samples, pictures of the registration borders on the cover plate of the sample holder are taken under all 24 rotations. Then, the quadrilateral for rectification is detected with subpixel precision. This enables rectification and automatic determination of exposure times during the measurement. This procedure is not possible for 3D objects. Here, the necessary exposure times are selected manually before starting the measurement.

The measurement process is a combination of view-parallel and serialized acquisition. Different inclination angles of the view direction are acquired in parallel. For covering different azimuthal view angles, the sample is rotated into the correct pose by the turntable. We execute our measurement procedure with the goal to minimize the time spent waiting for slow operations to finish. Recording eleven images in parallel takes as long as the maximal exposure time T over all cameras plus a constant amount of $375\ \mathrm{ms}$ for clearing the sensors and transmitting the data. Switching on a light source requires an additional $250\ \mathrm{ms}$ delay for the LED to reach stable characteristics. Rotating the turntable by $15°$ creates an average delay of $9\ \mathrm{s}$. Switching on a projector takes the longest. With the LG projectors, we wait for 15 minutes until the projection stops shifting. In case of the Casio projectors, we wait an average of $60\ \mathrm{s}$ until the projector shows the pattern.

For planar samples, we therefore first rotate the turntable. Then, we consecutively cycle through the light sources, always illuminating the sample with exactly one LED, and take an HDR exposure series for each one with all cameras simultaneously. We either use predetermined exposure times or employ an automatic exposure compensation. For the auto exposure, we first determine the *region of interest* (ROI) by detecting the registration borders on the sample holder prior to starting the measurement. During measurement, the control computer keeps track of oversaturated and underexposed pixels in the ROI. On that basis, it automatically decides whether additional exposure steps have to be captured. This is done separately for each camera, as the amount of reflected light, and hence the necessary exposure compensation, depends on the camera's viewing angle.

For reconstructing 3D geometry, we also perform a structured light measurement. Here, we first switch on the projector and then rotate through the desired poses. For every rotation, we capture a pattern sequence with all cameras. We use a sparser set of eight azimuthal angles in 45° steps for the geometry acquisition.

Unfortunately, the brightness of the 2.5 W LEDs as well as the brightness of the LED projectors cannot compete with the 575 W lamp of the gonioreflectometer or the flashes of the Dome 1. Hence, the exact measurement time strongly depends on the dynamic range and albedo of the captured material, requiring exposure times of up to five seconds or more in disadvantageous cases. To reduce the necessary exposure times, we may perform an electric preamplification of the CCD signal, gaining a factor of two. The effective dynamic range of a single image is then reduced to 25 dB, which is still acceptable. In our experiments, acquiring a BTF with this configuration took between three and ten hours. Additionally capturing the 3D geometry took another one to three hours (see Table 5.1). Typically, we take three differently exposed images to cover a total dynamic range of 60 dB with sufficient overlap. Therefore, the total number of images per measurement amounts to $11 \times 3 \times 198 \times 24 = 156{,}816$ for the reflectance and $11 \times 3 \times 42 \times 8 \times 4 = 44{,}352$ for the geometry. We found that, given the hardware of the acquisition PC, it is fastest to store the images as Bitpacked Raw (see Section 3.7) instead of applying an additional elaborate image compression. With this lightweight lossless compressed data format, the images require about 6 MB. Therefore, in this scenario the total size of the raw measurement data amounts to 918.8 GB for reflectance and 259.9 GB for geometry.

4.8 Comparison of Designs

In the previous sections, we presented three different setup implementations for the acquisition of digital material appearance. In the course of this, we already highlighted some important differences and similarities as well as advantages and disadvantages. In this section, we will now juxtapose the qualities of our setups together with devices found in the literature. We primarily focus on the different design requirements established in Section 4.2.

4.8.1 Quantitative Comparison Experiments

All three of our setups have generated a fair share of valuable measurements, be it for commercial, scientific or conservation purposes. First material measurements, captured with the gonioreflectometer setup, have been published as the UBO2003[1] and ATRIUM[2] data sets. They have become quite popular for benchmarking and comparing reflectance related tasks, such as compression, fitting or editing. Recently, a new systematic database, containing seven material classes with twelve specimens in each, has been captured with the Dome 1 and released as UBO2014[3]. Furthermore, with the SPECTRAL[4] data sets, four multispectral BTF measurements have been made publicly available to the research community. Finally, we published several of the digitized 3D objects listed in Table 5.1 as the OBJECTS2011[5] and OBJECTS2012[6] data sets.

Our setups have also been used regularly for commercial purposes. Several companies – mostly in the automotive industry – use measured data from our setups for visual prototyping, visualization and marketing. Finally, the Dome 2 setup of Bonn has been placed at the disposal of the Cultural Informatics research group at the University of Brighton in the scope of a one month exhibition in Brighton, UK [3D-12]. Here, the local organizers operated the device on their own and scanned several dozen artifacts that were contributed by interested visitors as well as local cultural heritage institutions.

In the context of this work, we additionally conduct a series of experiments in order to allow for a quantitative comparison of our setups. We evaluate the following attributes: the achievable dynamic range, the repeatability of a measurement and finally the overall accuracy of the measured reflectance.

First, the dynamic range is assessed in two ways. Since we employ HDR imaging in two of our setups, we report the dynamic range achievable with a single LDR image for each of our measurement instruments and the typical dynamic range that we obtain using HDR photography. In both cases, we consider the strongest radiance L_h that can be observed by the camera without the pixels becoming overexposed and the weakest radiance L_l that can still be distinguished from random sensor noise. The difference $L_h - L_l$ is the maximum detectable interval. Both values are determined manually from a large series of differently exposed images. Furthermore, we regard the radiance L_n that corresponds to the strength

[1] http://cg.cs.uni-bonn.de/en/projects/btfdbb/download/ubo2003/
[2] http://cg.cs.uni-bonn.de/en/projects/btfdbb/download/atrium/
[3] http://cg.cs.uni-bonn.de/en/projects/btfdbb/download/ubo2014/
[4] http://cg.cs.uni-bonn.de/en/projects/btfdbb/download/spectral/
[5] http://cg.cs.uni-bonn.de/en/projects/btfdbb/download/objects2011/
[6] http://cg.cs.uni-bonn.de/en/projects/btfdbb/download/objects2012/

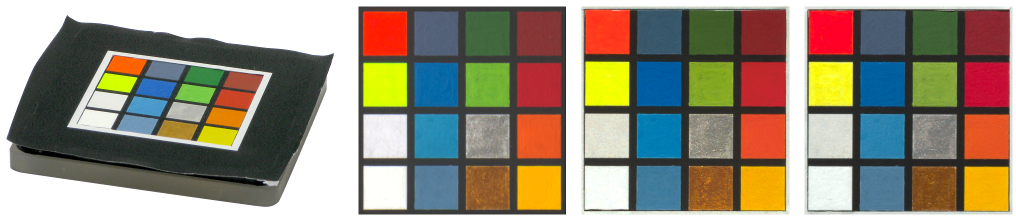
of noise found in a completely black image as the lower bound for the minimal resolvable value (actually any change in value must slightly exceed L_n in order to be distinguishable from random noise). The dynamic range is then expressed in decibels as $10\log_{10}((L_h - L_l)/L_n)$.

For HDR images that are obtained via exposure bracketing, we pick L_h from the image with the lowest exposure and L_l and L_n from the image with the highest exposure. The results are reported in Table 4.5. The diverse cameras employed in the gonioreflectometer all have a comparably high dynamic range in single LDR images. Due to the already long measurement times and rapid wear on the involved components, e.g. mirrors of the DSLRs and lifetime of HMI light bulbs, no additional exposure bracketing is employed. This restricts the dynamic range of a gonioreflectometer measurement to the dynamic range of the employed camera's sensor. The cameras of the Dome 1 show a lower dynamic range for single images. Via exposure bracketing this range is increased and surpasses the gonioreflectometer, but is ultimately limited by the available ISO speeds and flash intensity quantities. For single images, the Dome 2 has a dynamic range comparable to the Gonioreflectometer. However, here we employ HDR imaging with varying exposure times, measuring almost arbitrarily high dynamic ranges.

We determine the repeatability of the measurement using three criteria. First, we assess the repeatability of the imaging process. This is evaluated by repeatedly switching the cameras off and on, each time taking a picture, and comparing the subpixel precise positions of the automatically detected corners of the border markers. We report the standard deviation of the detected corners in pixels.

Second, we consider the angular repeatability. We bring all movable components of the setups into a series of configurations that were chosen to reflect the movements during measurement. Then, the respective hardware parts are homed again and the sequence is repeated. Note that due to its completely rigid nature, in the Dome 1 setup, only the cameras were turned off and on again. We report the standard deviation of the taken up poses in degrees. In the dome setups, we capture the sample holder and find its pose by detecting and triangulating its corners, relying on the geometric camera calibration. For the gonioreflectometer, we used Zhang's algorithm [Zha00] to obtain the relative poses of a checkerboard that was mounted on the robot arm.

Finally, we also look at the radiometric repeatability by taking a series of pictures of a white standard under the same illumination condition. During that time, no mechanical movement is carried out. However, in analogy with the reported measurement procedure, the lights of the Dome 2 setup are switched off and on again. We apply the corresponding radiometric correction procedure for each setup and report the remaining variance of the recorded values in percent.



(a) overview picture **(b)** gonioreflectometer **(c)** Dome1 **(d)** Dome2

Figure 4.16: *A handmade material chart (a), used for comparing the accuracy of our three setups. Fields column-wise: fluorescent red, fluorescent yellow, "paper", white, blue 3, blue 2, blue 1, turquoise, green 2, green1, silver, gold, red 2, red 1, orange, yellow. (b), (c) and (d) show rectified and radiometrically corrected measurement images ($\theta_i = 45°, \phi_i = 0°, \theta_o = 0°, \phi_o = 0°$) from each of our three setups. For illustration purposes, the result images are tone-mapped and converted to sRGB colorspace.*

All test-sequences were executed 15 to 25 times with all available cameras and for radiometric repeatability of the dome devices with a selection of six different light sources. The resulting figures are reported in Table 4.5.

As expected, the fixed optics of the gonioreflectometer and Dome 2 achieve a good imaging repeatability, whereas the motor-driven lenses of the Dome 1 perform about four times worse. Still, the angular repeatability of the Dome 1 by far surpasses the one of the gonioreflectometer due to the otherwise completely rigid nature. Despite the reintroduction of a moving part, i.e. the turntable, the high-quality components of the Dome 2 outperform the prior setups in both respects.

Concerning the radiometric repeatability, it is apparent that the variation in the flash discharges of the Dome 1 lead to a high deviation, despite the elaborate radiometric correction. The gonioreflectometer performs better, as it uses a a steady light source. Unfortunately, the employed gas discharge lamp is not completely flicker-free. The LED illumination of the Dome 2 provides the best radiometric repeatability.

To determine the overall accuracy, we capture a handmade material chart (first shown in [RSK10]) with all three systems. Figure 4.16 shows a picture of the prepared chart as well as rectified images. To be more robust to small alignment errors, we compute a single average reflectance distribution per material field. We compare the reflectance functions for each field on a set of discrete samples $(\theta_i, \theta_o, \phi_i, \phi_o) \in \{-75°, -74°, \ldots, 75°\} \times \{45°\} \times \{0°, 180°\}^2$. Values for direction combinations that have not been measured are computed via linear interpolation.

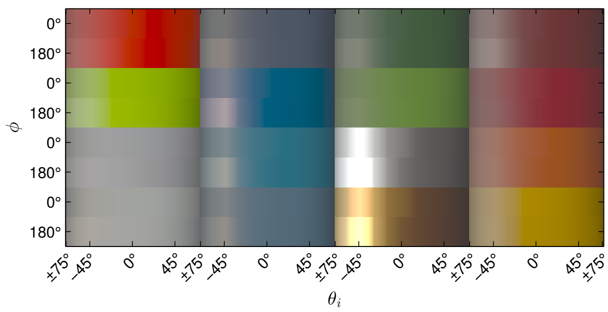
Note that the obtained reflectance values from the different devices are in no way normalized for this comparison. The only additional operation that is carried out is the conversion of the multispectral measurements of the gonioreflectometer to sRGB, using the CIE 1964 standard colorimetric observer spectrum. The accuracy of each of the setups is assessed by comparing the reflectance functions of the each field for fixed $\phi_i = \phi_o = 0°$ with those that have $\phi_i = \phi_o = 180°$. We make the assumption that each material field shows a homogeneous opaque material. Furthermore, the materials do not necessarily need to be isotropic but should be invariant for 180° rotations. Then, both functions should be identical and deviations between the functions indicate inaccuracies.

The results of the experiment are presented in figures 4.17, 4.18, 4.19 and 4.20. The gonioreflectometer only meets this condition on rather diffuse materials and reveals inaccuracies in the presence of highlights (see the large deviations for the silver and gold field in Figure 4.18). The Dome 1 performs better but still shows easily perceivable deviations for many materials. In the measurement of the Dome 2, almost all values are similar (indicated by the two rows in Figure 4.17c and the curves in Figure 4.20 being in good alignment). The only exception is the slight misalignment in the presence of the strong specular peak on the gold material.

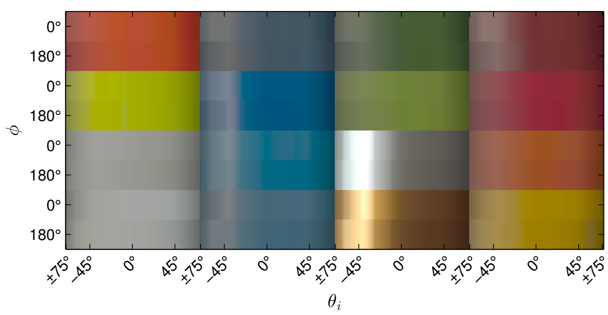
In Table 4.5, we provide a comprehensive comparison between our three devices, considering the results of our experiments and the other design attributes.

4.8.2 Comparison with Related Work

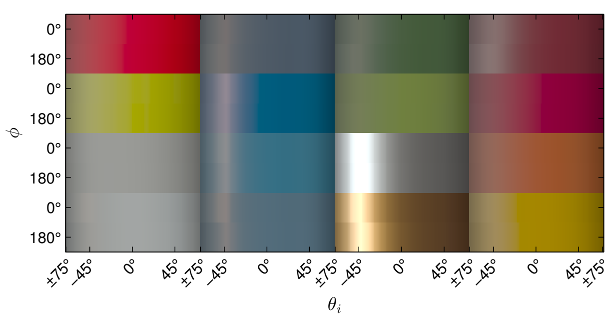
The values in Table 4.5 nicely underpin the practical value and applicability of our setups with respect to the initially established criteria. We continue our evaluation by comparing our solutions with the related work found in the literature. Our analysis follows the categorization into the three basic setup classes (gonioreflectometer, mirror and kaleidoscope setups, and camera and light array setups). Note that many publications do not report figures for all design attributes that we described in Section 4.2. This hampers a purely quantitative comparison. We will thus discuss the differences between the reported methods and qualitatively compare the advantages and disadvantages. Still, an overview of the most important common characteristics can be found in Table 4.6, Figure 4.21 and Figure 4.22 at the end of this section.



(a) gonioreflectometer



(b) Dome 1



(c) Dome 2

Figure 4.17: *Reflectance measurements of the handmade material chart (shown in Figure 4.16). All images and plots depict measurements taken at the fixed camera elevation of $\theta_o = 45°$. The reflectance is visualized as tone-mapped sRGB color with a gamma of 3. Measurement values are taken for a semicircle of illumination directions with $\theta_i \in \{-75°, -74°, \ldots, 75°\}$ under two different azimuthal view and light direction angles $\phi_i = \phi_o \in \{0°, 180°\}$. Each field of the material marker is divided in two rows, corresponding to $\phi = 0°$ and $\phi = 180°$. The illumination direction θ_i is varied along the X-axis. Almost all materials, especially gold and silver, exhibit a specular highlight in the perfect mirror direction at $\theta_i = -45°$. The Dome 2 (c) shows the best alignment and color stability between the two series. Please refer to figures 4.18, 4.19 and 4.20 for polar plots of the depicted values.*

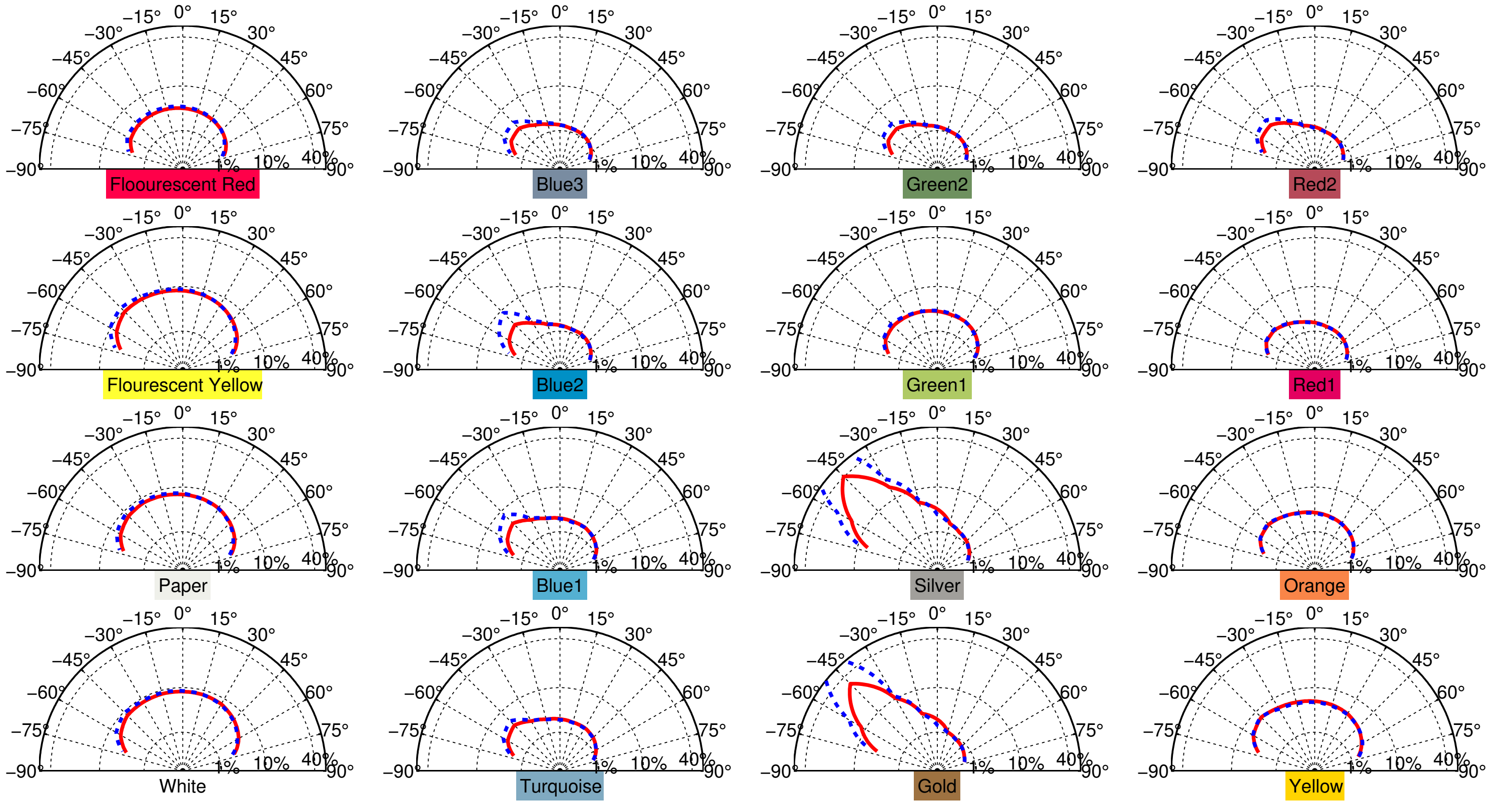


Figure 4.18: *Polar plots of the reflectance distributions of the handmade material chart (see Figure 4.16) for the* **gonioreflectometer**. *The plots show the average reflectivity over all color channels in percent for the respective elevation angles of the light directions. Samples for azimuthal angle $\phi = 0°$ are drawn red, samples for $\phi = 180°$ are blue and dotted. Please refer to the caption of Figure 4.17 for details about the depicted angles. For the purpose of visualization, the axes are scaled with $\sqrt[3]{}$. Note the large difference in the two recorded reflectivity series for the silver, gold and blue fields.*

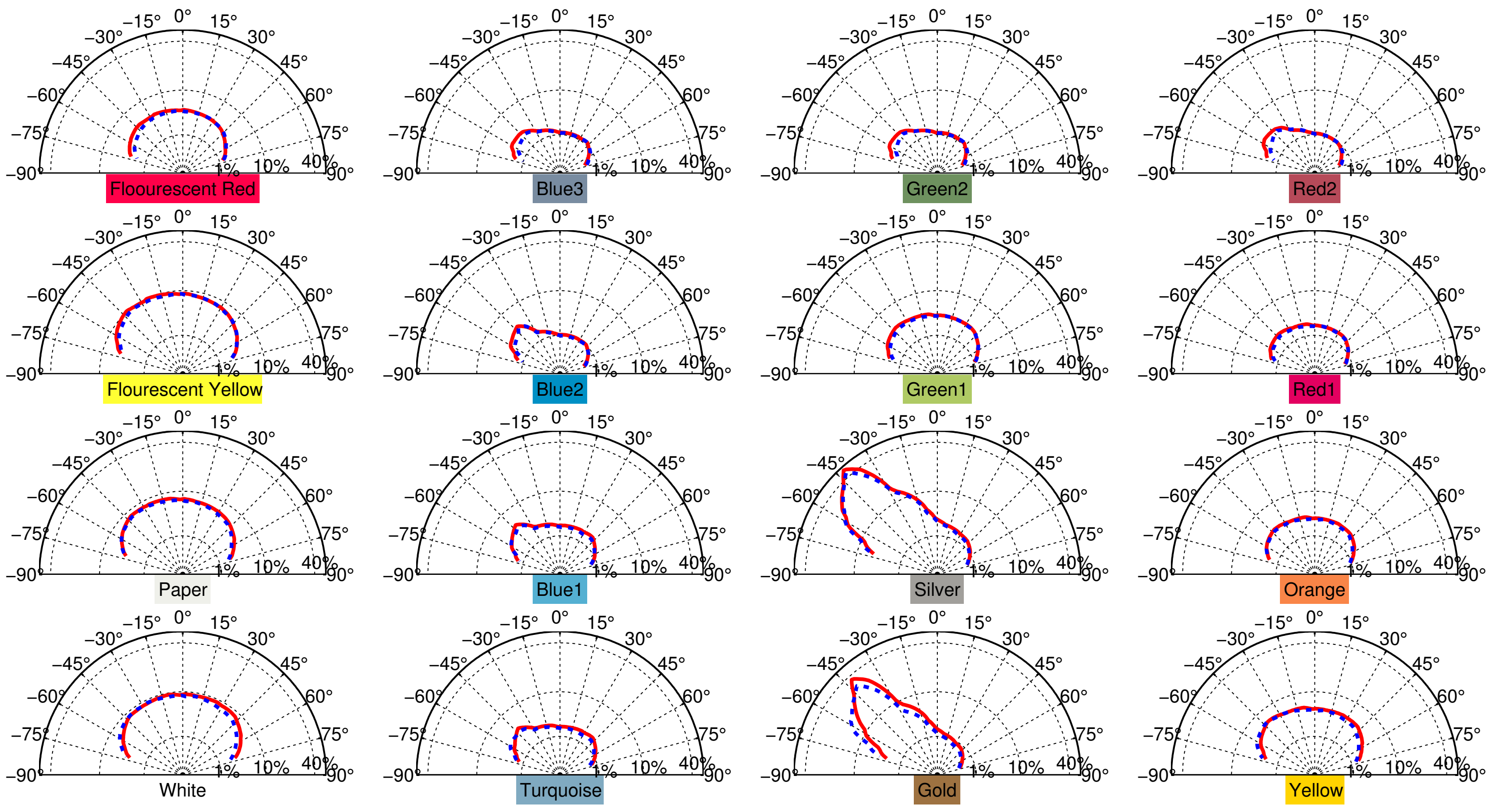


Figure 4.19: *The reflectance distributions of the handmade material chart for the* **Dome 1**. *Please refer to the caption of Figure 4.18 for a detailed explanation of the plots' axes. The general shapes of the recorded reflectivity match those of the gonioreflectometer (Figure 4.18) and Dome 2 (Figure 4.20). However, the measured values – especially those of the silver and gold fields – are more noisy.*

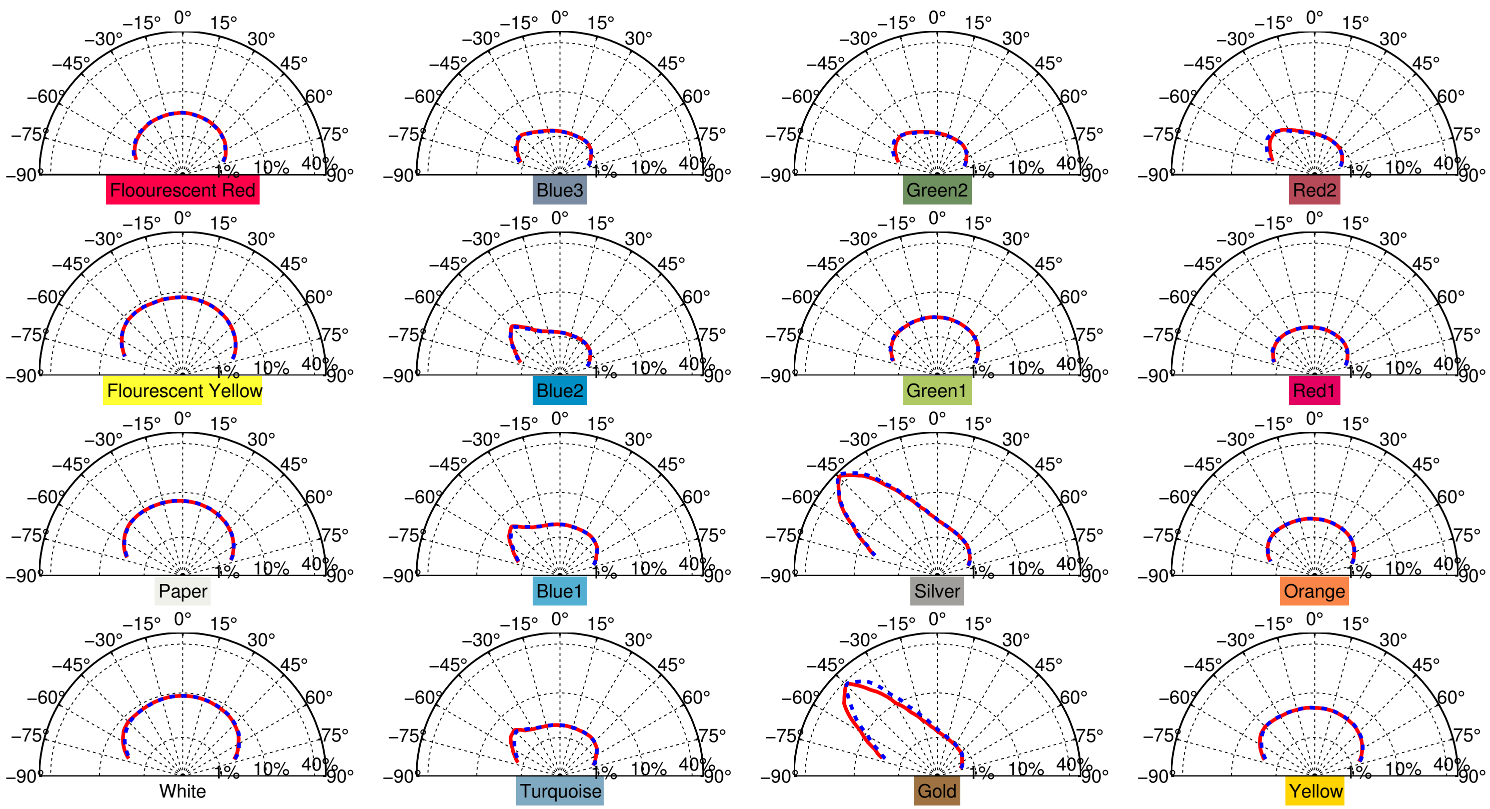


Figure 4.20: *The reflectance distributions of the handmade material chart for the* **Dome 2***. Please refer to the caption of Figure 4.18 for a detailed explanation of the plots' axes. In contrast to the measurements of the gonioreflectometer (Figure 4.18) and Dome 1 (Figure 4.19), the recorded reflectivity of all fields is basically free of noise and the two series are always in good alignment.*

	Gonioreflectometer			**Dome 1**			**Dome 2**
configuration	2002 [SSK03]	2004 [MMS*04]	2010 [RSK10]	2004 [MMS*04]	2008	2011 [SWRK11]	2012 [SK12]
dimensions (L×W×H) [cm]	410×170×90			190×190×190			340×250×250
distance to sample [cm]	170 (camera) / 240 (lightsource)			65			100
directions $\omega_i \times \omega_o$	81×81			151×151			**198**×**264**
resolution ω_i	14.7° ±0.4° / 16° ±0.8°			9.4° ±1°			**9**° ±**1.2**°
resolution ω_o	14.7° ±0.4° / 16° ±0.8°			9.4° ±1°			**7.6**° ±**2.6**°
maximum θ	Sampling 1: 75° / **Sampling 2: 85°**			75°			75°
35 mm equivalent focal length [mm]	240	180	270	116	104	52 – 104	95 / 190
spatial resolution [DPI]	280	330	290	235	**450**	225 – **450**	190 / 380
dynamic range[1] [dB]	**35** / – / ∞	31 / – / ∞	32 / – / ∞	28 / 33 / 33	25 / 44 / 44		32 / **60** / ∞
spectral bands	RGB		**32**	RGB			RGB
camera type	DSLR		Industrial	P&S			Industrial
recorded image data	**12 BPP raw**			8 BPP JPEG			**12 BPP raw**
light source type	gas discharge lamp			flash			LED
measurement volume [cm]	8×8		6.5×6.5	10.5×10.5		10.5^3 – 20.5×20.5×20.5	7.5×7.5 / **24**×**24**×**24**
direction variation across sample	1.9° / 1.4°		1.5° / 1.1°	6.5°		7.8° – 18°	3° / 12.6°
BTF measurement raw images #	6,561		209,952	91,204			156,816
BTF measurement time [hours]	14		60	**1:50**			3 – 20
BTF measurement size [GB]	83		1,228	22	281		918
3D acquisition raw images #						171,234	44,352
3D acquisition time [hours]						**1:25**	1 – 4
3D acquisition size [GB]						282	260
radiometric repeatability[2]	–‡‡		1.1‰	–‡‡	7.4‰		**0.1**‰
geometric repeatability[3]	–‡‡		0.17 pixel / 0.61°	–‡‡	0.81 pixel / 0.006°		**0.12 pixel / 0.002°**
sampling flexibility	**full: arbitrary ω_i and ω_o**			none			some: arbitrary ϕ_o
radiometric calib. procedure	easy but inaccurate			complex			**easy**
geometric calib. procedure	manual			automatic but unstable			**automatic**
durability (# measurements)	>12†	≈27‡	>1000*	–§	≈265 / >347¶		**>3650**††

†Camera use discontinued at about 83,000 exposures. Probably limited by wear on mirror. ‡Defect of mirror at about 180,000 exposures. *Assuming one measurement every three days and continuous camera operation for ten years. Note that the HMI bulb has a lifetime of 1,000 hours and therefore has to be replaced about every 16 measurements. §Not determined due to systematic defect of the CCD chips in the whole camera series [Can05]. ¶Two camera CCDs became defective and were replaced after about 160,000 exposures. The other 149 are counting 210,000 exposures and are probably limited by the wear of the flashes. ††Assuming one measurement per day. The camera manufacturer asserts continuous operation for at least ten years. Tests with the LEDs indicate a lifetime of at least 4,000 measurements. ‡‡The repeatability tests could no longer be performed for outdated configurations. [1]Single exposure / performed HDR measurements / theoretical maximum. [2]Given as variance in measured reflectivity for SphereOptics Zenith UltraWhite. [3]Standard deviation in imaging condition in pixels / Standard deviation in angular configuration in degrees.

Table 4.5: *Comparison of our setups with respect to the design requirements. Best values in a category are printed in bold.*

4.8.2.1 Gonioreflectometer Setups

Gonioreflectometers are the most common design used for BTF or related reflectance measurements (SVBRDFs, BRDFs). Hence, there is a large body of work available on these setups, varying in different aspects of the implementation.

As in our gonioreflectometer, several authors propose to have either the light source or the detector at a fixed position and achieve the necessary angular configurations by changing the orientation of the material sample. Most closely related to our approach are the works of Dana *et al.* [DVGNK97] and Kimachi *et al.* [KTT06], as in both cases the light source position is kept fixed while the material is turned into different poses by a robot arm.

In [DVGNK97], the view direction is changed by manually repositioning the camera. Due to the manual operation, only a limited set of seven view directions with different inclination is captured and isotropy of the sample is assumed. Anisotropic materials are considered by measuring them once again with a second azimuthal orientation. In total, either 205 or 410 LDR pictures of a material are taken with a video camera in about one or two hours respectively. The authors measured and published a total of 61 BTF materials as part of the `CUReT`[7] database, which was an enormous endeavor and provided valuable and popular data sets.

In [KTT06], the camera is repositioned automatically by a second robot arm. Furthermore, the light source of the setup is equipped with a spectral filter wheel, allowing a multispectral acquisition with eight spectral bands. In their experiments, the authors use a sparse sampling with fixed azimuthal angles of $\phi_i = 180°$ and $\phi_o = 0°$ and varying inclination angles, amounting to a total of $\theta_i \times \theta_o = 8 \times 71 = 568$ combinations with a higher resolution close to the perfect reflection direction (see Figure 4.22e). Unfortunately, Kimachi *et al.* do not report too many details about the acquisition process or the data and never seemed to have captured a full BTF, making it hard to judge the other qualities of the setup. Therefore, we do not consider it in our comparison in Table 4.6.

A variation of this design is used in the works of McAllister [McA02], Koudelka *et al.* [KMBK03] and Tsuchida *et al.* [TAN*05, TSA*05]. Here, the camera is placed at a fixed position and the light source is moving. McAllister [McA02] employs two rotary stages to turn the material sample into different poses and uses a movable arm holding the light source. He reports the capture of 311 to 7,650 angular samples (see Figure 4.22b) within 45 minutes to 36 hours, being slightly slower than our setup. Since the installed camera did not give access to the raw data, he performs HDR imaging via exposure bracketing with up to three different shutter times.

[7] `http://www.cs.columbia.edu/CAVE/software/curet/`

In [TAN*05, TSA*05], a setup with the same principle is extended for multispectral measurement with 16 spectral bands by using a band-pass filter wheel in front of the light source. That setup has been designed for utilization as a desktop device. It has a small extent of 80 cm×80 cm×80 cm and is lightweight. As a consequence, the maximum material sample size is restricted to only 4 cm×4 cm. The authors report the measurement of 6,500 angular samples in 13 hours, which is a good result. However, they only acquired BRDFs and not BTFs.

In [KMBK03], Koudelka *et al.* mount the light source on a robot arm so it can reach all positions on the hemisphere. The sample is presented to the camera in different angular configurations by a pan-tilt head. They use a video camera and capture a sampling of $120 \times 90 = 10{,}800$ light and view combinations within ten hours (reported in [HF11]). In their experiments, they acquired a total of nine material samples. However, the employed video camera captures LDR images with 640×480 pixels, yielding a low spatial resolution and dynamic range of the measured data.

Finally, Holroyd *et al.* [HLZ10] and Filip *et al.* [FVH*13] presented setups that put both, light source and camera, on robot arms. This eliminates the necessity to tilt the sample holder and hence allows to acquire the same range of delicate materials and 3D objects as leveled mirror setups and camera array setups do.

Being the most recent device, the setup proposed by Filip *et al.* [FVH*13] in 2013 has a very impressive maximal spatial resolution of 1,071 DPI. This resolution, however, is only achieved for rather small material samples of 4.4 cm×4.4 cm. The setup uses the same angular sampling of 81×81 uniformly distributed directions as our gonioreflectometer and captures RGB HDR data with a full-frame industrial camera. An additional turntable is employed to rotate the sample, as the camera arm only allows movement along one axis. The time needed for acquisition is with 18 hours and thus slightly higher than with our setup, possibly because the employed LED light source required longer exposure times. The authors published data sets of six measured materials with the high spatial resolution.

The setup of Holroyd *et al.* [HLZ10] has one design detail that is different to all other presented gonioreflectometers: It uses a light source and camera with a beam splitter on each robot arm to allow a coaxial arrangement of light and view direction. This gives view to the important retro-reflection configuration and allows to directly create reciprocal image pairs. As another advantage, they use self-made fringe projectors as light sources, making it possible to perform structured light reconstruction and utilize high-frequency patterns to separate direct from indirect lighting [NKGR06]. Since their setup is primarily intended for 3D object acquisition, they capture a structured light sequence for each view and light combination, rendering the acquisition process extremely slow. The authors report that they capture 6×7 different poses, taking seven minutes each. This amounts to about five hours for as few as 84 angular combinations.

In general, gonioreflectometers offer a great flexibility, as the employed robot arms, tilt heads or rotation stages can be brought into almost arbitrary angular configurations. Furthermore, the application of only a single light source and a single sensor allows the usage of high-quality components with favorable radiometric attributes, good optics and high resolutions at reasonable costs. Multispectral measurement can be integrated without much effort, using a computer-controlled band-pass filter in front of the light source or sensor [TAN*05, TSA*05, KTT06, RSK10].

However, the frequent utilization of moving parts easily introduces inaccuracies. Therefore, a thorough registration of each individual image and calibration of the light and view directions are mandatory. For this, all setups employ additional registration markers next to the sample. Furthermore, while the spatial domain is captured in parallel, the angular configurations have to be measured sequentially, requiring at least one mechanical movement for each. For that reason, the sampling resolution in the angular domain is often considerably lower than in the other two device classes. Measurement times for a sufficient number of view and light directions vary from ten hours [KMBK03] to 60 hours [RSK10]. To cope with the high number of images shot by a single camera, many setups employ video cameras or industrial machine vision cameras instead of still cameras. However, their sensors often show significantly lower resolutions.

Depending on the design of the gonioreflectometer, measurement of 3D objects as well as easily deformable or hard to fixate material samples, e.g. sand, granules, grass, foliage or fur, may be impossible. Devices that require the sample itself to be rotated into a slope orientation [DVGNK97, McA02, SSK03, KMBK03, TAN*05, TSA*05, KTT06] cannot be used for these kinds of specimens. However, setups that either employ only a horizontal turntable for the sample [FVH*13] or do not move the sample at all [HLZ10] could be used for this task (and have so in the latter case).

4.8.2.2 Mirror and Kaleidoscope Setups

Shortly after the introduction of gonioreflectometers for BTF measurements, setups based on mirrors – either curved or arranged as a kaleidoscope – have been proposed to overcome some of the fundamental shortcomings.

In [Dan01, DW04, WD06], Dana and Wang propose to use a parabolic mirror to capture multiple view directions at once, similar to earlier BRDF measurement setups. To obtain the reflected radiance for different illumination directions, they use a translation stage to move an aperture in the beam of a directional light source.

This way, only a small spot on the mirror is lit by the light, which is thereby focused on the material as a cone of illumination with a small solid angle. Furthermore, using a second translation stage, they also move the material to capture the spatial variation of the reflectance at different points of the surface. Translation stages offer rather reliable spatial positioning and registration. Thus, Dana and Wang do not employ additional registration markers.

Capturing the mirror with a VGA video camera yields the simultaneous acquisition of about 185,500 unique view directions per image. Furthermore, the small iris diaphragm in the aperture covering the light source allows for a generation of about 1,008 unique illuminations. Unfortunately, the employed parabolic mirror design only allows to capture directions between $\theta \leq 23°$ and $\theta \leq 37°$ elevation (depending on the azimuth angle, see Figure 4.22g). Hence, reflectance under grazing angles cannot be measured by the device. Furthermore, since the parabolic mirror has a specific point of focus, the device can only capture samples from flat surfaces or geometries with very shallow depths. Handling larger 3D shapes is not possible with this apparatus at all.

Despite that, this particular device design suffers from the severe drawback that light directions and spatial dimension again have to be sampled sequentially. In [WD06], the authors report a sampling of 200×200 surface points, which is a rather low number compared to other setups. Although the XY-stage can be moved extremely fast, the acquisition speed is limited by the *frames per second* (FPS) of the camera and takes about one hour per light direction. The authors do not report their exact measurement procedure. However, when conducting a measurement at full extent and highest resolution in all dimensions, the measurement time would amount to 840 days. It is safe to assume that in practice, similar to other sequentially operating devices, a compromise between resolution and acceptable measurement time was found.

Alternatively, a piecewise planar mirror geometry can be employed, with each facet showing the complete material sample from a constant direction. In this case, the spatial domain is captured in parallel as well. Thus, a complete outgoing light field is comprised in a single image, allowing a considerable speed-up of the measurement compared to gonioreflectometer devices ([HF11] reports one hour for the setup of Han and Perlin [HP03]). This concept has straightforwardly been employed by Levoy *et al.* [LCV*04], Garg *et al.* [GTLL06] and Mukaigawa and Tagawa *et al.* [MTK*10, TMY12]. In all three cases, planar facets have been arranged in a parabolic or ellipsoidal layout. A digital projector is employed together with a beam splitter to obtain a coaxial arrangement of projector and camera similar to Holroyd *et al.* [HLZ10] in Section 4.8.2.1. Different illumination directions are imposed on the sample by activating only those projector pixels that fall onto one particular mirror-facet. This has the advantage, that all parts of the setup remain fixed, eliminating time-consuming and possibly imprecise mechanical movements. However, similar to the parabolic design of Dana *et al.* [Dan01, DW04], they only portrait a subset of all directions on the hemisphere, with [MTK*10] having the largest coverage (see Figure 4.22h). We therefore only consider [MTK*10] in Table 4.6. Direct reflections from 50 planar mirrors are employed, leading to 50×50 bidirectional samples.

Han and Perlin [HP03] and Ihrke *et al.* [IRM*12] instead propose kaleidoscope-based setups. Here, rather than directly applying many mirror facets on an elliptical shape, a clever arrangement of three planar mirrors provides a set of recursive interreflections that provide a multitude of virtual viewpoints at once to the camera. Again, digital projectors are used for creating different light directions. There is no mechanical movement in these setups. In [HP03], two different angles of taper have been explored, forming 22×22 and 79×79 direction combinations, respectively. Figure 4.22f demonstrates the obtained direction sampling with 22 directions. Ihrke *et al.* [IRM*12] use their setup to capture 3D objects instead of flat material samples. They employ an additional mirror below the object that gives view to the lower hemisphere. This way, they obtain 246 virtual views and 144 virtual light sources, yielding 35,424 direction pairs distributed over the full sphere. Note that for flat samples only those pairs that lie in the upper hemisphere can be considered, reducing the number of useful direction combinations to 8,856.

Similar to gonioreflectometers, the fact that only one camera and one light source (a projector) is necessary facilitates the usage of high-quality components. Multi-spectral measurements should be directly possible as well, but this has not yet been topic of active research. However, a single camera also means that four measurement dimensions (ω_o and $\mathbf{x}$) are embedded into the same two-dimensional space on the sensor. This implies a trade-off between spatial resolution and the number of possible direction combinations. Bangay and Radloff provide a detailed analysis of this issue for kaleidoscopic configurations [BR04]. Furthermore, similar to some camera arrays, such as our Dome 1 setup, the resolution of the angular dimensions is ultimately fixed during construction time. The spatial resolution therefore directly depends on the resolution of the employed sensor. Han and Perlin [HP03] use 3.1 megapixels and achieve about 65 pixels for a sample size of 19 mm (for the 79×79 sampling). Mukaigawa and Tagawa *et al.* [MTK*10, TMY12] report a similar number of 60 pixels for a sample size of 6 mm using a 5 megapixel sensor. The resolution of the employed projector is a minor detail, since in principle – for BTF measurements – even a single pixel per mirror-facet would suffice.

The usage of a camera and a projector suggests a possible application of structured light for 3D reconstruction. While the coaxial arrangement of light source and camera does not provide the necessary stereo basis for triangulation, the multiple virtual viewpoints formed by the interreflections could be used instead. Still, a multiview triangulation is complicated by the fact that a 3D object will occlude parts of the mirrors and hence overlay the image in the virtual view points. It therefore depends on the unknown shape of the object which parts of an image depict the object from the correct perspective. So far, only Ihrke *et al.* [IRM*12] tackled this problem, eventually reconstructing geometry and surface reflectance using a kaleidoscope.

The piecewise planar mirror-based setups [HP03, LCV*04, GTLL06, MTK*10, IRM*12] do not employ any moving parts, making it possible to establish a registration of the data a priori and thus avoid auxiliary registration markers. Yet, a precise calibration poses a harder problem than for gonioreflectometers or camera arrays, as at least one if not several levels of indirection due to interreflections have to be considered. In [New71], Isaac Newton observed *"[...], that every irregularity in a reflecting superficies make the rays stray 5 or 6 times more out of their due course, than the like irregularities in a refracting one"* (pp. 3079–3080), arguing that mirror-based optics have to be manufactured with much higher precision than lens-based ones to achieve similar accuracy.

This is also one of the reasons why we opted against implementing a piecewise planar mirror-based setup on our own and rather explored two camera array devices. The major reason, however, can be found in the fact that all presented mirror-based setups exhibit either an extremely restricted measurement volume, a very low spatial resolution or both (see Figure 4.21b). Still, the number of sampled directions is often similar or worse than what our gonioreflectometer was able to obtain in an acceptable amount of time (compare figures 4.22f and 4.22h to Figure 4.22c). Of course, employing higher resolution sensors would directly increase the spatial resolution while measurement times would stay constant. Similarly, the measurement volume can be increased using larger mirrors. But then again, both modifications would further raise concerns about accuracy. The apparatus proposed by Dana and Wang [Dan01, DW04, WD06] does in theory not show these tight restrictions. However, their approach only captures a small portion of the hemispherical directions and comes at the cost of impractically long measurement times.

4.8.2.3 Camera and Light Array Setups

Consequently, setups based on camera arrays follow a different avenue to cope with the acquisition time drawback of the gonioreflectometer. Using multiple cameras, parts of or even the full captured outgoing light field are captured in parallel without sacrificing sensor resolution or accuracy. In contrast to the more restricted mirror setups, lenses with different focal lengths make camera arrays flexible with respect to resolution and measurement volume.

Debevec *et al.* [DHT*00] proposed a setup, introduced as "Light Stage", that utilized two cameras with fixed positions together with a light source mounted on a two-axes rotation system to capture reflectance fields of human faces. The setup was later extended to utilize a rotating arc with an array of 27 lights [HCD01] and eventually a fixed dome with 156 light sources [DWT*02, WGT*05]. The authors aim for real-time acquisition of reflectance data. For this, they utilize costly high-speed cameras, capturing all possible samples within few seconds (60 s [DHT*00], 15 s [HCD01], 83 ms [WGT*05]). The high price of these cameras leads to an insufficient sampling of the outgoing light field with only one or two view directions. We therefore disregard the Light Stage setups in our further comparison.

Weyrich *et al.* [WMP*05, WMP*06] follow up on the last approach by Debevec and Wenger [DWT*02, WGT*05] and present a system with 150 light sources evenly distributed on a geodesic dome. Here, however, 16 cameras are employed to simultaneously capture the reflectance samples, leading to a slightly better coverage of the view direction domain (see Figure 4.22l). Nonetheless, this approach still shows an insufficiently low view direction sampling for purely data-driven bidirectional material representations.

Recently, Hu *et al.* [HQS10, HWQ13] proposed a design very similar to Weyrich's [WMP*05]. They employ a geodesic dome with twelve cameras and 238 light sources. However, their cameras are all mounted on a single vertical arc. Therefore, all cameras lie on a single azimuthal angle, eventually sampling a 5D slice of the BTF. Yet, as the cameras are arranged at different inclination angles, this still adequately captures isotropic reflection. Wu *et al.* [WLDW11] also present a system that captures only a 5D slice of the BTF. Their design has a horizontal ring of 20 cameras and a dome of 290 LED light sources. In contrast to the setup of Hu *et al.*, this 5D slice does not correspond to a meaningful subset of the reflectance function, as the cameras are all arranged on a single inclination angle.

In [FKIS02], Furukawa *et al.* employ five cameras and six light sources, equidistantly installed on two separate vertical arcs. The sample is placed on a rotation stage at the center and the arc containing the light sources can be horizontally rotated as well. This way, the full sphere is covered for both directional domains with 72×60 samples. For flat materials, this number would be reduced, as view and light directions have to be located on the upper hemisphere in this case, yielding 36×36 samples (see Figure 4.22i). Similar to our Dome 1 design, Furukawa *et al.* employ point-and-shoot cameras.

Tong *et al.* [TWL*05] use the very same principle with eight industry cameras and eight light sources. They capture BTF data with a comparable angular sampling as our gonioreflectometer in two hours.

Matusik *et al.* [MPZ*02, MPN*02] employ a similar setup design to capture the appearance of complete objects. They use four light sources and six cameras and additionally capture matting images and make use of two computer screens, placed below and behind the object, to sample light that is transmitted through the surface more densely. In our comparison, we only consider the set of view and direct light directions that would be used for BTF capture. Here, they acquire 60×216 (=12,960) angular samples. Although the total amount of combinations lies between our gonioreflectometer and the Dome 1, the set of directions is not as well-balanced, putting a considerably higher emphasis on the view domain. Due to the large amount of mechanical movement (at least 540 operations), Matusik *et al.* require about 14 hours to capture all samples, despite the use of industrial video cameras. Similar to our Dome 2 setup, Matusik *et al.* employ two sets of prime lenses to account for objects of different size. Unfortunately, none of the three papers reports on measurement volume and achievable resolution.

Recently, Köhler *et al.* [KNRS13] and Nöll *et al.* [NJRS13] presented a setup called "OrCam". Their device combines an array of seven cameras that can be rotated to acquire different inclination angles with a spherical gantry, mounting a total of 633 light sources at fixed equidistant positions. Similar to [FKIS02, MPZ*02, MPN*02, TWL*05], the sample is placed on a turntable to capture different azimuthal angles. They report to capture 133 different view directions. The LED light sources are combined to a total of 19 illumination patterns per view, resulting in 2,527 combinations.

In contrast to the Dome 1 and Dome 2, their setup is explicitly designed to capture large 3D objects. It has a larger diameter and uses wide-angle lenses. This way, the authors can cover a measurement volume with a diameter of 80 cm with a moderate spatial resolution of about 127 DPI. With a measurement time of up to 69 minutes, the OrCam takes about three times longer per direction than the Dome 2.

Similar in spirit, Neubeck *et al.* [NZG05, NZG06] present the "KULETH Dome". It utilizes a rotation-stage together with a single camera on a robot tilt arm to capture different viewing directions. They also employ a dome of 169 affixed light sources, in this case a quarter sphere, to sample the different illumination directions. However, using only one camera this setup does not offer the advantage of simultaneous acquisition but only benefits from the reduced mechanical effort in sampling the different light directions.

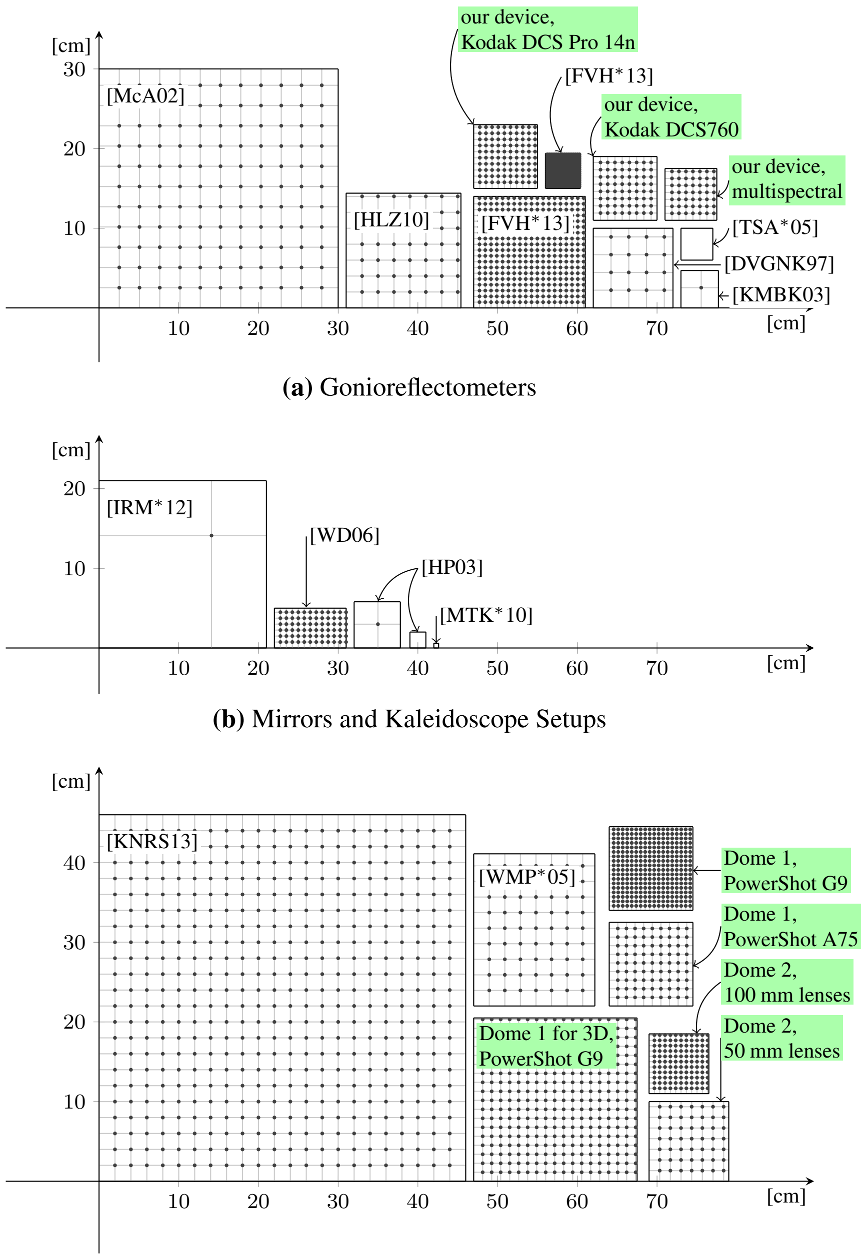
Setups that make use of camera and light arrays have the potential to gain a considerable speed-up compared to sequential gonioreflectometers without the necessity to trade-off spatial with angular resolution as mirror-based setups do. Three devices attempt to capture all view directions on the full hemisphere simultaneously and abandon any moving parts: our Dome 1 setup [MMS*04] as well as the setups of Weyrich *et al.* [WMP*05, WMP*06] and Hu *et al.* [HQS10, HWQ13]. Not surprisingly, those devices are also the fastest camera array setups and among the fastest BTF capturing setups in general. Weyrich *et al.* capture 2,400 images with 1.3 megapixels in only 25 seconds.

However, Weyrich and Hu only have a sparse set of directions covering the view hemisphere (16 and 12 cameras, respectively), whereas our Dome 1 setup provides as much as 151 directions. Unfortunately, the necessary amount of cameras to densely cover the full view direction hemisphere leads to increased costs as well as control and synchronization issues. Therefore, many camera array devices instead follow a hybrid approach, combining a smaller arrangement of cameras with a turntable [FKIS02, MPZ*02, MPN*02, TWL*05, SSWK13] and sometimes also an additionally movable tilt arm [NZG05, KNRS13, NJRS13]. Yet, in these cases the need for additional sequential capture in the view domain leads to an increase in measurement time.

Due to the reduction of mechanical sample movement to at most a rotation of a turntable as well as the usually large dimension to host all hardware parts, camera and light array setups lend themselves for the acquisition of reflectance of 3D objects. Almost all presented setups with the exception of [NZG05, TWL*05, NZG06, HQS10, HWQ13] have therefore reported their successful application for this task.

In [FKIS02, MBK05, MPZ*02, MPN*02] the silhouette of the object for each view direction is extracted in the images. The visual hull is constructed via volume carving [MA83] or using an image-based visual hull technique [MBR*00]. This approach has the advantage that the reconstructed geometry is correctly aligned with the captured images for different view directions. Unfortunately, the visual hull cannot reconstruct concavities correctly. Furthermore, inaccuracies in the silhouette extraction can lead to rather crude approximations of the actual shape. To improve upon these drawbacks, in [FKIS02] an additional laser scanner is employed. Yet, in turn, this requires to register the 3D geometry obtained by the laser range scanner with the reflectance measurement. Similar, in [WMP*05, WMP*06] an auxiliary structured light-based 3D scanner is employed. Here, feature points are matched between the scanner-generated texture data and the images of the reflectance measurement. The geometry captured from the separate scanner is registered to the cameras using these correspondences.

More recent devices [SWRK11, SK12, SSWK13, KNRS13, NJRS13] instead employ an integrated 3D reconstruction based on structured light, using the same views that are used for reflectance acquisition to capture the 3D surface via triangulation. This way, the geometry is already registered with the reflectance measurements, at the expense of requiring accessory fringe projectors in the setup.



(a) Gonioreflectometers

(b) Mirrors and Kaleidoscope Setups

(c) Camera Array Setups

Figure 4.21: *Sampling of the spatial domain for selected setups. Our setups are shaded in green. The size of the rectangles shows the maximum spatial extent of the sampling area. The raster inside the rectangles corresponds to the sampling density with a factor of 1:100 on both axes. For [FVH*13] and [HP03]. Please refer to Table 4.6 for the exact numbers. Note how in (b) all piecewise planar mirror setups [HP03, MTK*10, IRM*12] exhibit a low spatial sampling density and often cover only a small area.*

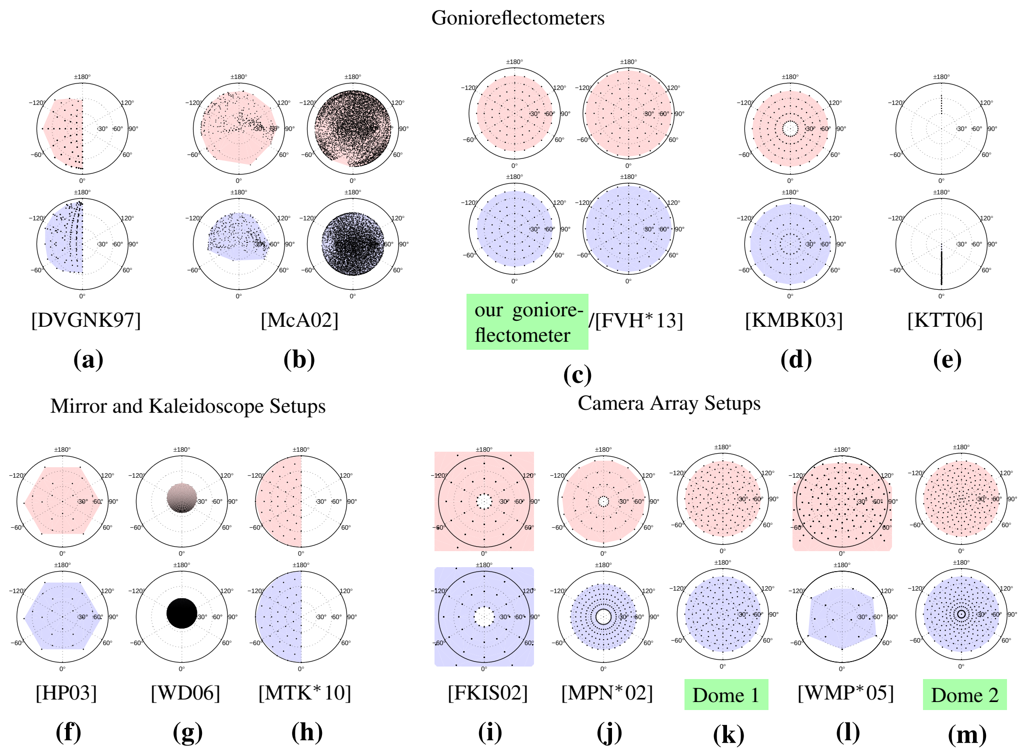


Figure 4.22: *Sampling of the angular domain for selected setups (top disc: light hemisphere; bottom disc: view hemisphere). Our setups are shaded in green. The angle corresponds to ϕ, the radius to θ. The reported measurement directions $\omega = (\phi, \theta)$ are plotted as black dots, shaded regions emphasize the achieved directional coverage for the sake of easy visual comparison. For (b), directions from the two measurements with fewest and most reported samples are shown. In (c), we illustrate both direction samplings from Table 4.1. Note that Filip et al. [FVH*13] adopted the sampling from our setup. The color-shading in (i) and (l) exceeds the plot, because the setups also capture samples at the lower hemisphere. Almost all setups capture the Cartesian product of the indicated view and light directions, i.e. all possible pairs. In contrast, in (a) and (b), each direction participates in exactly one sample. Many setups do only cover parts of the directions on the hemispheres (a), (b), (e), (g), (h)) have holes (d), (i), (j) or show an extremely sparse sampling (f), (l). Our setups (c), (k), (m) all have a wide direction coverage with densely and equally distributed samples.*

publication		year of publication	sample size [cm^2]	spatial resolution [DPI]	# direction samples	measurement time [hours]	speed[1] [MSamples/s]	# cameras	camera type	# light sources	light source type[2]	HDR capture	# spectral bands	3D objects
Gonioreflectometers														
[DVGNK97]	Dana *et al.*	1997	10×10	114	205 † 410	1 2	1.4	1	Video	1	GDL	✗	3	✗
[McA02]	McAllister	2002	30×30	100	311 7,650	0:45 36	20 10	1	DSLR	1	GDL	✓	3	✗
[SSK03]	Sattler *et al.*	2002 2004	8×8	280 330	6,561	14	4.9 6.8	1	DSLR	1	GDL	✓	3	✗
[KMBK03]	Koudelka *et al.*	2003	4.7×4.7‡	100	10,800	10	1.3	1	Video	1	LED	✗	3	✗
[TSA*05]	Tsuchida *et al.*	2005	4×4	?	6,500	13	?	1	Industrial	1	GDL	✓	16	✗
[RSK10]	Rump *et al.*	2010	6.5×6.5	290	6,561	60	22.3	1	Industrial	1	GDL	✓	**32**	✗
[HLZ10]	Holroyd *et al.*	2010	14.4×14.4*	127	84	5	0.3¶¶	2	Industrial	2	Projector	✓	3	✓
[FVH*13]	Filip *et al.*	2013	4.4×4.4 14×14	**1,071** 350	6,561	18	43.5 47.1	1	Industrial	1	LED	✓	3	✗
Mirror and Kaleidoscope Setups														
[HP03]	Han and Perlin	2003	5.8×5.8§ 2×2§	85§	484 6,241	1	0.6 1.0	1	P&S	1	Projector	✗	3	✗
[WD06]	Dana and Wang	2006	9×5	339	$\mathbf{1.9{\cdot}10^{8}}$ ¶	20,160††	262	1	Industrial	1	Tungsten	✗	3	✗
[MTK*10]	Mukaigawa *et al.*	2010	0.6×0.6	250	2,500	?	?	1	Industrial	1	Projector	?	3	✗
[IRM*12]	Ihrke *et al.*	2012	21×21 *	18	35,424	93:30	0.3†††	1	DSLR	1	Projector	✓	3	✓
Camera Array Setups														
[FKIS02]	Furukawa *et al.*	2002	?	?	4,320	?	?	5	P&S	6	Tungsten	✗	3	✓
[MPN*02]	Matusik *et al.*	2002	?	?	12,960	14	?	6	Industrial	4	GDL	✓	3	✓
[MMS*04]	Müller *et al.*	2004 2008	10.5×10.5	235 450	22,801	1:50	414 1,520	151	P&S	151	Flash	✓	3	✗
[WMP*05]	Weyrich *et al.*	2005	≈15.2×19.1**	130§§	2,400	**0:00:25**	**9,044**	16	Industrial	150	LED	✓	3	✓
[NZG05]	Neubeck *et al.*	2005	?	230‡‡	44,616	?	?	1	?	169	?	✗	3	✓
[TWL*05]	Tong *et al.*	2005	?	?	7,056	2	?	8	Industrial	8	Tungsten	✓	3	✗
[HQS10]	Hu *et al.*	2010	?	?	2,856 †	0:30	?	12	?	238	LED	✓	3	✗
[SWRK11]	Schwartz *et al.*	2011	20.5×20.5 * 10.5×10.5 *	225 450	22,801	3	1,617‡‡‡ 1,696‡‡‡	151	P&S	151	Flash	✓	3	✓
[SSWK13]	Schwartz *et al.*	2013	10×10 7.5×7.5	90 380	**52,272**	3 – 20	132‡‡‡ 297‡‡‡	11	Industrial	198	LED	✓	3	✓
[KNRS13]	Köhler *et al.*	2013	**46×46** *	127	2,527	0:39 – 1:09	493 [12,13]	7	DSLR	633	LED	?	3	✓

† Only isotropic reflectance sampling. ‡ Estimated from the Lego brick sample, depicting 6×6 nobs. * Projection of reported measurement volume on the ground plane. § Estimated using the size of a penny coin depicted in a camera image in the article. ¶ Estimated from given mirror diameter 25.4 mm/484 pixel and aperture diameter 0.8 mm. Directions are limited to $\theta \leq 23°$ to $\theta \leq 37°$ (depending on ϕ, see Figure 4.22g). Light samples are cones with 2.5°–6.6° diameter. †† Theoretical value for a complete measurement (probably never attempted). See discussion in Section 4.8.2.2. ‡‡ Estimated from the texture resolution reported in the paper and the size of a depicted M&M candy. ** Median head breadth and menton-crinion length of a male Caucasian. §§ Estimated using camera images of male Caucasian faces depicted in the article. ¶¶ Based on timings including the 3D acquisition, as this cannot be separated in this approach. ††† Based on timings including the 3D acquisition, because no separate timings are available. ‡‡‡ Average speed over all available measurements. [1] The speed is given for monochromatic bidirectional reflectance samples. Possible multiple exposures for HDR combination are not considered. For oblique views the full resolution is assumed as well. [2] GDL stands for gas discharge lamp.

Table 4.6: *Comparison with other setups. Our setups are shaded in green. The numbers of other setups are compiled from publicly available sources: the cited publications, associated technical reports, state of the art reports, courses as well as websites of accompanying databases or laboratories. Best values in a category are printed bold.*

4.9 Summary

In this chapter, we have identified a list of basic attributes that should be fulfilled by BTF capturing devices. Subsequently, we surveyed the literature for existing approaches that meet the established requirements and found that these setups can be categorized in three primary device classes: gonioreflectometers, mirror-based setups and camera array setups. Each of the classes has its distinct advantages and drawbacks. We illustrated this by discussing one gonioreflectometer and two different camera array designs in great detail. Furthermore, we compared them with each other with respect to all of the identified attributes. Finally, we also took the approaches from the surveyed literature into account and pointed out similarities as well as unique solutions found in the variety of proposed setups.

Most parts of the content and insights presented in this chapter have already been made available in two publications. We have published the details about the newly developed Dome 2 setup in a workshop paper as “DOME II: A Parallelized BTF Acquisition System” [SSWK13]. Furthermore, we published the full technical documentation and comparison with other designs drawn in this chapter in the open access journal “Sensors” under the title “Design and Implementation of Practical Bidirectional Texture Function Measurement Devices focusing on the Developments at the University of Bonn” [SSW*14].

In the end, there is no single device that outperforms the others on all disciplines. There is not even a clear tendency towards one of the main device classes. Instead, different approaches focus on different aspects of the BTF acquisition.

We believe that our most recent Dome 2 setup provides state-of-the-art performance and a well balanced compromise between many of the practical aspects. However, which device class or particular setup design is best suited depends on the application at hand. The presented comparison of the basic attributes in Table 4.6 can be an aid for decision-making. Still, it is hard to grasp the practical applicability of many of the setups, as very little is reported on the topics of reliability, durability, etc.. In this case, the in-depth discussion of our three implemented devices can serve as an indicator what problems can be expected, which device class handles them best and and how much effort is necessary to tackle them.

Considering the scope of this thesis, i.e. the 3D digitization, it seems to be clear that camera array designs are the most suitable. Although it has been implemented in all three device classes, the other two show drastically lower speed of acquisition for 3D objects. This would prohibit the desired dense sampling for an image-based material appearance. In contrast, with the presented Dome 1 setup acquisition times of less than two hours can be achieved. This has proven to be fast enough for the time-critical application scenario of digitizing fresh food in Section 2.3.

4.9.1 Lessons Learned

Many considerations for an acquisition setup depend on its intended application. For pursuing truly general appearance capture on a larger scale than a handful of samples with a laboratory prototype, we would recommend the following:

- Use steady light sources. While strobe light sources might provide a good photon yield and avoid unnecessary exposure, they enormously complicate the radiometric calibration, which eventually leads to increased effort for every single measurement and probably reduced repeatability and accuracy.
- Use sufficiently strong light sources. Comparably weak LED light sources are the major reason why our Dome 2 setup is far beyond its capture frame rate potential. In the case of the spectral gonioreflectometer setup, even 575 watt are not enough. Due to the narrow spectral band filtering, long exposure times of several seconds per image can be necessary.
- Avoid mechanical movement whenever possible. This improves measurement speed as well as reliability and accuracy. The fastest capture setups in the literature follow exactly this strategy. Mechanical movement is also one of the reasons why our Dome 2 setup lacks the speed of its predecessor.
- Do not use point-and-shoot cameras or similar consumer grade devices, such as smart-phones. Those devices usually require a lot of compromises and perform unavoidable unwanted operations. Long transmission times, missing raw capture support, "image improvements", bad repeatability and the necessity for using the autofocus are just a few of the drawbacks we encountered in our Dome 1 setup. Furthermore, although the nominal spatial resolution of the Dome 1 setup is higher, images taken with the Dome 2 are still sharper, because of the better optics and the access to raw images without JPEG compression artifacts.
- The camera array plus light dome design is probably the way to go. It has recently been adopted by other groups as well (e.g. [WMP*05, NZG05, HQS10, KNRS13]) and view-parallel acquisition seems to be the most promising approach to keep measurement times in balance for capturing a high number of direction samples.
- For larger camera arrays, plan a distributed acquisition setup with a client-server architecture and sufficiently many camera control computers. To avoid the bottleneck of transmission with USB 2.0, we equipped our control computers with additional USB 2.0 PCI cards. However, we found that the PCs could not handle more than 20 simultaneously connected cameras without occasional hiccups. Although we did not yet reach a similar limit with the 13 gigabit Ethernet connections in the Dome 2 setup, the throughput

of the employed bus system (Dome 1: PCI; Dome 2: PCIe) will at some point become a bottleneck as well.

- Consider the trade-off between bandwidth (and storage requirements) and CPU load. In the Dome 1 setup, the cameras' internal processor applies a JPEG compression, allowing the transmission and storage of the images within a few seconds and without significant load on the control computers. However, the Dome 2 cameras deliver a raw data stream. Here, we found ourselves in a dilemma: On the one hand, a too elaborate compressed image format (e.g. OpenEXR) would reduce the throughput due to limited CPU capabilities. On the other hand, directly storing the raw data lets the hard disk's write speed become a considerable bottleneck. We eventually employ a lightweight self-written lossless image compression (see Section 3.7) but are still occasionally limited by the disk speed.
- Even special purpose white standards, such as SphereOptics Zenith Ultra-White or Labsphere Spectralon are not completely Lambertian. Both show specular reflectance behavior for grazing angles. To account for this, we approximate their reflectance with a Cook-Torrance BRDF model. The parameters of the model are obtained from fitting to measurement. The measurements, however, are performed with our devices which we intend to calibrate using the model. We resolve this circular dependency with a bootstrapping strategy that performs a few iterations of fitting and calibration until the parameters stabilize.

4.9.2 Limitations & Future Work

There are still several limitations and possibilities for future work that can be found throughout the entire spectrum of existing methods.

The time requirement for a single measurement is still a limiting factor for the widespread application of BTFs. Our Dome 2 setup reveals that one main concern are prolonged exposure times due to the low amount of reflected radiance for many directions. A simple solution to this problem are more powerful light sources. For instance, the setup of Weyrich *et al.* [WMP*05] uses 103 LED emitters per lamp to provide sufficient illumination for capturing the reflectance at 12 FPS. However, this compromises the assumption that the light at each point is coming from a single direction. Furthermore, it can severely impact costs in case a light dome is employed. Another possibility to increase the brightness is an illumination from multiple light sources at the same time. Using an appropriate set of light source combinations, the appearance under a single light source can later be reconstructed by solving a linear equation. The "Light Stage 5" setup in [WGT*05] already implements this idea. However, this approach also amplifies measurement noise and is more susceptible to outside sources of error, such as stray light.

Almost all of the discussed setups are bulky laboratory devices. It will require further research until fast and comprehensive appearance measurement becomes applicable directly on the desks of designers or in easily deployable tools for digitization in cultural heritage or other industries. Our Dome 2 setup and the OrCam [KNRS13] begin to tackle the issue of on-site usability by being demountable into separate parts that can be transported. However, there are also other approaches towards compact and transportable setups, such as a the desktop setup of Tsuchida *et al.* [TAN*05], a single-view light dome that fits into a briefcase [WVM*05] or an SVBRDF acquisition toolset that even fits in a pocket [RWS*11].

Another limitation, common to almost all discussed setups, is the sampling resolution in the angular domains. Whereas the prevalent high spatial resolution with millions of sampled points is sufficiently dense to provide a continuous impression of the material's surface, the highest complete angular resolution that we found is 198×264 directions in our Dome 2 setup, i.e. a resolution of about $8°$ and $9°$, respectively. However, a study about data-driven BRDF models [MPBM03] shows that preserving the highlight of specular materials requires resolutions considerably below $1°$. Unfortunately, their approach to utilize a denser sampling close to the highlight is not applicable for measurement from inhomogeneous surfaces as the highlight direction might be different in every single point. Ruiters and Klein [RK13] argue that a shift in paradigm away from capturing discrete samples towards measuring weighted integrals might help to solve this problem. Thus, the utilization of spatially extended pattern illumination (see [TFG*13, AWL13] for two recent SVBRDF approaches) for BTF measurement would be an interesting avenue of future research.

Ultimately, the discussed measurement setups can only cope with mostly opaque materials. For strongly globally subsurface scattering, translucent or completely transparent materials or objects, a new class of BSSRDF measurement devices would be required. Some of the presented setups [TWL*05, WMP*05, WMP*06] tackle this problem using an auxiliary measurement for subsurface scattering approximation. Existing experimental setups for full BSSRDF measurement [LCV*04, GTLL06, CNR08, MTK*10, TMY12] only capture fractions of the angular domains. Moreover, none of the setups considers the wavelength- and time-dependent redistribution of energy, found in the full twelve dimensional scattering function S (Equation 3.10). Here, bispectral measurement setups and the possibility to capture repeatable series of measurements with a controlled time-shift to the illumination are necessary. It thus remains a challenging problem of future research, how to effectively sample such a high-dimensional appearance space within reasonable acquisition times and disk space requirements.

CHAPTER 5

INTEGRATED ACQUISITION OF GEOMETRY AND REFLECTANCE

After the in-depth discussion of BTF measurement setups with an integrated 3D acquisition in Chapter 4, this chapter will focus on the details of the postprocessing. We follow the steps of the digitization pipeline outlined in Figure 5.1. In doing so, we explain how the digital model is created from the measurement data. Eventually, we obtain a 3D geometry in combination with a BTF (see Figure 5.2), which is the representation we already identified as favorable in Section 3.3.2.

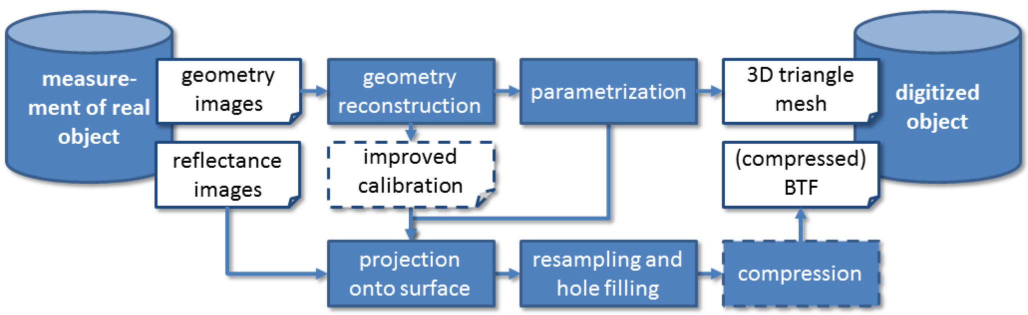
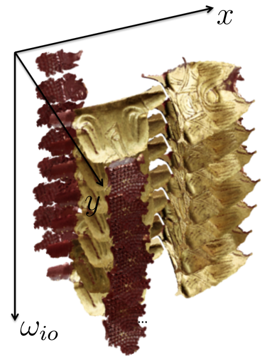


Figure 5.1: *Overview of our proposed processing pipeline for object digitization.*

5.1 Introduction

After performing an acquisition with either one of the dome devices, a large amount of images has been captured. In our digitization experiments, an average object measurement with the Dome 2 consists of about 230,000 single images, occupying more than 1 TB of disk space. Several consecutive processing steps have to be performed to boil this massive amount of data down to a manageable representation, consisting of a 3D triangle mesh and a matching BTF (see Figure 5.2).

(a) triangle mesh



(b) BTF

Figure 5.2: *The digitized object representation consists of a triangle mesh (a) and the BTF (b). Here, the BTF is symbolically visualized as a stack of surface textures for different view and light direction combinations ω_{io}.*

The 3D geometry serves as a proxy for the macroscale appearance effects such as shape, shadowing and occlusion. The BTF depicts the object appearance on the mesoscopic and microscopic scale. We therefore create a 3D geometry with a comparably low number of triangles. The resulting mesh occupies less than 10 MB of disk space and can directly be used for real-time rendering. The intricate details are captured within the BTF. Following the considerations in Section 3.3.2, the captured reflectance samples are resampled and expressed with respect to the surface of the 3D geometry. For the local directions we choose the same sampling as the Dome 1, presented in Table 4.2. The BTF thus results in a stack of 22,801 HDR surface textures; one for every possible combination of view and light direction. If the available camera resolution is fully exploited, the resampled reflectance data in our experiments amounts up to 2 TB, depending on the surface area. To facilitate real-time rendering and reduce storage requirements to a more economic level, we therefore eventually perform a lossy compression step.

The compressed BTF files created by the process described in this chapter still occupy between 821 MB and 6.4 GB. While this is just within the capabilities of the very latest graphics hardware, we rather understand this form of representation to be a “master” file, much like digital masters in audio recordings. From it, lower-quality versions can be created to support rendering the digitized object on older hardware or mobile platforms. Alternatively, Chapter 7 introduces an approach to use hierarchical level of detail rendering to cope with the still large amount of compressed BTF data. First, however, this chapter presents the necessary processing steps to obtain this master representation.

In summary, our main contributions are

- a practical and robust pipeline to obtain a 3D geometry and surface parameterized BTFs from raw measurement data,
- a novel resampling and hole filling approach for BTFs,
- a small extension to two state-of-the art BTF compression schemes to cope with the high dynamic range of the data,
- an extensive experimental evaluation of the pipeline on 27 challenging objects.

After a discussion of the related work in Section 5.2, we briefly recapitulate the important steps and quantities of the image acquisition process in 5.3. Section 5.4 gives the necessary details on our procedure to obtain a 3D geometry and an improved camera calibration from the structured light images. Then, in Section 5.5.1, the mesh is parameterized and a texture atlas is created. Using the 3D mesh and an improved camera calibration from the geometry reconstruction step, we can project the view- and light-dependent HDR reflectance images onto the surface, which is described in Section 5.5.2. Next, a resampling and hole filling step is performed in Section 5.5.3. This brings the reflectance samples of every point on the surface into a dense regular grid of directions and thereby creates the desired surface parameterized BTF representation. Then, Section 5.5.4 describes two lossy compression methods to reduce the large amount of data from the raw BTF while still maintaining a high visual quality. We evaluated the approach by digitizing a total of 27 challenging real-world objects. The results are discussed in Section 5.6. Finally, we summarize the findings in Section 5.7.

5.2 Related Work

Section 3.3 provides an overview on other methods found in literature that digitize the appearance of complete objects. There, we identified two previous publications that are very closely related to our approach. Furukawa *et al.* [FKIS02] and Müller *et al.* [MBK05] both also acquired images of 3D objects and obtained a 3D geometry and a BTF from the measured data. Therefore, we will discuss these approaches here in more detail.

Both methods rely on a shape from silhouette approach to reconstruct their 3D geometry. The major drawback to using such a method is that most concave regions of an object cannot be reconstructed, even when using an infinite number of views. Additionally, as Figure 5.3 demonstrates, a practical number of images leads to a very coarse approximation of the 3D shape. This is problematic, as macroscopic occlusion, parallax and shadowing effects that have not been captured by the visual hull geometry are then instead incorporated into the BTF. However, the BTF has only a finite directional resolution, leading to a blurred appearance if real surface and proxy geometry are too far apart. This effect has been described for outgoing light fields by Gortler *et al.* [GGSC96]. In the case of BTFs this consideration applies to the captured illumination-dependent effects as well. Gortler *et al.* also already make the observation that an inexact proxy geometry impairs compression performance.

To overcome this weakness, in [FKIS02], additional more accurate geometry information was obtained with a laser scanner. However, the registration of geometry and images is a challenging task on its own and poses a serious problem. In contrast, our approach does not necessitate a registration by utilizing an integrated structured light measurement instead.

In addition to the distinction in geometry acquisition, we also propose different strategies for projection, resampling and hole filling of the reflectance data.

For the projection, Furukawa *et al.* simply map each triangle to half a square of a lattice in the texture domain. The resulting texture atlas is extremely fragmented. However, our results in Part III of this thesis show that having larger connected parts in the parameterization is beneficial, as the resulting texture atlas shows more characteristics of a natural image. This can for instance be exploited for additional image compression (see Chapter 6) or for virtual texturing (see Chapter 7). Therefore, we, as well as Müller *et al.*, instead employ a more elaborate parameterization algorithm, leading to a largely connected texture atlas layout.

Furukawa *et al.* write that they perform a *"reparameterization"* such that *"Both of these view and light directions are represented by certain local coordinates, which are fixed onto the object"* [FKIS02]. However, this is not a resampling in our sense, as they do not take any local orientation frame of the surface points into account. The authors merely map the acquisition device inherent capturing parameters, containing the turntable and arm rotation, to light and view directions in a common global coordinate system. Furthermore, Furukawa *et al.* make no effort to mask parts of the reflectance that show occlusions and shadows caused by the macroscale geometry. Consequently, there is also no notion of missing values and hence no hole filling. Instead, during rendering the required pair of directions ω_i, ω_o at each surface patch is transformed into the global coordinate system of the capture device. Then, the final color is computed by a linear interpolation of the closest captured values.

This approach introduces undesirable additional computational load during rendering. Instead, we propose to perform direction transformation and interpolation once as a postprocessing step of the digitization, speeding up any actual rendering procedure. Furthermore, the lack of occlusion handling in [FKIS02] produces visible ghosting artifacts in the presented result images. We instead propose to mask such cases as missing values and perform a hole filling step during postprocessing.

Similar to Furukawa *et al.*, Müller *et al.* do not resample the BTF data but consider the samples at each point to be given in the directions of the measurement setup, which in their case is the Dome 1 shown in Section 4.6. However, they do implement a hole filling approach, filling in masked data due to occlusion and shadowing. They first perform a statistical analysis based on the *principal component analysis* (PCA) on all points on the surface that do not suffer from missing data, i.e. upwards oriented and unoccluded geometry. In particular, they use the *local PCA* (LPCA) [MMK03], which spatially segments the data during analysis into a predefined number of clusters. In doing so, the authors construct a lower dimensional basis of ABRDFs for each cluster to describe the reflectance of the analyzed points. Then, they find a set of coefficients for these basis ABRDFs for each remaining point by minimizing the distance to the measured samples. This approach has the severe drawback that only those surface patches that face the zenith of the measurement setup will have measured values for all directions. Hence, the statistical analysis is only performed for points with this one orientation. It is doubtful whether such an ABRDF basis could correctly describe (and fill-in) the reflectance of surface points that have a significantly different orientation (e.g. on the side of objects).

In his dissertation [Mül09], Müller therefore proposes to use his method presented in [MSK06] instead of the LPCA. In [MSK06], Müller *et al.* present a data-driven estimation of local coordinate frames, aligning the reflectance values prior to PCA. This would solve the mentioned problem of bad matching reflectance values at differently oriented points on the surface. Indeed, Müller reports an improved visual quality of the digitized 3D object [Mül09]. However, one problem still remains: Only points that are facing the zenith of the measurement device participate in creating the ABRDF basis. If all materials that are encountered on the surface of the object are well represented by exactly these points, this might be tolerable. This is the case for the 3D object in [MBK05, Mül09], which is made from a single homogeneous material. Yet, our application scenarios foresee objects that exhibit significantly different surface materials (e.g. Figure 5.10). Here, it is not the case that all possible material variations occur in upward facing areas that are never occluded or in shadow.

As a consequence, although we also follow the idea to perform a statistical analysis for data completion, we want to avoid such a situation. Therefore, we perform the analysis on data that has been resampled into directions in the local hemisphere of the surface point. Furthermore, we choose the representative points to compute the ABRDF basis uniformly distributed over the object's surface, avoiding a bias for certain directions.

Recently, we also proposed a reparameterization and hole filling approach for SVBRDF data together with Ruiters *et al.* [RSK12]. Similar to [MBK05], it directly integrates into a compression technique. A factorized tensor representation is estimated in an iterative process from a set of irregularly measurement values. The method utilizes three basic prior assumptions: low separation rank, spatial self-similarity and isotropy. The resulting representation can potentially show high resolutions in the spatial and angular domain while still being very compact. Figure 5.16 shows a comparison between BTFs and the tensor fitting technique.

However, the iterative optimization to estimate the factorized values of the tensor is computationally very costly. As a consequence, we only presented results for low spatial resolutions and a very sparse set of input samples. The processing takes about 16 hours for a data set with 256×256 texels and 150 direction-dependent samples, each. In contrast, the digitized objects created with the approach proposed in this thesis have a spatial resolution of up to 4096×4096 texels with up to 52,282 samples per texel. This is about 90,000 times as much. On this amount of data, the approach presented in [RSK12] would require an impractically long computation. Furthermore, in the approach discussed in this thesis, we omit the restriction to isotropy. Actually, most of our digitized objects show an anisotropic reflectance that could otherwise not be reproduced.

5.3 Image Acquisition Procedure

To have a good impression of the input data to the described pipeline, we now briefly recapitulate the important properties of the image acquisition. We will be working with data obtained from two measurement devices: the Dome 1 and the Dome 2. In both cases, the real object is placed in the center of the device. Then, in an automatic measurement procedure, both devices capture a large number of images of the object. A detailed description of the respective acquisition process can be found in sections 4.6.3 and 4.7.3.

For postprocessing, we group the resulting measurement data into geometry and reflectance images. The reflectance images show the object captured from different viewpoints under varying light positions. They are enumerated by their light and view index combination $\omega_{io} \in \mathfrak{L} \times \mathfrak{V}$ for the sets of light sources $\mathfrak{L}$ and camera views $\mathfrak{V}$. In the case of the Dome 1 setup, both sets have the same number of entries, i.e. $|\mathfrak{L}| = |\mathfrak{V}| = 151$. The Dome 2 setup provides a denser set of directions with $|\mathfrak{L}| = 198$ and $|\mathfrak{V}| = 264$. However, while the Dome 1 captures the full Cartesian product $\mathfrak{L} \times \mathfrak{V}$ in 22,801 reflectance maps, the Dome 2 by construction only captures partially different light directions sets for different azimuthal angles of the viewing directions. Still, eventually a total of 198×264 = 52,272 images is captured. Please refer to Section 4.7.1 for more details. This slightly irregular distribution of directions is no problem, as the proposed processing pipeline can in principle handle arbitrary angular combinations.

The geometry images depict the object under a structured light illumination. They use the same set of camera views $\mathfrak{V}$. However, instead of homogeneous illuminations from a set of different light sources $\mathfrak{L}$, the projection of different fringe patterns onto the object is recorded. The patterns $g \in \mathfrak{G}$ are cast consecutively from different projectors $p \in \mathfrak{P}$ at different positions around the object. In our experiments, we used different numbers of projector positions from $|\mathfrak{P}| = 4$ to $|\mathfrak{P}| = 8$. Please refer to Table 5.1 for the exact numbers. As with the reflectance maps, the Dome 1 captures the full Cartesian product $\mathfrak{G} \times \mathfrak{P} \times \mathfrak{V}$. Again, the Dome 2 has by design a different set of projector positions $\mathfrak{P}$ for different view directions.

Calibration: For the realistic depiction of the digitized object, a precise calibration of the measurement setup is of high importance. A fine-grained 3D reconstruction of the geometry of an object requires an accurate geometric camera calibration, i.e. intrinsic and extrinsic parameters of the cameras' perspective projections. Here, we rely on the good calibration of the devices, which is described in great detail in sections 4.6.2.1 and 4.7.2.1. Since we captured structured light images for obtaining the 3D geometry, we also perform the mentioned additional SBA step proposed in [WSRK11] for an increased accuracy of the Dome 1.

We also need to consider the radiometric calibration to obtain meaningful radiance values. We follow the procedures described in sections 4.6.2.2 and 4.7.2.2. Note that the radiometric correction factors (see equations 4.13 and 4.17) are dependent on the surface position $\mathbf{x}$ and hence already require geometry information. Fortunately, for the structured light images, we are not interested in the reflectance but rather the radiance. Furthermore, the fringe projections consist only of binary black and white stripes, making the 3D reconstruction very robust (see Section 5.4). Here, the dependency on an exact surface position is not that important. When computing the geometry images' radiance maps, we thus make the crude approximation of the position to be at the center of the sample holder. Then, we first reconstruct the 3D geometry and use it to obtain the correct positions $\mathbf{x}$ in the consecutively performed radiometric correction of the reflectance images.

For the Dome 1 setup, it needs to be taken into account that a 3D object might cast a shadow onto the radiometric calibration markers (see Figure 4.6c). In our current implementation, we do not consider radiometric calibration markers that are in shadow. That is, we set their weight to $w = 0$ in Equation 4.11. For the results presented in this thesis, we manually identified the images and markers in question, although this could probably also be solved algorithmically, e.g. by thresholding or deduction from the reconstructed 3D geometry.

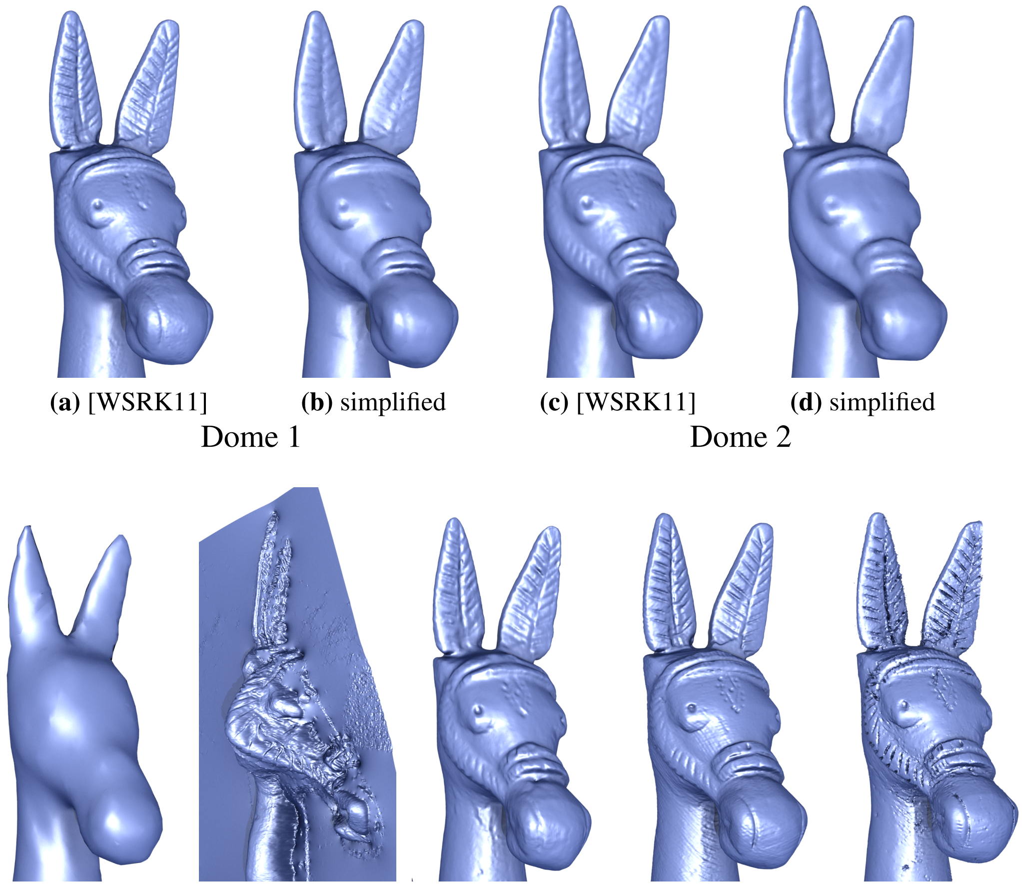
Note that this weighting approach is not completely unproblematic. It might impact the radiometric correction accuracy, since all remaining markers could be oversaturated or underexposed. This could potentially be avoided by placing redundant markers on opposite sides of the sample holder. Fortunately, the more recent Dome 2 setup does not suffer from this drawback, because here no additional radiometric calibration markers are required. Hence, we did not further investigate this issue, but instead manually reviewed critical images for their correctness.

5.4 Geometry Acquisition

A multitude of possible methods exist to capture the 3D shape of an object. We already provided an overview in Section 3.4. There, we opted against the use of several approaches for practical reasons. Figure 5.3 presents a comparison of reconstruction results obtained with the most relevant remaining methods.

Figure 5.3e depicts the result of the shape from silhouette approach employed by Müller *et al.* in [MBK05]. The geometry is reconstructed as the visual hull from 151 silhouettes, generated from a Dome 1 acquisition. The mesh is obviously very coarse and lacks a significant amount of details. Figures 5.3g and 5.3h show geometries obtained via laser scanning. The first is the result of the commodity NextEngine 3D Scanner HD [Nex], whereas the second is produced by a professional measurement setup, combining a Perceptron ScanWorks V5 scanner head with a Romer Infinite 2.0 high-precision arm [Met]. Although the scanners are not the same model as employed by Furukawa *et al.* [FKIS02], they provide an impression of the quality of laser scanned geometry. Note that the Perceptron scanner captures only a single scanline per arm position, leading to noticeable stripe artifacts in the reconstruction.

We also tested the reconstruction with photometric stereo (see Figure 5.3f). The geometry was generated by the University of Leuven's "Minidome" [WVM*05]. At first glance, this method might seem to be an appropriate choice, as the reflectance acquisition provides the necessary images under different illumination directions. However, our test revealed that the employed single-view photometric stereo reconstruction is not well suited for optically complicated materials, such as the specular metal of the Donkey object, and produces an unusable result.



(a) [WSRK11] **(b)** simplified **(c)** [WSRK11] **(d)** simplified

Dome 1 Dome 2

(e) [MBK05] **(f)** [WVM*05] **(g)** [Nex] **(h)** [Met] **(i)** [WRO*12]

Figure 5.3: *Comparison of different geometry reconstruction methods on the Donkey object (see Table 5.1 for details). Only the head part of the object is shown to emphasize the difference in the reconstruction of small geometric details. Please refer to Figure 5.10 for a depiction of the full object. The first row (a) – (d) shows the geometry obtained with the described pipeline. We eventually employ the smoothed and simplified geometries shown in (b) and (d) as representation of the macroscale geometry. The second row illustrates results achieved with different alternative geometry acquisition approaches proposed in the literature. Image (e) shows a reconstruction with shape from silhouette with 151 different views obtained by the Dome 1. In (f), a single-view photometric stereo method was used. The geometries in (g) and (h) were obtained using off-the-shelf laser scanners. (i) shows a result obtained with the Dome 2 using an alternative reconstruction method. Here we employed an approach based on Helmholtz stereopsis and structured light consistency, we explored together with Weinmann et al. .*

In our approach, we instead use structured light to obtain the macroscopic geometry of the objects. Surveys show that structured light approaches robustly produce accurate 3D models [Bla04]. Instead of employing an auxiliary off-the-shelf structured light scanner, we use the cameras of the Dome devices to capture the patterns projected by ordinary DLP projectors. As argued in Section 4.1, this has the advantage that the geometry and the reflectance measurement are both undertaken using the same sensors, avoiding registration of the measurements. We employ the super-resolution method that we developed together with Weinmann *et al.* [WSRK11]. The result is a point cloud from which we eventually reconstruct a triangle mesh, depicted in figures 5.3a and 5.3c. Using the Dome 1, our results achieve a similar fidelity as that of a professional laser scanner(Perceptron ScanWorks V5 head on a Romer Infinite 2.0), depicted in Figure 5.3h.

5.4.1 Structured Light Reconstruction

Techniques based on structured light project a sequence of patterns onto the object, assigning every point on the surface a code. Recognizing these codes in one or more cameras provides correspondences, which allow the reconstruction of the surface points via triangulation. The standard approaches in the literature propose to triangulate points using the rays from the the projector and a single camera. While this usually makes sense, as it reduces the amount of required hardware in a scanner, it requires the projector to be well calibrated. In our first acquisition experiments, however, we manually repositioned the projector on a tripod (see Section 4.6.1.4). Performing an accurate calibration procedure for each position would have been very time-consuming. Additionally, we found that the optics of the employed low-cost consumer projectors (both, the Acer C20 and LG HS200G) showed a lot of distortions, hampering a good calibration in the Dome 1 as well as the Dome 2. Fortunately, both setups show a sufficiently large number of cameras to completely abandon the projector rays during triangulation.

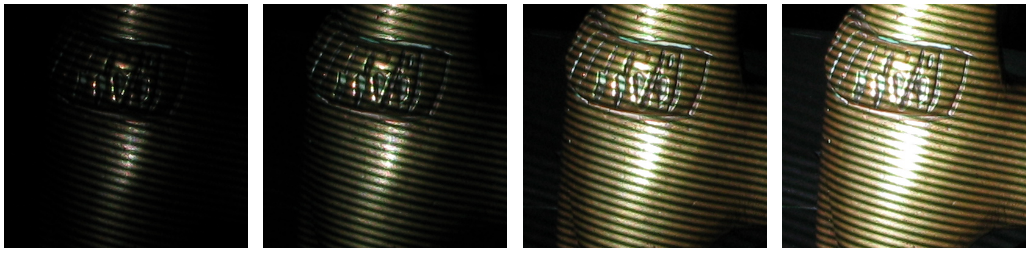
There are several well-studied pattern codification strategies for structured light. Salvi *et al.* [SPB04] distinguish between the fundamental classes of time-multiplexed (or *temporal*) coding, spatial neighborhood coding and direct coding. Temporal coding employs a sequence of patterns to form the code words, whereas spatial neighborhood coding and direct coding aim to convey sufficient correspondence information within a single pattern. The two latter classes are certainly a good choice for dynamic scenes. However, they impose strong requirements on the object, such as local smoothness of the geometry or uniformity of color. For our scenario of capturing objects that on the one hand are likely to violate these requirements but on the other hand are static, we therefore utilize temporal coding.

For temporal coding, the survey by Salvi *et al.* lists three different approaches: First, *binary codes* use a series of 1 bit black-and-white patterns to convey their code words. Again, there are several possible options for the binary representation. They all have in common that the highest positional resolution is that of a single projector pixel. Furthermore, $\lceil \log_2(n) \rceil + \lceil \log_2(m) \rceil$ patterns are necessary to uniquely identify $n \times m$ pixels. The authors highlight Gray code [Gra53] as a particular good choice for structured light 3D reconstruction [ISM84]. Consecutive code words have a Hamming distance of one [SPB04]. Therefore, small localization errors of 0–1 crossings cannot result in large absolute code changes [SS03]. This restricts the maximum reconstruction error and hence the noise in the resulting point cloud. Second, there is the extension of binary to *n-ary* codes. However, this is not applicable to our case, as these codes require the reflectance to be constant over the whole object.

Finally, there is *phase shifting*. Here, a periodic grayscale pattern (usually a sine wave) is projected n times, each time shifted by $\frac{1}{n}$ of the period. Usually, n is chosen as 3 or 4, resulting in a very low number of patterns ($2n$ for horizontal and vertical position determination). The relative phase value for each image pixel can easily be reconstructed from the captured shifted patterns. This technique has the advantage that a higher precision than one projector pixel can be achieved. If the projection is brought slightly out of focus, a suitable signal (e.g. a sine wave) seemingly appears to be continuous on the object's surface [SPB04]. However, the phase is only unique within a single period of the signal. Thus, an additional *phase unwrapping* is required to obtain meaningful absolute values as correspondences. For this, optimization techniques can be employed [ARGB*07], possibly aided by stereo camera constraints [WLVG07, GZ11a]. Unfortunately, this again requires the surface to be locally smooth. Salvi *et al.* [SPB04] point out that Gray codes can be used to resolve the ambiguity. Yet, this abolishes the advantage of requiring fewer patterns. In addition to that, in [SS03] there is a comparison of the usage of Gray codes and sine wave patterns. They point out that sine patterns are more susceptible to nonlinearities in the color processing of the cameras and projector and to interreflections in the scene.

In order to be sufficiently robust and as general as possible, we thus employ Gray code pattern sequences in our structured light approach. To be impervious to highly varying albedo and reflectance of the objects, we employ HDR imaging when acquiring the patterns. In contrast to Lambertian materials, over- and underexposure of the pattern is an issue when dealing with more complex reflectance behavior (see Figure 5.4). This necessity was also observed and handled in [SS03], but only by selecting the better of merely two exposures.

In [SL00], Skocaj *et al.* create HDR images of the patterns by varying the intensity of their projector. However, the employed LED-DLP projectors do not offer any



(a) 125ms **(b)** 500ms **(c)** 2000ms **(d)** 4000ms

Figure 5.4: *An exposure time series of the same fringe projection on a specular object. The images were captured by the Dome 1 setup, using the Canon PowerShot G9 cameras and the Acer C20 projector. Images taken from [WSRK11].*

control over the brightness of the lamp. We could instead modify the pixel values of the fringe images, e.g. use darker tones of gray instead of white. However, this fails due to the insufficient contrast of the projectors. As Figure 5.4c demonstrates, in a single camera image, the intensity of the black stripes of the fringe pattern can lead to oversaturated pixels, while the white stripes are still not bright enough to fully illuminate regions at steep angles.

We instead employ the approach we proposed together with Weinmann *et al.* in [WSRK11]. We vary the exposure time of the cameras and use additional inverse patterns as well as a fully lit and fully black image to achieve a reliable classification of whether a pixel is illuminated or not.

With precisely calibrated cameras, the epipolar geometry of the cameras and scene points restricts the search space for correspondences between images onto a single line, the epipolar line. Therefore, projecting the Gray code in only one pattern dimension, e.g. only as horizontal stripes, would in principle suffice: Matching points would be exactly determined by the intersection of the stripe and the epipolar line in the image. However, as described in Section 4.6.2.1, the intrinsic parameters of the Dome 1 setup cannot be precalibrated with sufficient precision. Instead, we utilize the correspondences obtained from the structured light projection to refine the initial calibration via SBA. As a consequence, this requires us to project horizontal as well as vertical stripe patterns to robustly determine the position of corresponding points in the camera images. This results in a total number of $|\mathfrak{G}| = 2(\lceil \log_2(848) \rceil + \lceil \log_2(480) \rceil) + 2 = 40$ and $|\mathfrak{G}| = 2(\lceil \log_2(800) \rceil + \lceil \log_2(600) \rceil) + 2 = 42$ different patterns for the Acer C20 and LG HS200G projectors respectively. Each pattern is captured between one and four times with different exposure times. In the end, we achieve a robust decoding, which avoids mis-classification very well.

As the employed projector models have mainly been chosen due to the size and positioning constraints within the acquisition devices (see Section 4.6.1.4), we had to compromise on the available projector resolution. This would usually restrict the achievable geometric accuracy for Gray code-based structured light. However, in order to capture the object's geometry from all sides, we use a series of projections from multiple directions. This is especially important in the Dome 1 setup, as the object will not be moved during the acquisition. Here, five to eight projector directions have been distributed over the hemisphere. Yet, even in the Dome 2 setup, four projectors at different inclinations are employed. That allows us to utilize the superresolution idea we proposed together with Weinmann *et al.* in [WSRK11]. In our approach, we combine the codes from multiple projectors to generate larger code words and thus obtain dense and more precise correspondences. Given a low projector resolution and high camera resolution as in our case, this increases the amount of captured geometric details significantly.

Following the procedure in [WSRK11], we perform a sparse bundle adjustment [LA09] on the numerous correspondences. For computational complexity and accuracy reasons, only a subset of the given correspondences that we identified as highly confident is used. From this, we obtain a very precise point cloud and improved camera calibration. Please note that due to the combination of codes from all projection directions into unique code words, the superresolution method computes a single set of correspondences instead of one per projection direction. This eliminates the need for a combination of different point clouds and conveniently considers all available information in a global optimization.

Unfortunately, the superresolution method can only be applied if cameras, projectors and object are rigid with respect to each other. In the case of the Dome 2 setup, this means that the approach has to be employed for each rotation of the turntable separately and the resulting point clouds have to be registered and merged. Here, we rely on the accurate calibration of the turntable to rotate the partial results onto each other rather than performing a separate registration step.

A better alternative would be to use a volumetric approach with a consistency measure, as we explored together with Weinmann *et al.* in [WRO*12]. Here, all available projections for all rotations are considered at once in a common volumetric representation. Figure 5.3i demonstrates the reconstruction quality that was achieved with this method using the Dome 2 setup. However, this approach assumes a precise calibration of the cameras and projectors. While this is available for the Dome 2, the method could only be used as a consecutive step after the SBA on the superresolution correspondences for the Dome 1. At any rate, the volumetric consistency approach is very computationally demanding and memory consuming. In this thesis, we only apply the approach in [WSRK11], as it provides sufficient quality for our use-case.

5.4.2 Surface Reconstruction

For further handling, such as editing and rendering, a mesh representation has several advantages over point clouds, being more compact and better supported by current graphics hardware. Therefore, we generate a triangle mesh using the Poisson surface reconstruction [KBH06, KH13]. This method generates closed surfaces, robustly dealing with remaining outliers, noise and holes. The latter occur in regions of the object that have not been lit by any projector or have not been observed by sufficiently many cameras, e.g. small but deep concavities or low downward facing regions. Even though the Poisson reconstruction produces a slightly smoothed surface, fine details are eventually represented by the BTF.

The Poisson reconstruction works on oriented point clouds, i.e. it requires a normal direction at every triangulated point. We estimate the normals by fitting a plane to the k nearest neighbors of each point via a PCA. In our experiments, we usually set k to values between 100 and 500. We employ a Gaussian kernel as distance-dependent weighting function to avoid over smoothing. The standard deviation of the kernel is determined by a local density estimation. The sign of the normal is then chosen in such a way that it is facing the centers of projection of those cameras that were used to triangulate the point in the first place. Alternatively, the captured reflectance data could be used to estimate better normal directions via Helmholtz stereopsis (see [WRO*12]). However, we found the quality of the PCA method to be sufficient for the Poisson reconstruction and did employ any more elaborate strategies.

In the employed implementation of the Poisson reconstruction[1], the final triangle mesh is obtained from the isosurface found in the volumetric representation using marching cubes [LC87]. This creates a regular and dense set of triangles. This might be beneficial for a lot of applications, such as physics simulations or storing per-vertex attributes. In our particular case, however, we instead use a parameterization to map the reflectance information. Thus, the dense triangle structure contains unnecessary redundancies on smooth parts of the surface. It possibly even encodes unwanted noise from the 3D reconstruction. We therefore perform a Laplacian smoothing [SCOL*04] and a consecutive quadric edge collapse simplification [GH97] on the meshes. The resulting final triangle meshes (see figures 5.3b and 5.3d) have a considerably reduced triangle count and smoother surface and at the same time still show almost all of the macroscopic details (see RMSE with respect to the point cloud in Table 5.2). Smaller mesoscopic scale details that might have been lost in the mesh during smoothing will be represented by the BTF during rendering.

[1] http://www.cs.jhu.edu/~misha/Code/PoissonRecon/Version5.5a/

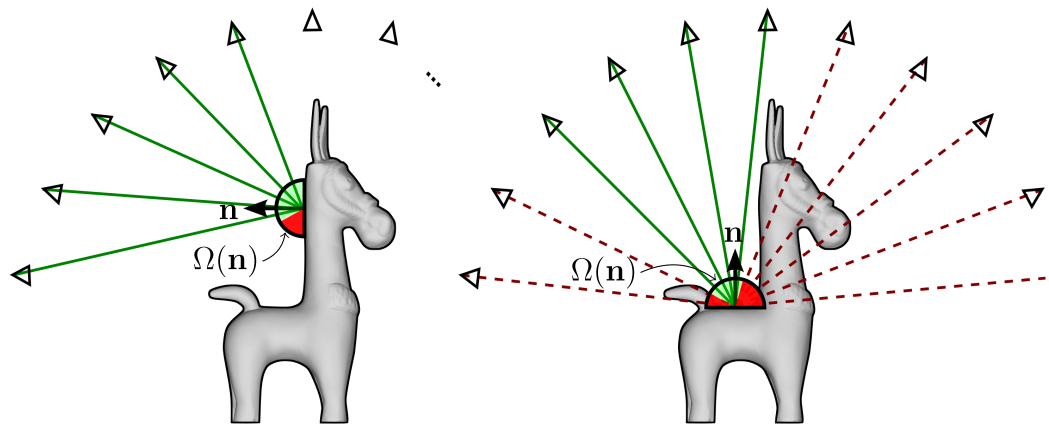


Figure 5.5: *Sketch illustrating the interpolation and hole filling problem. The image indicates captured directions at two surface points with orientation* $\mathbf{n}$*. Red shading in the local hemisphere* $\Omega(\mathbf{n})$ *indicates missing data for these directions. Left: missing data due to hemispherical setup. Right: missing data due to occlusion.*

5.5 Reflectance Acquisition

Originally, BTFs have been employed for planar proxy surfaces (see Section 3.2.6 and Figure 3.4b). Naturally, they have also mostly been captured from flat samples. In many approaches documented in the literature, including all of our setups in Chapter 4, flatness of the sample surface was even mechanically enforced by the use of special sample holders. In combination with orthographic or tele-zoom lenses and sufficiently directed illumination, this holds the advantage that the local light and view directions ω_i and ω_o are almost the same at every point on the sample surface. In [SSK03], for instance, the measurement directions have directly been employed for rendering computations without any further correction.

Unfortunately, for curved and protruding 3D surfaces the issue of data interpolation must be raised. In these cases, the surface has divergent orientations at different positions, rendering the respective local directions on the surface dissimilar to each other and to the measured ones. Depending on the geometry of the object, there may also be significant holes in the sampling of the local view and light hemispheres (see Figure 5.5).

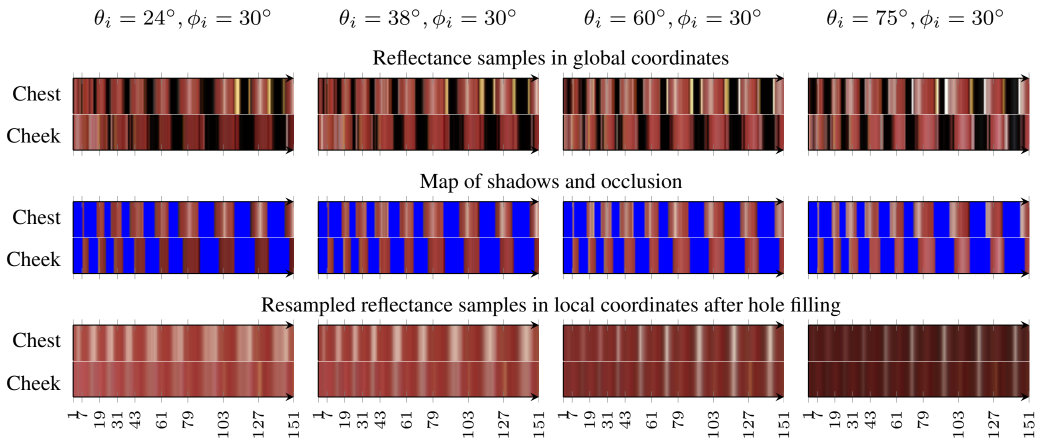
Without additional processing, this poses a hard problem for rendering. On the one hand, the application of elaborate resampling and hole filling for each reflectance lookup imposes an undesirable workload. On the other hand, simple real-time capable interpolation schemes between single views will easily introduce artifacts.

Furthermore, the surface of the virtual object could be locally deformed; Either partially rigid, e.g. for illustrating interactions with an articulated object, or even in a non-rigid manner, e.g. to simulate wrinkling fabrics. Unless the inverse transformation for each triangle is still known, correct rendering will only be possible with a representation parameterized in local coordinate frames.

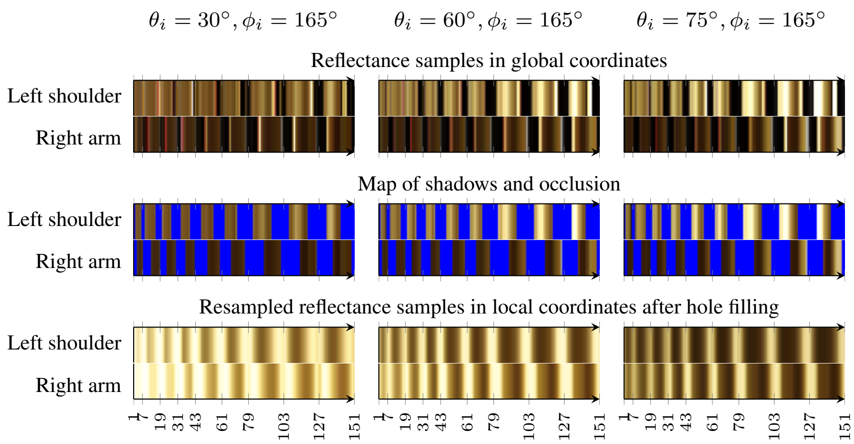
Our BTF representation therefore specifies the reflectance values for a regular grid of directions inside the local coordinate system. Further, we choose the same grid of local directions for every point. If the data is then arranged as a matrix as described in Section 5.5.4, all entries in one row of the matrix correspond to the same local direction. For one, this eases the random access during rendering. More importantly, this should align common characteristics (e.g. highlights, retro-reflections or the Fresnel effect) of the spatially distributed reflectance functions with each other. This is crucial for an efficient data-driven compression. In [Rus98], Rusinkiewicz observed that for BRDF data, an alignment of important features considerably reduces the number of basis coefficients in low-rank representations. In [MSK06], Müller *et al.* demonstrated that aligning the coordinate frames of the texels in a BTF significantly improves the efficiency of factorization-based compression.

In contrast to Müller *et al.*, in our case, parts of the orientation of local surface patches, i.e. the normal, is already known from the captured 3D geometry. Furthermore, our processed geometry also shows a good alignment of the local coordinate systems with respect to the tangent direction: Our employed parameterization maps the surface to large connected and ideally hardly distorted patches in $\mathbb{R}^2$. Thus, we obtain local coordinate systems that are consistent over large neighborhoods on the surface by aligning the tangent direction with one of the axes of the parameterization domain.

For these reasons, we propose one of the key contributions of this thesis: A novel resampling method for obtain hole-free reflectance data from the captured samples, parameterized over the local hemispheres. We perform the task of interpolation and hole filling as a preprocessing step rather than at render time. This makes more sophisticated hole filling and interpolation techniques applicable, which result in an improved overall rendering quality. Additionally, the hole filling removes the non-local macroscale effects, i.e. shadows and occlusion. In combination with the alignment of local coordinate frames this leads to a more regular structure of the reflectance data, as shown in Figure 5.6, and hence an improved compression efficiency and quality.

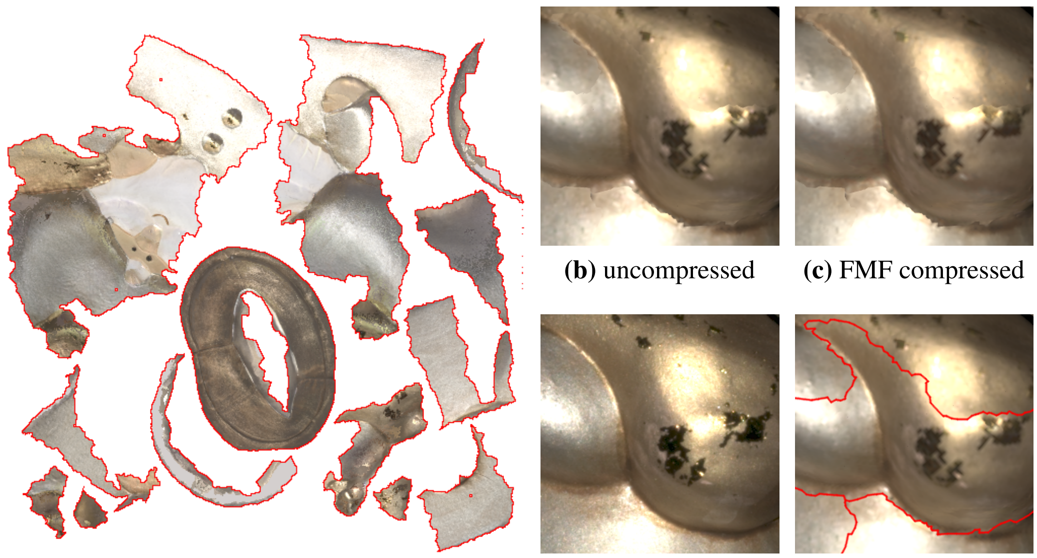


(a) indian red paint



(b) gold leaf

Figure 5.6: *Slices through ABRDFs of different materials found on the Buddha object. For each column, the illumination direction is fixed to the given value* (θ_i, ϕ_i)*. Reflectance samples for all 151 viewing directions are plotted along the horizontal axis, ordered by their indices (see Table 4.2). (a) shows ABRDFs of the red paint on chest and cheek, (b) ABRDFs of the gold leaf on the left shoulder and right arm. The first series shows the captured samples. The second series shows shadowed and occluded samples marked in blue. The third series shows the ABRDFs after resampling and hole filling. The polar coordinates for the samples are the same, but in the first two rows they refer to global world coordinates whereas for the bottom rows the local coordinate systems of the respective texels are used. The features of the respective ABRDF pairs (a) and (b), e.g. highlights or shading, are in much better alignment after resampling and hole filling. The remaining slight misalignment in (b) is probably caused by a different orientation of the mesoscale geometry.*



(a) texture atlas (b) uncompressed (c) FMF compressed (d) reference (e) seams

Figure 5.7: *Shading errors at seams in the surface parameterization of the Santa figurine (see Table 5.1 for details). (a) shows the texture atlas with borders of separate components highlighted in red. Images (b) and (c) show renderings of the arm and bag using the resampled and the additionally FMF compressed BTF respectively. The original measurement image is shown in (d) for reference. In both renderings, discontinuities in the shading – especially in the presence of highlights – become apparent at the borders (highlighted in (e) for illustration purposes). The errors are slightly more noticable when rendering with the compressed BTF.*

5.5.1 Parameterization

Since the BTF is defined over a surface $\partial V \subset \mathbb{R}^2$ (see Section 3.2.6), we need a parameterization $\Pi : \mathbb{R}^2 \mapsto \mathbb{R}^3$ that maps from the 2D plane to the points of the reconstructed 3D geometry. In practice, we compute and store the parameterized coordinates $\Pi^{-1}(\mathbf{v})$ for each vertex $\mathbf{v} \in \mathbb{R}^3$. We then use the barycentric coordinates of the mesh's triangles to compute values for either Π or Π^{-1} for arbitrary input positions in the triangle in $\mathbb{R}^2$ or $\mathbb{R}^3$, respectively. Without loss of generality, we restrict the parameterization domain to $[0, W) \times [0, H)$ with $W \in \mathbb{N}, H \in \mathbb{N}$. This way, Π directly maps 3D coordinates into texture images.

As the BTF is also in general anisotropic and depends on the full orientation of the local coordinate system, the orientations of the parameterization should be locally smooth and consistent. Finding a good parameterization for a mesh is a field of research on its own. The parameterization should impose a low stretch, in order not to distort spatial structures of the texture and waste resolution. Furthermore, for practicability the parameterization should be reasonably fast to compute.

In any case, it will in general be unavoidable to decompose the mesh into multiple parts parts in order to achieve a planar lay out. Unfortunately, artifacts can occur near the borders of these parts due to inconsistent orientations of the regular sample directions (see figures 5.7 and 5.11). This can happen at multiple stages of the pipeline, e.g. during resampling, compression or the linear interpolation of the tabulated values during rendering. Therefore, another important requirement on the parameterization algorithm is that the mesh should ideally be decomposed into few but large connected components.

We use ABF++ [SLMB05] for creating a parameterization and a texture atlas (see 5.7a). Other methods might also be applicable for this purpose. A survey of recent methods can be found in [HLS07].

Note that the methods presented in chapters 6 and 7 are also influenced by our choice of parameterization. In Chapter 6, large connected components in the texture atlas and low distortions lead to the important attribute that the computed textures for real-time rendering (see Section 5.5.4) show characteristics of natural images that can be exploited by image compression methods. In Chapter 7, the fact that visible parts of the mesh are usually mapped to localized regions in the textures allows to subdivide the BTF into tiles for efficient virtual texturing.

5.5.2 Projection

To obtain the BTF, we first project the captured reflectance values from the camera images into textures for the parameterized surface. For each texel in the parameterization domain $\mathbf{x} \in \mathfrak{X} \subset \{0, 1, \dots, W-1\} \times \{0, 1, \dots, H-1\}$, we calculate its corresponding 3D position in world space $\mathbf{v} = \Pi(\mathbf{x})$. Then, we compute the projection $\mathbf{v}'' = \mathcal{P}(\mathbf{v})$ into the image plane of each capturing camera using Equation 3.17. The reflectance value for $\mathbf{x}$ is then obtained from the reflectance image via bilinear interpolation. We utilize the GPU to efficiently perform the described projection for all texels in parallel.

As a result, we obtain a set of HDR reflectance textures $\mathcal{T}^{(l,c)}$, $l \in \{1, 2 \dots, |\mathfrak{L}|\}$, $c \in \{1, 2, \dots, |\mathfrak{V}|\}$ of size $W \times H$. Let $\mathbf{o}_c \in \mathbb{R}^3$ be the center of projection of camera c (see Section 3.5.1) and $\mathbf{i}_l \in \mathbb{R}^3$ be the point of origin of the illumination, i.e. the position of light source l. For a given texel $\mathbf{x}$, the color value in $\mathcal{T}^{(l,c)}(\mathbf{x})$ therefore depicts the reflectance of surface point $\mathbf{v} = \Pi(\mathbf{x})$ for illumination direction $\omega_l = \mathbf{R_v} \frac{\mathbf{i}_l - \mathbf{v}}{||\mathbf{i}_l - \mathbf{v}||}$ and viewing direction $\omega_c = \mathbf{R_v} \frac{\mathbf{o}_c - \mathbf{v}}{\|\mathbf{o}_c - \mathbf{v}\|}$. Here, $\mathbf{R_v} = (\mathbf{t}\,\mathbf{b}\,\mathbf{n}) \in \mathbb{R}^{3\times3}$ is the transformation from the global coordinate system into local coordinates, defined by tangents $\mathbf{t}, \mathbf{b}$ and normal $\mathbf{n}$ at point $\mathbf{v}$.

This means that the directions in a single texture are different for each texel. If these textures would directly be used for rendering, this would make bilinear interpolation between neighboring reflectance values more difficult. Hence, we additionally perform a resampling step, making sure that all texels have the same directions in the final textures that constitute the BTF.

Furthermore, if a point $\mathbf{v}$ has been occluded by other parts of the mesh geometry for a certain direction, care has to be taken not to use this direction sample. While mesoscale occlusion, i.e. masking, is an effect that should explicitly be represented by the BTF, macroscale occlusions pose a serious problem during rendering. Consider for example an observer that takes up a position between parts of the mesh, e.g. between the Donkey's legs or in the inside of the Mug (see Figure 5.10). If the user looks at the occludee, then erroneously the reflectance of the occluder behind the observer would be used.

Note that this issue is often not considered in image-based rendering. The distinction of our approach from other methods lies in the bounding volume for which the outgoing light field of a reflectance field is defined (see Section 3.2.5). As long as the observer is always outside the bounding volume, self-occlusions of geometry inside the volume are correctly captured and reproduced. In our case, however, we use the reconstructed mesh as the bounding volume and do not restrict the position of the virtual observer with respect to the mesh. A similar argument holds for virtual light sources and shadows cast by the macroscale geometry.

We therefore mask out occluded and shadowed parts during generation of the textures $\mathcal{T}^{(c,l)}$. We complete the missing direction values later during resampling with our hole filling approach. In our implementation, the necessary computations are performed on the GPU via shadow mapping.

For practical reasons, we employ one additional processing step subsequent to the projection and prior to resampling. The raw measurement data is given as a collection of single images. The GPU accelerated processing step can benefit from that, as it can process the single images independently, i.e. it takes a collection of photographic images and outputs a collection of texture images for each direction. However, the consecutive steps described in sections 5.5.3 and 5.5.4 instead operate on independent ABRDFs, i.e. they the values for a single texel from all images as an input. This would require access to information from all images at once. Since the data is usually too large to fit into main memory, we need to employ out-of-core processing. Reading all values for a single texel from a set of images would require a lot of scattered read operations. Furthermore, we employ `OpenEXR` images that are compressed via LZ77 and Huffman coding to save disk space and achieve higher throughput. This makes the situation even worse, as this format, like most other lossless image compression techniques, does not provide efficient random access. We therefore rearrange the data to better support the necessary ABRDF access. We iteratively go through the images, buffer as much of the texel values in RAM as possible and write continuous blocks of the rearranged ABRDFs to a temporary file. This avoids costly scattered disk operations and reduces the decompression overhead.

5.5.3 Resampling and Hole Filling

For our resampling and hole filling step, we consider each texel $\mathbf{x} \in \mathfrak{X}$ separately and regard its values in all projected images as a set of irregularly distributed reflectance samples given in the local coordinate system, i.e. $\rho(\omega_l, \omega_\mathbf{c}) := \mathcal{T}^{(l,c)}(\mathbf{x})$. As argued above, for efficient compression and rendering, the reflectance data should instead be arranged in a regular grid of directions. Here, any sampling, such as a regular 4D grid of polar coordinates $(\phi_i, \theta_i, \phi_o, \theta_o) \in \left([0, 2\pi] \times \left[0, \frac{\pi}{2}\right]\right)^2$, could be employed. This particular example, however, has an unnecessary oversampling at the poles. We choose the Cartesian product of the idealized Dome 1 directions for all of the results presented in this thesis. This sampling, given in detail in Table 4.2, uses the same 151 directions for view and light. It has the advantages that the directions are distributed quite uniformly over the hemisphere A comparison of a BRDF sampled with the idealized Dome 1 directions with other possible sets of directions can be found in Figure 3.2. In the following we denote this set of directions $\mathfrak{D}$ with cardinality $D := |\mathfrak{D}| = 151$. We further employ ω_{io} as a shorthand notation for direction combinations $(\omega_i, \omega_o) \in \mathfrak{D}^2$.

Resampling: We resample the available irregular samples into the target sampling using *radial basis functions* (RBFs) (Shepard's method [She68] with a Gaussian weight-function). For any bidirectional sample $\omega_{io} \in \mathfrak{D}^2$, we compute the resulting reflectance as

$$\tilde{\rho}(\omega_{io}) = \frac{\displaystyle\sum_{\omega_{lc} \in \mathfrak{L} \times \mathfrak{V}} \exp\left(-\frac{1}{2}\sqrt{\frac{1}{\lambda \mathrm{d}_{\min}} \mathrm{d}\left(\omega_{io}, \omega_{lc}\right)}\right) \rho(\omega_{lc})}{\displaystyle\sum_{\omega_{lc} \in \mathfrak{L} \times \mathfrak{V}} \exp\left(-\frac{1}{2}\sqrt{\frac{1}{\lambda \mathrm{d}_{\min}} \mathrm{d}\left(\omega_{io}, \omega_{lc}\right)}\right)}. \tag{5.1}$$

Here, $\mathrm{d}_{\min} := \min_{\omega_{lc}} \mathrm{d}\left(\omega_{io}, \omega_{lc}\right))$ denotes the distance to the closest captured sample. This makes the RBF kernel size dependent on the closeness to a valid sample and a user controlled parameter λ. In all of our experiments we set $\lambda = 1.5$.

For the distance metric d, we utilize the Rusinkiewicz parameterization (see Section 3.1.3.1). We use the notation $\mathbf{h}_{io}, \mathbf{d}_{io} \in \Omega_{\text{Cartesian}}$ and $\mathbf{h}_{lc}, \mathbf{d}_{lc} \in \Omega_{\text{Cartesian}}$ to denote the *halfway* and *difference* vectors for direction pairs ω_{io} and ω_{lc}. The distance is then defined as

$$\mathrm{d}\left(\omega_{io}, \omega_{lc}\right) = \sqrt{\arccos^2\left(\frac{\mathbf{h}_{io} \cdot \mathbf{h}_{lc}}{\|\mathbf{h}_{io}\| \, \|\mathbf{h}_{lc}\|}\right) + \arccos^2\left(\frac{\mathbf{d}_{io} \cdot \mathbf{d}_{lc}}{\|\mathbf{d}_{io}\| \, \|\mathbf{d}_{lc}\|}\right)}. \tag{5.2}$$

In practice, not all of the $|\mathfrak{L} \times \mathfrak{V}|$ input samples can be used, due to occlusion and back-facing orientation. This could be accounted for by introducing a weighting term per direction sample. For the sake of simplicity, we only use those samples of the texel that are identified as valid. However, the RBF interpolation would most of the time still be operating on a large subset of the 22,801 (Dome 1) or 52,272 (Dome 2) input samples. In order to avoid oversmoothing and heavy computational workload, we thus limit the RBF support to the n nearest neighbors of the target sample. The choice of n is user controlled. If we want to consider all samples in the 1-ring on the 4D manifold, we can approximate n by the number of neighbors on a regular 4D grid. This is either $n = 4 \cdot 2 = 8$ if neighborhood is only considered along the coordinate axes or $n = 3^4 - 1 = 80$ if diagonal connections are considered as well. We observed that the method is robust w.r.t. to the choice of n. Hence, we simply use $n = 10$ for all reported results.

As we aim to compute 22,801 samples per texel, we use an k-d tree data structure to accelerate the n nearest neighbor search. The tree has to be rebuilt for each texel, since the number and distribution of input samples is in general different. Although the ABRDF is a 4D function, the neighborhood information in 4D, e.g. as polar or parabolic coordinates, contains discontinuities and singularities or is distorted. We therefore use an embedding into 6D using the simple L^2-norm in order to obtain meaningful neighborhoods and distances. Instead of directly using Cartesian coordinates of $\mathcal{S}^2$, we use the halfway and difference vectors of the Rusinkiewicz parameterization, expressing in Cartesian coordinates. This way, important BRDF features are aligned with the coordinate axes and could be given different weights. However, we did not explore this option and instead use a weight of 1 for all axes.

Hole filling: The hemisphere of directions in the local coordinate system may contain significant holes, i.e. no meaningful reflectance data has been captured for a large cone of directions (see Figure 5.5). Furthermore, as argued above, we have to omit reflectance data from directions that show macroscale occlusions. Using RBF interpolation alone for these cases would produce a very dull and blurry appearance, as possibly contradictory reflectance values from far away directions are blended together.

To avoid this, we perform a separate hole filling. Our approach is based on the well explored best practice of using additional information from neighbors in appearance space. In many SVBRDF fitting approaches (e.g. [LKG*03, RK09b, HLZ10, WDR11, RSK12, PCDS12]), samples from multiple spatial positions on the surface are considered to estimate the BRDFs. For this, the points on the surface are either assigned a unique base material [HLZ10, PCDS12] or an affine combination of base materials [LKG*03, RK09b, WDR11, RSK12]. The often very sparse sampling for one point on the surface is then completed by samples of the same base material from other spatial locations. Criteria for the partition of the materials can be either spatial neighborhood, similarity in appearance or a combination thereof.

We follow a similar approach as Holroyd *et al.* [HLZ10] and partition the captured ABRDFs via k-mean clustering. Then, a d-dimensional material basis is established for each cluster. Finally, the ABRDF of each texel are expressed using the material basis of the assigned clusters. Yet, in contrast to Holroyd *et al.*, we do not fit analytical BRDF models but consequently maintain the data-driven paradigm. In the following we describe the steps of our novel hole filling in detail.

To keep the computational workload for the k-means clustering at a reasonable level, we run it on a representative subset of texels. First, we compute a confidence $\sigma(\omega_{lc}) \in [0, 1]$ for all direction pairs in the device sampling at every texel, depending on visibility and inclination angle of the view and light directions. We then draw a random subset of texels with a high overall confidence. For these texels, we perform an appearance space clustering based on their RBF interpolated ABRDFs $\tilde{\rho}$. As metric we employ the L^2 distance on the tabulated reflectance values. Since the representative texels have been chosen to have a high number of visible directions, they should not need the additional hole filling. The size of the random subset, the confidence threshold and the number of clusters k are user-defined parameters. In our experiments, we used between 200 and 1000 representative texels with an average confidence of $\frac{1}{|\mathfrak{L}\times\mathfrak{V}|}\sum_{\mathfrak{L}\times\mathfrak{V}}\sigma(\omega_{lc}) \geq 0.2$ and partitioned them into one to four clusters, depending on the object.

Whereas Holroy *et al.* propose to fit a mixture of d analytical BRDF models to the captured reflectance ρ, we instead follow the idea presented in [MBK05] to construct a low dimensional basis from the different ABRDFs $\tilde{\rho}$ of each cluster via statistical analysis. For this, we use *non-negative matrix factorization* (NMF) [LS00] to represent the reflectance values of all texels $\mathfrak{C}$ in the cluster as a product of two matrices

$$\mathbf{W}\,\mathbf{H} \approx \begin{pmatrix} \tilde{\rho}_1(\omega_{1,1}) & \tilde{\rho}_2(\omega_{1,1}) & \cdots & \tilde{\rho}_{|\mathfrak{C}|}(\omega_{1,1}) \\ \tilde{\rho}_1(\omega_{2,1}) & \tilde{\rho}_2(\omega_{2,1}) & \cdots & \tilde{\rho}_{|\mathfrak{C}|}(\omega_{2,1}) \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\rho}_1(\omega_{D,D}) & \tilde{\rho}_2(\omega_{D,D}) & \cdots & \tilde{\rho}_{|\mathfrak{C}|}(\omega_{D,D}) \end{pmatrix}, \tag{5.3}$$

such that all entries in $\mathbf{W}, \mathbf{H} \geq 0$. Here, $\tilde{\rho}_t(\omega_{io})$ is the reflectance for a specific direction pair $\omega_{io} \in \mathfrak{D}^2$ in the RBF-interpolated ABRDF at texel $\mathbf{x}_t \in \mathfrak{C}$. We obtain a low-rank basis $\mathbf{B}$ from the d most important columns of $\mathbf{W}$. We use $d = 10$ in all of our examples. This data-driven approach has the advantage that it does not impose the restrictive assumptions of an analytical model. We only assume that the material reflectance data is of low rank and can sufficiently be represented by the chosen number of columns.

In principle, a low-rank basis $\mathbf{B}$ could also be found using other matrix decompositions approaches, such as the SVD. However, by using NMF to compute a basis and *non-negative least squares* (NNLS) optimization to obtain the projection (see Equation 5.4), we omit filling the holes with negative reflectance values. This would be physically implausible and – depending on the rendering algorithm – introduce artifacts in the generated images. In our experiments we also observed that NMF handles ringing caused by overfitting of highlights more gracefully than SVD.

We furthermore avoid overfitting the data at highlights in the first place by applying a dynamic range compression of the reflectance values with $\sqrt[4]{}$ prior to the factorization in Equation 5.3. This operation has to be performed on all ABRDFs prior to projection in Equation 5.4 and reverted during blending in Equation 5.6. However, for better readability, we refrain from writing it explicitly.

Given the NMF basis, we can perform the hole filling for each texel. We interpolate its direction-dependent confidence σ via RBF, similar to the reflectance values. Then the RBF interpolated reflectance $\tilde{\rho}$ is projected into the basis $\mathbf{B}$ of each cluster, using NNLS optimization

$$\hat{\mathbf{x}} = \arg \min_{\substack{\mathbf{x} \\ \forall_i \mathbf{x}_i \geq 0}} \left(\|\mathbf{\Sigma} \mathbf{B} \mathbf{x} - \mathbf{\Sigma} \mathbf{r}\|_2^2 + \alpha \, \|(\mathbf{I} - \mathbf{\Sigma}) \mathbf{B} \mathbf{x} - (\mathbf{I} - \mathbf{\Sigma}) \bar{\mathbf{r}}\|_2^2 \right) \tag{5.4}$$

Here, $\mathbf{\Sigma}$ is a diagonal matrix, weighting the terms according to the confidence $\tilde{\sigma}$. The vectors $\mathbf{r}$ and $\bar{\mathbf{r}}$ contain the interpolated reflectance $\tilde{\rho}$ and the cluster's mean reflectance $\bar{\rho}$, i.e. its cluster center from the k-means algorithm:

$$\begin{aligned}
\mathbf{\Sigma} &:= \begin{pmatrix} \tilde{\sigma}(\omega_{1,1}) & & & \\ & \tilde{\sigma}(\omega_{2,1}) & & \\ & & \ddots & \\ & & & \tilde{\sigma}(\omega_{D,D}) \end{pmatrix}, \\
\mathbf{r} &:= \begin{pmatrix} \tilde{\rho}(\omega_{1,1}) \\ \tilde{\rho}(\omega_{2,1}) \\ \vdots \\ \tilde{\rho}(\omega_{D,D}) \end{pmatrix}, \qquad \bar{\mathbf{r}} := \begin{pmatrix} \bar{\rho}(\omega_{1,1}) \\ \bar{\rho}(\omega_{2,1}) \\ \vdots \\ \bar{\rho}(\omega_{D,D}) \end{pmatrix}.
\end{aligned}$$

The second term serves as a data-driven regularization prior to avoid artifacts in sparsely sampled texels. In these cases, or in the extreme case that no samples are available at all, the algorithm thus gracefully falls back to the average cluster reflectance $\bar{\rho}$. The strength of the prior is chosen for each entry in inverse relation to the confidence and is further controlled via the parameter α. In our experiments, we set $\alpha = 10^{-10}$.

Because this projection step is performed for each combination of texel and cluster, it should be reasonably fast. The two terms in Equation 5.4 can be converted to a single expression

$$
\begin{aligned}
\hat{\mathbf{x}} &= \arg \min_{\substack{\mathbf{x} \\ \forall_i \mathbf{x}_i \geq 0}} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 \\
\text{with } \mathbf{A} &= \sqrt{\boldsymbol{\Sigma}^2 + \alpha \left(\mathbf{I} - \boldsymbol{\Sigma}\right)^2}\, \mathbf{B} \\
\mathbf{y} &= \sqrt{\boldsymbol{\Sigma}^2 + \alpha \left(\mathbf{I} - \boldsymbol{\Sigma}\right)^2}^{-1} \left(\boldsymbol{\Sigma}^2 \mathbf{r} + \alpha \left(\mathbf{I} - \boldsymbol{\Sigma}\right)^2 \bar{\mathbf{r}}\right),
\end{aligned}
\tag{5.5}
$$

and $\hat{\mathbf{x}}$ can efficiently be computed by standard NNLS solvers, such as [LH95, LD11, KSD12]. We use the C implementation of [LH95].

The texel is then assigned to the cluster for which the NMF basis explains the reflectance values $\tilde{\rho}$ best, i.e. $\|\boldsymbol{\Sigma}(\mathbf{r} - \mathbf{B}\hat{\mathbf{x}})\|_2^2$ is minimal. Its final ABRDF is then computed by blending the RBF-interpolated reflectance and the reconstructed NMF-projection according to the confidence:

$$
\rho = \boldsymbol{\Sigma}\mathbf{r} + (\mathbf{I} - \boldsymbol{\Sigma})\mathbf{B}\hat{\mathbf{x}}. \tag{5.6}
$$

5.5.4 Compression

Our highest resolution test data sets have 4096×4096 texels of spatial resolution, 151×151 direction combinations and three color channels. When arranging the colors as part of the ABRDFs, this results in a BTF matrix $\mathbf{B} \in \mathbb{R}^{68{,}403 \times 16{,}777{,}216}$ with about 1.15 trillion entries. The number of actually used texels may vary depending on the occupancy of the texture atlas, see Table 5.2. Compactly storing only the non-zero entries of the matrix in half-precision floating-point values still results in a data size of up to two terabytes. Therefore, the BTF data set has to be compressed to enable efficient storage, transfer and rendering.

A large number of different techniques for BTF compression have been proposed. See [FH09, HF11, HF13] for recent surveys. We use either the *full matrix factorization* (FMF), proposed by Koudelka *et al.* [KMBK03], or the *decorrelated FMF* (DFMF), presented by Gero Müller [Mül09]. Even though there are several other techniques available, FMF-based compression offers several important advantages for our purposes. It provides good compression ratios while allowing for decompression at a reasonable cost. In addition, the FMF and DFMF are well suited for real-time rendering on the GPU (see Section 6.3). Both techniques have the considerable advantage that the texture mapping units of the GPU can be utilized to perform interpolation both in the angular and spatial domain. This reduces the decompression costs during rendering considerably in comparison with techniques using clustering [MMK03], sparse representations [RK09a] or vector quantization [HFM10], to name just a few.

We also prefer the FMF-based techniques over tensor factorization, because they are based on the SVD of the data. The resulting data of the compressed formats has important mathematical attributes that will be exploited in Chapters 6 and 7.

In the case of the FMF, the complete reflectance data is organized in a single matrix $\mathbf{B} \in \mathbb{R}^{M \times N}$ with $M = |\mathfrak{D}^2 \times \Lambda|$ directions and colors arranged as rows and $N = |\mathfrak{X}|$ texels organized as columns. In this thesis, only measurements with $|\Lambda| = 3$ color channels are considered. Koudelka *et al.* argue that reflectance values of textures should exhibit a high redundancy and can thus be well approximated by a matrix with a lower rank [KMBK03].

For their approach, Koudelka *et al.* first compute the *singular value decomposition* (SVD) $\mathbf{U\Sigma V}^T = \mathbf{B}$ of the BTF matrix. They then obtain a more compact ABRDF basis $\mathbf{U}' \in \mathbb{R}^{M \times C}$ by truncating $\mathbf{U}$ after the first C columns. Finally, the values for all texels are projected into the ABRDF basis, obtaining the spatial distribution $\mathbf{V}' = \mathbf{B}^T \mathbf{U}' \in \mathbb{R}^{N \times C}$. In this factorized form, $C(M + N)$ values need to be stored instead of $M \cdot N$. When choosing C sufficiently small, this leads to a considerable reduction of entries. Note that $\mathbf{V}'$ is the same as the first C columns of $\mathbf{V\Sigma}$. The compressed result $\mathbf{U}'\mathbf{V}'^T \approx \mathbf{B}$ is the optimal rank C approximation of the original BTF data with respect to the L^2-norm: According to the Eckart-Young-Theorem [EY36] the SVD computes the best possible rank C approximation of the original matrix under the Frobenius norm:

$$\arg \min_{\{\mathbf{U}_c, \mathbf{V}_c\}} \left\| \mathbf{B} - \sum_{c=1}^{C} \mathbf{U}_c \sigma_{c,c} \mathbf{V}_c^T \right\|_F^2 . \tag{5.7}$$

Here $\mathbf{U}_c$ and $\mathbf{V}_c$ denote the c-th column vector of the matrices $\mathbf{U}$ and $\mathbf{V}$ respectively and $\sigma_{c,c}$ denotes the c-th singular value.

As proposed in [KMBK03], we compute the ABRDF basis $\mathbf{U}'$ only for a subset of texels to keep the SVD computationally feasible. For this, we randomly select about 1% of the columns in $\mathbf{B}$, assuming that their values are representative of the reflectance on the whole surface. For some of the BTFs in this thesis, this smaller matrix still has up to six billion entries. We therefore employ the fast parallel matrix factorization algorithm by Ruiters *et al.* [RRK09] to further speedup computations. Due to these modifications, not all of the SVD properties are guaranteed any more. However, in practice we found no noticeable difference between the computation of an EM-PCA [Row97] on the full matrix $\mathbf{B}$ and our approximation.

Because of the close relation of the SVD to the *principal component analysis* (PCA), we also refer to the compressed data in PCA terminology. The singular vectors $\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_N$ and $\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_M$, i.e. the column vectors of $\mathbf{U}$ and $\mathbf{V}$, are in analogy to eigenvectors denoted as *eigen-ABRDFs* and *eigen-textures*, respectively. The number of remaining columns C in $\mathbf{U}'$ and $\mathbf{V}'$ is also denoted as the number of considered *components* of the PCA.

In contrast to the naive FMF, the DFMF compression [Mül09] first decorrelates the color related features of the reflectance information and seeks to compress the resulting individual signals each with a separate FMF. In our case, we separate the data into one luminance and two chrominance signals by transforming the RGB values into YUV color space and arranging each color channel's reflectance information as separate matrices $\mathbf{B}^{\mathrm{Y}}, \mathbf{B}^{\mathrm{U}}, \mathbf{B}^{\mathrm{V}} \in \mathbb{R}^{|\mathfrak{L}\times\mathfrak{V}|\times|\mathfrak{X}|}$. The rationale here is that for many materials there exists a strong dependence of the intensity on the view and light direction, whereas the color remains largely constant. Furthermore, the human visual system is more sensitive to differences in intensity than color, which is for instance exploited in JPEG image compression [Wik14]. The DFMF allows to account for these effects by employing a higher approximation rank for $\mathbf{B}^{\mathrm{Y}}$ at the expense of a lower rank for the two chrominance matrices $\mathbf{B}^{\mathrm{U}}$ and $\mathbf{B}^{\mathrm{V}}$. In our experiments, we used C components for $\mathbf{B}^{\mathrm{Y}}$ and $\frac{C}{2}$ components for $\mathbf{B}^{\mathrm{U}}$ as well as $\mathbf{B}^{\mathrm{V}}$. This amounts to storing $2k(|\mathfrak{D}^2| + |\mathfrak{X}|)$ values, i.e. twice as much values for the spatial distribution but one third less for the material basis when compared to the FMF with the same rank C.

However, since both techniques are based on the minimization of the Frobenius norm $\|\mathbf{B} - \mathbf{U}'\mathbf{V}'^T\|_F$, the high dynamic range of specular objects poses a difficult challenge. Often, the diffuse parts of the reflectance are mainly responsible for the overall appearance of the material. Yet, the FMF compression will sacrifice their approximation accuracy in order to keep the error of the considerably larger but visually less important values of the highlights in check. Like Matusik *et al.* [MPBM03], we address the problem by first performing a *dynamic range compression* (DRC) and computing the factorization for the dynamic-compressed data. We also use the natural logarithm, as this resembles the human brightness perception according to the Weber-Fechner law [Gla95].

In the case of the FMF compression, each element $b_{i,j}$ in $\mathbf{B}$ is compressed to $b_{i,j}^{\log} = \log(b_{i,j} + \varepsilon)$. For the DFMF compression, we apply the logarithm only to the elements of the intensity matrix $\mathbf{B}^{\mathrm{Y}}$ and employ a normalization with the

(a) no DRC **(b)** proposed DRC **(c)** reference

Figure 5.8: *Rendering using a DFMF compressed BTF with 16 components without (a) and with (b) the proposed dynamic range compression applied prior to the factorization. (c) shows a reference rendering with the uncompressed data. Without DRC, factorization leads to clearly noticeable rendering artifacts and an SSIM index of 0.95 with respect to the reference rendering. In comparison, if DRC is applied prior to computing the SVD, the SSIM index is 0.98. In the regions shown in the insets the SSIM index is 0.75 versus 0.94 respectively.*

intensity for the values of the two chromaticity matrices $\mathbf{B}^{\mathrm{U}}$ and $\mathbf{B}^{\mathrm{V}}$, i.e.

$$b_{i,j}^{\log \mathrm{Y}} = \log(b_{i,j}^{\mathrm{Y}} + \varepsilon), \qquad b_{i,j}^{\mathrm{U/Y}} = \frac{b_{i,j}^{\mathrm{U}}}{b_{i,j}^{\mathrm{Y}} + \varepsilon}, \qquad b_{i,j}^{\mathrm{V/Y}} = \frac{b_{i,j}^{\mathrm{V}}}{b_{i,j}^{\mathrm{Y}} + \varepsilon}. \tag{5.8}$$

To avoid numerical instability when taking the logarithm and dividing, we introduce a small constant $\varepsilon = 10^{-5}$. Our experiments indicated that the proposed dynamic-compression for DFMF is better at avoiding artifacts from overfitting of highlights than taking the logarithm for all three matrices. The DRC is reversed during rendering by applying the respective inverse transformations to the reconstructed values in $\mathbf{U}'\mathbf{V}'^T$.

Figure 5.8 illustrates the necessity of the DRC. Without this transformation, the reflectance characteristics outside the cone of the highlight are not sufficiently taken into account, since they do not contribute much to the squared error compared to the highlight itself. As a result, the error in the diffuse reflecting regions on the object, resulting from ringing due to overfitting of the highlights, can lead to noticeable artifacts during rendering (see Figure 5.8a). Figure 5.8b shows that the original appearance (depicted in Figure 5.8c) is better preserved by our proposed $\log \mathrm{Y}, \mathrm{U/Y}, \mathrm{V/Y}$ scaling.

Although the proposed logarithmic DRC can be easily inverted after decompressing a sample, its usage has some influence on the results of the real-time rendering

Figure 5.9: *Four objects from different measurements, composed in a common virtual scene, rendered with full simulated light transport (path tracing). The objects are true to scale.*

algorithm described in Section 6.3. To obtain a smooth appearance, the BTF values are usually linearly interpolated in all six dimensions. To improve real-time rendering performance, the proposed algorithm computes this interpolation separately on the spatial and angular components before reconstructing the BTF sample. More details can be found in Section 6.3. For plain FMF-based compression this course of action is completely unproblematic, as the reconstruction of the sample itself only employs linear operations as well, making it possible to change the order of operations without influencing the result. In the case of our nonlinear DRC, performing the linear interpolation prior to sample reconstruction effectively interpolates the samples in logarithmic space. In direct comparison, subtle changes in the resulting color gradients between single texels can be perceived. However, at practical viewing distances we did not notice any degradation in the overall material appearance.

5.6 Evaluation

We test our proposed acquisition system and processing pipeline using several challenging examples. The 27 objects involved in the evaluation in this thesis are depicted in Figure 5.10 and listed in Table 5.1. Several more objects have been measured with the Dome 2 during the scanning campaign at the Brighton Exhibition [3D-12]. However, since in these cases the objects were measured by a local operator, we had no control over the thoroughness of the calibration and measurement procedure. Thus, we only include three exemplary objects measured in Brighton in our evaluation: the Fish, the Inkwell and the Teal.

Our chosen test cases cover a large variety of geometrical shapes and surface materials. The complexity of the shapes range from almost perfect spheres (Billiard Ball and Tennis Ball) over mostly convex and smooth objects (Donkey, Buddha, Apple, Pyramid, etc.) to surfaces with many small protrusions (Terracotta Soldier, Strawberry, Almond Horn, Chess Piece, Fish) or deep concavities (Minotaur, Shoe, Mug, Epithelioma Moulage).

Yet, even the overall simpler shaped objects, such as the Donkey and the Buddha, exhibit regions with more complicated geometric details, e.g. the hair of the Buddha or the carvings on the Donkey's chest and head. In addition, the surfaces are covered with mesoscale geometric details, such as:

- The Donkey exhibits many small scratches and cracks with patina.
- The Buddha shows brush strokes on the red paint and bumps and cracks in the gold leaf (see figures 3.7 and 5.14).
- The surface of the Tennis Ball is covered with a fibrous felt.
- The "skin" on the hand of the Psoriasis Moulage appears to have plaques.

The complexity of the material appearance varies from uniform and diffuse (Terracotta Soldier, Chess Piece, Pyramid) over glossy and spatially varying (Minotaur, Buddha, Almond Horn, Santa, Shoe, Fish) to specular (Donkey, Billiard Ball, Micrometer, Inkwell). Several objects also exhibit subsurface scattering. Among them are most of the food items (Strawberry, Pudding Pastry, Apple, Crispy Roast Pork), both Moulages, the Ganesha and the second Ammonite. The last two also show the overall most challenging reflectance behavior: Their surfaces are subsurface scattering, have strong specular highlights and show iridescense, i.e. they shimmer in bright colors in some areas for certain direction combinations. We demonstrate this effect in figures 1.1 and 3.6. Still, even those objects that appear to have a single uniform diffuse material exhibit weak view- and light-dependent effects, e.g. at gazing angles, and vary spatially due to patina.

Note that all three Brighton objects suffer to some degree from the carelessness of the operators: First, the height of the Fish exceeds the defined measurement volume. As a result, a large part of the bottom side of the tail fin is completely lacking any reflectance samples. Furthermore, in all cases the setup has geometrically not been calibrated as precisely as in our measurements. This leads to holes in the 3D reconstruction of fine details and registration errors in the projected reflectance textures Nonetheless, this shows that our proposed approach is sufficiently robust to obtain a convincing appearance representation despite the disadvantageous circumstances.

Figure 5.10: *Pictures of digitized objects (in the order of first digitization). The shown pictures are taken from the set of BTF measurement images. There is, however, one exception: due to a hard drive defect, the measurement data of the Shoe object was lost shortly after processing. A rendering of the digitized object is shown as a substitute. Please refer to Table 5.1 for more details on the digitized objects. The depictions in this figure are not true to scale. Please refer to Figure 1.2 to get an impression of the relative dimensions of the objects.*

dataset	**dimensions**[7] [cm×cm×cm]	**apparent materials**	**setup**	**focal length**[6] [mm]	**resolution** [DPI]	**projectors** #	**HDR**[1] #	**acquisition time**[1,8] [hours]	**size** [GB]
Donkey	$10 \times 4 \times 17$	specular brass	Dome 1[2]	52	225	5	3/4	2:49 / 1:55	368
			Dome 2	95	190	4×8	2/3	3:18 / 4:50	850
Minotaur	$4.5 \times 4.5 \times 10$	bronze, green paint, marble	Dome 1[2]	61	264	7	4/4	2:45 / 2:56	326
Terracotta Soldier	$6.4 \times 7.5 \times 22$	black terracotta	Dome 1[2]	38	165	7	2/2	1:45 / 1:03	172
Buddha	$6.7 \times 11.8 \times 13.3$	red paint, gold leaf, wood	Dome 1[2]	61	264	8	3/3	2:41 / 1:30	245
Strawberry	$3.6 \times 4.6 \times 4.4$	strawberry skin and leafs	Dome 1[3]	104	450	8	2/2	0:46 / 0:57	279
Pudding Pastry	$14 \times 14.6 \times 3.3$	pastry, sugar-coating, vanilla pudding	Dome 1[3]	80	346	8	3/3	1:12 / 1:21	373
Apple	$5 \times 4.7 \times 7.5$	apple skin and flesh	Dome 1[3]	80	346	8	3/3	1:14 / 1:21	398
Almond Horn	$11.5 \times 11 \times 2.3$	almonds, pastry, chocolate	Dome 1[3]	104	450	8	3/4	1:12 / 2:14	509
Crispy Roast Pork	$15.4 \times 13.3 \times 5$	pork, bacon, crust	Dome 1[3]	80	346	8	3/3	1:00 / 1:22	300
Billiard Ball	$5.7 \times 5.7 \times 5.7$	red, back and white phenolic	Dome 2	?[4]	?[4]	?[4]	?[4]	?[4]	?[4]
Santa	$8.7 \times 8.4 \times 17$	mixed glossy paints	Dome 2	95	190	4×8	4/4	3:59 / 6:10	?[4]
Psoriasis Moulage	$26.8 \times 13 \times 5.5$	wax, paint, fabric, lacquered wood, paper	Dome 2	95	190	4×12	3/3	3:07 / 7:02	1,141
Chess Piece	$6 \times 3.5 \times 9.6$	resin, matte white finish	Dome 1	80	346	8	1/3	0:26 / 1:29	255
			Dome 2	95	?[4]	?[4]	?[4]	?[4]	?[4]
Tennis Ball	$6.5 \times 6.5 \times 6.5$	synthetic fabric (fluorescent)	Dome 2	95	190	4×8	4/3	1:10 / 7:08	1,073
Shoe	$13.4 \times 15 \times 7.8$	synthetic fabric, rubber, plastic	Dome 2	95	190	4×8	?[4]	?[4]	?[4]
Mug	$11 \times 10 \times 11$	ceramics	Dome 2	95	190	4×8	2/3	0:50 / 2:52	862
Ganesha	$3.5 \times 5 \times 7$	labradorite	Dome 2	190	380	4×8	3/4	1:47 / 9:12	1,122
Paintbrush	$5.3 \times 18.5 \times 8$	lacquered wood, metal, bristles	Dome 2	95	190	4×8	3/5	1:01 / 8:27	1,420
Micrometer	$5.3 \times 14 \times 2.6$	polished and rough metal, plastic	Dome 2	95	190	4×8	3/5	2:04 / 16:21	1,447

Table 5.1 – continued on next page

Table 5.1 – continued from previous page

dataset	**dimensions**[7] [cm×cm×cm]	**apparent materials**	**setup**	**focal length**[6] [mm]	**resolution** [DPI]	**projectors** #	**HDR**[1] #	**acquisition time**[1,8] [hours]	**size** [GB]
Fish[5]	$10.3 \times 15 \times 26.4$	gold and red paint	Dome 2	95	190	4×8	$3/4$	1:44 / 7:34	1,236
Inkwell[5]	$71 \times 13.3 \times 7.5$	silver	Dome 2	95	190	4×8	$2/4$	0:51 / 5:25	1,043
Teal[5]	$31 \times 16 \times 19$	feathers, beak, green paint	Dome 2	95	190	4×8	$2/2$	1:13 / 4:22	724
Epithelioma Moulage	$16 \times 11.4 \times 5.5$	wax, paint, fabric, lacquered wood, paper	Dome 2	95	190	4×17	$3/4$	5:21 / 13:20	1,498
Pyramid	$7.5 \times 8 \times 6.8$	clay, patina	Dome 2	190	380	4×24	$1/2$	1:39 / 3:01	732
Ammonite 1	$13.4 \times 15 \times 5.5$	fossil	Dome 2[3]	95	190	4×8	$2/3$	1:04 / 6:24	923
Ammonite 2	$4.2 \times 3.4 \times 1.1$	ammolite	Dome 2[3]	190	380	4×8	$3/5$	3:20 / 20:13	1,434
Rhinoceros Teeth	$8.1 \times 5.2 \times 4.3$	fossil	Dome 2[3]	190	380	4×8	$2/3$	2:19 / 4:57	887

[1]Geometry acquisition / reflectance acquisition. [2]Manual projector positioning. [3]Manual projector switching. [4]Original measurement data damaged. [5]Scanned at the Brighton Exhibition [3D-12]. [6]$35\ \mathrm{mm}$ equivalent focal length. [7]Estimated via the bounding-box diagonal of the reconstructed mesh. [8]Not including projector warm-up.

Table 5.1: *List of digitized objects (in the order of first digitization). In case of the Dome 2 setup, the number of employed projectors is given as projectors × rotations. For some objects, the measurement data was damaged shortly after processing due to hard drive failures, making it impossible to fill out all cells. Please refer to Figure 5.10 for pictures of the listed objects.*

5.6.1 Geometry Reconstruction

Reconstruction of the shape using the structured light approach worked well on almost all objects. We observed that the decision to employ high dynamic range imaging to capture the structured light images was well justified (see Figure 5.4) and led to a remarkably robust and complete detection of the projector pattern in the camera images. Finally, point clouds with about 250,000 to 22,000,000 points on the object surface were obtained. The point clouds show an average density of about 300 points per mm^2 for measurements conducted with the Dome 1 setup and 100 points per mm^2 for the Dome 2. To fully exploit the quality of the point clouds, we employed the Poisson surface reconstruction with an octree depth of nine to eleven, depending on the dimension and geometric details of the respective objects. Finally, we simplify the resulting triangle meshes to contain between 50,000 and 100,000 triangles. We assess the quality of the simplified mesh by computing the RMSE to the measured point cloud. Exact numbers can be found in Table 5.2. The RMSE values indicate that the error introduced by the simplification is still below the size of one millimeter. We deem this very acceptable as structures at this size are located in the mesoscopic scale anyway.

There were only two noteworthy obstacles we encountered during reconstruction: First, the strong subsurface scattering on the Ganesha and Strawberry led to a rather noisy point cloud. We also observed a bias towards points lying slightly beneath the surface. We manually corrected for these effects by choosing an outbound offset for the isosurface during surface reconstruction and applying a more aggressive smoothing. Second, the thin and translucent leafs of the Strawberry made the patterns appear clearly visible on both sides of the leafs. As a result, points were triangulated by cameras on both sides, becoming located in the middle of the leaf. We manually accounted for this by orienting the normals of the points towards the respective projectors. Then we slightly shift the points outward along the normal.

5.6.2 Parameterization, Projection and Resampling

We employ the freely available `Graphite`[2] toolkit to perform the ABF++ parameterization and texture atlas generation. The generated atlases occupy between 30% and 70% of their bounding rectangles, i.e. only about half of the BTF texels will have meaningful reflectance values. This ratio could be improved by using better texture atlas layouting algorithms, e.g. Nöll *et al.* [NS11]. However, we did not investigate this issue any further, as the advanced compression, streaming

[2] `http://alice.loria.fr/index.php/software.html`

data set	geometry reconstruction & parameterization				HDR combination, projection & data rearrangement				resampling & hole filling			total
	triangles #	surface area [mm^2]	RMSE [mm]	size [MB]	texture W×H	texels #	reso. [DPI]	time[1] [hours]	clusters #	time [hours]	size [GB]	time [hours]
Donkey (Dome 1)	71,907	23,968	0.09	6.78	2048 × 2048	1,834,814	222	2:06 + 0:55	1	9:34	233	12:35
Donkey (Dome 2)	71,952	24,184	0.18	6.68	2048 × 2048	1,297,628	186	4:28 + 1:16	1	8:54	165	14:38
Minotaur	49,999	8,389	0.09	4.66	1600 × 1600	1,298,994	316	1:45 + 1:05	3	7:33	165	10:23
Terracotta Soldier	99,952	42,118	0.01	9.36	2048 × 2048	2,204,273	183	2:11 + 2:42	1	10:24	280	15:17
Buddha	49,999	28,561	0.10	4.58	3072 × 3072	3,038,700	262	3:45 + 1:56	2	17:14	387	22:55
Strawberry	60,000	5,006	0.22	5.61	1600 × 1600	1,227,850	397	1:43 + 0:40	2	6:03	156	8:26
Pudding Pastry	53,380	24,315	0.34	4.84	2560 × 2560	4,447,916	343	3:30 + 3:25	3	25:03	566	31:58
Apple	49,984	8,198	0.09	4.64	1600 × 1600	1,314,710	322	1:54 + 0:44	1	6:55	167	9:33
Almond Horn	77,312	14,080	0.44	7.09	2048 × 2048	2,195,611	317	2:10 + 1:53	2	11:54	279	15:57
Crispy Roast Pork	74,978	22,631	0.19	6.85	2560 × 2560	3,926,807	334	2:58 + 3:56	1	19:44	500	26:38
Billiard Ball[2]	79,600	10,142	?[3]	8.50	256 × 256	65,176	64	?[3]	1	?[3]	8	?[3]
Santa[2]	199,999	38,885	0.38	18.70	512 × 512	130,960	46	?[3]	3	1:44	16	?[3]
Psoriasis Moulage	74,963	49,629	0.16	7.05	2560 × 2560	2,933,812	195	7:59 + 3:06	4	23:46	373	34:51
Chess Piece (Dome 1)	99,992	12,442	0.04	9.50	2048 × 2048	1,829,682	308	2:18 + 0:57	1	9:37	233	12:52
Chess Piece (Dome 2)[2]	500,000	13,535	?[3]	49.87	1024 × 1024	470,615	149	?[3]	1	3:12	60	?[3]
Tennis Ball[2]	100,000	13,367	0.20	9.20	1024 × 1024	470,409	150	4:44 + 0:31	1	3:41	60	8:56
Shoe[2]	99,999	38,217	0.12	9.31	1024 × 1024	555,139	97	?[3] + 0:57	3	4:19	71	?[3]
Mug	49,986	54,172	0.17	4.58	2048 × 2048	2,534,347	173	6:23 + 2:16	1	20:18	322	28:57
Ganesha	34,953	9,434	0.51	3.16	2048 × 2048	2,124,918	381	6:19 + 1:59	1	14:21	270	22:39
Paintbrush	9,996	7,732	0.60	0.87	1024 × 2048	401,282	183	6:23 + 0:38	4	3:17	51	10:18
Micrometer	37,161	6,654	0.15	3.35	1024 × 2048	484,563	216	6:06 + 0:37	3	3:48	62	10:31
Fish	74,968	65,120	0.47	6.99	3072 × 3072	3,110,228	176	11:19 + 3:17	2	21:36	396	36:12
Inkwell	41,324	16,288	0.56	3.79	1600 × 1600	737,149	171	5:22 + 0:57	1	4:22	94	10:41
Teal	74,770	126,068	0.51	7.26	4096 × 4096	7,092,675	191	17:57 + 7:24	3	49:27	903	74:48
Epithelioma Moulage	49,958	27,097	0.18	4.49	1600 × 1600	1,775,179	206	6:05 + 1:49	4	14:18	225	22:12
Pyramid	49,975	12,648	0.06	4.53	2560 × 2560	3,094,942	397	8:47 + 3:01	1	20:55	394	32:43
Ammonite	99,412	25,500	0.39	9.12	1600 × 1600	1,806,928	214	5:16 + 1:47	1	12:33	231	19:36
Ammonite 2	24,976	1,525	0.17	2.15	800 × 800	440,023	431	5:54 + 0:35	1	2:59	56	9:28
Rhinoceros Teeth	49,891	8,398	0.10	4.49	1600 × 1600	1,655,700	357	5:09 + 1:34	1	11:18	210	18:10

[1]HDR combination and projection + rearrangement of the data on disk for improved access. [2]Inferior or deviating quality due to data loss. [3]No information available due to data loss.

Table 5.2: *Results of the proposed postprocessing. The spatial dimension W×H of the BTFs have been chosen, such that the resolution (denoted in column "*reso.*") comes close to the optical resolution of the employed acquisition system (see Table 5.1). The resolution depends on the number of occupied texels in the texture map (column "*texels*") and the surface area of the triangle mesh. The file sizes refer to the resulting 3D mesh and uncompressed BTF, respectively. All timings are measured on the system described in Section 5.6.4.1.*

and rendering techniques presented in chapters 6 and 7 will mostly ignore the empty parts. To facilitate efficient real-time rendering, we employ GPU friendly texture resolutions, such as 2048×2048 or 4096×4096 texels, for the full lattice. Furthermore, we choose the dimension of the texture atlas such that it exploits the available resolution of the measurement setup as good as possible. The exact number of texels, the resulting average resolution as well as other processing parameters and timings can be found Table 5.2.

Our proposed resampling approach produces overall reasonable results when restricting the virtual light source and camera to the hemisphere of measurement. However, we observed that for extreme extrapolations outside the measured data, changes in color and specularity can occur (see Figure 5.11). One possible work-around might be taking additional measurements of the object turned upside down. This avoids the complete lack of data for large parts of the local hemispheres during resampling and hole filling. A full spherical setup, as proposed by Köhler *et al.* [KNRS13], would also be a viable solution. Another avenue of future research could be to develop an improved hole filling that is capable of handling these cases more robustly, e.g. based on the tensor fitting approach we proposed together with Ruiters *et al.* in [RSK12].

5.6.3 Compression

All models were compressed with FMF and DFMF, applying the proposed dynamic range reduction first. We employed very cautious quality setting. In the case of the FMF, we keep 100 components for $\mathbf{B}^{\log}$. For the DFMF, we keep 100 components for $\mathbf{B}^{\log \mathrm{Y}}$ and 50 for $\mathbf{B}^{\mathrm{U/Y}}$ and $\mathbf{B}^{\mathrm{V/Y}}$ each. Using this data, we are able to create path traced novel images of arbitrary scenes in very high quality (see figures 1.2, 2.2b and 5.9) or provide interactive real-time viewing on the GPU, still depicting the objects photorealistically (see Figure 7.1). As we will demonstrate in Chapter 6, this representation even facilitates web-based dissemination with moderate bandwidth requirements.

As the angular sampling is the same for all objects, the achieved compression ratios depend only on the spatial resolution of the BTF. By employing the described FMF compression on all occupied texels of an uncompressed BTF, we could in principle achieve compression ratios between 1:321 (Billiard Ball) and 1:676 (Teal). With DFMF compression, which in our case stores exactly twice as many components, the compression ratios would be between 1:160 and 1:338.

However, we directly want to use the compressed BTF data in GPU-based rendering. For this, we also store the components for the unoccupied texels of the texture maps. This omits the necessity for an additional indirection during texture lookup

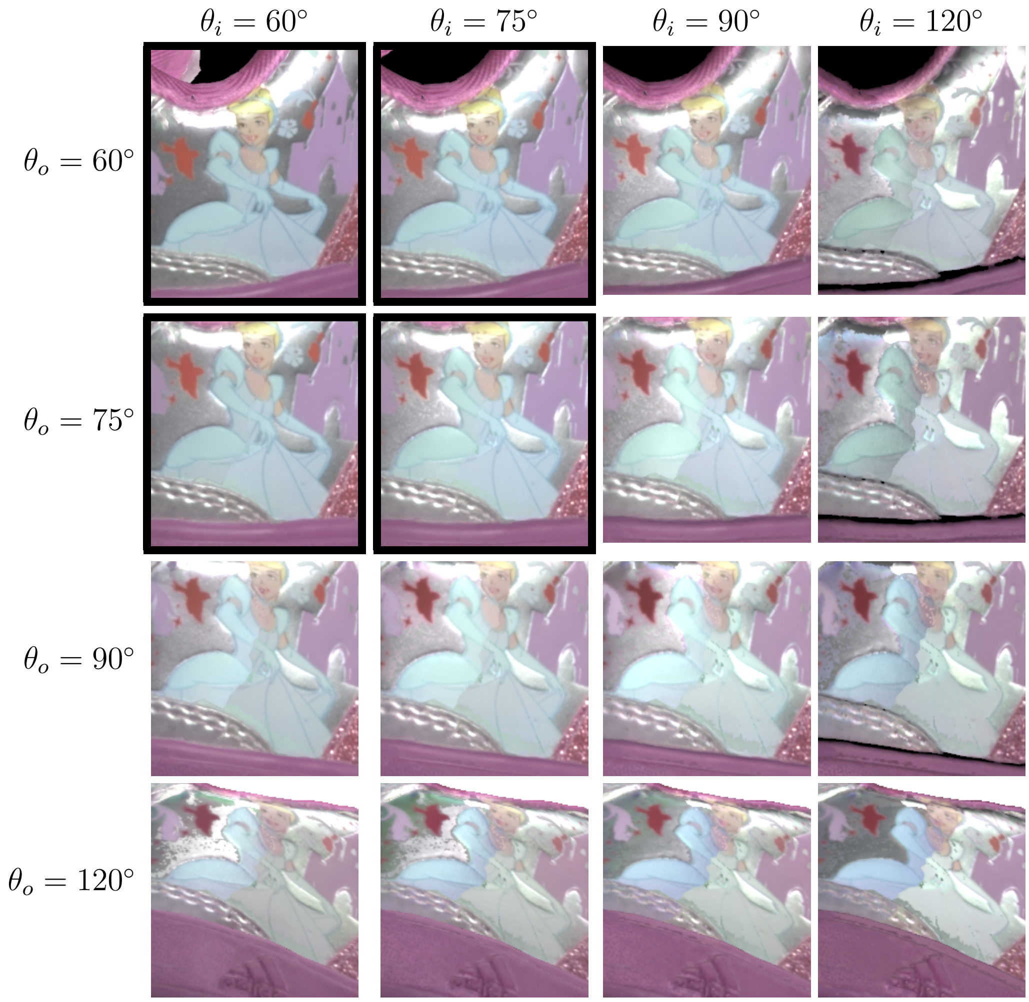


Figure 5.11: *Renderings of the Shoe object with extrapolation beyond measured angles. Only samples for inclination angles up to $\theta = 75°$ with respect to the upright shoe were actually measured (framed images). However, the data is extrapolated during the resampling and hole filling step to bring the data into a regular grid in the local coordinate systems, such as here on the depicted side of the shoe. As a result, the hue of the princess' dress erroneously shifts to a more greenish tone for larger θ_i and more to blueish tone with increased gloss for larger θ_o. Due to a vertical seam in the parameterization and the resulting difference in the local tangent directions on both sides, the hue shift even diverges for the left and right sides of the dress.*

and allows to use the linear interpolation of the GPU's texture mapping units, because the texels of the spatial components are arranged in the correct spatial layout. As a consequence, the compression ratio also depends on the utilization of the texture atlas. The real compression ratios lie between 1:192 (Inkwell) and 1:471 (Ammonite) for FMF compression and thus between 1:96 and 1:235 for the DFMF.

We perform a series of numerical comparisons to quantify the quality obtained from BTFs compressed with FMF and DFMF compression. First, we compare the relative L^2 error — a standard measure of signal fidelity. For FMF compression this error is defined as

$$\frac{\|\mathbf{B} - \mathbf{U}'\,\mathbf{V}'\|_F}{\|\mathbf{B}\|_F}. \tag{5.9}$$

For DFMF compression, we use the same measure,

$$\frac{\sqrt{\|\mathbf{B}^{\mathrm{Y}} - \mathbf{U}'^{\mathrm{Y}}\,\mathbf{V}'^{\mathrm{Y}}\|_F^2 + \|\mathbf{B}^{\mathrm{U}} - \mathbf{U}'^{\mathrm{U}}\,\mathbf{V}'^{\mathrm{U}}\|_F^2 + \|\mathbf{B}^{\mathrm{V}} - \mathbf{U}'^{\mathrm{V}}\,\mathbf{V}'^{\mathrm{V}}\|_F^2}}{\sqrt{\|\mathbf{B}^{\mathrm{Y}}\|_F^2 + \|\mathbf{B}^{\mathrm{U}}\|_F^2 + \|\mathbf{B}^{\mathrm{V}}\|_F^2}}, \tag{5.10}$$

i.e. the root of the sum of squared errors of all entries divided by the Frobenius norm of the original signal. Due to the use of dynamic range compression, the algorithms actually operate on the matrices $\mathbf{B}^{\log}$ or $\mathbf{B}^{\log \mathrm{Y}}$, $\mathbf{B}^{\mathrm{U/Y}}$ and $\mathbf{B}^{\mathrm{V/Y}}$, respectively. During rendering, however, the quality depends on the reconstruction of the original linear BTF data. We thus compute the relative L^2 error for both. In all cases, the FMF compression provides the best approximation with respect to this error metric. In logarithmic space, the the error lies between 1.4% for the clay Pyramid, which is mostly diffuse, and 7.1% for the silver Inkwell, which is extremely specular. In linear space, we obtain errors of 2.9% (Pyramid) up to 132.6% (Minotaur). For the DFMF compression, the best DRC space value is 2.5% for the diffuse Terracotta Soldier whereas the worst value is 33.9% for the Billiard Ball. In linear space, values range from 10.8% (Terracotta Soldier) to 112.3% (Minotaur).

Using only the L^2 measure, the DFMF compression does not seem to provide much benefit. However, as argued in Section 3.8, the error perceived by a human observer might differ from the prediction of the L^2 error. We thus conduct a second test series, for which we use the perceptually motivated SSIM index (see Section 3.8) as an indicator of perceived similarity. We evaluate the SSIM on renderings showing the objects from six different viewing directions and lit from six different lighting directions. The chosen directions and example renderings for each combination are shown in Figure 5.13. Here, both compression techniques achieve similar values between 0.962 (Inkwell) and 0.997 (Pyramid). The appearance is in any case perceptually extremely close to the that of the uncompressed BTF, since a SSIM value of 1 denotes complete similarity.

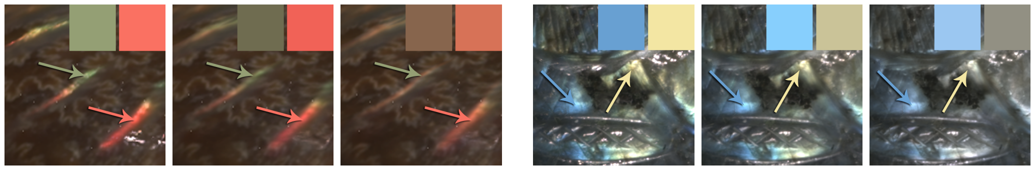


Figure 5.12: *Direction-dependent color preservation of the two utilized BTF compression algorithms The images depict details of renderings used for the SSIM comparison in Table 5.4. The first three depict the Ammonite 2, the other three the Ganehsa. Both objects show direction-dependent iridescence appearance, i.e. the colors change in dependence of the light and viewing angles. For the purpose of illustration, all images show two insets of colors found in the iridescent regions. The exact positions are highlighted by arrows. While the renderings with both compression techniques show slightly washed-out colors, the DFMF compression manages to preserve the general hues considerably better. In case of the Ammonite, the FMF compression even changes the hue to a completely different color.*

Quantitatively, the FMF compression is often an equally good or better choice than DFMF. Still, in Figure 5.12, we demonstrate that renderings using DFMF compressed BTFs achieve a considerably better reproduction of direction-dependent shifts in color than renderings with FMF compressed BTFs. This is probably not reflected in the SSIM index, because it is computed on monochrome images.

Please refer to tables 5.3 and 5.4 for more details on the required compression time and achieved size and quality for the individual objects. Additionally, qualitative comparisons are shown figures 5.18 and 5.16.

5.6.4 Discussion

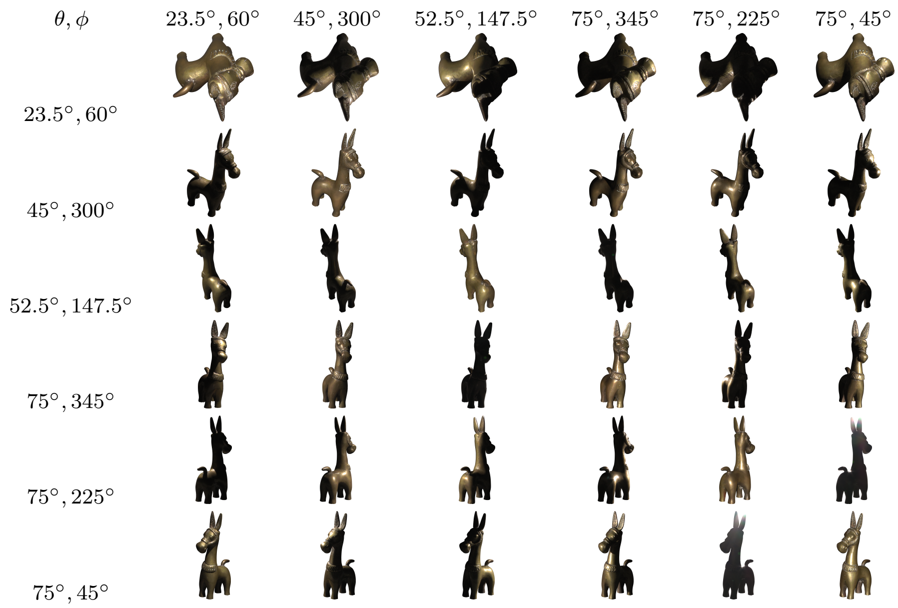
In the following, we discuss the computational costs, achievable resolution and limitations of the proposed processing pipeline.

5.6.4.1 Performance

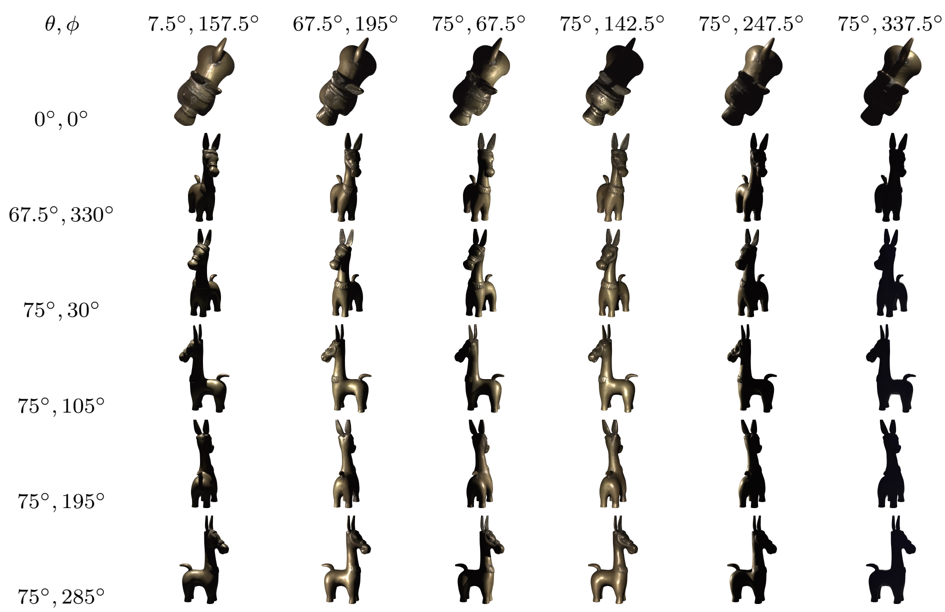
All timings have been measured on a system with two Intel Xeon E5-2650 CPUs (total of 16 cores with 2 GHz each), 128 GB RAM and an NVIDIA GeForce GTX 680 GPU. The raw input images were directly accessed from a storage server via gigabit Ethernet and the intermediate and final data was stored on a local RAID volume. It is very apparent from Table 5.2 that the overall processing times can vary greatly, depending on the desired spatial resolution and on the number of available

data set	FMF compression					DFMF compression				
	time [hours]	size [MB]	${L^2}_{\log}$ %	${L^2}_{\mathrm{lin}}$ %	SSIM	time [hours]	size [MB]	${L^2}_{\log}$ %	${L^2}_{\mathrm{lin}}$ %	SSIM
Donkey (Dome 1)	2:07	821	**2.8**	**40.2**	0.991	3:48	1,633	9.9	46.5	0.991
Donkey (Dome 2)	2:03	821	**1.6**	**18.5**	0.996	3:25	1,633	7.9	32.1	0.996
Minotaur	1:20	506	**3.2**	132.6	0.986	2:17	1,000	4.1	**112.3**	0.986
Terracotta Soldier	2:09	821	**1.9**	**8.6**	0.996	4:14	1,633	2.5	10.7	0.996
Buddha	2:56	1,831	**3.2**	**12.7**	0.990	5:53	3,663	20.9	39.4	**0.991**
Strawberry	1:19	506	**3.7**	**15.1**	**0.974**	2:20	1,000	33.0	64.9	0.973
Pudding Pastry	3:28	1,276	**4.0**	**21.9**	0.979	6:41	2,546	22.1	51.1	0.979
Apple	1:16	506	**1.7**	**8.5**	0.992	2:03	1,000	24.8	36.1	0.992
Almond Horn	2:06	821	**3.7**	**26.3**	0.981	3:45	1,633	15.8	45.6	0.981
Crispy Roast Pork	3:21	1,276	**2.5**	**6.7**	0.991	5:34	2,546	20.9	30.1	**0.992**
Billiard Ball	0:03	26	**2.7**	**80.1**	**0.997**	0:03	34	33.9	90.7	0.978
Santa	0:08	64	**1.0**	**17.7**	0.993	0:13	110	2.0	22.0	**0.994**
Psoriasis Moulage	3:42	1,276	**2.9**	**8.5**	0.993	7:38	2,546	7.9	19.5	0.993
Chess Piece (Dome 1)	2:00	821	**1.9**	**3.7**	0.996	3:49	1,633	14.9	28.1	0.996
Chess Piece (Dome 2)	0:33	215	**1.8**	**3.6**	0.995	1:01	415	10.8	23.1	0.995
Tennis Ball	0:32	215	**3.2**	**18.8**	**0.971**	0:54	415	12.4	39.0	0.968
Shoe	0:35	215	**3.5**	**44.8**	0.986	1:04	415	14.0	46.5	0.986
Mug	2:06	821	**1.5**	**5.5**	0.996	3:51	1,633	7.1	14.5	0.996
Ganesha	2:03	821	**3.2**	67.2	0.976	3:18	1,633	3.8	67.2	0.976
Paintbrush	0:58	417	**4.7**	**62.8**	0.970	1:36	821	12.4	63.3	0.970
Micrometer	0:57	417	**3.9**	**34.7**	0.979	1:36	821	4.0	35.0	0.979
Fish	5:05	1,831	**3.4**	**21.1**	0.990	8:33	3,663	16.0	35.3	0.990
Inkwell	1:16	506	**7.1**	**65.9**	0.962	2:06	1,000	7.5	69.3	0.962
Teal	8:24	3,245	**2.5**	**7.2**	0.994	13:43	6,505	5.4	11.6	**0.995**
Epithelioma Moulage	1:24	506	**1.8**	**6.6**	0.995	2:35	1,000	4.3	15.7	0.995
Pyramid	3:08	1,276	**1.4**	**2.9**	0.997	5:13	2,546	5.7	12.1	0.997
Ammonite	1:22	506	**2.4**	**5.4**	0.994	2:20	1,000	6.0	16.2	0.994
Ammonite 2	0:21	136	**3.4**	**77.3**	0.973	0:36	257	10.5	79.1	**0.974**
Rhinoceros Teeth	1:23	506	**2.0**	**4.1**	0.995	2:38	1,000	7.9	20.7	0.995

Table 5.3: *Results from compressing the processed BTFs. All FMF compressed BTFs use $C = 100$ components. All DFMF compressed BTFs use $C = 100$, $C = 50$ and $C = 50$ components for $\log Y$, U/Y, V/Y respectively. The relative L^2 errors in columns ${L^2}_{log}$ are computed in the respective dynamic range compressed matrices $\mathbf{B}^{\log}$ or $\mathbf{B}^{\log Y}$, $\mathbf{B}^{U/Y}$ and $\mathbf{B}^{V/Y}$. Analogously, the values in columns ${L^2}_{lin}$ are computed in linear RGB color space with respect to the entries of the original resampled BTF matrix $\mathbf{B}$. The SSIM is evaluated on renderings from 6×6 directions (see Figure 5.13), depicting the complete object at the resolution of the original input images. The numerically better numbers are printed in bold. Despite the higher compression ratio, the FMF compression shows a lower relative RMSE. However, the DFMF compression provides a slightly better SSIM value in some cases. All timings are measured on the system described in Section 5.6.4.1.*



(a) Dome 1



(b) Dome 2

Figure 5.13: *The 6×6 direction combinations employed to evaluate the SSIM. The directions were chosen to provide an evenly spaced sampling of the view and light hemispheres. Rows correspond to view directions, columns to light directions. For the Dome 2 case (b), the light angle ϕ_i is given with respect to the fixed light hemisphere. The angle with respect to the object is obtained as $\phi_i + \phi_o$.*

	BTF			**alternative approaches**			**silhouette**
data set	uncomp.	DFMF	FMF	tensor fit	SVBRDF	texture	
							
Donkey (Dome 1)	**0.707**	0.699	0.699			0.658	0.512
Donkey (Dome 2)	**0.711**	0.710	0.710			0.663	0.525
Minotaur	**0.784**	0.775	0.775			0.733	0.553
Terracotta Soldier	**0.612**	0.608	0.608			0.570	0.343
Buddha	**0.663**	0.650	0.651	0.584	0.579	0.542	0.393
Strawberry	**0.583**	0.564	0.562			0.504	0.317
Pudding Pastry	**0.643**	0.630	0.629			0.543	0.229
Apple	**0.674**	0.665	0.666			0.637	0.327
Almond Horn	**0.701**	0.686	0.686			0.582	0.267
Crispy Roast Pork	**0.698**	0.670	0.671			0.574	0.303
Billiard Ball	0.668	0.673	0.671	**0.684**	0.670	0.676	0.303
Santa	**0.629**	0.628	0.629	0.598	0.581	0.563	0.401
Psoriasis	**0.729**	0.725	0.725			0.682	0.430
Chess Piece (Dome 1)	**0.611**	0.606	0.606			0.564	0.312
Tennis Ball	**0.349**	0.354	0.355			0.326	0.183
Mug	**0.504**	0.506	0.506			0.498	0.208
Ganesha	**0.496**	0.484	0.484			0.417	0.245
Paintbrush	**0.827**	0.817	0.817			0.759	0.531
Micrometer	**0.742**	0.736	0.736			0.647	0.443
Fish	**0.683**	0.677	0.677			0.619	0.514
Inkwell	**0.647**	0.642	0.642			0.600	0.485
Teal	**0.744**	0.742	0.742			0.712	0.477
Epithelioma	**0.775**	0.769	0.769			0.726	0.478
Pyramid	**0.732**	0.731	0.731			0.658	0.368
Ammonite	**0.714**	0.707	0.707			0.613	0.303
Ammonite 2	**0.788**	0.784	0.783			0.742	0.312
Rhinoceros Teeth	**0.649**	0.642	0.642			0.581	0.338

Table 5.4: *Quantitative comparison of appearance reproduction. We give the SSIM index between measured radiance maps and ray traced HDR renderings with different material appearance reproduction techniques. For the comparison, all images are tone-mapped with gamma correction using $\gamma = 2$. The numerically better numbers are printed bold. For all objects except for the Billiard Ball, uncompressed BTFs yield the largest similarity to the reference images, followed by DFMF and FMF compressed BTFs. The presented figures are the arithmetic mean based on 6×6 different combinations of light and view directions. All employed direction combinations are depicted in Figure 5.13. The data-driven tensor fit representation and the analytical Cook-Torrance SVBRDF data sets used in this comparison are depicted in Figure 5.16 and described in [RSK12]. Comparisons with the objects' silhouettes are given as a baseline for judging the reported SSIM index values. The silhouettes are purely black-and-white and were also used to mask out the background in the measured radiance maps.*

input direction samples. The total processing time, excluding compression, lies between eight and 75 hours. This is substantially slower than the acquisition of the data. Processing is thus currently the major bottleneck of the proposed data-driven digitization approach.

The projection step is limited by GPU, disk and network throughput. The execution time therefore mostly depends on the number of input directions $|\mathfrak{L} \times \mathfrak{V}|$ and the selected texture map resolution. The data rearrangement is purely dominated by disk operations. Since we employ a buffer in RAM that only holds the non-zero entries of the texture map, the execution time depends on the number of non-zero texels $|\mathfrak{X}|$ and the number of input direction. Thus, the biggest remaining issue is the massive amount of data which makes the use of out-of-core algorithms mandatory. However, we expect this to become more manageable in the future, with fast solid-state drives with sufficiently high storage capacities becoming affordable.

During resampling and hole filling, several computationally demanding operations have to be performed, making the CPU the bottleneck. Most computations are performed per output direction pair for each texel. As a result, the timings depend on $|\mathfrak{D}|$ and $|\mathfrak{X}|$.

The compression timings depend on the number of matrix entries, i.e. on $|\mathfrak{D}|$ and $|\mathfrak{X}|$. The runtime characteristics of the employed GPU accelerated compression algorithm are in detail investigated in the original publication by Ruiters *et al.* [RRK09]. The additional dynamic range compression employed in this thesis has some influence on the overall runtime, because the proposed logarithm and division are both among the more costly arithmetic operations.

In our experiments, the DRC led to an increase in compression times of about 38% for the FMF and 24% for the DFMF method. Similarly, rendering times are increased due to the additional costly operation of taking the exponent. In our CPU implementation, which was used to create the path traced images and the SSIM evaluation renderings, we perform a full hexalinear interpolation from 36 samples[3]. Here, the increase in rendering time due to the DRC is about 5% to 8% for both methods. In our real-time rendering implementation, the angular and spatial interpolation is performed separately and directly on each component by the GPU's texture mapping units (see Section 6.3). The exponent has to be taken only once per sample reconstruction. As a result, we did not observe any significant influence of the DRC on the achieved frame rate, which is mostly limited by the large amount of texture accesses.

[3]Bilinear spatial interpolation from 4 points $\times$ view direction interpolation from 3 barycentric coordinates $\times$ light direction interpolation from 3 barycentric coordinates.

All three time-consuming steps, i.e. projection, resampling and compression, are well parallelized and should scale easily to server farms or cloud computing services. However, in-between projection and resampling, the data rearrangement step requires access to all data. Similarly, although the ABRDF basis can be partially computed on different machines, the final basis $\mathbf{U}'$ has to be merged from the partial results and then redistributed. Thus, in such an acceleration scenario the processing computers would need to be connected with high bandwidth and synchronized twice. This renders the option to use a cloud computing service impractical at the moment.

5.6.4.2 Resolution

The total achievable physical accuracy in terms of geometry and BTF resolution is object-dependent, since we utilize different focal lengths to cover different sized objects. Table 5.1 lists the focal length employed for each object and gives the achievable resolution in DPI, considering the camera resolution and distance.

It is difficult to assess the overall accuracy of the simplified 3D geometry, since we do not have any ground truth measurements for our test objects. In [WSRK11], we evaluated the accuracy for the point clouds and found that our chosen structured light approach is able to achieve a very low RMSE of 23.3 $\mu\mathrm{m}$ on a test object with ground truth geometry. The measurements for this test were obtained with the Dome 1 and the Acer C20 projector, so they should at least be transferable to the opaque objects considered in this thesis. Thus, we can at least provide an approximation by computing the RMSE of the simplified geometry with respect to the obtained point cloud (see Table 5.2).

The remaining mesoscopic geometry details that have been captured by the camera become part of the BTF. We aim to use the highest sensible number of DPI in order to depict the mesoscopic details as precise as possible. On the one hand, this number is limited by the resolution of the cameras. On the other hand, we also try to utilize "GPU friendly" texture resolutions for the texture atlas, i.e. quadratic textures with edge lengths of powers of two or at least multiples of 16. The reasons are improved performance of real-time rendering as well as the GPU accelerated projection step. Especially our WebGL renderer presented in Chapter 6, has strict requirements with respect to texture sizes. We therefore choose a compromise between exactly matching the input resolution and the desired texture properties. The exact resolution of the processed BTFs can be found in Table 5.2.

Considering for example the Buddha with dimensions of roughly 6.7 cm$\times$11.8 cm$\times$ 13.3 cm and a surface area of about 286 cm^2, the resolution of the cameras would allow for a depictions of details the size of 96 $\mu\mathrm{m}$. Maybe, this limitation of

Figure 5.14: *The Buddha with a DFMF compressed BTF rendered under point light illumination. The insets show detail enlargements to demonstrate how well mesoscopic surface and material features are preserved.*

camera resolution could even be relaxed by a superresolution approach. However, to resolve the 96 $\mu\mathrm{m}$ accuracy, we need a texture of at least 1760×1760 pixels resolution. Unfortunately, due to the low-distortion parameterization, which leads to a partially empty atlas, in practice we even require 3072×3072 pixels resolution for the texture map to exploit the full potential of the measurement setup. Figure 5.14 demonstrates the reproduction of fine mesoscale details on the Buddha.

Tables 5.2 and 5.3 demonstrate that this resolution already leads to very large uncompressed as well as compressed data sizes. These sizes could potentially be further reduced, e.g. by using a consecutive LZMA compression. However, we identified the large number of texels to be a major hurdle during processing and did thus not explore any further resolution enhancements. We instead focus on approaches to cope with the high amount of data. A more efficient compression and a solution for memory efficient rendering are discussed in chapter 6 and 7.

5.6.4.3 Limitations

We have to point out that there are certain limits to the classes of objects that can be captured with the chosen BTF approach. As demonstrated in Figure 5.15, transparent or translucent objects cannot be represented fully faithfully this way. Under strongly directional illumination, shadow boundaries appear to be too hard and shadowed areas appear to be too dark (see Figure 5.15a). Furthermore, thin

picture BTF rendering

(a) soft shadow boundaries

picture BTF rendering

(b) translucency

Figure 5.15: *Despite the realistic impression of subsurface scattering objects under light probe illumination (e.g. Figure 2.2), their appearance under strongly directed illumination is only insufficiently reproduced using the proposed method. In (a), the resampled BTF rendering does not show the soft shadow boundaries on the pudding. In (b), the translucency of the leafs and flesh of the fruit under back-light is not reproduced. In both cases, this leads to an artificial and unappetizing impression.*

objects that usually appear to be translucent, e.g. the leafs or the flesh of the Strawberry, are not correctly portrayed under back-light illumination (see Figure 5.15b). Yet, if additional light sources are present, e.g. when using a light probe, this lack of translucency is sufficiently hidden to produce convincing images (see Figure 2.2).

Similarly, the structured light reconstruction approach also restricts the possible classes of objects. In addition to the BTF constraints, objects exhibiting a perfect mirroring or strong subsurface scattering appearance cannot be captured as well. However, our proposed system is robust enough to reconstruct the geometry of objects that exhibit strong specularities and weak subsurface scattering. Optically complicated materials, e.g. the ammolite of the second Ammonite or the labradorite of the Ganesha, can be captured and faithful renderings under the distant illumination assumption can be generated from the BTF data.

The amount of specularity that can be reproduced with the proposed BTF approach is restricted by the number of direction samples of the measurement device as well as the direction resolution of the tabulated resampled data. This can for example be observed in Figure 5.16 in the comparison images of the Billiard Ball. First, if specular highlights are too narrow, they might have been been missed, depending on the sampled directions. However, the set of local directions is usually different for each spatial position (see Section 5.5.2). Therefore, the BTFs of more specular objects, such as the Donkey, the Ganesha or the Billiard Ball, exhibit a distinct splat pattern close to highlight directions (see Figure 5.17).

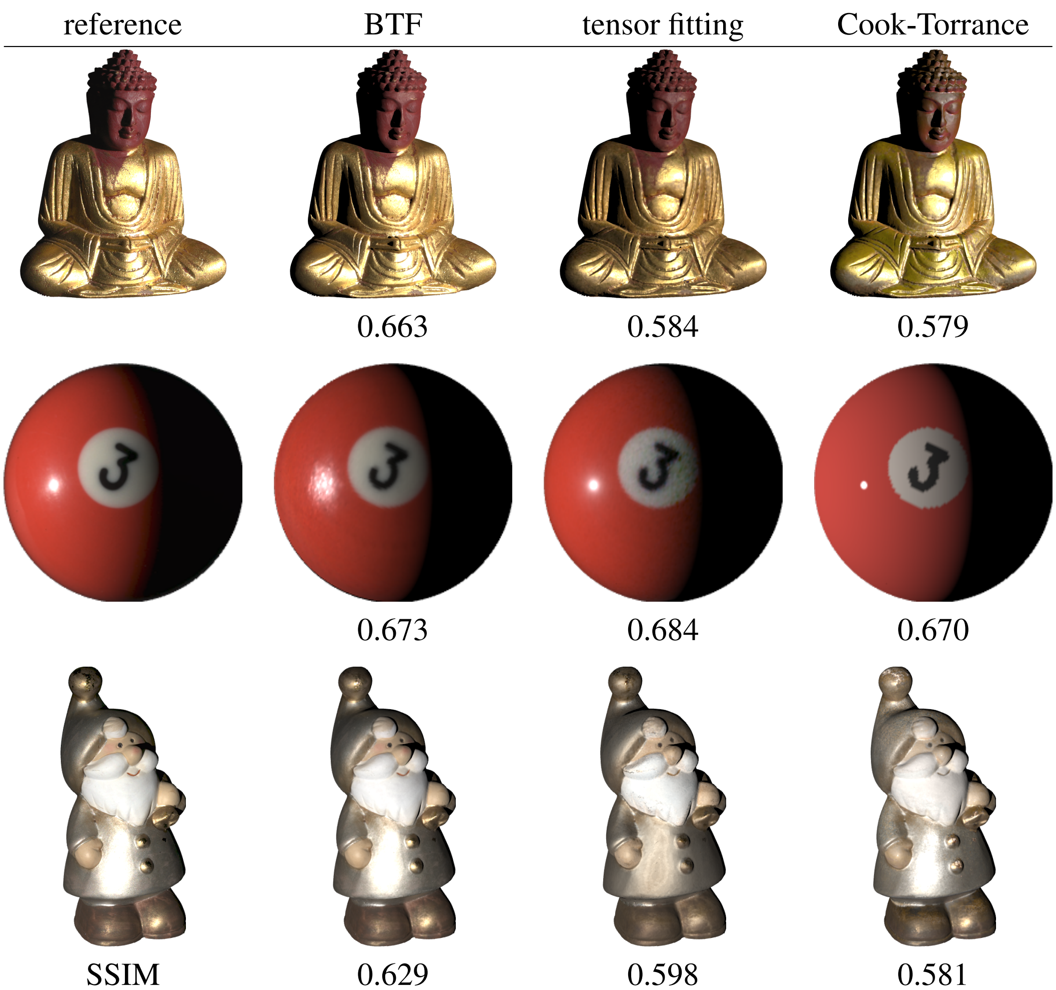
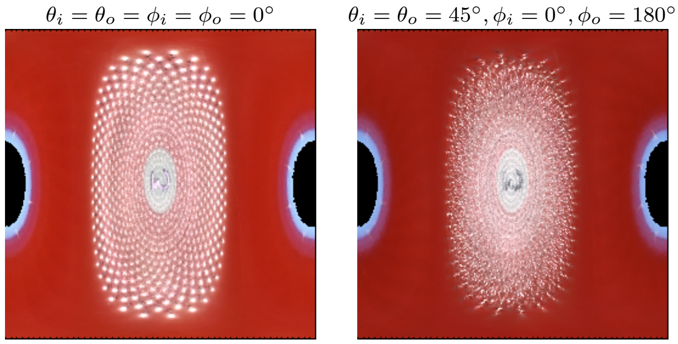


Figure 5.16: *Comparison between BTF, tensor fitting and SVBRDF. The data sets used for this comparison are those presented in [RSK12]. They show a lower spatial resolution than the BTFs used in this thesis. The Cook-Torrance SVBRDFs employ a spatially varying mixture of a basis of fitted Cook-Torrance BRDFs [CT82]. Please refer to [RSK12] for more details.*
The numbers below the renderings give the average SSIM index with respect to the reference (computed on 6×6 *directions, see Figure 5.13). For Buddha and Santa, the BTF provides the best reproduction of the reference appearance. However, the resampled BTF is not capable of accurately reproducing the sharp highlight on the Billiard Ball in the middle. Here, the also data-driven tensor fitting approach provides a more accurate reconstruction. In any case, the fitted Cook-Torrance SVBRDF distribution produces the least convincing results.*

(**a**) BTF rendering



(**b**) BTF slices

Figure 5.17: *Specularity reproduction issues due to the restricted number of captured direction combinations. Image (a) shows a rendering of the Billiard Ball under a uniformly bright illumination from all directions. The red material shows a clearly visible pattern of bright spots, although it should have a uniform appearance under this illumination. Image (b) depict slices through the BTF along the spatial dimension at specular reflection direction combinations. Here, the source of the bright spots is clearly identifiable as the inhomogeneity in the stored reflectance values, whereas the real billiard ball in fact exhibits a mostly homogeneous surface appearance.*

The pattern is mainly produced during resampling. In some texels a direction combination showing the highlight was captured by the measurement setup. Here, the neighborhood of the local highlight direction configuration contains an extremely high value due to the specular peak and influences the RBF interpolation accordingly. In other texels the highlight direction was simply not observed in any measurement image, resulting in a reflectance reconstruction from the interpolation of generally more diffuse samples.

The second limitation arises from the restricted number of direction samples in the resampled data. During rendering, values for intermediate directions need to be interpolated from the available data. In this thesis, we use a hexalinear interpolation. This results in a blurred appearance of highlights if the specular peak does lie in-between directions: It should be higher than the six samples, which is not modeled by the linear combination. As a consequence, global light transport simulation on the virtual replicas will also not be able to produce mirror-like reflections that might have been visible on the real object. These considerations obviously apply to any other effect that has a strong directional characteristic as well. Please note that this problem will also occur if rendering is performed purely image-based without prior resampling to a surface.

While we eliminate macroscale shadows and occlusions from the resampled data, macroscale interreflections are still captured within the BTF representation. Although this adds to the realistic impression when viewing a single object under local illumination, e.g. in a real-time viewer application, it introduces some systematic errors when rendering with global illumination. Path tracers and other global illumination algorithms simulate the macroscale interreflections as well, resulting in a duplicate integration of these light paths and thus a too bright appearance (see Figure 5.18). Using a setup with projectors as light sources, such interreflections can be eliminated at capture time (see [HLZ10]). Alternatively, this problem might also be handled in a postprocessing step by removing the geometry induced interreflections from the BTF. Another alternative would be to modify the path tracer to ignore indirect light contributions from other parts on the same object.

With respect to real-time rendering, the resulting file sizes from the utilized data-driven compression approach presents a limitation as well. Larger scenes consisting of multiple objects will not even fit into the memory of recent high-end GPUs. This could be tackled by converting the data into a more compact representation for such purposes. However, we instead propose to use a virtual texturing scheme, presented in Section 7, which preserves the high quality of the materials.

Finally, the high-quality results presented in this chapter require a large amount of reflectance measurements as input data. This necessitates rather complex and expensive automated capturing setups, which might hinder the widespread use of this approach. However, as there is a certain demand for high-quality virtual replicas, building or renting such a setup could be worthwhile.

5.7 Summary & Future Work

In this chapter, we presented a processing pipeline capable of reconstructing a detailed geometry along with extensive view- and light-dependent reflectance information. By treating appearance at different feature scales differently and representing it with a triangle mesh and a BTF, we facilitate the creation of digital replicas of real-world objects that can be viewed from almost arbitrary directions and illuminated by arbitrary illuminations. The only restriction is given by the decrease in quality due to extrapolation for view and light directions that are too far away from the sampled ones.

We furthermore successfully tested our approach on a variety of objects, exhibiting different challenging characteristics in terms of geometry or reflectance behavior. The consequent usage of HDR data throughout all the steps of our pipeline, although introducing new issues for the factorization that needed to be addressed via

0.772	0.802	0.952	
(a) PTM	**(b)** texture	**(c)** BTF	**(d)** reference

Figure 5.18: *Comparison between PTM and path traced renderings with texture and DFMF compressed BTF. (d) shows a measured radiance map for reference. The PTM in (a) was created from tone-mapped BTF measurement images taken from the depicted viewpoint. While the spatial resolution is similarly good, reflectance properties are best preserved using the BTF (c). Slight differences between BTF (a) and reference (d) occur in regions with large contributions of indirect illumination. These are discussed in Section 5.6.4. The SSIM index with respect to (d) is given below the images. The PTM probably scores lowest, because the technique fails to reproduce sharp features such as highlights and shadows.*

DRC, allows for a robust high-quality reconstruction of even specular objects. Our presented results show a considerable amount of detail at a very high resolution that was not achieved before using a reflectance capturing approach. Furthermore, the results demonstrate that, after acquisition and processing, BTFs can be used in similar applications as conventional 2D textures or SVBRDF models, while allowing a much more faithful reproduction of the appearance.

We have published the proposed postprocessing approach in three conference proceedings:

- Integrated High-Quality Acquisition of Geometry and Appearance for Cultural Heritage [SWRK11] as a research paper at the VAST 2011.
- Capturing Shape and Reflectance of Food [SWR*11] as an application sketch at the SIGGRAPH Asia 2011.
- Acquisition and Presentation of Virtual Surrogates for Cultural Heritage Artefacts [SK12] as an invited talk at the EVA 2012.

Furthermore, for the Donkey, Minotaur, Terracotta Soldier and Buddha objects, the full raw measurement data has been made publicly available as the `OBJECTS2011` data sets. Another batch of digitized objects has been published in processed form as OBJ geometry with DFMF compressed BTF under the name `OBJECTS2012`. All data sets can be downloaded from `http://btf.cs.uni-bonn.de`.

So far, we only used reflectance samples captured for directions on one hemisphere. For a faithful reproduction in completely arbitrary view and illumination situations, it is however necessary to have reflectance data for the whole sphere. The acquisition of this data would be possible with our employed setup by turning the object and taking multiple measurements. As mentioned in Section 5.6.2, a full spherical setup [NJRS13, KNRS13] would be a better alternative. We did not explore this option in the scope of this work, but instead relied on the proposed hole filling technique to fill in the missing data. Still, incorporating full spherical data will be an important future endeavor, since even when rendering the object from view directions that lie in the captured hemisphere, reflectance information for the missing angles are needed for global illumination computations and thus for a faithful rendering of the object.

Yet, even with full spherical reflectance information available, there will always be a demand for hole filling and interpolation because of macroscale occlusions and shadowing. Therefore, it would also be an important avenue of future research to improve the presented resampling and hole filling algorithm, further enhancing the visual quality. One possibility would be to extend our proposed approach to simultaneously consider samples from multiple points on the surface. The idea is to increasing the number of available directions and covering a better distribution. Additionally, while the clustering works reasonably well for clearly distinct materials, for some examples, e.g. the half-transparent icing on the Pudding Pastry, an exclusive cluster assignment is a less-than-ideal solution. Here, approaches that consider neighborhoods of multiple texels, similar in spirit to the global fitting scheme we explored together with Ruiters *et al.* in [RSK12], might be more beneficial.

Furthermore, it would be worthwhile to consider more elaborate parameterization techniques. For example, the visual quality of the compressed BTFs could be improved by having more consistent tangents along seams, e.g. using a technique similar to *Invisible Seams* [RNLL10]. In this work, the authors employ additional constraints to enforce consistent alignment across borders of parameterization parts. This gets rid of the artifacts caused by differences in linear interpolation at the opposing sides of a parameterization seam in classical texture rendering. Similarly, additional constraints might be used to enforce consistency in the orientation of local coordinate systems. The rendering artifacts shown in Figure 5.7 might this way be avoided. Orthogonal to the seam considerations, better texture atlas layouting approaches, e.g. [NS11], could be employed to reduce the memory footprint of the BTF on the GPU during real-time rendering.

Moreover, we see the removal of macroscale interreflection in the BTF as a relevant direction for future work. One possible approach that has already been successfully applied for heightfield geometry in [RK09b] would be to alternate between computation of reflectance and removal of interreflections until the appearance converges to a stable state.

Part III

Transmission and Rendering

CHAPTER 6

WEBGL-BASED STREAMING AND RENDERING OF BTFS

In this chapter, we present a novel progressive transmission method for the visualization of digitized objects given in the representation established in Chapter 5, i.e. a triangle mesh and a compressed BTF. The envisioned application is the fast interactive inspection of remote object collections over the Internet. Instead of relying on server-side rendering, we perform the real-time rendering directly in the web browser. This has the advantage that for using this interactive viewer, neither the host nor the client side need any kind of special equipment, such as dedicated rendering or streaming servers or software. On both sides, everyday web technology, i.e. a standard HTML web server and a fairly modern browser, are the only requirements.

This offers the considerable advantage that the high-quality virtual surrogates can directly be linked in other hypertext documents, such as museum websites or encyclopedic articles, or found and referred to by search engines like Europeana or Google. Nonetheless, this technique could also be employed for browser plug-in-based 3D viewers or full-scale stand-alone applications, e.g. kiosk viewers, information panels, collection browsers, etc..

For this, the BTF data needs to be transmitted over the Internet. Here, a progressive download is desirable due to the still comparatively large size of the DFMF compressed BTFs. The full transmission of a compressed BTF can otherwise require several minutes. By employing a suitable compression and streaming scheme, we are able to provide a high-quality display of the object within a few seconds. Interactive exploration is already possible while the remaining details are transmitted in the background. The quality of the real-time visualization is successively updated. To achieve this, we both progressively increase the number of components C and the resolution of the textures by utilizing a wavelet codec. An overview over the complete streaming and rendering architecture is given in Figure 6.1.

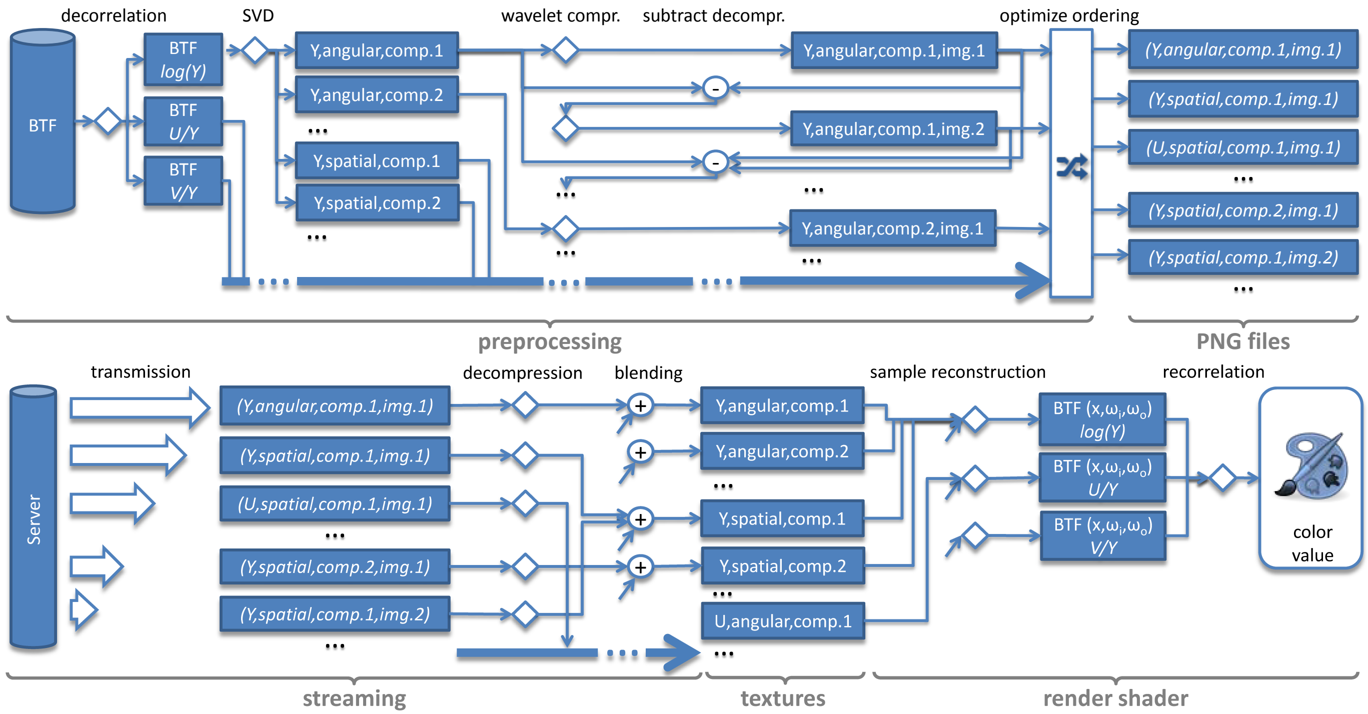


Figure 6.1: *Overview of our proposed compression (top) and streaming and rendering pipeline (bottom).*

Of course, exact timings depend on the connection speed. In the remainder of this chapter, we will for simplicity of conversion consider a net data rate of 8 Mbit/s. This is for example about the bandwidth that can be achieved via commonplace 3G mobile phone networks using HSDPA (7.2 Mbit/s – 42.2 Mbit/s) [HT06].

To demonstrate the practical applicability, we provide an implementation of our technique in a HTML5-based viewer, using the emerging web standard *WebGL*. This way, the viewer works cross-platform on all standard compliant browsers without the need for installing any apps, plug-ins or extensions. We test our approach with several of the objects obtained in Chapter 5. The utilized texture resolution and file sizes of the uncompressed and compressed data can be found in Table 6.1.

6.1 Introduction

In 2010, the introduction of `WebGL`, a JavaScript-based variant of `OpenGL`, suddenly opened the door for efficient and platform independent real-time rendering in a web browser. It is has since been standardized by the Khronos Group [Web13]. Today, the young standard is already widely adopted by modern browsers. All mainstream browser manufacturers (Microsoft, Mozilla, Google, Apple, Opera) already support WebGL.

Although not yet at the visual quality level of native graphics applications, this recent development already finds application in browser-based 3D games or 3D engine tech demos, such as the technically very advanced "Unreal Engine 3: Epic Citadel" demo[1]. This impressively shows, that the presentation of 3D objects or even complete virtual worlds in the web browser have in principle become a possibility. In our application example in the field of cultural heritage (see Section 2.1), this creates completely new means of public dissemination of 3D objects. It enables the creation of a virtual exhibition of objects or could even give access to whole collections.

While WebGL-based frameworks for textured 3D geometries or PTMs are already available [DBPGS10], up to our first publication on this topic in 2011 [SRWK11] (which is part of this thesis' work), the public presentation of BTFs via the Internet had not been realized. This was probably due to the sheer size of the data sets, which are too large for a direct transmission and visualization, even if compression is applied (see Table 5.3).

The main application demonstrated in this chapter is the presentation and interactive inspection of single objects for public dissemination of cultural heritage artifacts.

[1] `https://www.unrealengine.com/showcase/epic-citadel`

However, the presented technique is of course also applicable in other scenarios: scholarly use, e.g. for fast browsing of artifact databases or collaboration between institutions over the Internet, presentation of products in online shops and possibly even photorealistic texturing of virtual worlds in the browser with measured BTF materials.

In summary, the main contributions in this chapter are

- a two-tiered BTF streaming scheme via successive transmission of SVD singular vectors, each of which is progressively downloaded,
- the considerable improvement of BTF compression ratio by employing an additional image compression on the singular vectors,
- a wavelet-based codec for HDR image compression, optimized for the shader-based decoding in WebGL,
- a sophisticated heuristic for determining an optimized streaming order, prioritizing the perceptually most important information by balancing the number of transmitted singular vectors against their respective accuracy,
- an inexpensive preintegration approach to improve the rendering quality using a view-dependent ambient light term,
- a real-time WebGL object exploration demo application as proof of concept, supporting concurrent rendering and transmission of BTFs out of the box on standard compliant browsers.

First, we give a brief overview over the previous work in the areas of presentation of tangible cultural heritage and web-based dissemination of 3D content in Section 6.2. Then, we explain the utilized state-of-the-art method for BTF rendering in Section 6.3, which lays the foundation for our novel work. In addition, in Subsection 6.3.2, a small technical contribution to the real-time rendering of BTFs is discussed. Section 6.4 will tackle the major technical contribution of this chapter, i.e. streaming of BTFs over the Internet and rendering in the browser. We evaluate the feasibility of our technique in Section 6.5 on several examples. Finally, in Section 6.6 we summarize our results and point out possible avenues of future work.

6.2 Related Work

For the display of 3D content in web applications, a wide variety of technical solutions and APIs is readily available. For a more comprehensive overview on 3D content in web applications, we refer the reader to the survey in [BEJZ09].

The most widespread modern technique for interactive inspection of an object from arbitrary viewpoints is the use of textured 3D meshes. There are already several web-based presentation applications in the context of cultural heritage, that make use of this technique, for example [DBPGS10, JBG11]. However, these approaches are not really suitable for a photorealistic representation of objects with complex reflectance behavior.

In addition to the use of still images, there are also image-based techniques which take pictures of an object from several viewpoints on an orbiting trajectory and allow the user to make an interactive selection. Often, either `Apple Quicktime VR` [Che95] or `Adobe Flash`[2]-based solutions have been employed for the presentation. While these approaches allow a very realistic depiction, the selection of viewpoints is limited to those views for which images have been captured.

A different avenue was followed in [DBPGS10], where a web-based viewer for PTMs was presented. PTMs, are the complementary technique for the photorealistic depiction in the sense that they provide a fixed view under arbitrary illumination. By employing progressive downloads, the user is able to view large images and interactively change the light direction. There are also works on multiview PTMs [GWS*09], but, to the best of our knowledge, there is no solution for web-based distribution and viewing. In [MSE*10], an offline viewer was presented. However, due to the fact that it is difficult to take advantage of the coherence between different views and because flow fields are used as an implicit geometry representation, a rather large amount of storage is required for every view. Additionally, a large number of views would be needed for high-quality view interpolation, especially if a completely arbitrary viewpoint selection is desired. As a consequence, their viewer is limited to a predefined orbiting trajectory at fixed distance.

We instead advocate the use of BTFs parameterized over a triangle geometry of the object to convey view- and light-dependent appearance. However, BTFs are notorious for requiring huge amounts of data, making them an unconventional choice for web-based transmissions. In the following sections, we show that BTFs can be compressed sufficiently, so that streaming a dense sampling of view and light directions becomes practically attainable. Furthermore, unlike multiview PTMs, our appearance representation is suited for working with scenes composed of multiple objects and free camera movements.

Alternatively, due to their compactness and real-time rendering capabilities, SVBRDFs are sometimes regarded as a reasonable choice for web-based representation [KNRS13]. However, since analytical BRDF models are employed, the complexity of reflectance behavior that can be represented is more restricted than for BTFs. A recent SVBRDF estimation approach presented in [WDR11] accom-

[2] `http://www.adobe.com/products/flash.html`

modates for that fact by representing the reflectance as a mixture of several different analytical BRDFs with varying normals and a tabulated 4D residual function. Yet, in contrast to factorization-based FMF or DFMF BTF compression, this form of representation does not provide a level of detail hierarchy and thus does not lend itself as easily for a progressive streaming.

We use DFMF compressed BTFs [Mül09] as a basis for our additional wavelet compression. A thorough description of BTF compression with DFMF can be found in Section 5.5.4. There are also tensor factorization-based approaches [WXC*08], which are, like our wavelet compression, capable of further compressing the spatial dimensions. However, with the application of streaming in mind, we prefer the DFMF technique over tensor factorization, as it will guarantee the best approximation with incompletely transmitted data (see discussion in Section 6.4). Even though the simple FMF compression would be equally suitable for the proposed transmission, the DFMF offers the advantage to treat the luminance and chrominance of the material appearance separately. This allows a prioritization of the luminance components, to which the human perception is more sensitive.

6.3 Real-time BTF Rendering

When compressing a BTF via matrix factorization, the original data is represented as a matrix $\mathbf{B}$. Then the SVD $\mathbf{B} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ is computed. The diagonal matrix $\mathbf{\Sigma}$ can be stored by multiplication either with $\mathbf{U}$ or $\mathbf{V}$. We choose $\mathbf{V}$. However, this can also be understood as the approximation of the reflectance function ρ by a sum of products of two functions, one of them depending only on the view and light directions ω_o, ω_i, the other on the spatial position $\mathbf{x}$:

$$\rho(\mathbf{x}, \omega_i, \omega_o) \approx \hat{\rho}(\mathbf{x}, \omega_i, \omega_o) = \sum_{c=1}^{C} \mathcal{U}_c(\omega_i, \omega_o) \cdot \mathcal{V}_c(\mathbf{x}). \tag{6.1}$$

The c-th column of $\mathbf{U}$ is regarded as a tabulated representation of the function $\mathcal{U}_c(\omega_i, \omega_o)$ and analogously the columns of $\mathbf{V}\mathbf{\Sigma}$ as representations of $\mathcal{V}_c(x)$. The approximation quality versus the compression ratio is controlled by C, i.e. the number of functions/columns used. Due to the DFMF compression, we employ fewer components to approximate the color information than for the luminance.

If no columns are left out (i.e. C equals the rank of $\mathbf{B}$), the original data can be reproduced exactly. Since many points on the surface usually exhibit similar reflectance behavior, there is a large redundancy between the columns of $\mathbf{B}$, allowing for a good approximation via a low rank matrix. We denote the left and right singular matrices that are truncated after c columns $\mathbf{U}'$ for $\mathbf{U}$ and $\mathbf{V}'$ for $\mathbf{V\Sigma}$. For more details, we refer to Section 5.5.4.

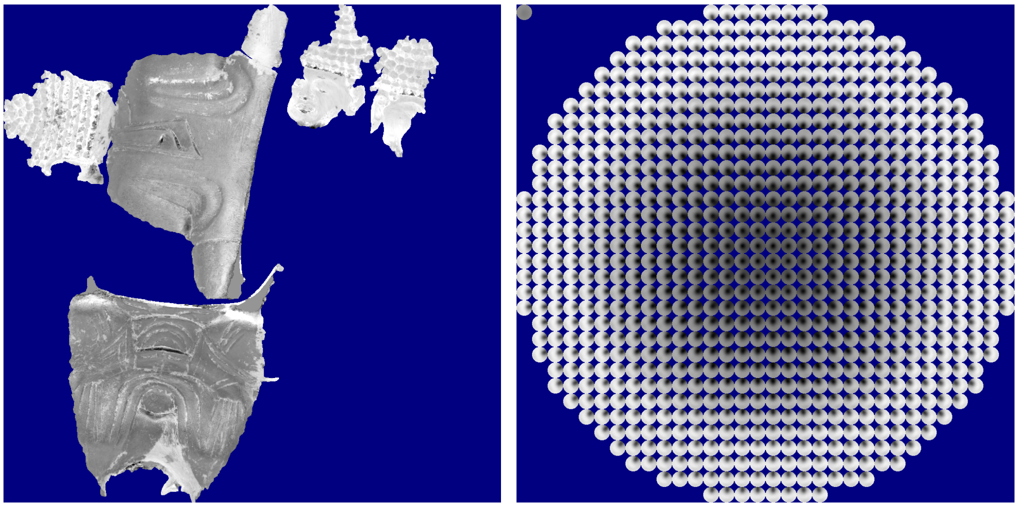
Before performing the DFMF compression, we apply a conversion from RGB to YUV color space and a dynamic range compression to the data matrix. Both operations have to be inverted at rendering time in the fragment shader after reconstructing the value for $\hat{\rho}(\mathbf{x}, \omega_i, \omega_o)$. In addition, the data is stored and the reconstruction is performed for each of the three data signals (one luminance and two chrominance signals) separately. For the simplicity of notation, we will refrain from explicitly writing this out in any of the equations in this chapter.

6.3.1 Fast Sample Interpolation on the GPU

For rendering, it is necessary to reconstruct samples for arbitrary positions on the surface and arbitrary view and light directions. Therefore, one has to interpolate the available discretized representation. Here, the DFMF has the advantage that the interpolation can be performed independently for the spatial and angular dimensions by interpolating the 2D functions $\mathcal{U}_c$ and the 4D functions $\mathcal{V}_c$, respectively, instead of the actual 6D function $\hat{\rho}(\mathbf{x}, \omega_i, \omega_o)$.

The GPU can be used to evaluate the sum in Equation 6.1 in real-time. Here, the tabulated functions $\{\mathcal{U}_c, \mathcal{V}_c\}_c$ are stored in textures and can thus be evaluated per fragment by simply performing adequate texture fetches. Considering an appropriate texture layout (see Figure 6.2a), the 2D texture interpolation for $\mathcal{V}_c$ can be performed directly on the GPU, whereas for the angular components a 4D interpolation is necessary, which is not supported in hardware. In [GMSK09], Guthe *et al.* propose to use texture fetches into 3D textures, which provide hardware support for interpolation in three of the four dimensions, and perform the linear interpolation in the last dimension manually. Unfortunately, 3D textures are not yet well supported by WebGL in general and can not be used as render targets. Yet, our GPU-based wavelet decompression necessitates rendering into textures. Therefore, we instead have to perform four fetches into 2D textures.

For this, we resample the stored reflectance values into view and light directions represented via a parabolic parameterization (see Section 3.1.3.1). The parabolic light direction coordinates $\omega_i \in \Omega_{\text{parabolic}}$ are mapped onto a quadratic image patch with $P \times P$ pixels, i.e. $\mathbf{p}_i = \lfloor \frac{P-1}{2}(\omega_i + 1) \rfloor$. Similarly, the view directions $\omega_o \in \Omega_{\text{parabolic}}$ are mapped to coordinates $\mathbf{p}_o$. The final 2D texture containing the bidirectional samples is obtained by inserting the respective light hemisphere samples at each view direction coordinate. The resulting 2D layout is exemplarily shown in figures 3.2d and 6.2b. In summary, we store the eigen-ABRDF value $\mathcal{U}_c(\omega_i, \omega_o)$ at texel $\mathbf{t} = P \cdot \mathbf{p}_o + \mathbf{p}_i$ of the texture for angular component c.



(a) spatial component texture **(b)** angular component texture

Figure 6.2: *First spatial (a) and angular (b) luminance components for the Buddha data set, prior to the transmission compression. Texels without any significance for the rendering process are marked in blue. The angular component texture in fact exhibits rather low frequencies, as the borders to the here depicted blue background do not need to be taken into account. The disk in the left upper corner of (b) is the first component of the normalized integrated ambient term.*

As we explained in Section 3.1.3.1, parabolic coordinates do not show any issues of "wrap-around" or singularities. Furthermore, within the neighborhood around a point, the ratio of Euclidean distances provide a good approximation of actual direction distance ratios. Thus, we can obtain samples for arbitrary directions by simple bilinear interpolation of the pixel values in the texture. Due to the described pixel layout, we can directly utilize the texture mapping units of the GPU for light interpolation. This way, we obtain four samples for the given light direction via texture fetches and perform the bilinear view interpolation explicitly in the shader. Figure 6.2b shows the resulting texture for the first angular component.

6.3.2 Preintegrated View-dependent Ambient Lighting

In real-time graphics, it is common to use either directional light sources, i.e. all light rays come from the same direction, or point lights, i.e. all light rays come from the same point of origin, as this allows fast enough lighting computations. However, in reality, light conditions that are either perfect point lights or perfect directional lighting are almost never encountered. Instead, often one dominant

(a) correct shadows **(b)** attenuated shadows **(c)** correct shadows + ambient

Figure 6.3: *The proposed preintegrated ambient lighting (c) provides an increased level of realism in comparison to no ambient lighting at all (a). (b) demonstrates a common alternative approach of just attenuating the brightness of the object by a factor in the shadowed regions, leading to a wrong light-dependent behavior, such as the highlights on the shoulder and neck of the Buddha.*

light source, e.g. the sun shining through a window, and additional ambient light, for example coming from the reflections of the dominant light at the surrounding surfaces, can be observed. A common technique to increase the level of realism and let scenes look less artificial is the introduction of an *ambient term* into the lighting computation (see Figure 6.3 for a comparison). Here, the (weaker) ambient light is approximated by a single intensity value.

The underlying assumption is that the incident ambient light comes equally distributed from all directions. In the case of the BTF, this means that the contributions of the light direction-dependent reflectance need to be integrated for all directions (in our implementation, the cosine term from the rendering equation is already included in the BTF ρ). This eliminates the light direction dependency for the evaluation of the final intensity and allows a precomputation of the integrated value. However, the view direction still needs to be considered during rendering, as many effects in the BTF, e.g. masking in the material, are heavily view-dependent. Due to storage requirements, it would not be feasible to have separate textures for each view direction of the preintegrated ambient term. Instead, we can again employ the truncated factorized representation $\mathbf{U}'\mathbf{V}'^T$ for the BTF:

$$\begin{aligned}\int_\Omega \hat{\rho}(\mathbf{x},\omega_i,\omega_o)\,\mathrm{d}\omega_i &= \int_\Omega \sum_{c=1}^{C} \mathcal{U}_c(\omega_i,\omega_o)\,\mathcal{V}_c(\mathbf{x})\,\mathrm{d}\omega_i \\ &= \sum_{c=1}^{C}\left(\int_\Omega \mathcal{U}_c(\omega_i,\omega_o)\,\mathrm{d}\omega_i\right)\mathcal{V}_c(\mathbf{x}) \\ &= \sum_{c=1}^{C} \mathcal{U}_c^{(a)}(\omega_o)\,\mathcal{V}_c(\mathbf{x}). \end{aligned} \tag{6.2}$$

Thus, we only have to compute the integrated ambient functions $\mathcal{U}_c^{(a)}(\omega_o) = \int_\Omega \mathcal{U}_c(\omega_i,\omega_o)\,\mathrm{d}\omega_i$, with $c = 1, 2, \ldots, C$. These integrals can simply be precomputed by summing up all row-vectors corresponding to a particular view direction ω_o in $\mathbf{U}'$.

Evaluation of the rendering equation for the local illumination consisting of the combined direct and ambient lighting terms then reads as follows:

$$\begin{aligned} L_o(\mathbf{x},\omega_o) &= \sum_{c=1}^{C}\left(L_a\cdot\mathcal{U}_c^{(a)}(\omega_o)\,\mathcal{V}_c(\mathbf{x}) + \sum_i L_i(\omega_i)\cdot\mathcal{U}_c(\omega_i,\omega_o)\,\mathcal{V}_c(\mathbf{x})\right) \\ &= \sum_{c=1}^{C}\mathcal{V}_c(\mathbf{x})\left(L_a\cdot\mathcal{U}_c^{(a)}(\omega_o) + \sum_i L_i(\omega_i)\cdot\mathcal{U}_c(\omega_i,\omega_o)\right) \end{aligned} \tag{6.3}$$

with L_o denoting the radiance reflected in outgoing direction ω_o towards the observer from point $\mathbf{x}$, L_i denoting the radiance of the i-th light source with the incident direction ω_i and L_a denotes the ambient light factor of the scene.

Since $\mathcal{U}_c^{(a)}$ is only view-dependent, the necessary 2D texture interpolation can be performed on the GPU. When using the same parabolic parameterization to store the integrated values as for the other angular components, both of them can be stored in the same texture without any overhead. This can be achieved by choosing the texels outside the circle addressed by parabolic coordinates to store the ambient term. In the texture depicted in Figure 6.2b, the values for the integrated ambient function are stored in the left upper corner. Even for a single light source, evaluating the ambient lighting does increase the number of necessary texture fetches and scalar product computations by only 20% and is hence rather inexpensive. The proportional overhead decreases further if more direct light sources are used.

The above considerations also apply when using any linear transformations on the BTF data, such as the proposed color transformation in YUV color space. If the also proposed nonlinear dynamic range reduction prior to factorization is applied, the integral in the SVD basis does not equal the integral of the original BTF data any more. However, computing the ambient light preintegration from a second linear SVD basis would require the transmission of additional, separate eigen-texture components. The same is true for computing the function on the original data and performing a subsequent factorization. Fortunately, in all our experiments the result obtained with nonlinear dynamic range reduction applied was perceptually very close to the correct result . We therefore ignore the remaining difference and apply the method for the dynamic range reduced SVD basis.

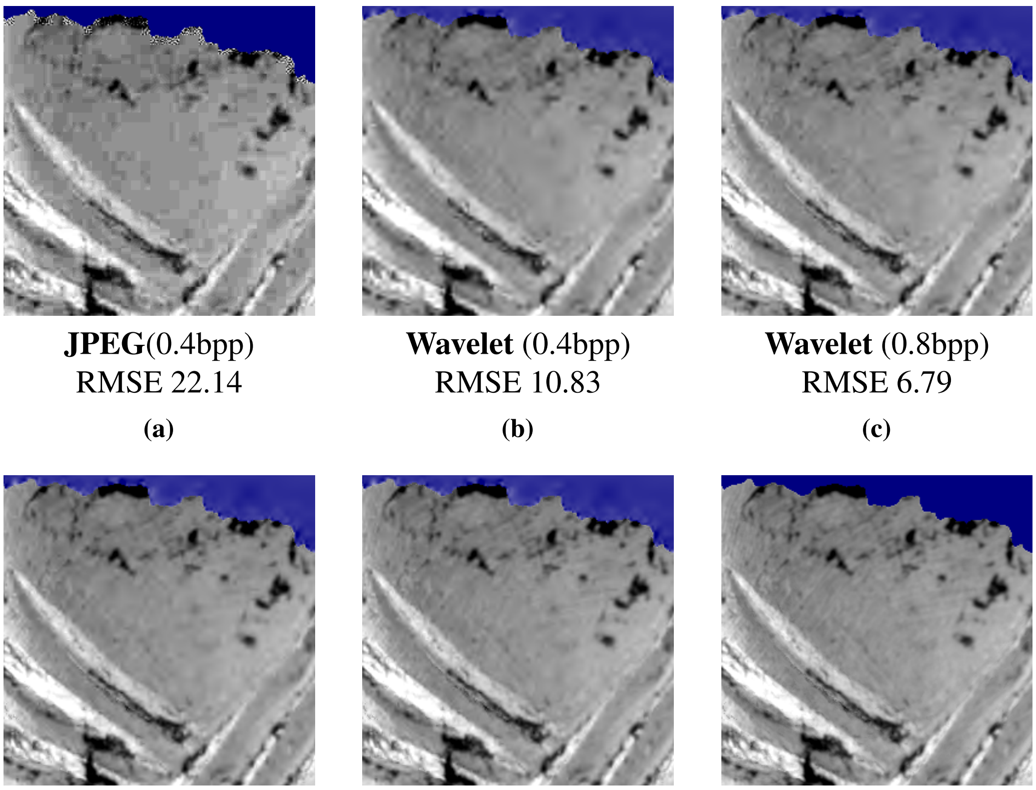
6.4 Streaming

By transmitting more components, the quality of the approximation is successively increased. In fact, it is possible to prove that by using only the first C components, one obtains the best possible rank-C approximation of the original matrix under the Frobenius norm [EY36]. This can obviously be directly utilized for the progressive transmission of a BTF, by successively transferring the individual columns of the matrices $\mathbf{U}'$ and $\mathbf{V}'$. Each column effectively increases the rank of the approximation available for rendering.

6.4.1 Wavelet Compression

The rendering can start as soon as the first column for each of the six matrices (i.e. $\mathbf{U}'^{\log \mathrm{Y}}, \mathbf{V}'^{\log \mathrm{Y}}, \mathbf{U}'^{\mathrm{U/Y}}, \mathbf{V}'^{\mathrm{U/Y}}, \mathbf{U}'^{\mathrm{V/Y}}, \mathbf{V}'^{\mathrm{V/Y}}$) are available. Each additional component that has been transmitted can then be utilized directly for rendering to further increase the quality of the approximation. The individual columns $\mathbf{U}_c$ and $\mathbf{V}_c$, however, are still very large. Especially the spatial components (eigen-textures) can require considerable space, as each one is a 16 bit grayscale image (see Section 3.7) with the full resolution of the texture. Thus, for a 2048×2048 pixel BTF, uncompressed transmission of only one spatial component for one of the channels still requires 8 MB.

Since the angular components (eigen-ABRDFs) show rather low frequencies and the spatial components exhibit frequency characteristics similar to natural images (see Figure 6.2), usual image compression and transmission techniques can be applied here. We thus utilize a wavelet codec to send each of the individual component textures progressively. We start with a highly compressed version and then gradually send several difference images, each also encoded with the wavelet codec, until the original has been reconstructed to a sufficient level of accuracy.



JPEG(0.4bpp) RMSE 22.14 (a) | **Wavelet** (0.4bpp) RMSE 10.83 (b) | **Wavelet** (0.8bpp) RMSE 6.79 (c)

Wavelet (1.2bpp) RMSE 4.99 (d) | **Wavelet** (1.6bpp) RMSE 3.93 (e) | **Original** (16bpp) (f)

Figure 6.4: *Comparison of compression ratio and quality of the proposed wavelet image compression. The enlarged views show a detail in the first eigen-texture of the Buddha data set (see Figure 6.2a). Image (a) shows as comparison a uniformly quantized and JPEG compressed LDR version. Images (b), (c), (d) and (e) show how our wavelet codec continuously refines the texture compared to the original (f). The RMS error with respect to the uncompressed original image (f) is given below the images. It is computed for only those regions of the texture which are occupied (unoccupied texels are marked in blue).*

The difference images are created by subtracting the reconstructed compressed version from the original texture. Thus, subsequent difference images contain the residual to the reconstruction resulting from all previous images.

There is a wide range of techniques, both for image compression and for progressive transmission of images. Even giving a short overview would by far exceed the scope of this thesis, and hence, as possible starting points, we refer the reader to the overviews given in [DN99, RY00]. However, for our purposes, many of these techniques are not directly applicable, as we have two important constraints.
Firstly, we have to transmit 16 bit textures, as a quantization of the data to 8 bit integers is not an option. Even though the dynamic range of the resulting textures has been reduced considerably by the logarithmic transform, for full quality display of the BTF the precision of floating-point values is still desirable.
And secondly, our codec must be suitable for fast decompression in the browser. There are several elaborate encoding schemes, such as the SPHIT codec [SP96], which achieve very good compression ratios and allow for elegant progressive transmission. However, JavaScript is an interpreted language and does not show the preformance of native code. Although *just-in-time compilation* (JIT) is available in newer browsers, it is still not fast enough for decompressing data encoded with these techniques. Instead we need a codec that can either be decoded by native browser functions or via shaders in WebGL.

Unfortunately, none of the main browsers does support a decompression codec which is directly suited to our purposes. Usually, one can only rely on support for JPEG and PNG images, both providing neither HDR encoding nor progressive transmission (though possible in both PNG and JPEG, it cannot be used for WebGL textures).

We therefore decided to implement a simple wavelet codec ourselves. The restoration of the HDR signal from LDR images and the decompression of the wavelet transform is performed in a fragment shader using the render-to-texture capabilities of WebGL. Consequently, this step is no longer limited by the execution speed of the JavaScript interpreter. For the actual stream-decoding, on the other hand, WebGL is not suitable at all, since a sequential decompression of the bit-stream is necessary, which cannot easily be performed on the GPU. Hence, we store the quantized wavelet components in a PNG image, utilizing the LZ77 compression and Huffman entropy encoding that is part of PNG. This approach is not the best available image-codec with respect to compression ratio. However, while still compressing reasonably well it allows for the progressive transmission of HDR data and can be efficiently decoded with the limited resources available to a JavaScript application. In our experiments, it performed better than JPEG compression regarding RMS error (see Figure 6.4).

For image compression we apply a straightforward wavelet transform coder. As the first compression step, we perform a dyadic wavelet decomposition of our texture using the popular CDF 9/7 wavelet [CDF92], which has found widespread application in image compression, for example in the JPEG2000 image format. This decomposition is performed directly on the HDR floating-point data. To compress these floating-point values, we then use a deadzone uniform scalar-quantizer to obtain 16 bit integer values. For each of the wavelet bands, we choose an individual threshold, using the algorithm from [SG88] to perform the bit allocation for obtaining chunks of a fixed size. During this allocation, we compute the rates under the assumption that each of the subbands is compressed individually via LZ77 and Huffman encoding. This is obviously only an approximation, since the coefficients for all subbands are stored together in the final PNG, but we found that it is an acceptable approximation, resulting in files of almost correct size. Finally, the quantized coefficients are stored in one PNG image, storing the high and low bytes separately in two halves, as this provided the best compression results in our experiments.

As the ABRDF values are only defined on the upper view and light hemispheres, the values occupy only circular regions when given in parabolic parameterization. In addition, often the spatial components contain an only partially filled texture atlas. Therefore, in both textures we have entries which are of no importance to the final appearance of the object. Please refer to Figure 6.2 for an example: here, these areas marked in blue. However, when these entries are just set to zero, the resulting texture contains sharp edges. These are not well-suited for the wavelet compression, as many coefficients are necessary to encode the resulting high frequencies. To avoid this problem, we use the approach described in [BHH*98] to fill these areas in such a way that the number of coefficients needed for compression is minimized as far as possible.

The PNG file format uses LZ77 compression and Huffman encoding. Thus, the compression ratio depends on the exact implementation and parameters of the encoder. The standard implementation of the compression algorithm in `libpng`[3] is based on `zlib`[4], but better results can be obtained using a more sophisticated implementation. Thus, we first use MATLAB's libpng-based `imwrite` command. Then we strip the resulting PNGs from metadata and recompress them with `AdvanceCOMP`[5], which in addition to zlib offers the compressor implementations `7Zip`[6] and `Zopfli`[7]. This improved the compression ratio by about 22%.

[3]`http://www.libpng.org/`
[4]`http://www.zlib.net/`
[5]`http://advancemame.sourceforge.net/`
[6]`http://www.7-zip.org/`
[7]`http://code.google.com/p/zopfli/`

6.4.2 Transmission and Decompression

The individual components are loaded by the JavaScript running in the browser. For this, the script successively requests the compressed chunks from the server. The quantization thresholds can be loaded separately or embedded into the HTML file. As soon as it has been received, each chunk is decompressed from PNG into a WebGL texture by the built-in functionality of the browser. This reverses the LZ77 compression and Huffman encoding.

Further decompression is then performed in two steps by a shader. First, the low and high bytes are combined and the quantization is reversed to obtain a floating-point texture of the wavelet transformed image. Secondly, the original texture is reconstructed by applying the inverse wavelet transform. This is done successively for the horizontal and vertical directions on each scale. We perform the transformation via direct convolution, instead of using a lifting scheme, to reduce the number of necessary render passes. Each of these steps is performed in a fragment shader on the GPU, using a WebGL framebuffer to render a suitable quad into a texture. The number of wavelet decomposition levels is adapted to the image dimensions. However, we restrict it to be at most eight levels, bounding the number of render passes during reconstruction. Care has to be taken to correctly interleave these individual decoding tasks with the actual rendering of the object to avoid noticeable drops in the frame rate. For progressive transmission, we successively transmit encoded difference images, which are then joined by the shader using additive blending.

To achieve a high throughput with the available bandwidth, we already request and transmit further components from the server while the decompression of the previous component is still in progress. Arriving data is buffered in a queue in order to be available as soon as the processing capabilities of the GPU are ready.

6.4.3 Transmission Order

One important remaining question is the order in which the components are to be transmitted to the client. At any time during streaming, there are several possibilities for choosing the next chunk. For each of the channels $\log$ Y, U/Y and V/Y, it is either possible to transmit a new component or to increase the quality of an already transmitted component by loading the next difference image for either the angular or spatial domain.

We determine this order in a preprocessing step. Here, we sort the textures by employing a greedy scheme in such a way that the total RMS error for the whole BTF is decreased as much as possible with each transmitted chunk. This order could be found by computing the *sum of squared errors* (SSE) between the original BTF and the reconstruction with the transmitted components explicitly. However, this would be prohibitively costly, as it would require decompressing and computing the SSE for the whole BTF file for every possible decision. We instead use an approximation, which takes advantage of the fact that the BTF is represented via an SVD. For this approximation, we consider the errors in $\mathbf{U}$ and $\mathbf{V}$ independently. Assuming that the compression had only been performed for the columns in matrix $\mathbf{U}$, this results in a distorted matrix $\tilde{\mathbf{U}}$, the error of which is given by

$$\left\|\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T - \tilde{\mathbf{U}}\mathbf{\Sigma}\mathbf{V}^T\right\|_F^2 = \left\|\left(\mathbf{U} - \tilde{\mathbf{U}}\right)\mathbf{\Sigma}\mathbf{V}^T\right\|_F^2 = \left\|\left(\mathbf{U} - \tilde{\mathbf{U}}\right)\mathbf{\Sigma}\right\|_F^2. \tag{6.4}$$

By construction, $\mathbf{V}$ is orthogonal and the second equality holds as a result of the fact that the Frobenius norm is invariant under orthonormal transformations. We can thus compute the SSE for the whole BTF data set by computing the errors for each of the component textures individually and weighting them with the entries in $\mathbf{\Sigma}$. The same computation is also possible for $\mathbf{V}$, under the assumption that $\mathbf{U}$ is orthonormal. When there are distortions in both matrices, the equality no longer holds exactly. However, we found that deviations are very small. The difference between the correctly computed error $\|\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T - \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^T\|_F^2$ and the approximation $\|(\mathbf{U}\mathbf{\Sigma} - \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}})\|_F^2 + \|\mathbf{V}\mathbf{\Sigma} - \tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}}\|_F^2$ was below 0.3% when comparing the DFMF compressed data set to one with additional wavelet compression applied. Currently, we simply compare the errors in the channels $\log \mathrm{Y}$, U/Y and V/Y directly. However, since luminance and chrominance are actually represented using different units, one being the logarithm of the intensity, the other being normalized color components, proper weighting factors, obtained by perceptual experiments, should be employed instead.

Even better results might be possible by using a more sophisticated BTF error metric, such as BTF-CIELab, proposed in [GMSK09]. Unfortunately, for such an approach our heuristic would no longer be applicable. This would increase the computation time drastically, as the error would have to be recomputed for many different combinations of transmitted components with varying approximation qualities.

6.5 Evaluation

We evaluate our approach on the digitized objects presented in detail in Section 5.6. All data sets offer an angular sampling of 151×151 view and light direction combinations, high-quality geometries, a high spatial resolution as well as a high dynamic range.

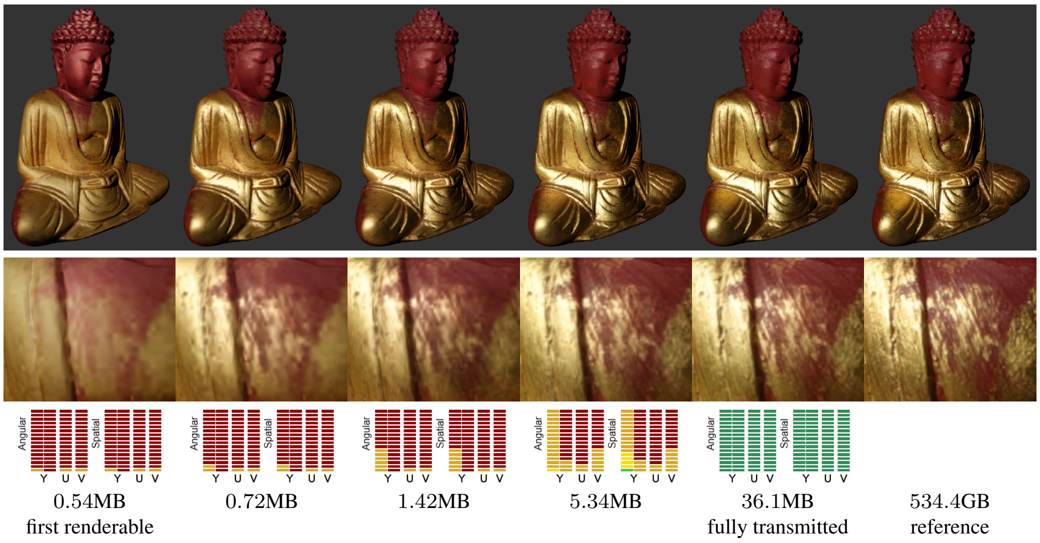


Figure 6.5: *A sequence of screenshots showing the refinement of the appearance of the 2048×2048 pixel Buddha data set over the streaming progress (rendered without ambient lighting). With 0.54 MB, enough chunks were transmitted to start rendering. From 5.34 MB to 36.1 MB only minor changes in appearance are noticeable and merely remaining fine details are loaded (improving the SSIM index from 0.961 to 0.975 on the full image and from 0.875 to 0.917 on the detail view). As reference, an image of the uncompressed data set, computed out-of-core using deferred rendering, is shown on the right. The bars below the images represent the component transmission status with more greenish colors representing that a higher number of refinement chunks were received for that component.*

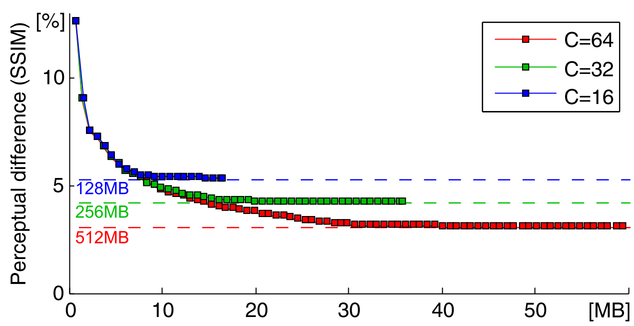


Figure 6.6: *The perceptual error (evaluated via SSIM) in dependence of the amount of transmitted data for different versions of the 1024×1024 pixel Buddha BTF. The dashed lines correspond to the DFMF compressed BTFs without employing further wavelet compression (assuming half-precision floating-point values). The error is computed with respect to the uncompressed (133 GB) data set and averaged over five representative view and light combinations.*

spatial resolution [pixel]	uncompressed [GB]	components #	DFMF [MB]	PNGs [MB]	GPU memory [MB]	performance [FPS]	preprocessing¹ [hours]
2048×2048	534.4†	32	514.8†	36.1	640† / 1280‡	25*/36§	8:45 + 3:25
1024×1024	133.6†	64	261.6†	71.6	512† / 1024‡	10*/10§	7:22 + 3:51
1024×1024	133.6†	32	130.8†	36.1	256† / 512‡	42*/47§	3:42 + 1:58
1024×1024	133.6†	24	98.1†	27.1	192† / 384‡	49*/55§	2:44 + 1:30
1024×1024	133.6†	16	65.4†	18.1	128† / 256‡	57*/60+§	1:53 + 1:00

† half-precision. ‡single-precision. *during streaming. §fully transmitted and unpacked.
¹Creation of PNG files + additional PNG compression.

Table 6.1: *Evaluation results: The GPU memory consumption is computed for parabolic parameterization of the angular components and additionally given for single-precision values, which is the amount currently needed due to WebGL restrictions. The achievable FPS are measured with Chrome 37 on Windows 7 using an NVIDIA GeForce GTX 780 GPU and a rendering resolution of 640×640 pixels. The preprocessing times are given for a 2 GHz Intel Xeon E5-2650 (singlethreaded).*

To utilize the texture interpolation capabilities of the GPU, the entries in $\mathbf{U}'$ are resampled into a parabolic map (see Section 6.3). The new directions are arranged in a regular 32×32 grid in the parabolic coordinate space. This increases the number of directions on the hemisphere to more than 800, which turned out to be necessary in order not to loose too much quality during this resampling step. For the combination of both hemispheres, this results in angular textures of 1024 ×1024 pixels (with about 63% occupancy). Please note that the components in this parameterization do require considerably more GPU memory than the original matrix $\mathbf{U}'$. Details are listed in Table 6.1.

6.5.1 Performance

The PNG files, which are stored on the server and used for streaming, are directly created from a factorized representation of the BTF and do not require the uncompressed data set. The processing of DFMF BTFs to streaming-ready PNG files takes three to twelve hours on a 2 GHz Intel Xeon CPU, depending on the number of components and spatial resolution (see Table 6.1). Please note that all operations have been performed by a singlethreaded MATLAB implementation. There is a lot of potential for future optimization, as all steps can easily be parallelized.

For testing the transmission performance and real-time capabilities, we created an HTML5-based BTF viewer that uses WebGL for rendering. We do not separately transfer the additional data, e.g. the 3D geometries or the quantization thresholds, as this would be beyond the scope of this thesis. Instead, we include them directly in the HTML file. Even though in actually deployed applications these files could be transmitted asynchronously on request (or in the case of geometry even progressively downloaded [LJBA13]), we found this setting suitable for evaluating the performance of our rendering and BTF streaming technique. Please note that we do not include the quantization thresholds ($\approx$ 35 KB – 64 KB) in our transmission size considerations.

Unfortunately, the missing support for 3D textures forces us to store the angular and spatial components in several tiles of one 2D texture, restricting the maximum renderable spatial and angular resolution and increasing the number of necessary texture fetches. Although these constraints could be partially handled by using large textures and multiple texture mapping units, the most crucial remaining limitation is the available GPU memory. This is especially since half-precision floating-point textures are not yet supported in any browser. For example, just a 2048×2048 pixel BTF alone, disregarding any additional buffer textures, would require 3.9 GB of GPU memory for 100 components for the Y and 50 for the U and V channels (the amount used in the quality evaluation of Chapter 5). This far exceeds current mainstream device capabilities.

Therefore, we tested our streaming approach with different feasible numbers of components C for $\log$ Y, using always half as many for U/Y and V/Y. The respective GPU memory consumption is shown in Table 6.1. In principle higher numbers of components as well as higher texture resolutions could also be streamed efficiently. This is especially true because the wavelet codec performs a multiscale analysis and transmits the most important wavelet coefficients more accurately (see Figure 6.4). Thus, a perceptually acceptable version, only lacking high-frequency details, is usually available very fast.

To allow for a meaningful visualization as early as possible, we constrain the wavelet compression to produce chunk sizes of 100 KB (in average 82.5 KB after PNG optimization). Rendering can start as soon as six chunks are available. As shown in Figure 6.5, after transmitting just 0.72 MB of the chunks, i.e. less than one second of transmission, the overall impression of the object's appearance is already successfully recovered. The progressive download of additional fine details is perceivable until about 5.3 MB are transmitted. The full data set with applied wavelet compression occupies as little as 36.1 MB, while approximating the captured appearance with an SSIM index of 0.975 (1 denotes an identical appearance, see Section 3.8 for details). A more detailed analysis of the convergence of perceived appearance quality is given in Figure 6.6.

In this thesis, we only considered a fixed amount of chunks per component (four for spatial components, three for angular components). Choosing the number of chunks adaptively could lead to even better compression ratios for the fully transmitted BTF. However, due to the optimized transmission order, there should be no difference in the rendering quality for a given amount of transmitted megabytes.

While rendering the BTF is mainly limited by the number of components and screen resolution, decompression depends on the resolution of the angular and spatial textures. The achieved rendering performance on a recent GPU (NVIDIA GeForce GTX 780, released in May 2013) are reported in Table 6.1 in *frames per second* (FPS). The decompression times directly influence the maximum possible transmission rate, which was about 1.3 MB/s for the smallest data set and 0.34 MB/s for the largest. Even on eight-year-old graphics hardware (NVIDIA GeForce 8800 GTX, released in November 2006) the $C = 16$, 1024×1024 data sets were transmitted with 20 FPS and rendered with 25 FPS. Therefore, we recommend to offer data sets at multiple quality levels to accommodate older hardware.

6.5.2 Compatibility

Currently, all five major browsers have committed themselves to supporting WebGL. In the current release versions of Microsoft Internet Explorer 11, Google Chrome 38, Mozilla Firefox 33, Opera 25 and Apple Safari 8, WebGL is readily available. As our renderable BTF representation makes use of floating-point data, our current viewer implementation uses the `oes_texture_float` extension to WebGL. In our experiments, we found that the extension is provided by all of the above browsers except for the Internet Explorer 11. It should be possible to circumvent this requirement by storing the single bytes of the floating-point numbers in separate textures and combining them again in the shader. However, we have not attempted to implement such an approach. We hope that the support for this feature will be added in future versions of the Internet Explorer.

We would like to stress that the basic streaming technology we presented is in no way limited to WebGL. Our decompression and viewing shaders are based on OpenGL ES2.0, a standard specifically designed to be supported by as many devices as possible, including tablets and mobile phones.

In our proposed implementation, a major challenge for mainstream deployment would be the large amount of texture memory that is needed to render even one object with BTF material appearance, making scenes composed of multiple objects not yet feasible. Thus, a more memory efficient rendering scheme would be desirable. We will tackle this problem in Chapter 7 of this thesis.

Figure 6.7: *A screenshot of our WebGL viewer implementation running in Firefox. The enlarged views show the presented virtual surrogate of a Minotaur figurine from two arbitrarily selected viewpoints with two freely chosen light directions, respectively.*

6.6 Summary

We presented a WebGL framework for the interactive viewing of digitized 3D objects textured with bidirectional texture functions. This representation allows for the display of highly accurate virtual surrogates with complex reflectance behavior on the Internet. By streaming the individual components, obtained by the SVD-based compression of the BTF together with a wavelet-based image compression, we are able to present a high-quality display of the BTF after just a few seconds. The remaining data is progressively loaded until a full quality presentation is obtained. Even though we used a WebGL-based implementation, the presented technique is not limited to web browsers, but could be used by a wide range of clients for the streaming and photorealistic depiction of objects.

The techniques and results presented in this chapter have been published in the proceedings of a conference and as an extended journal publication:

- WebGL-based Streaming and Presentation Framework for Bidirectional Texture Functions [SRWK11] as a research paper at the VAST 2011. It was awarded as best paper.
- WebGL-based Streaming and Presentation of Objects with Bidirectional Texture Functions [SRWK13] as an extended journal article in ACM JOCCH.

Source code for both the preprocessing as well as streaming and rendering using WebGL can be downloaded at

`http://cg.cs.uni-bonn.de/en/publications/additional-material/btfstreaming-source-code/`.

Furthermore, an instance of the interactive viewer website with several of the presented data sets is available at `http://btf.cs.uni-bonn.de/viewer`.

In our implementation, we include the geometry of the objects in the HTML file. It would be an obvious first extension of our technique to also employ progressive geometry transmission. Furthermore, the concept of using the successive transmission of factorization components for progressive detail refinement could be extended to a full-fledged hierarchical level of detail renderer, employing also a view-dependent refinement. This would allow for the presentation of large objects in extremely high resolution or even complete scenes, such as virtual excavation sites or historical settings, at very high levels of detail. We explore such an approach in the next chapter.

CHAPTER 7

Level of Detail Streaming and Rendering using BSVTFs

In this chapter, we approach the problem of GPU texture memory limitation that has been raised in the previous chapter. Even compressed, the high-resolution BTFs for digitized objects are so large that even the most recent generation of GPUs (e.g. the NVIDIA GeForce GTX 780) can barely fit one object into memory. To solve this issue, we present a new hierarchical level of detail approach for BTFs that allows to render virtual scenes containing several digital replicas or other BTF materials (see figures 7.1 and 7.4). For this, BTFs are combined with virtual texturing to *bidirectional sparse virtual texture functions* (BSVTFs). We show that this new representation can efficiently be streamed and rendered in real-time.

7.1 Introduction

As discussed in previous chapters, factorized BTFs are our representation of choice for high-quality materials. Chapter 6 also demonstrates the application of this format for interactive viewing. Unfortunately, so far the usefulness of BTFs in real-time graphics is greatly hampered by the still rather large data sizes of up to several gigabytes per material.

An additional entropy coding or lossy compression can be employed to improve the compression ratio over the factorized BTF, e.g. for the fast transmission over the Internet. However, the data needs to be unpacked into the factorized representation again to support efficient random access for real-time rendering. In Chapter 6 we demonstrated that, using a lossy wavelet compression for transmission, factors of ten for no perceivable up to 60 for a noticeable but still acceptable error can be achieved. However, we also pointed out that a BTF with a compressed size of 36.1 MB for the transmission had then to be stored in 2.5 GB of GPU memory.

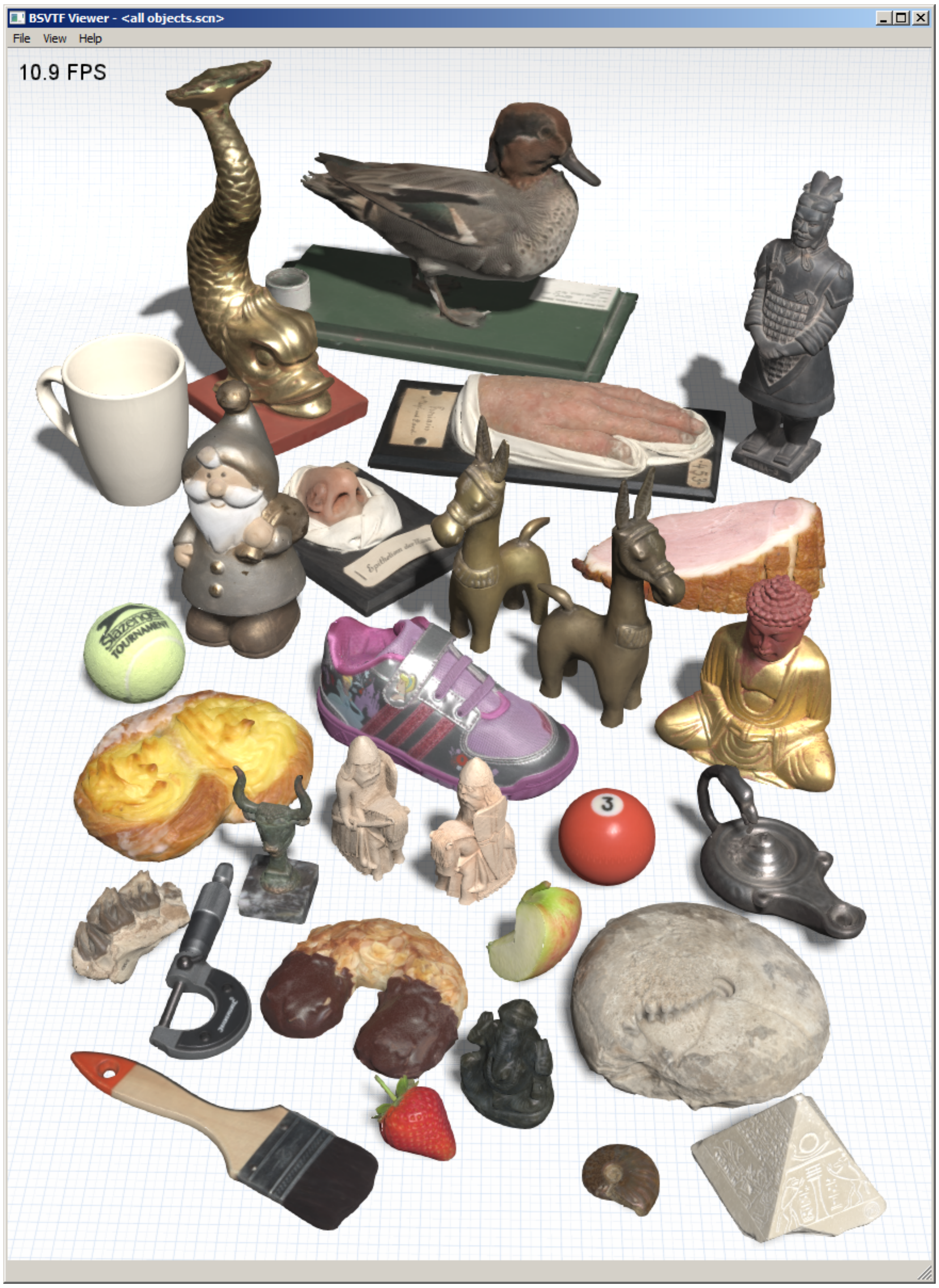


Figure 7.1: *Screenshot of our BSVTF viewer application. It shows a scene with 29 objects, all textured with high-resolution BTF materials, rendered in real-time on the GPU. For comparison, a fully path traced rendering of this scene (requiring* 96 GB *RAM and several hours) can be found in Figure 1.2.*

This makes rendering even a small scene containing a few objects with high-resolution BTF materials on the GPU simply impossible. The high memory requirements for even a single object can hardly be met by the latest professional hardware. When considering the trend towards high-quality 3D graphic on tablets and mobile phones, which nowadays have performant graphics chips but a drastic shortage of memory, the problem of GPU memory consumption becomes even more severe.

Therefore, we propose a new hierarchical level of detail approach to BTF rendering. We show that this way the bottleneck of GPU memory can effectively be circumvented, allowing for rich virtual scenes that are textured with several high-resolution materials, such as the scene shown in Figure 7.1. Furthermore, we present an efficient streaming solution to load the necessary data for rendering on-the-fly from disk or from a network connection.

The familiar problem of rendering very large textures that exceed the available memory has already been successfully handled by employing a technique that is known as Clipmapping [TMJ98] or *sparse virtual texturing* (SVT) [Bar08]. SVT utilizes a level of detail hierarchy in the spatial domain to only keep the required parts of the texture in the necessary resolution in GPU memory.

In this chapter, we propose the *bidirectional sparse virtual texture function* (BSVTF), an adaption of the SVT technique to the context of real-time BTF rendering. In contrast to plain textures, which only have a level of detail hierarchy in their spatial resolution, a factorized BTF representation inherently includes a second level of detail domain of the ABRDF approximation quality. We demonstrate that both level of detail hierarchies can be combined in a consistent manner by reducing them to a single spatial level of detail problem.

In contrast to several hundreds of megabytes per high-resolution BTF, in our case the CPU and GPU memory demand is very moderate. More importantly, the memory demand grows only marginally with higher resolutions. Also for increasing number of materials most additional storage space is required for the angular rather than the spatial part of the factorized matrix. Furthermore, the computational overhead on the GPU introduced by the approach remains constant regardless of the number of BSVTFs. We demonstrate that BSVTFs can also be used for the efficient streaming over a network, allowing to display scenes with multiple high-resolution materials without considerable delay. For this, we apply an additional streaming compression that utilizes the redundancy found in the level of detail hierarchy. To facilitate the fast start of rendering, we interleave the angular factorization components with the transmission of the level of detail tiles of the spatial information.

In summary our contributions are

- a hierarchical level of detail approach for memory friendly real-time BTF rendering,
- the inherent weighting of the BTF compression approximation error and the spatial level of detail error of the SVT by formulating the approximation problem as a unified error minimization,
- a streaming approach utilizing a transmission compression based on the level of detail hierarchy, allowing rendering of scenes with BTF materials transmitted over a network without significant loading times.

In Section 7.2, we first discuss the related work in the areas of level of detail and out-of-core rendering as well as compression of BTF for real-time viewing. Then, we explain the sparse virtual texturing technique and the employed real-time rendering using factorized BTFs in more detail in sections 7.3 and 7.4, as this constitutes the foundation of the proposed approach. Our main technical contribution of this chapter is presented in Section 7.5, which explains the extension of SVT to BSVTFs, and Section 7.6, describing our BSVTF-based streaming implementation. In Section 7.7, we present results and evaluate the proposed novel streaming and rendering method. Finally, Section 7.8 summarizes the insights obtained in this chapter and outlines directions of future research.

7.2 Related Work

To the best of our knowledge, there exists no previous literature on a similar level of detail application on BTFs. However, there is a large body of related work in the separate fields of level of detail rendering as well as real-time rendering and streaming of BTFs.

7.2.1 Hierarchical Level of Detail

As early as 1976, Clark introduced the concept of hierarchical level of detail on geometric models [Cla76]. Here, the problem of considering only that parts of the geometry of a synthetic scene that are actually relevant for rendering the users viewport is solved by using an object hierarchy. The hierarchy holds the geometry of objects in the scene in different levels of detail. A *graphical working set* is built from the hierarchy by choosing exactly those objects that are visible on the screen in a level of detail that is sufficient for the required rendering resolution of the object.

Since then, hierarchical level of detail has found a lot of application for scene geometries and terrain visualization and also for streaming these types of data over the Internet. More information on these research topics can be found in [LWC*03].

In real-time graphics, another level of detail hierarchy has also found very widespread application: In combination with trilinear interpolation, mip-maps of a texture, first introduced in [Wil83], are commonly applied to avoid aliasing artifacts arising from undersampling textured areas. In [TMJ98], Tanner *et al.* first make use of the mip-map hierarchy to allow for arbitrarily large virtual textures maintaining an active working set, similar to Clark. While Tanner *et al.* propose the use of a specialized graphics workstations, the concept of virtual texturing has in recent years regained popularity (e.g. [Bar08, Mit08, vW09, OvWS12, SOC*13]) due to the increasing flexibility and general availability of GPUs.

7.2.2 BTF Compression, Streaming and Rendering

For the task of real-time rendering of BTF materials, a number of different solutions have been proposed. For a comprehensive overview we refer to [HF13]. At their core, almost all approaches have in common that they aim to reduce the huge amount of data in a BTF description to a more compact representation that will fit on the GPU. One approach is to fit SVBRDFs to the BTF data. While this representation is well suited for evaluation on the GPU, the quality can suffer drastically by the reduction to an SVBRDF, as the non-local effects of the light scattering in the material are lost. In a recent publication, Wu *et al.* therefore combine a mixture of several fitted SVBRDF models with residual ABRDFs and propose to compress those via vector quantization [WDR11].

A second group of compression techniques is based on factorization. Here, the BTF is considered as a matrix or tensor of which a low-rank approximation is found. Recent comparisons [PSR13] indicate that on BTF data, *full matrix factorization* (FMF) [KMBK03] often yields the best RMSE for a given compression ratio. The only mentioned exception is a BTF compression scheme based on *K-SVD* [RK09a] that outperforms the FMF by a factor of three to four at comparable quality. However, an efficient real-time rendering technique for this compression has not yet been found.

In [GMSK09], Guthe *et al.* employ a perceptually motivated BTF compression based on matrix factorization. Compression rates of about $500:1$ are achieved with a high approximation quality. The authors observe that GPU memory can be saved by employing downsampled versions for some of the factorized data. In this chapter, we will also save GPU memory by exploiting the fact that lower resolution versions of factorization components can be used. However, instead of reducing the level of detail once at compression time, based on assumptions about viewing distance and angles, we store the factorized BTF data at multiple precomputed resolutions. This allows us to dynamically decide at runtime which level of detail is necessary and can thus consider the actual viewpoint of the user.

Recently, data-driven compression methods for BTFs that are not based on factorization have been proposed as well. In [HF07], the authors follow a statistical modeling approach that achieves impressive compression ratios but in its nature is not capable of exactly reproducing the surface features of a given BTF. While this might be tolerable or even desired for the purpose of texture synthesis, it would for example not be applicable in the case of virtual surrogates for cultural heritage. In [HFM10], Havran *et al.* employ a compression based on multilevel vector quantization and in [TFLS11] Tsai *et al.* propose to use a decomposition in multivariate radial basis functions. Both methods provide high-quality results for the reproduction of material reflectance at real-time frame rates. Unfortunately, no direct quality comparisons to FMF are given. However, the reported compression ratios are in the same region as achieved with FMF, so it is not to be expected that these techniques will reduce memory demand sufficiently to eliminate the memory issues of BTF rendering. In the following, we use FMF compression, as it greatly facilitates the simplicity of the proposed progressive streaming and L^2-norm-based error approximation for tile prioritization. In future work, one might consider the applicability of other compression methods for BSVTFs as well.

In Chapter 6, we already proposed a factorization-based approach for rendering BTFs in the web browser via WebGL. There we utilized the level of detail hierarchy implicated by the factorization to perform a progressive streaming of the BTF data over the Internet. An additional lossy image compression has been applied to facilitate the efficient transmission. However, the image compression did not allow fast random-access reconstruction of the compressed data any more, which is mandatory for the purpose of real-time rendering. Therefore, after transmission, the factorized data has to be unpacked into GPU memory again, occupying up to 3.9 GB for a single high-quality BTF with 2048×2048 texels and the amount of components used in the quality evaluation of Chapter 5. In contrast, with the technique proposed in this chapter, scenes that contain several high-resolution BTF materials with an equally high number of factorization components, such as the one shown in Figure 7.1 and 7.3, can be rendered in real-time with a much lower memory footprint (629 MB for a scene with 22.3 GB of factorized data).

7.2.3 Out-of-Core Rendering of Reflectance Data

For their editing system *BTFShop* [KBD07], Kautz *et al.* proposed an out-of-core rendering architecture for BTFs. For this, the uncompressed BTF data is split into tiles which are successively streamed to memory for editing and rendering. However, BTFShop is an editing application and not meant for interactive viewing purposes. The rendering relies on lazy updates and assumes that usually only a subset of pixels on the screen are changed and light and view directions remain constant. A slight rotation around the object would require to completely swap the cached tiles. This severely restricts the achievable frame rates and prohibits streaming over a limited bandwidth network connection. In contrast, the proposed BSVTFs allow changing the light and view directions even for scenes with many materials in real-time with moderate bandwidth requirements.

On the related topic of surface light field rendering, Chen *et al.* presented the technique of *light field mapping* [CBCG02]. Here, the authors proposed to perform a spatial partition of the object's surface. In combination with factorization, vector quantization and image compression this allowed combining the surface light fields for each such spatial part into textures that are suitable for rendering with the GPU. Images are then generated using multipass rendering, rasterizing the triangles of one spatial part at a time. While this algorithm allows for memory-friendly out-of-core rendering, it does not include level of detail and therefore requires the costly successive swapping of all textures for the visible parts of the object's surface in every frame.

Ruiters proposed to use surface light fields in a view-dependent out-of-core level of detail approach in the context of terrain rendering [Rui08]. He employs factorized surface light fields as imposters for far-away heightfield geometry to improve the rendering performance. For this, he utilizes the spatial level of detail hierarchy of the terrain data. In contrast to our approach, however, the additional level of detail hierarchy found in the factorization is not considered.

7.3 Sparse Virtual Texturing

In this section, we briefly discuss the SVT algorithm [Bar08] and introduce our notation and implementation details.

SVT considers the problem of representing a very large image $\mathbf{I} \in \mathbb{R}^{M \times N}$ using a considerably smaller image $\mathbf{C} = \mathbb{R}^{O \times P}$, $O \ll M$ and $P \ll N$ as a cache. The technique exploits the fact that the display resolution itself is usually much smaller than the dimensions of $\mathbf{I}$. For rendering, it is therefore sufficient to hold only those parts of the image in memory, i.e. in the cache $\mathbf{C}$, that are visible on the screen at a given time. Furthermore, the parts only have to be held in memory at the displayed resolution, which has the additional benefit of avoiding aliasing artifacts due to undersampling.

To this end, the original image $\mathbf{I}$ is decomposed into a set of disjoint quadratic tiles $\mathfrak{T} = \{\mathbf{T_i} \in \mathbb{R}^{T\times T} \subset \mathbf{I} | \forall_{\mathbf{i}\neq\mathbf{j}} \mathbf{T_i} \cap \mathbf{T_j} = \emptyset \wedge \mathbf{I} = \bigcup \mathbf{T_i}\}$ of size T. The tiles $\mathbf{T_i}$ are indexed by a two-dimensional multiindex $\mathbf{i}$. Similar sets of tiles are generated for downsampled versions of the original image $\mathbf{I}$. The sets for different levels of resolution $l = 0, 1, \ldots, L$, with L referring to the resolution of the original image $\mathbf{I}$ are then denoted as $\mathfrak{T}_l$. In case portions of the image can sufficiently be represented in a lower resolution l, tiles from the set $\mathfrak{T}_l$ can be used. Note that tiles from this set will allow a larger coverage of the virtual image $\mathbf{I}$ at the same size T. We compute and decompose all downsampled versions of the original image with resolutions of $\frac{M}{2^{L-l}} \times \frac{N}{2^{L-l}}$, $l \in \{0, 1, ..., L\}$ until its content can eventually be expressed using the single tile in $\mathfrak{T}_0 = \{\mathbf{T}_{(0,0)}\}$, i.e. $\max(\frac{M}{2^L}, \frac{N}{2^L}) \leq T$. The content of the cache $\mathbf{C}$ is then compiled from that subset of tiles that form the visible part of the image $\mathbf{I}$ at a sufficient resolution. Hence, $\mathbf{C}$ is also referred to as the *tile cache*. If all space in the cache is already occupied on arrival of a new tile, free space will be made available by unloading existing tiles based on their priority (see Section 7.5.2). Tiles from multiple virtual textures are handled in a single tile cache. In our implementation, we take special care when manipulating the tile cache that at all times all parts of $\mathbf{I}$ are covered at least on a low-resolution level. This strategy prevents drastic drawing errors due to cache misses in case of rapid user interaction.

To determine the information which tiles of which level have to be displayed, a *feedback image* $\mathcal{F} : (x, y) \in [0, X) \times [0, Y) \mapsto (\mathbf{i}, l, \tau) \in \mathbb{R}^4$ is computed in regular time intervals. Let $\Pi : \mathbb{R}^2 \to \mathbb{R}^2$ be a function that maps screen pixel coordinates (x, y) to texel coordinates $(s, t) \in [0, M) \times [0, N)$ (i.e. a similar mapping as described in Section 5.5.2 for images of a virtual instead of a real camera). For each pixel (x, y), the down-sampling level l and the index $\mathbf{i}$ of the tile $\mathbf{T_i}$ with the content for that pixel can be computed as

$$
\begin{aligned}
l &= L - \log_2 \max\left(\left\|\frac{\partial\Pi(x,y)}{\partial x}\right\|, \left\|\frac{\partial\Pi(x,y)}{\partial y}\right\|\right), && (7.1)\\
\mathbf{i} &= \left(\left\lfloor 2^l \frac{s}{M}\right\rfloor, 2^l, \left\lfloor 2^l \frac{t}{N}\right\rfloor, 2^l\right)^T. && (7.2)
\end{aligned}
$$

To allow the texturing of larger surfaces with a repetitive pattern, we employ texel coordinates $(s,t) \in \mathbb{R}^2$. These coordinates would then map to the original texture domain as $(s,t) \mapsto (\mathrm{mod}(s,M), \mathrm{mod}(t,N))$. For this, we modify Equation 7.3 as follows:

$$\mathbf{i} \;=\; \left(\mathrm{mod}\left(\left\lfloor 2^l \frac{s}{M} \right\rfloor, 2^l\right), \mathrm{mod}\left(\left\lfloor 2^l \frac{t}{N} \right\rfloor, 2^l\right)\right)^T . \tag{7.3}$$

A fragment shader is used to evaluate Equations 7.1 and 7.3. To support multiple source textures, an additional texture index τ is stored in the fourth channel.

In order to reassemble the original appearance of $\mathbf{I}$ from the possibly fragmented tiles that might also exhibit different resolutions, an indirection has to be performed for all texture fetches during rendering. For each screen pixel (x,y), the texel coordinates $(s,t) = \Pi(x,y)$ are mapped to coordinates in the tile cache where the value for $\mathcal{I}(s,t)$ is stored. This requires to locate the appropriate tile in the tile cache and find the correct offset within the tile itself. For this purpose, we maintain a *lookup table* $\mathcal{L} : \mathbb{N}^2 \to \mathbb{R}^3$ that holds the level l' in which a tile for (s,t) is available in the tile cache (which might differ from the optimal level l) and the texel coordinates $\mathbf{i}'$ of its top-left corner in $\mathbf{C}$. Please note that $\mathcal{L}$ is considerably smaller than the original texture $\mathbf{I}$, as only one entry for every T^2-th texel is required. From this information, the coordinate $\mathbf{x}$ of the texel in the tile cache that needs to be fetched is computed as

$$\mathbf{x} = \mathbf{i}' + \left(M\left(2^{l'} \frac{s}{M} - \left\lfloor 2^{l'} \frac{s}{M} \right\rfloor\right), N\left(2^{l'} \frac{t}{N} - \left\lfloor 2^{l'} \frac{t}{N} \right\rfloor\right)\right)^T . \tag{7.4}$$

We employ a separate lookup table for each virtual texture.

7.4 BTF Real-time Rendering

We base our rendering approach on the compact FMF representation [KMBK03] that can be obtained from a BTF data matrix $\mathbf{B}$ via *singular value decomposition* (SVD). Given the full SVD $\mathbf{B} = \mathbf{U}\Sigma\mathbf{V}^T$, a low-rank approximation $\mathbf{B} \approx \mathbf{U}'\mathbf{V}'^T$ is obtained by truncating the matrices $\mathbf{U}$ and $\mathbf{V}$ after C columns. Being a diagonal matrix, Σ can be multiplied with $\mathbf{V}$ prior to truncation, i.e. $\mathbf{V} := \mathbf{V}\Sigma$. Please refer to the description in Section 5.5.4 for additional details about the compression.

In the context of real-time rendering, the factorized representation has the important benefit of allowing random access to arbitrary values of the BTF without the necessity to reconstruct the full matrix $\mathbf{B}$. Consider the c-th column of the matrices $\mathbf{U}$ and $\mathbf{V}$ as images $\mathcal{U}_c$ and $\mathcal{V}_c$. Then, the BTF ρ can be approximated according to Equation 6.1. If $\mathcal{U}_c$ and $\mathcal{V}_c$ are stored as textures on the GPU, the equation can be efficiently evaluated in a shader program. For directions and positions other than the discrete samples stored in $\mathbf{B}$, the values have to be interpolated.

As in Section 6.3, we rely on the texture mapping units of the GPU to perform the spatial 2D interpolation for us when accessing the textures $\mathcal{V}_c$, by choosing a suitable layout of the eigen-textures. Regarding the 4D interpolation in $\mathcal{U}_c$, however, we follow a different approach. Instead of resampling the ABRDF basis samples into parabolic coordinates, which creates a significant memory overhead (see Section 6.5), we follow an idea presented in [ND06]. For this, we precompute two separate 2D Delaunay triangulations D_l and D_v for the sets of light and view direction samples of the BTF given in parabolic coordinates. We then raster each triangulation D into two RGB textures $\mathcal{D} : \Omega_{\text{parabolic}} \rightarrow \mathbb{N}^3$ and $\mathcal{B}\Omega_{\text{parabolic}} \rightarrow \mathbb{R}^3$, containing the three direction indices of the enclosing Delaunay triangle and the three barycentric weights respectively. This way, during rendering the interpolated value for arbitrary view and light directions given in parabolic coordinates, can be evaluated in a GPU shader: For all 9 combinations of direction indices from $\mathcal{D}_l(\omega_i)$ and $\mathcal{D}_v(\omega_o)$, we perform a lookup into $\mathcal{U}_c$ and blend the values according to the barycentric weights. The small GPU memory overhead reintroduced by the index and weight textures can further be reduced in the case of rendering multiple BTF with the same angular sampling by sharing the textures between them.

7.5 Extension of SVT to BSVTFs

While for curved surfaces and perspective cameras almost all entries of the bidirectional reflectance properties in $\mathbf{U}'$ have to be accessed, the utilization of parts of the eigen-textures stored in $\mathbf{V}'$ follow the same consideration as conventional textures. Therefore, the idea of sparse virtual texturing could be directly applied in this case. The eigen-textures could be treated as an image with C channels and a spatial level of detail hierarchy can be constructed and decomposed into tiles.

However, this would not provide the best approximation as it does not take advantage of the property that the SVD compacts most of the information in the first few columns of $\mathbf{U}'$ and $\mathbf{V}'$, so that the contribution of later columns to the quality of the approximation decreases quickly. This observation has already found application in Chapter 6, where the columns of $\mathbf{U}'$ and $\mathbf{V}'$ were transmitted sequentially. In the case of BSVTFs, the situation is far more general. Using SVT introduces a new degree of freedom, since every column of $\mathbf{V}'$ could be stored in a different spatial resolution.

We aim to combine both, the spatial resolution and the approximation rank level of detail, in a consistent manner. Instead of considering the matrix $\mathbf{V}'$ as one texture with multiple channels, we regard every column as an individual virtual 2D texture $\mathcal{V}_c$. This way, the tiles of different columns are weighted against each other for utilization of the tile cache. From now on, we will use $\tilde{\mathcal{V}}_c$ to refer to the reconstruction of the virtual texture $\mathcal{V}_c$ from the contents of the tile cache. We make this distinction, as in general with a limited tile cache size we cannot guarantee for a perfect reconstruction of the visible parts of every texture.

In principle, the goal of hierarchical level of detail rendering can be defined as the minimization of the rendering error that can result from the restriction to a fixed tile cache size. In the case of BTFs, possible sources of error are an insufficient spatial resolution or an insufficient number of factorization components for the low-rank approximation. Let image $\mathcal{S}$ denote the content of the screen when directly using $\mathcal{V}_c$ for rendering. Let further $\tilde{\mathcal{S}}$ be the content of the screen when using SVT to reconstruct $\tilde{\mathcal{V}}_c$. The rendering error under the L^2-norm can be expressed as

$$\sum_{y=0}^{Y-1}\sum_{x=0}^{X-1}(\mathcal{S}(x,y)-\tilde{\mathcal{S}}(x,y))^2, \tag{7.5}$$

i.e. the sum of squared differences over all screen pixels.

In our implementation, we do not directly minimize this term but instead propose a simplification. Let Π be the mapping function used during rendering that maps from screen pixels (x, y) to the spatial position $\mathbf{x}$ in the BTF. Let furthermore the vector $\mathbf{a}^{\mathbf{x}}$ denote the tabulated ABRDF encoded at that position in the factorized BTF and $\mathbf{U}_c$ be the c-th column of $\mathbf{U}$. Then

$$\begin{aligned}
&\sum_{y=0}^{Y-1}\sum_{x=0}^{X-1}\left\|\mathbf{a}^{\Pi(x,y)}-\tilde{\mathbf{a}}^{\Pi(x,y)}\right\|^2\\
=\;&\sum_{y=0}^{Y-1}\sum_{x=0}^{X-1}\left\|\sum_{c=0}^{C}\mathbf{U}_c\mathcal{V}_c(\Pi(x,y))-\sum_{c=0}^{C}\mathbf{U}_c\tilde{\mathcal{V}}_c(\Pi(x,y))\right\|^2
\end{aligned}$$

denotes the L^2-error of the ABRDF vectors for every pixel. The ABRDFs that are reconstructed directly from the factorization are designated $\mathbf{a}$ whereas the ABRDFs that are reconstructed using SVT are denoted as $\tilde{\mathbf{a}}$.

Utilizing the SVD property of matrix $\mathbf{U}$ being unitary, this minimization can be expressed as

$$
\begin{aligned}
& \sum_{y=0}^{Y-1}\sum_{x=0}^{X-1}\left\|\sum_{c=0}^{C}\mathbf{U}_c\mathcal{V}_c(\Pi(x,y)) - \sum_{c=0}^{C}\mathbf{U}_c\tilde{\mathcal{V}}_c(\Pi(x,y))\right\|^2 \\
= & \sum_{y=0}^{Y-1}\sum_{x=0}^{X-1}\left\|\sum_{c=0}^{C}\mathbf{U}_c\left(\mathcal{V}_c(\Pi(x,y)) - \tilde{\mathcal{V}}_c(\Pi(x,y))\right)\right\|^2 \\
= & \sum_{y=0}^{Y-1}\sum_{x=0}^{X-1}\sum_{c=0}^{C}\left\|\left(\mathcal{V}_c(\Pi(x,y)) - \tilde{\mathcal{V}}_c(\Pi(x,y))\right)\right\|^2.
\end{aligned}
$$

Therefore, it is sufficient to consider the error for every single virtual 2D texture $\mathcal{V}_c$ individually:

$$E = \left\|\left(\mathcal{V}_c(\Pi(x,y)) - \tilde{\mathcal{V}}_c(\Pi(x,y))\right)\right\|^2. \tag{7.6}$$

Our proposed BSVTF rendering algorithm minimizes this error under the constraint of limited memory.

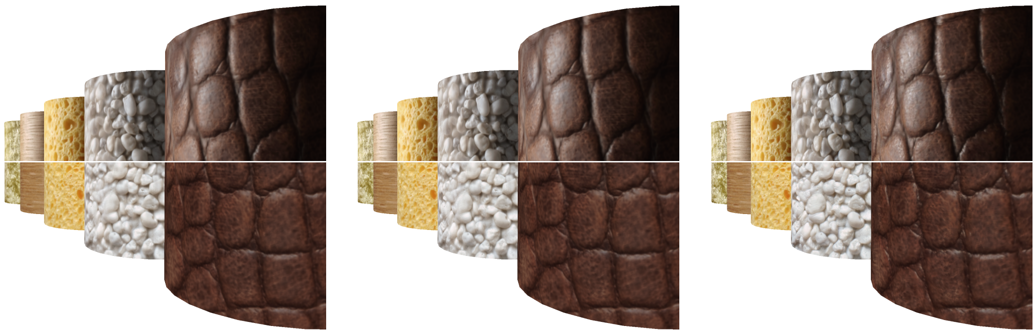
Please note, that even though the different columns in $\mathbf{U}'$ and $\mathbf{V}'$ have different importance for the quality of the BTF approximation, using the proposed minimization formulation we elegantly avoid the introduction of additional weighting terms to balance the individual textures against each other. Furthermore, the proposed simplification from a rendering error minimization to an ABRDF error minimization has the additional advantage that the lighting in the virtual scene can be changed without the necessity to change anything in the tile cache utilization. Changes in view direction benefit from the availability of ABRDFs as well, as not all tiles in the tile cache have to be exchanged but only those which are affected by changes in visibility or mip-level.

Without loss of generality, in the remainder of this chapter we will assume that the individual eigen-textures $\mathcal{V}_c$ are laid out side by side in a sufficiently large virtual texture $\mathcal{I}$ that will be used for SVT.

7.5.1 Level of Detail Strategy

While in the case of level of detail for geometry a variety of strategies for the artifact free refinement without inconsistencies exists, for SVT not too many details can be found in the literature. In this work, we essentially distinguish between two operations: `add` and `swap`.

The operation `add` will insert a tile $\mathbf{T}$ at free space in the tile cache. As a postcondition, we check whether any ancestor tile of $\mathbf{T}$ in the tile hierarchy is now



BSVTF (244 MB) $C = 100$ 4032^2 pixel tile cache size

FMF BTF (316 MB) $C = 100$

uncompressed (167 GB)

Figure 7.2: *BTF renderings using BSVTFs (left), FMF compression (center) and no compression (right). Upper and lower half of the images are lit from different light directions. All of the depicted materials (front to back: leather, gravel, sponge, wood, velvet) exhibit complex view- and light-dependent material appearance. While the uncompressed materials visibly appear to be sharper, there is hardly any noticeable difference between our technique and the FMF.*

completely covered by its children. If so, the ancestor is removed from the tile cache, as it will not contribute to the pixels drawn on screen any more. An `add` operation is only performed on tiles that have an ancestor in the tile cache. After the operation one or none (if an ancestor has been removed) of the free entries in the tile cache will be occupied.

The operation `swap` will remove two tiles $\mathbf{T}_{\mathbf{i}_1,l_1}, \mathbf{T}_{\mathbf{i}_2,l_2}$ from the tile cache and instead insert a tile $T_{\mathbf{i}',l'}$ from a lower level $l' < \min(l_1, l_2)$ in the tile hierarchy that covers those parts of $\mathbf{I}$ that were shown in $\mathbf{T}_{\mathbf{i}_1,l_1}$ and $\mathbf{T}_{\mathbf{i}_2,l_2}$, that is $\mathbf{i}' = \left\lfloor 2^{l'-l_1}\mathbf{i}_1 \right\rfloor = \left\lfloor 2^{l'-l_2}\mathbf{i}_2 \right\rfloor$. This operations will result in one free entry in the tile cache.

After all operations have been performed, the lookup table is updated accordingly.

7.5.2 Tile Prioritization

In order to minimize the ABRDF error from Equation 7.6, we weight the possible tiles that can be loaded into the tile cache against each other. For this, we roughly follow two measures:

1. the number of the pixels on the screen covered by the tile,
2. the average reduction of the approximation error E for those pixels.

As long as there is still free space in the tile cache, we perform the `add` operations on tiles prioritized by these two criteria. For this, we set the priority P of a tile $\mathbf{T_i}$ at level l as $P = w(\mathbf{i}, l, l-1) \cdot v(\mathbf{i}, l)$, where

$$w(\mathbf{i}, l, l') = \frac{1}{T^2} \sum_{x=0}^{T-1} \sum_{y=0}^{T-1} \left(\mathcal{T}_{\mathbf{i},l}(x,y) - \mathcal{T}_{2^{l-l'}\mathbf{i},l'}(x',y') \right)^2 \tag{7.7}$$

designates the average L^2-difference of the tile to its lower resolution ancestor at level l' in the tile hierarchy and

$$v(\mathbf{i}, l) = \left| \left\{ (\mathbf{i}', l') \in \mathcal{F} | l' \geq l \wedge \mathbf{i} = \left\lfloor 2^{l-l'} \mathbf{i}' \right\rfloor \right\} \right| \tag{7.8}$$

denotes the number of votes, i.e. pixels in the feedback image (see Section 7.3) that show the index values $\mathbf{i}$ and l of the tile or its descendants in the tile hierarchy. The point (x', y') in Equation 7.7 identifies the coordinate in the lower resolution tile $\mathcal{T}_{2^{l-l'}\mathbf{i},l'}$ that maps to the same position in the virtual texture as (x, y) does in $\mathcal{T}_{\mathbf{i},l}$. The value P approximates the reduction in the error E in Equation 7.6 if $\mathbf{T}_{\mathbf{i},l}$ would be in the tile cache.

This definition for P is only valid for `add` operations on tiles of level l for which the parent at level $l-1$ is currently visible. Otherwise computing the votes v would be more complex, as several in-between steps would have to be considered. Since, `add` operations for a tile with a directly available parent are favorable for the application of streaming in Section 7.6, we restrict ourselves to this simple case.

The weight w is precomputed for every tile of every eigen-texture. The pixel votes v are obtained at runtime from the feedback buffer and apply to all eigen-textures. Our particular choice of v will also make sure that no space is wasted on tiles with unnecessarily high resolutions, i.e. levels that are higher than the ones in the feedback buffer, since those will have a priority of $P = 0$.

In case there is no free space left in the tile cache but further tiles could be added, we have to decide whether a `swap` operation should be performed to free space or not. Naturally, the `swap` operation will increase the error E, as it replaces higher resolution tiles $\mathbf{T}_{\mathbf{i},l}$ with a lower resolution substitute $\mathbf{T}_{\mathbf{i}',l'}$. We can approximate the rise in error by

$$\begin{array}{rcl} c(l', \mathbf{i}, l) & = & \sum_{k=l'}^{l} w(\left\lfloor 2^{k-l}\mathbf{i} \right\rfloor, k, k-1) v(\mathbf{i}, l) \\ & \approx & w(\mathbf{i}, l, l') v(\mathbf{i}, l) \end{array}, \tag{7.9}$$

which is the accumulated approximated error of the portion of all in-between tiles with levels $k = l', \ldots, l$ that are currently covered by pixels from the high-resolution tile and would therefore be revealed in case of a `swap`. Even though this particular approximation is not very accurate, it has the benefit that only one weight value $w(\mathbf{i}, l, l-1)$ – the same as we employ for computing P – has to be computed and stored per tile. Since we will replace exactly two tiles $\mathbf{T}_{\mathbf{i}_1,l_1}$ and $\mathbf{T}_{\mathbf{i}_2,l_2}$, the total increase in error or *cost* of this operation can be expressed by $c = c(l', \mathbf{i}_1, l_1) + c(l', \mathbf{i}_2, l_2)$.

In order to decide whether to perform a `swap` operation, we first find the three candidates with the lowest cost $c^\star$ and compare this value with the highest priority $P^\star$ of the tiles that could be added. If $c^\star < P^\star$, this means that the approximated error of not having the tile with priority $P^\star$ in the tile cache is higher than the error induced by performing the `swap` operation with cost $c^\star$. Hence, we will reduce the total error E by first performing the least costly `swap` operation to obtain free space and then performing the highest priority `add` operation. Otherwise, we are already displaying the best solution and will not perform any operation.

7.6 Streaming

Similar to other hierarchical level of detail techniques, the proposed BSVTFs are very well suited for streaming over a network. Tiles that have to be inserted into the tile cache by the `swap` or `add` operation are in this case requested from a streaming server.

To facilitate the transmission of tiles over a low-bandwidth network, we employ an additional compression to the tiles prior to submission that is inverted before the tile is inserted into the tile cache. As observed in Chapter 6, the eigen-textures obtained by the SVD show similar image statistics as natural images. Therefore, in principle every image compression technique could be employed for this purpose. For example, in [KMBK03] Koudelka *et al.* utilize JPEG compression while we employed a wavelet codec similar to JPEG2000 in Chapter 6.

Although we demonstrated that the wavelet codec outperformed JPEG compression on the full eigen-textures, this performance breaks down in the case of the small tiles used for BSVTFs. Thus, we perform a *discrete cosine transformation* (DCT) on 8×8 pixel blocks of the tiles $\mathcal{T}$ and then apply a quantization with respect to a quality threshold similar to JPEG. The quantized data is then stored using deflate. The only mentionable difference to other off-the-shelf implementations is the fact that our compression operates on floating-point values (half precision).

To further improve the compression ratio and exploit the large redundancy present in the sets of tiles for different resolutions, instead of directly compressing the tiles, we compress the differences $\mathcal{T}'$ of a tile to its upsampled parent in the tile hierarchy $\mathcal{T}'_{\mathbf{i},l}(x,y) = \mathcal{T}_{\mathbf{i},l}(x,y) - \mathcal{T}_{\lfloor\frac{1}{2}\mathbf{i}\rfloor,l-1}(\frac{x}{2},\frac{y}{2})$. This procedure exploits the fact that due to our construction of the tile hierarchy, most of the low-frequent components of the DCT are already covered by the parent tile. Thus, the amount of information that needs to be compressed is drastically reduced by using the difference image. The size of a compressed tile depends on the choice of T and the user-determined quality threshold for the quantization. In our experiments we were able to obtain a compression ratio up to 6:1 with no perceivable artifacts.

In order to unpack the DCT compressed difference images after transmission, the respective parent tile is required. During an `add` operation this does not pose a problem, since we decided in Section 7.5.2 that this operation should only be performed if a parent of the tile is still in the tile cache and hence available at the client side. When performing a `swap` operation, in the worst case, all ancestor tiles will have to be requested as well in order to sequentially unpack all of them until the parent is available. However, in order for a `swap` operation to occur in the first place, higher resolution tiles had to be added to the tile cache first. In turn, this means that the full branch of the level of detail hierarchy up to this resolution and thus also all ancestors of the tile that has to be swapped in, had to be previously transmitted to the client. We therefore employ a *least recently used* cache to keep as many received tiles as possible in the client-side RAM.

Even before applying the transmission compression the size of the tiles is only in the order of a few kilobytes. The eigen-ABRDFs in $\mathbf{U}'$ on the other hand have a combined size of a few megabytes per color channel (4.4 MB for all data sets considered in this thesis). Fortunately, in contrast to the tiles that have to be swapped in and out on demand, $\mathbf{U}'$ only has to be transmitted once and does not change during the rendering process. Still, loading this amount of data for multiple objects in advance over a low-bandwidth connection is not a good solution.

We therefore transmit the eigen-ABRDFs $\mathcal{U}_c$ sequentially and interleave them in the tile stream. This way, only a few hundred kilobytes have to be transmitted at once, allowing to start rendering considerably faster. In this case, the images $\mathcal{U}_c$ have to be prioritized in a similar fashion as the tiles to decide whether to stream the next tiles for $\tilde{\mathcal{V}}$ or another column of $\mathbf{U}'$. From Equation 6.1, it becomes apparent that $\mathcal{U}_c$ can only contribute to the BTF approximation if $\mathcal{V}_c$ is available as well. We can therefore approximate the priority of the eigen-ABRDFs by the sum of votes for all tiles $\mathbf{T}_{\mathbf{i},l}$ in the tile cache that are currently used to represent $\tilde{\mathcal{V}}_c$, weighted by the average intensity of the tile, i.e. $\sum v(\mathbf{i}, l)\|\mathbf{T}_{\mathbf{i},l}\|_F^2$. This weighting can be understood as the contribution the tile makes to not having a value for $\tilde{\mathcal{V}}_c$ available at all, which would in turn render the request for $\mathcal{U}_c$ pointless.

In this work, we did not undertake any further effort to improve the transmission of the eigen-ABRDFs. However, a first obvious extension would be to apply a lossless compression, such as deflate, to the data.

7.7 Evaluation

To assess the feasibility of our approach, we tested the level of detail rendering and streaming on the 29 digitized objects from the evaluation in Section 5.6 as well as a collection of 100 measured material samples. Details on the objects can be found in tables 5.1, 5.2 and 5.3.

The 100 planar material samples were acquired with the Dome 1 setup and processed in a similar fashion as the objects. However, instead of a 3D geometry, a planar proxy surface is used and in contrast of covering the complete sample surface, a representative, well-tileable, quadratic section was taken. The resulting resampled BTFs have 512×512 texels and a resolution of about 290 DPI. Most of them have recently been made publicly available as the `UBO2014` database[1].

All of our BTF materials have a high dynamic range and are represented in RGB color. In all cases, the angular sampling contained the same set of 151×151 directions ω_i, ω_o (that of the Dome 1, see Table 4.2). Before uploading $\mathbf{U}$ to the GPU we furthermore compute an additional 152-nd basis illumination in which we stored a preintegrated value of all other lights for the efficient evaluation of a view-dependent ambient term in the fragment shader (see Section 6.3.2). We use the FMF compression as described in Section 5.5.4 with $C = 100$ components. Note that this also includes the dynamic range compression by computing the logarithm of the BTF data. We will subsequently refer to this compressed format as *FMF BTF*.

From the FMF BTFs, we generate the BSVTF by first creating a layout of the eigen-textures. To save texture fetches in the shader, we store four values $\mathcal{V}(\mathbf{x})$ per texel as RGBA channels. Then we compute the sets of tiles for different resolutions $\mathfrak{T}_l$. In our experiments, we use a tile size $T = 64$. We additionally extend the tile with four pixels of padding at each border to allow for trilinear filtering using the tile cache texture. This results in 40.5 KB per tile when employing half-precision floating-point numbers. Using the DCT compression, this size is reduced to about 7 KB to 10 KB. We also precompute the weights between direct descendents $w(\mathbf{i}, l, l-1)$ from Equation 7.7 and store them in half precision as well. Finally, the eigen-ABRDFs with 151×151 angular directions are stored for the three color channels. Here, we employ the strategy of packing four components into the RGBA channels as well, resulting in packets of 534 KB that are interleaved with the tile transmission. Details on the processing times and the resulting total file sizes can be found in Table 7.1.

The costs for generating the level of detail representation from factorized BTFs is negligible compared to the time requirements of the factorization. While computing

[1] `http://cg.cs.uni-bonn.de/en/projects/btfdbb/download/ubo2014/`

1008^2 tile cache, 77 MB, SSIM 0.96

(a) BSVTF

4032^2 tile cache, 230 MB, SSIM 0.99

(b) BSVTF

8064^2 tile cache, 718 MB, SSIM 0.99

(c) BSVTF

$C = 6$, 238 MB, SSIM 0.95

(d) FMF

$C = 100$, 3.85 GB, SSIM 0.99

(e) FMF

uncompressed, 2.53 TB

(f) reference

Figure 7.3: *Quality comparison on a* 20 *megatexel BTF scene that would exceed the memory of most GPUs. With a too small tile cache size (a), our technique is not able to resolve the fine mesoscopic details. Using an appropriate tile cache size (b), the BSVTFs still have a small GPU memory footprint and at the same time achieve a comparable quality to directly rendering the FMF BTF data (e). Larger tile cache sizes (c) do not significantly improve the quality any further. Image (d) demonstrates the loss in quality when using FMF with a higher compression ratio to achieve a similarly small memory footprint as (b). Due to the insufficient number of* $C = 6$ *columns, this rendering shows blurred mesoscopic details, washed-out highlights and shifted colors. The SSIM values are given with respect to the rendering with uncompressed BTFs (f).*

resolution	uncompressed [GB]	FMF [MB]	BSVTF [MB]	preprocessing [hours]
4096×4096	2,138	3,213	2,365	3:14
3072×3072	1,202	1,813	952	1:44
2560×2560	835	1,263	1,052	0:47
2048×2048	534	813	640	0:31
1600×1600	326	501	435	0:18
1024×2048	267	413	161	0:13
1024×1024	134	213	197	0:06
800×800	82	135	146	0:04
512×512	33	63	74	0:01
256×256	8	26	33	0:01

Table 7.1: *Average processing times and file sizes for all employed spatial resolutions. The columns* Uncompressed, FMF *and* BSVTF *give the file sizes for the different levels of compression. Here,* BSVTF *designates the streaming-ready file including headers, precomputed weights, level of detail hierarchy and DCT compression. The column* preprocessing *refers to the time required to obtain the BSVTF from the FMF BTF.*

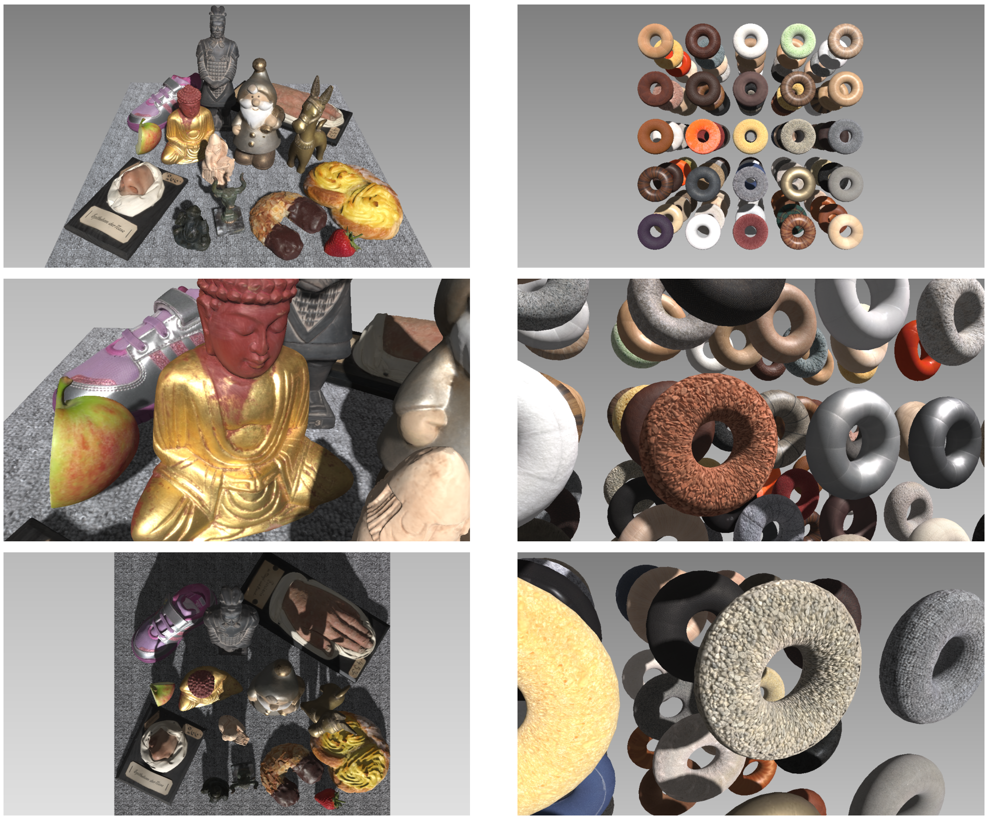
the FMF compression for a 800×800 texel BTF took 21 minutes using a highly optimized GPU implementation [RRK09], generating the DCT compressed tiles with a singlethreaded CPU implementation took only four additional minutes on the same hardware (two 2 GHz Intel Xeon E5-2650 CPUs with eight cores each, 128 GB RAM, NVIDIA GeForce GTX 680 GPU).

We compiled seven scenes from the available data sets:

1. all captured objects (see Figure 7.1),
2. 14 of the captured objects on a BTF textured plane (see Figure 7.4a),
3. all 100 materials, arranged on a grid of tori (see Figure 7.4b),
4. the four objects of the `OBJECTS2011`[2] data set (see figures 7.3 and 7.5),
5. only the Buddha object,
6. only the Terracotta Soldier object,
7. five selected materials, presented on cylinders (see Figure 7.2).

The performance of the BSVTFs was measured using animations of a camera moving along a predefined path. Videos of the sequences for scenes two to seven can be found in the additional multimedia material of the publication "Level-of-Detail Streaming and Rendering using Bidirectional Sparse Virtual Texture

[2] `http://cg.cs.uni-bonn.de/en/projects/btfdbb/download/objects2011/`



(a) scene 2 **(b)** scene 3

Figure 7.4: *Example screenshots of two animation sequences used to evaluate the BSVTFs' rendering performance. The evaluation results can be found in Table 7.2.*

Functions" [SRK13]. All rendering tests were conducted on an Intel Core i7 4930K (hexa-core, $3.4\,\mathrm{GHz}$) machine with an NVIDIA GeForce GTX 780 GPU with 4 GB of GPU memory. We measured the performance with screen resolutions of 1280×720 and 1448×1448 pixels. Using a tile cache size of 4032×4032 pixels, we achieve a comparable quality to FMF BTFs at real-time frame rates for the lower screen resolution. Details on rendering performance and GPU memory consumption can be found in Table 7.2.

In all of our experiments with BSVTFs, the average total CPU utilization of the system was at 17%. This amount was distributed with 50% of the time spent on rendering the scene and the user interface, 47% on evaluating the feedback image and deriving the list of operations, 1.9% in image decompression and the remaining 1.1% in network or disk IO. As expected, rendering with FMF BTFs resulted in only 9% CPU load, since here no other task than rendering is performed.

Scene	number of megatexels	materials	triangles	FPS BSVTF[1]	FMF[1]	BSVTF[2]	FMF[2]	GPU memory BSVTF	FMF	savings
1	112.0	29	2,412,160	**21** $\pm$**4**	-	**18** $\pm$**6**	-	629 MB	22.3 GB	97.2%
2	52.8	15	1,072,597	**33** $\pm$**5**	-	**26** $\pm$**8**	-	404 MB	10.5 GB	96.2%
3	25.0	100	180,000	**51** $\pm$**8**	-	**39** $\pm$**9**	-	1,743 MB	6.2 GB	72.4%
4	19.4	4	271,857	**60** $\pm$**5**	55 $\pm$9	44 $\pm$9	**50** $\pm$**10**	230 MB	3.9 GB	94.2%
5	9.0	1	49,999	64 $\pm$1	**64** $\pm$**0**	45 $\pm$8	**57** $\pm$**5**	181 MB	1.8 GB	90.0%
6	4.0	1	99,952	**64** $\pm$**0**	**64** $\pm$**0**	45 $\pm$8	**57** $\pm$**6**	181 MB	0.8 GB	77.7%
7	1.3	5	4,640	64 $\pm$1	**64** $\pm$**0**	57 $\pm$8	**61** $\pm$**4**	244 MB	0.3 GB	23.1%

[1]Rendering resolution of 1280×720. [2]Rendering resolution of 1448×1448.

Table 7.2: *Results of the performance evaluation on the test scenes described in Section 7.7. The first three columns denote the number of total megatexels, individual materials and triangles in the scene. The* FPS *columns give the average and standard deviation on the tested animation sequence. Values for FMF BTFs are only available for scenes that fit into the GPU memory of our test system. The maximum frame rate was bounded at* 64 *FPS by the GPU.*

Unfortunately, the maximum frame rate was restricted to 64 FPS by the GPU. When rendering with a screen resolution of 1280×720 pixels, the frame rates for scenes four to seven often peaked at this limit. Here, the tests with the higher resolution reveal that the BSVTFs achieve between 7% and 21% fewer FPS in these scenes. Performance seems to be mainly correlated with the triangle count as well as the memory footprint, but not with the number of different BTFs.

Interestingly, for scene four at 1280×720 pixels, rendering with FMF BTFs achieves less FPS than the BSVTFs. Since the BTFs in this scene nearly consume all of our GPU's on-board memory, the loss in frame rate might be caused by swapping of textures to main memory and back during rendering. It could also be an effect of the highly incoherent memory access pattern. The BTF shader requires to perform scattered reads throughout the occupied GPU memory during texture access, rendering the GPU's caching mechanisms less effective.

Figures 7.2 and 7.3, which depict scenes seven and four respectively, offer a qualitative comparison of BSVTFs with FMF BTFs and uncompressed BTFs. While the uncompressed BTFs appear to be visibly sharper, there is hardly any noticeable difference between BSVTFs and FMF BTFs. Note that rendering the uncompressed BTFs has been performed using deferred shading from out-of-core data and is prohibitively costly. The hard disk is a severe bottleneck, resulting in several hours per image with solely local illumination (and ambient term).

While we employed a tile cache with 4032×4032 pixels for our evaluation of the performance, the GPU memory footprint could be reduced even further by choosing a smaller cache size. Figure 7.3 demonstrates the influence of a reduced tile cache size. Although the most obvious difference can be observed in the spatial resolution of surface details, the quality of the reflectance also suffers from too small tile cache sizes. For example, the copper parts of the Minotaur object show a shift in color and appear more dull.

In Figure 7.5, we demonstrate the streaming over the network on scene four. After a transmission of 25 MB (11% of the total footprint) the scene already achieves a high perceptual similarity – indicated by the SSIM index of 0.95 – to the converged BSVTF (i.e. no `add` or `swap` operation would further reduce the error). After transmitting about 100 MB (43% of the total footprint) the images become virtually indistinguishable.

Limitations: Although our evaluation shows that the proposed BSVTF is applicable in a number of scenarios and provides real-time rendering performance, the method also has a few limitations that need to be considered as well. First, the additional buffer updates, the regular tile uploads, and the additional texture fetches due to the indirection in the fragment shader have an unavoidable and significant impact on the frame rate. Second, our current approach only uses a level of detail hierarchy on the eigen-textures. While this is very feasible for few but high-resolution BTFs (e.g. scenes one, two and four), it is less efficient in scenes with many but comparably low spatial resolution materials (e.g. scene three). It will not help at all if, instead of spatial resolution, a high angular resolution of the BTF data would become the bottleneck. Finally, unless the movement of the user is somehow anticipated, a prefetching of data is hard to implement and resolution popping artifacts can not completely be eliminated, especially when streaming from a network connection with high latency.

7.8 Summary

In this chapter, we demonstrated that by adapting sparse virtual textures to factorized BTFs it becomes possible to render scenes with a large number of high-resolution BTFs efficiently on the GPU. For this, we suggested a strategy to trade of spatial resolution and the accuracy of the reflectance representation. Furthermore, we demonstrated that this technique can be combined with an additional image compression codec and used for network transmission.

We presented the BSVTF method proposed in this chapter "Level-of-Detail Streaming and Rendering using Bidirectional Sparse Virtual Texture Functions" [SRK13] as a research paper at the Pacific Graphics 2013. It was published in Computer Graphics Forum (Proceedings of Pacific Graphics). Please note that the evaluation for the published article was performed on slightly older hardware and with differing BTF resolutions.

Figure 7.5: *Rendering quality after streaming different amounts of data over the network. The SSIM values predict the perceptual similarity between the images and are computed with respect to the converged version.*

An important consideration of the proposed BSVTFs with regard to GPU memory is that only the spatial domain is covered by the level of detail hierarchy. We envision to overcome this limitation by extending the level of detail approach to the eigen-ABRDFs as well, keeping only those that are most important to the current viewpoint in GPU memory. For this, a hierarchical factorization could be used. First, the whole BTF is represented by a small number of columns C. Then the residuum is subdivided into smaller subsets which are factorized individually.
Another direction of future research will be improving the network streaming by integrating a progressive refinement of the tiles, similar to what we proposed in Chapter 6. This would allow smaller chunk sizes and thus faster responses to changes in viewpoint over low-bandwidth networks.

Part IV

Closure

CHAPTER 8

CONCLUSIONS

In this work, we presented all necessary steps to compose a feasible and practical set of tools (in form of hardware and algorithms) for the acquisition, transmission and presentation of 3D objects with optically complicated material appearance. The previous four chapters approached different aspects of this general goal. Chapter 4 discussed the requirements and practical implementation of a measurement apparatus. Chapter 5 created a manageable representation from the raw measurement data that can be utilized directly for rendering but also serves as a foundation for further steps. Chapters 6 and 7 tackled different aspects of a more efficient representation of the objects for streaming and real-time rendering. Chapter 6 focused on a more efficient compression and progressive transmission while Chapter 7 provided a solution to cope with the high demand of BTFs on GPU memory.

Due to the different directions, each of the four chapters already contains a summary, reciting the most important insights and showing possibilities for improving the presented approaches in future work. In this chapter, we will instead take a look at the overall picture, give a broader summary and provide a more visionary outlook to possible future avenues of research.

8.1 Summary

At the beginning of this thesis, we made the deliberate choice to use a triangle mesh for the representation of the macroscopic and the data-driven BTF for the meso- and microscopic appearance effects. The reasons have been well explained in Section 3.3. We consequently employed this hybrid representation between purely image-based and traditional computer graphics. This allowed us to use existing and well-supported state-of-the-art path tracers, e.g. Mitsuba, and widespread real-time graphics APIs, e.g. WebGL, to create images that convey a faithful appearance of the digitized objects under novel points of views and illumination.

Although far from being completely general, the presented approach covers a large class of objects. While it is not applicable for objects that are made from transmissive materials, it works well for most opaque or locally subsurface scattering surfaces. We will discuss the applicability of the representation in more detail and point out the remaining limitations and possible failure cases in Section 8.2.

Using the presented integrated measurement setups, i.e. the Dome 1 and Dome 2, objects can be acquired and processed automatically, requiring very little human oversight. Although the average acquisition takes longer than just taking a few representative pictures or using a laser scanner, the additional value gained by being able to faithfully reproduce the overall appearance certainly outweighs this aspect in many applications. Both acquisition setups are designed with practical applicability in mind. The proposed Dome 2 can even be taken apart and reassembled on-site. This is especially valuable for the presented application scenario of cultural heritage digitization. Rather than moving precious cultural heritage items to the location of the setup, the setup can be brought to the objects. Still, both employed devices utilize elaborate radiometric and geometric calibration procedures to obtain accurate measurements of the objects' reflectance.

The raw measurement data takes up to 1.4 TB of disk space. In our proposed processing pipeline, this unhandy amount is broken into its final form, a triangle mesh and a compressed BTF, requiring between 260 MB and 6.5 GB. Although every processing step can be controlled by a number of additional parameters, the employed algorithms are reasonably robust. In our experiments, we use a single conservative set of parameters to cover a wide range of objects. This allows the processing to take place automatically but still enables the operator to intervene and improve the results, if necessary.

The need to compromise between accuracy and manageable effort can in the end of course only allow for an approximation of the appearance. However, in our experience the chosen set of captured directions and the lossy compression work reasonably well for the faithful reproduction of a large number of materials. In analogy to audio recordings, we consider the resulting files a digital master format that can directly be used for storage and high-quality rendering. Depending on the purpose, it can be drastically compacted, for example from 512.8 MB to a 36 MB set of PNG files for Internet streaming or converted to the memory-friendly BSVTF representation.

In conclusion, the presented approach can be considered as a viable choice for the high-quality digitization of the appearance of objects. However, the insights presented in this thesis are not limited to the digitization of 3D objects. In fact, BTFs are today most commonly employed as a versatile and high-quality representation for materials that are mapped on arbitrary geometries. Most of the investigated topics, such as practical acquisition devices and novel rendering methods, can directly be transferred to the domain of flat material samples. However, even not so obviously applicable aspects of integrated 3D acquisition with subsequent resampling and hole filling are useful in this context. For example, together with Ruiters *et al.*, we employ our proposed digitization pipeline to acquire and represent parts of the materials' mesoscopic geometry explicitly as a heightfield in [RSK13]. We show that this is an important prerequisite for the synthesis of novel materials via BTF interpolation.

8.2 Limitations and Use Cases

From a theoretical standpoint, the main limitation of the faithfulness of the chosen representation is the restriction to far field illumination. As argued in more detail in Section 3.3.2, the far field assumption is especially violated in the presence of fast changes in illumination intensity or direction. Often encountered examples of such a situation are hard shadow boundaries or light sources that are close to the material surface, e.g. global illumination from neighboring geometry.

In a practice, further limitations arise from the angular resolution of the stored BTF. In our experiments, especially the digital appearance of the the Donkey, the Billiard Ball the Inkwell and the Tennis Ball suffer from a insufficient number of directions. In the first three cases, the surrounding environment was clearly discernible as a mirroring reflection on the surfaces of the real objects. The digitized objects mostly fail to reproduce this behavior in a global illumination simulation (see Figure 1.2). Only on the Inkwell, blurred reflections of the Billiard Ball and Ammonite are indicated. On the Tennis Ball, the angular resolution of the view direction is too low to reproduce the appearance of the many small fibers that stick out the surface. As a result, the surface of the digitized object appears blurred. Yet, further increasing the measurement resolution is not trivially possible, as this would be prohibitively costly in terms of required acquisition time or hardware, storage requirements and computational effort during postprocessing.

Moreover, we also observed that the limited angular resolution and coverage of the proposed measurement devices together with the employed resampling can lead to rendering artifacts. Due to the limited resolution, the separate sample interpolation in each texel can lead to incorrect bright spots on specular materials (see Figure 5.17). The difference depends on the highlight directions of the particular texel to be covered by the set of measurement directions or not. In addition, because of the limited angular coverage of the hemispherical measurement devices, some of the resampled directions are actually extrapolated from the measurement data. Here, false extrapolation can lead to undesired shifts in appearance.

Finally, the comparably large file sizes, long measurement times, limited editing capabilities and of course cost of the hardware restrict the applicability of the approach. If accuracy of appearance is not a top priority, traditional modeling might be a more flexible alternative to obtain digital objects and materials. Additionally, the prolonged measurement time prohibits capturing fast changing or non-static items, e.g. ice cream, wet surfaces or living subjects.

However, despite the mentioned limitations, there are a lot of practical applications in which exact reproduction is a key goal and the presented acquisition setups are sufficiently fast. In Chapter 2, we pointed out several possible scenarios involving the fields of cultural heritage, medical education and even food photography. The achieved results throughout this thesis demonstrate that our techniques are fit for these envisioned challenging applications. Moreover, we believe this technique also has the potential to find application in fields such as online shopping, the entertainment and movie industry as well as advertisement.
For online shopping, digitized versions of the objects at sale could be presented in a virtual display using the WebGL-based rendering and streaming technique proposed in Chapter 6. This would take up on the already existing endeavors by some online shops to capture and present 360° orbiting sequences. In movies, props that were used by the actors or in the background of a scene are today already subject of digitization to be used in computer generated special effects sequences. Here, our proposed pipeline and representation would lend itself as a better automatized and more faithful alternative to currently employed approaches [BB11].

Finally, objects digitized with our method can also serve as synthetic ground truth, e.g. for computer vision applications. The faithful reproductions come close to real-world images, while test sequences using the digitized objects are much easier to obtain under controlled conditions. We recently explored this possible application in [GKSK14] together with Güssefeld *et al.*. Here, we evaluated the performance of different optical flow algorithms on real-world sequences and different synthetic data sets. Our results indicate that BTF textured objects are indeed well suited for predicting the performance. The tested algorithms produced similar artifacts on a synthetic sequence as on a recorded sequence of the real object.
In [WGK14], Weinmann *et al.* use material samples that were digitized with our approach for material classification. With the virtual materials they could systematically synthesize a huge amount of training data. Capturing such a comprehensive training set manually by taking photographs of the real material samples would have been infeasible. As a consequence, the synthetically trained classifiers achieve higher classification rates.

8.3 Future Work

Due to the image-based nature of the BTF, the proposed form of representation is mainly suitable for reproduction. However, the creation of novel objects from the captured data is also desirable. Whereas the geometry can to a certain degree already be modified with classical mesh editing techniques, adjusting the texture and reflectance behavior realistically, e.g. adding additional seeds when resizing the Strawberry, requires novel BTF editing and synthesis methods. Here, the approach we explored together with Ruiters *et al.* [RSK13] presents a first step into this direction. Several other general problems of the BTF have still to be solved as well. The limited angular resolution of the tabulated representation, e.g. leading to blurred highlights, is one of them. We already investigated an alternative data-driven approach together with Ruiters *et al.* in [RSK12]. However, the generation of renderings that show realistic reflections or caustics from such data-driven materials would still be very impractical, due to the current lack of suitable importance sampling strategies. A good solution to this problem is still to be found.

Although not as pressing as the increase in angular resolution, the spatial resolution of the measured BTFs could potentially be improved, too. The redundancy found in the many different view directions might be exploited by a multiview superresolution approach, similar to Goldlücke and Cremers [GC09]. However, instead of assuming a diffuse reflectance, a different regularization prior has to be used. Maybe the assumption that the reflectance is of low rank could be employed. In [MSK06], Müller *et al.* did so successfully for the optimization of local coordinate system orientations. We also exploit it during the factorization-based compression.

However, there is also a completely different direction of future work that is worth pursuing. The techniques presented in this thesis could be combined to create a unified streamable hierarchical level of detail representation. Currently, the BTF is parameterized over the surface via a texture atlas. The BSVTFs and the progressive transmission utilize the level of detail found in this texture atlas. However, we did not address the remaining issue how the macroscale geometry should be treated. Of course, an already existing level of detail approach for meshes could additionally be employed. However, this would for instance not directly allow to combine the refinement of material and geometry decisions. Furthermore, geometric details that are removed on a lower level of detail mesh should ideally instead become part of the mesoscopic detail in the BTF material.
Here, a volumetric approach would probably be a beneficial choice. In analogy to the usage of surface light fields in a quadtree structure [Rui08] or fitted Lambertian BRDFs in a voxelized binary space partition tree [GM05], the BTF would not

be projected in a texture atlas but into cells of an octree. The octree is refined according to the captured macroscale geometry.

If a local orientation for each cell is given, the resampling and hole filling could be carried out the same way as described in Chapter 5. Similarly, the BTF matrix $\mathbf{B}$ could be organized as "cells $\times$ ABRDF" instead of "texels $\times$ ABRDF" and treated the same way as before for compression, transmission and rendering.

However, in contrast to the current approach, this representation would have several advantages. First, it does not require a 2D parameterization of the surface. Hence, there are no distortions, such as stretch or sheer, and no seams at edges of texture patches that could lead to visible artifacts. Note that singularities in local tangent directions are unavoidable due to the Poincaré-Hopf theorem. Thus, some seam artifacts will probably remain on watertight objects. Yet, the local coordinate frames in each octree cell could be chosen separately to minimize the visible error, e.g. by using the data-driven method described in [MSK06].
Second, the reflectance samples of cells could be obtained using the footprint of the cell in the measured images. This way, a cell at a certain depth always considers finer resolved geometry as mesoscopic and captures its appearance in image-based form.
Third, if all cells of the octree are compressed using matrix factorization, the redundancy between different scales, e.g. in case of a (partially) fractal structure, can be exploited to further increase the compression ratio.
Finally, the resulting factorized octree holds a geometry as well as an appearance level of detail hierarchy: the tree structure and the factorization components. The refinement decision could be unified similar to our BSVTF approach.

Rendering with such a data structure would be very memory efficient, avoid aliasing and maintain a distinguishable silhouette, even when zooming in – at least until the resolution of the original 3D geometry used to construct the octree is reached. Furthermore, it would represent a truly view-dependent distinction between macroscale and mesoscale geometry.

Another possible avenue of future research would be the extension of digital material appearance beyond the BTF, including more of the variables found in the general 12D scattering function in Equation 3.10. First attempts to include the dependence on the wavelength have already been made by several groups by performing multispectral BTF measurements (e.g. [TAN*05, TSA*05, KTT06, RSK10]). All of the techniques proposed in this thesis should be directly applicable for this kind of measurement data. An interesting future questions would be whether full spectral data can be inferred from a sparse set of spectral samples, e.g. different wavelength bands for different captured directions combinations, during resampling. Here, for example the work of Rump *et al.* [RK10] shows a first attempt into this direction. Another question would also be if a more efficient compression can be found for this kind of data. We believe that, given the installation of the necessary tunable filters in our setup, our method could also be straightforwardly extended to handle bispectral measurements as well as polarization as an additional attribute. The biggest issue would probably be the exploding amounts of data and measurement time.

Naturally, the extension of the presented approach towards full surface reflectance field rendering comes into mind as well. This would especially be useful to overcome the restriction to opaque materials. Devices that use a projector to freely vary the point of incidence are available, such as mirror-based setups [LCV*04, GTLL06, MTK*10] or gonioreflectometers [HLZ10]. Here, however, an integrated geometry acquisition approach is hard to achieve. Few 3D scanning methods are capable of handling transparent parts of objects (see the survey in [IKL*10]). Furthermore, while for the possible extension to (bi-)spectral data a coarse sampling is usually considered sufficient, the reproduction of transparency of objects requires a very dense sampling and representation of both, the spatial domain of incident and outgoing positions as well as the angular domain. Depending on the sampling resolution, different degrees of transparency, from frosty to clear, would be representable. Similar to the challenge mirroring appearance of opaque surfaces poses for BTFs, data-driven methods probably have their limit in the foreseeable future when it comes to clear transparent objects.

[3D-12] 3D-COFORM summer exhibition. Website, July 2012. Available online at `http://exhibition.3d-coform.eu/`; visited on November 7th 2014. 110, 163, 167

[AB91] ADELSON E. H., BERGEN J. R.: The plenoptic function and the elements of early vision. In *Computational Models of Visual Processing*. MIT Press, Cambridge, MA, 1991, pp. 3–20. 32

[AFG13] ACKERMANN J., FUHRMANN S., GOESELE M.: Geometric point light source calibration. In *Proceedings of Vision, Modeling, and Visualization (VMV)* (Lugano, Switzerland, September 2013), Eurographics Association, pp. 161–168. 105

[ARGB*07] ABDUL-RAHMAN H. S., GDEISAT M. A., BURTON D. R., LALOR M. J., LILLEY F., MOORE C. J.: Fast and robust three-dimensional best path phase unwrapping algorithm. *Applied Optics 46*, 26 (September 2007), 6623–6635. 145

[AWL13] AITTALA M., WEYRICH T., LEHTINEN J.: Practical SVBRDF capture in the frequency domain. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 32*, 4 (July 2013), 110:1–110:12. 134

[Bar08] BARRETT S.: Sparse virtual texture memory. In *Game Developer Conference* (San Francisco, CA, USA, February 2008). 213, 215, 217

[BB08] BELLINGHAM L., BYBEE J. A.: *Food styling for photographers: a guide to creating your own appetizing art*. Elsevier/Focal Press, 2008. 16

[BB11] BHAT P., BURKE S.: PhotoSpace: A vision based approach for digitizing props. In *ACM SIGGRAPH 2011 Talks* (Vancouver, British Columbia, Canada, July 2011), pp. 1:1–1:1. 240

[BEJZ09] BEHR J., ESCHLER P., JUNG Y., ZÖLLNER M.: X3DOM: a DOM-based HTML5/X3D integration model. In *Proceedings of the International Conference on Web 3D Technology* (Darmstadt, Germany, June 2009), ACM, pp. 127–135. 13, 192

[BHH*98] BOTTOU L., HAFFNER P., HOWARD P. G., SIMARD P., BENGIO Y., LECUN Y.: High quality document image compression with DjVu. *Journal of Electronic Imaging 7*, 3 (1998), 410–425. 202

[BHKD09] BARAK M., HERSCOVIZ O., KABERMAN Z., DORI Y. J.: MOSAICA: A web-2.0 based system for the preservation and presentation of cultural heritage. *Computers & Education 53*, 3 (November 2009), 841 – 852. 12

[Bla04] BLAIS F.: Review of 20 years of range sensor development. *Journal of Electronic Imaging 13*, 1 (2004), 231–243. 144

[Bli78] BLINN J. F.: Simulation of wrinkled surfaces. *ACM SIGGRAPH Computer Graphics 12*, 3 (August 1978), 286–292. 38

[BR04] BANGAY S., RADLOFF J. D.: Kaleidoscope configurations for reflectance measurement. In *Proceedings of the International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa (AFRIGRAPH)* (Stellenbosch, South Africa, 2004), ACM, pp. 161–170. 123

[Bra14] BRAY T.: The JavaScript object notation (JSON) data interchange format. RFC 7159 (Proposed Standard), March 2014. 56

[Can05] Service notice: CCD image sensor advisory. Website, October 2005. Available online at `http://canoncanada.custhelp.com/app/answers/detail/a_id/4304`; visited on November 7th 2014. 83, 118

[CBCG02] CHEN W.-C., BOUGUET J.-Y., CHU M. H., GRZESZCZUK R.: Light field mapping: Efficient representation and hardware rendering of surface light fields. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 21*, 3 (July 2002), 447–456. 217

[CDF92] COHEN A., DAUBECHIES I., FEAUVEAU J.-C.: Biorthogonal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics 45*, 5 (June 1992), 485–560. 202

[Che95] CHEN S. E.: QuickTime VR: An image-based approach to virtual environment navigation. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (Los Angeles, CA, USA, August 1995), ACM, pp. 29–38. 40, 193

[Cla76] CLARK J. H.: Hierarchical geometric models for visible surface algorithms. *Communications of the ACM 19*, 10 (October 1976), 547–554. 214

[CNR08] COSSAIRT O., NAYAR S., RAMAMOORTHI R.: Light field transfer: global illumination between real and synthetic objects. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 27*, 3 (August 2008), 57:1–57:6. 34, 134

[COM98] COHEN J., OLANO M., MANOCHA D.: Appearance-preserving simplification. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (Orlando, Florida USA, July 1998), pp. 115–122. 38

[Coo84] COOK R. L.: Shade trees. *ACM SIGGRAPH Computer Graphics 18*, 3 (July 1984), 223–231. 38

[CT82] COOK R. L., TORRANCE K. E.: A reflectance model for computer graphics. *ACM Transactions on Graphics 1*, 1 (January 1982), 7–24. 25, 77, 181

[CV01] CHAN T., VESE L.: Active contours without edges. *IEEE Transactions on Image Processing 10*, 2 (2001), 266–277. 69

[D'A06] D'APUZZO N.: Overview of 3D surface digitization technologies in europe. In *Proceedings of the SPIE* (February 2006), pp. 42–54. 45

[Dan01] DANA K. J.: BRDF/BTF measurement device. In *Proceedings of the International Conference on Computer Vision (ICCV)* (Washington, DC, USA, 2001), vol. 2, IEEE Computer Society, pp. 460–466. 67, 121, 123, 124

[DBPGS10] DI BENEDETTO M., PONCHIO F., GANOVELLI F., SCOPIGNO R.: SpiderGL: a JavaScript 3D graphics library for next-generation WWW. In *Proceedings of the International Conference on Web 3D Technology* (Los Angeles, California, July 2010), pp. 165–174. 13, 191, 193

[DCC*10] DELLEPIANE M., CALLIERI M., CORSINI M., CIGNONI P., SCOPIGNO R.: Improved color acquisition and mapping on 3D models via flash-based photography. *ACM Journal on Computing and Cultural Heritage (JOCCH) 2*, 4 (March 2010), 9:1–9:20. 43

[Deb98] DEBEVEC P.: Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (Orlando, Florida USA, 1998), ACM, pp. 189–198. 17, 271

[DHT*00] DEBEVEC P., HAWKINS T., TCHOU C., DUIKER H.-P., SAROKIN W., SAGAR M.: Acquiring the reflectance field of a human face. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*

(New Orleans, Louisiana, USA, July 2000), ACM Press/Addison-Wesley Publishing Co., pp. 145–156. 33, 34, 42, 64, 68, 125

[DM97] DEBEVEC P. E., MALIK J.: Recovering high dynamic range radiance maps from photographs. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (Los Angeles, CA, USA, August 1997), ACM Press/Addison-Wesley Publishing Co., pp. 369–378. 54

[DN99] DAVIS G. M., NOSRATINIA A.: Wavelet-based image coding: An overview. *Applied and Computational Control, Signals, and Circuits 1* (July 1999), 369–434. 201

[DVGNK97] DANA K. J., VAN GINNEKEN B., NAYAR S. K., KOENDERINK J. J.: Reflectance and texture of real world surfaces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (San Juan, June 1997), IEEE Computer Society, pp. 151–157. 8, 34, 35, 62, 67, 70, 71, 119, 121, 128, 129, 130

[DW04] DANA K. J., WANG J.: Device for convenient measurement of spatially varying bidirectional reflectance. *Journal of the Optical Society of America A 21*, 1 (2004), 1–12. 67, 121, 123, 124

[DWT*02] DEBEVEC P., WENGER A., TCHOU C., GARDNER A., WAESE J., HAWKINS T.: A lighting reproduction approach to live-action compositing. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 21*, 3 (July 2002), 547–556. 68, 97, 125

[EMM10] EARL G., MARTINEZ K., MALZBENDER T.: Archaeological applications of polynomial texture mapping: analysis, conservation and representation. *Journal of Archaeological Science 37*, 8 (August 2010), 2040–2050. 42

[EY36] ECKART C., YOUNG G.: The approximation of one matrix by another of lower rank. *Psychometrika 1* (1936), 211–218. 160, 199

[FB81] FISCHLER M. A., BOLLES R. C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM 24*, 6 (June 1981), 381–395. 91

[FH09] FILIP J., HAINDL M.: Bidirectional texture function modeling: a state of the art survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 31* (October 2009), 1921–1940. 45, 66, 159

[FKIS02] FURUKAWA R., KAWASAKI H., IKEUCHI K., SAKAUCHI M.: Appearance based object modeling using texture database: acquisition, compression and rendering. In *Eurographics Workshop on Rendering* (Pisa, Italy, June 2002), Eurographics Association, pp. 257–266. 35, 42, 43, 47, 68, 97, 125, 126, 127, 129, 130, 137, 138, 139, 142

[Foo97] FOO S. C.: *A gonioreflectometer for measuring the bidirectional reflectance of material for use in illumination computation.* Master's thesis, Cornell University, 1997. 66

[Fou92a] FOURNIER A.: *Filtering normal maps and creating multiple surfaces.* Tech. Rep. TR-92-41, Department of Computer Science, University of British Columbia, Vancouver, BC, Canada, 1992. 38

[Fou92b] FOURNIER A.: Normal distribution functions and multiple surfaces. In *Graphics Interface Workshop on Local Illumination* (Vancouver, BC, Canada, May 1992), Canadian Information Processing Society, pp. 45–52. 36, 38

[FVH*13] FILIP J., VÁVRA R., HAINDL M., ŽID P., KRUPIČKA M., HAVRAN V.: BRDF slices: Accurate adaptive anisotropic appearance acquisition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Washington, DC, USA, June 2013), IEEE Computer Society, pp. 1468–1473. 67, 120, 121, 128, 129, 130

[GAHO07] GHOSH A., ACHUTHA S., HEIDRICH W., O'TOOLE M.: BRDF acquisition with basis illumination. In *Proceedings of the International Conference on Computer Vision (ICCV)* (Washington, DC, USA, 2007), IEEE Computer Society, pp. 1–8. 67

[GC09] GOLDLÜCKE B., CREMERS D.: Superresolution texture maps for multiview reconstruction. In *Proceedings of the International Conference on Computer Vision (ICCV)* (Kyoto, Japan, September 2009), IEEE, pp. 1677–1684. 241

[Gei09] GEIGES M. L.: Traces of Marion B. Sulzberger in the museum of wax moulages in Zurich and their importance for the history of dermatology. *Journal of the American Academy of Dermatology 60*, 6 (June 2009), 980–984. 14

[Gei12] GEIGES M. L.: Hundert Blicke auf ein Püggeli - Museale Wachsmoulagen in der modernen Lehre. In *Exhibition Catalogue*. Moulagenmuseum der Universität und des Universitätsspitals Zürich, Zürich, Switzerland, 2012. 14

[GGSC96] GORTLER S. J., GRZESZCZUK R., SZELISKI R., COHEN M. F.: The lumigraph. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (New Orleans, Louisiana, USA, August 1996), ACM, pp. 43–54. 33, 41, 42, 47, 138

[GH97] GARLAND M., HECKBERT P. S.: Surface simplification using quadric error metrics. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (Los Angeles, CA, USA, August 1997), ACM, pp. 209–216. 148

[GHLS00] GOESELE M., HEIDRICH W., LENSCH H. P. A., SEIDEL H.-P.: Building a photo studio for measurement purposes. In *Proceedings of Vision, Modeling, and Visualization (VMV)* (November 2000), pp. 241–238. 69

[GKSK14] GÜSSEFELD B., KONDERMANN D., SCHWARTZ C., KLEIN R.: Are reflectance field renderings appropriate for optical flow evaluation? In *IEEE International Conference on Image Processing (ICIP)* (Paris, France, October 2014), IEEE. 240

[Gla95] GLASSNER A. S.: *Principles of digital image synthesis*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995. 21, 22, 26, 28, 29, 161

[GLL*04] GOESELE M., LENSCH H. P. A., LANG J., FUCHS C., SEIDEL H.-P.: DISCO: Acquisition of translucent objects. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 23*, 3 (August 2004), 835–844. 38, 39

[GM05] GOBBETTI E., MARTON F.: Far voxels: A multiresolution framework for interactive rendering of huge complex 3D models on commodity graphics platforms. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 24*, 3 (July 2005), 878–885. 241

[GMSK09] GUTHE M., MÜLLER G., SCHNEIDER M., KLEIN R.: BTF-CIELab: A perceptual difference measure for quality assessment and compression of BTFs. *Computer Graphics Forum 28*, 1 (February 2009), 101–113. 24, 58, 195, 204, 215

[Gra53] GRAY F.: Pulse code communication, March 1953. US Patent 2,632,058. 86, 145

[GTLL06] GARG G., TALVALA E.-V., LEVOY M., LENSCH H. P. A.: Symmetric photography: Exploiting data-sparseness in reflectance fields. In *Eurographics Symposium on Rendering*

(EGSR) (Cyprus, June 2006), Eurographics Association, pp. 251–262. 34, 67, 122, 124, 134, 243

[GWS*09] GUNAWARDANE P., WANG O., SCHER S., RICKARDS I., DAVIS J., MALZBENDER T.: Optimized image sampling for view and light interpolation. In *International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage (VAST)* (St. Julians, Malta, September 2009), pp. 93–100. 42, 193

[GZ11a] GARCIA R., ZAKHOR A.: Temporally-consistent phase unwrapping for a stereo-assisted structured light system. In *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)* (Hangzhou, China, May 2011), IEEE Computer Society, pp. 389–396. 145

[GZ11b] GIESEL M., ZAIDI Q.: Visual perception of material affordances. *Journal of Vision 11*, 11 (September 2011). 4

[HCD01] HAWKINS T., COHEN J., DEBEVEC P.: A photometric approach to digitizing cultural artifacts. In *International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage (VAST)* (Aire-la-Ville, Switzerland, November 2001), Eurographics Association, pp. 333–342. 68, 125

[HEE*02] HAUTH M., ETZMUSS O., EBERHARDT B., KLEIN R., SARLETTE R., SATTLER M., DAUBERT K., KAUTZ J.: Cloth animation and rendering. In *Eurographics 2002 Tutorials* (Aire-la-Ville, Switzerland, September 2002), Eurographics Association. Tutorial. 67, 70, 72

[HF07] HAINDL M., FILIP J.: Extreme compression and modeling of bidirectional texture function. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 29*, 10 (October 2007), 1859–1865. 216

[HF11] HAINDL M., FILIP J.: Advanced textural representation of materials appearance. In *SIGGRAPH Asia 2011 Courses* (New York, NY, USA, December 2011), ACM, pp. 1:1–1:84. 45, 66, 120, 122, 159

[HF13] HAINDL M., FILIP J.: *Visual texture: accurate material appearance measurement, representation and modeling*. Advances in Computer Vision and Pattern Recognition. Springer, New York, NY, USA, 2013. 45, 66, 159, 215

[HFM10] HAVRAN V., FILIP J., MYSZKOWSKI K.: Bidirectional texture function compression based on multi-level vector quantization. *Computer Graphics Forum 29*, 1 (January 2010), 175–190. 160, 216

[HHA*10] HULLIN M. B., HANIKA J., AJDIN B., SEIDEL H.-P., KAUTZ J., LENSCH H. P. A.: Acquisition and analysis of bispectral bidirectional reflectance and reradiation distribution functions. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 29*, 4 (July 2010), 97:1–97:7. 32

[HLS07] HORMANN K., LÉVY B., SHEFFER A.: Mesh parameterization: Theory and practice. In *ACM SIGGRAPH 2007 Courses* (San Diego, California, July 2007), ACM. 153

[HLZ10] HOLROYD M., LAWRENCE J., ZICKLER T.: A coaxial optical scanner for synchronous acquisition of 3D geometry and surface reflectance. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 29*, 4 (July 2010), 99:1–99:12. 38, 39, 67, 68, 120, 121, 122, 128, 130, 157, 183, 243

[HP03] HAN J. Y., PERLIN K.: Measuring bidirectional texture reflectance with a kaleidoscope. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 22*, 3 (July 2003), 741–748. 35, 67, 81, 122, 123, 124, 128, 129, 130

[HQS10] HU Y., QI Y., SHEN F.: Modeling spatially-varying reflectance based on kernel nyström. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST)* (New York, NY, USA, 2010), ACM, pp. 91–92. 68, 125, 127, 130, 132

[HT06] HOLMA H., TOSKALA A.: *HSDPA/HSUPA for UMTS: high speed radio access for mobile communications*. John Wiley & Sons, April 2006. 191

[HW11] HERBORT S., WÖHLER C.: An introduction to image-based 3D surface reconstruction and a survey of photometric stereo methods. *3D Research 2*, 3 (2011), 40:1–40:17. 45

[HWQ13] HU Y., WANG S., QI Y.: Efficient reconstruction, decomposition and editing for spatially-varying reflectance data. In *Proceedings of the Graphics Interface Conference (GI)* (Toronto, Ont., Canada, Canada, 2013), Canadian Information Processing Society, pp. 55–62. 68, 125, 127

[HZ04] HARTLEY R. I., ZISSERMAN A.: *Multiple view geometry in computer vision*, second ed. Cambridge University Press, 2004. 20, 51

[IEE85] IEEE standard for binary floating-point arithmetic. *ANSI/IEEE Std 754-1985* (1985). 56

[IEE08] IEEE standard for floating-point arithmetic. *IEEE Std 754-2008* (August 2008), 1–70. 56

[IKL*10] IHRKE I., KUTULAKOS K. N., LENSCH H. P. A., MAGNOR M., HEIDRICH W.: Transparent and specular object reconstruction. *Computer Graphics Forum 29*, 8 (2010), 2400–2426. 45, 243

[IRM*12] IHRKE I., RESHETOUSKI I., MANAKOV A., TEVS A., WAND M., SEIDEL H.-P.: A kaleidoscopic approach to surround geometry and reflectance acquisition. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (Washington, DC, USA, 2012), IEEE Computer Society, pp. 29–36. 43, 67, 68, 123, 124, 128, 130

[ISM84] INOKUCHI S., SATO K., MATSUDA F.: Range imaging system for 3-D object recognition. In *International Conference on Pattern Recognition (ICPR)* (Montreal, Canada, 1984), IEEE Computer Society, pp. 806–808. 145

[ITU94] ISO/IEC 10918-1 | ITU-T recommendation T.81: Information technology – digital compression and coding of continuous-tone still images: Requirements and guidelines, 1994. 56

[Jak10] JAKOB W.: Mitsuba renderer, 2010. `http://www.mitsuba-renderer.org`. 7, 52

[JBG11] JUNG Y., BEHR J., GRAF H.: X3DOM as carrier of the virtual heritage. In *4th ISPRS International Workshop 3D-ARCH 2011: "3D Virtual Reconstruction and Visualization of Complex Architectures"* (Trento, Italy, March 2011), pp. 475–482. 13, 193

[JMLH01] JENSEN H. W., MARSCHNER S. R., LEVOY M., HANRAHAN P.: A practical model for subsurface light transport. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (Los Angeles, CA, USA, August 2001), ACM, pp. 511–518. 31

[Kaj86] KAJIYA J. T.: The rendering equation. *ACM SIGGRAPH Computer Graphics 20*, 4 (August 1986), 143–150. 29, 30

[KBD07] KAUTZ J., BOULOS S., DURAND F.: Interactive editing and modeling of bidirectional texture functions. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 26*, 3 (July 2007). 217

[KBH06] KAZHDAN M., BOLITHO M., HOPPE H.: Poisson surface reconstruction. In *Proceedings of the Eurographics Symposium on Geometry Processing* (Cagliari, Sardinia, Italy, 2006), Eurographics Association, pp. 61–70. 148

[KH13] KAZHDAN M., HOPPE H.: Screened Poisson surface reconstruction. *ACM Transactions on Graphics 32*, 3 (July 2013), 29:1–29:13. 148

[KMBK03] KOUDELKA M. L., MAGDA S., BELHUMEUR P. N., KRIEGMAN D. J.: Acquisition, compression, and synthesis of bidirectional texture functions. In *International Workshop on Texture Analysis and Synthesis (Texture)* (Nice, France, October 2003), pp. 59–64. 35, 67, 119, 120, 121, 128, 129, 130, 159, 160, 215, 219, 225

[KNRS13] KÖHLER J., NÖLL T., REIS G., STRICKER D.: A full-spherical device for simultaneous geometry and reflectance acquisition. In *IEEE Workshop on Applications of Computer Vision (WACV)* (Washington, DC, USA, 2013), IEEE Computer Society, pp. 355–362. 68, 97, 126, 127, 128, 130, 132, 134, 170, 185, 193

[KSD12] KIM D., SRA S., DHILLON I. S.: A non-monotonic method for large-scale non-negative least squares. *Optimization Methods and Software (OMS) 28*, 5 (February 2012), 1012–1039. 159

[KTI*01] KANEKO T., TAKAHEI T., INAMI M., KAWAKAMI N., YANAGIDA Y., MAEDA T., TACHI S.: Detailed shape representation with parallax mapping. In *Proceedings of the International Conference on Artificial Reality and Telexistence (ICAT)* (Tokyo, Japan, December 2001), pp. 205–208. 38

[KTT06] KIMACHI A., TANAKA N., TOMINAGA S.: Development and calibration of a gonio-spectral imaging system for measuring surface reflection. *IEICE - Transactions on Information and Systems E89-D* (July 2006), 1994–2003. 67, 119, 121, 129, 242

[LA09] LOURAKIS M. I. A., ARGYROS A. A.: SBA: A software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software 36*, 1 (March 2009), 2:1–2:30. 52, 91, 104, 147

[lab] LABSPHERE: Spectralon© diffuse reflectance targets. Website. Available online at `http://www.labsphere.com/uploads/pb13025Targets.pdf`; visited on November 7th 2014. 107

[LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3D surface construction algorithm. *ACM SIGGRAPH Computer Graphics 21*, 4 (August 1987), 163–169. 148

[LCV*04] LEVOY M., CHEN B., VAISH V., HOROWITZ M., MCDOWALL I., BOLAS M.: Synthetic aperture confocal imaging. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 23*, 3 (August 2004), 825–834. 34, 67, 122, 124, 134, 243

[LD11] LUO Y., DURAISWAMI R.: Efficient parallel nonnegative least squares on multicore architectures. *SIAM Journal on Scientific Computing (SISC) 33*, 5 (October 2011), 2848–2863. 159

[Lev44] LEVENBERG K.: A method for the solution of certain non-linear problems in least squares. *Quarterly Journal of Applied Mathematics II*, 2 (1944), 164–168. 52, 90, 105

[LFD*08] LELOUP F. B., FORMENT S., DUTRÉ P., POINTER M. R., HANSELAER P.: Design of an instrument for measuring the spectral bidirectional scatter distribution function. *Applied Optics 47*, 29 (October 2008), 5454–5467. 66, 67

[LFTW06] LI H., FOO S. C., TORRANCE K. E., WESTIN S. H.: Automated three-axis gonioreflectometer for computer graphics applications. *Optical Engineering 45*, 4 (2006). 66, 67

[LH95] LAWSON C. L., HANSON R. J.: Appendix C: Description and use of fortran codes for solving problem LS. In *Solving Least Squares Problems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1995, ch. 30, pp. 248–283. 159

[LH96] LEVOY M., HANRAHAN P.: Light field rendering. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (New Orleans, Louisiana, USA, August 1996), ACM, pp. 31–42. 33, 41, 63

[LJBA13] LIMPER M., JUNG Y., BEHR J., ALEXA M.: The POP buffer: Rapid progressive clustering by geometry quantization. *Computer Graphics Forum (Proceedings of Pacific Graphics) 32*, 7 (October 2013), 197–206. 207

[LKG*03] LENSCH H. P. A., KAUTZ J., GOESELE M., HEIDRICH W., SEIDEL H.-P.: Image-based reconstruction of spatial appearance and geometric detail. *ACM Transactions on Graphics 22*, 2 (April 2003), 234–257. 38, 39, 157

[LS00] LEE D. D., SEUNG H. S.: Algorithms for non-negative matrix factorization. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)* (Denver, Colorado, USA, November 2000), MIT Press, pp. 556–562. 157

[LT03] LU W., TAN Y.-P.: Color filter array demosaicking: new method and performance measures. *IEEE Transactions on Image Processing 12*, 10 (October 2003), 1194–1210. 106

[LW93] LAFORTUNE E. P., WILLEMS Y. D.: Bi-directional path tracing. In *Proceedings of the International Conference on Computational Graphics and Visualization Techniques (Compugraphics)* (Alvor, Portugal, December 1993), ACM and ISGG, pp. 145–153. 30

[LW96] LAFORTUNE E. P., WILLEMS Y. D.: Rendering participating media with bidirectional path tracing. In *Eurographics Workshop on Rendering* (Porto, Portugal, August 1996), Eurographics Association, pp. 91–100. 30

[LWC*03] LUEBKE D., WATSON B., COHEN J. D., REDDY M., VARSHNEY A.: *Level of detail for 3D graphics*. Elsevier Science Inc., New York, NY, USA, 2003. 215

[MA83] MARTIN W. N., AGGARWAL J.: Volumetric descriptions of objects from multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 5*, 2 (March 1983), 150–158. 127

[Max88] MAX N. L.: Horizon mapping: shadows for bump-mapped surfaces. *The Visual Computer 4*, 2 (1988), 109–117. 38

[MB95] MCMILLAN L., BISHOP G.: *Shape as a perturbation to projective mapping*. Tech. Rep. TR95-046, Department of Computer Science, University of North Carolina at Chapel Hill, Chgapel Hill, NC, USA, 1995. 40

[MBK05] MÜLLER G., BENDELS G. H., KLEIN R.: Rapid synchronous acquisition of geometry and BTF for cultural heritage artefacts. In *International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage (VAST)* (Pisa, Italy, 2005), Eurographics Association, pp. 13–20. 35, 42, 43, 47, 68, 81, 127, 137, 139, 140, 142, 143, 157

[MBR*00] MATUSIK W., BUEHLER C., RASKAR R., GORTLER S. J., MCMILLAN L.: Image-based visual hulls. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (New York, NY, USA, July 2000), ACM Press/Addison-Wesley Publishing Co., pp. 369–374. 127

[McA02] MCALLISTER D. K.: *A generalized surface appearance representation for computer graphics*. PhD thesis, University of North Carolina at Chapel Hill, 2002. AAI3061704. 67, 119, 121, 128, 129, 130

[Met] METROLOGY E.: Romer Infinite 2.0 SC Scanshark 3D laser scanning portable CMM. Website. Available online at `https://web.archive.org/web/20131013114513/http://www.exactmetrology.com/products/romer/infinite-2-0/`; visited on November 7th 2014. 142, 143

[MGW01] MALZBENDER T., GELB D., WOLTERS H.: Polynomial texture maps. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (Los Angeles, CA, USA, August 2001), ACM, pp. 519–528. 42, 43

[Mit08] MITTRING M.: Advanced virtual texture topics. In *ACM SIGGRAPH 2008 Games* (Los Angeles, CA, USA, August 2008), pp. 23–51. 215

[MKRH11] MANTIUK R., KIM K. J., REMPEL A. G., HEIDRICH W.: HDR-VDP-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 30*, 4 (July 2011), 40:1–40:14. 58

[MMK03] MÜLLER G., MESETH J., KLEIN R.: Compression and real-time rendering of measured BTFs using local PCA. In *Proceedings of Vision, Modeling, and Visualization (VMV)* (München, Germany, November 2003), Akademische Verlagsgesellschaft AKA GmbH, pp. 271–280. 139, 160

[MMS*04] MÜLLER G., MESETH J., SATTLER M., SARLETTE R., KLEIN R.: Acquisition, synthesis and rendering of bidirectional texture functions. In *Eurographics 2004 State of the Art Reports* (Grenoble, France, September 2004), INRIA and Eurographics Association, pp. 69–94. 30, 66, 67, 68, 70, 72, 81, 82, 118, 126, 130

[MP95] MANN S., PICARD R. W.: On being 'undigital' with digital cameras: Extending dynamic range by combining differently exposed pictures. In *IS&T's 48th Annual Conference* (May 1995), pp. 422–428. 54

[MPBM03] MATUSIK W., PFISTER H., BR M., MCMILLAN L.: A data-driven reflectance model. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 22*, 3 (July 2003), 759–769. 37, 134, 161, 271

[MPN*02] MATUSIK W., PFISTER H., NGAN A., BEARDSLEY P., ZIEGLER R., MCMILLAN L.: Image-based 3D photography using opacity hulls. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 21*, 3 (2002), 427–437. 68, 97, 126, 127, 129, 130

[MPZ*02] MATUSIK W., PFISTER H., ZIEGLER R., NGAN A., MCMILLAN L.: Acquisition and rendering of transparent and refractive objects. In *Eurographics Workshop on Rendering* (Aire-la-Ville, Switzerland, 2002), Eurographics Association, pp. 267–278. 68, 97, 126, 127

[MRP98] MILLER G., RUBIN S., PONCELEON D.: Lazy decompression of surface light fields for precomputed global illumination. In *Eurographics Workshop on Rendering* (Vienna, Austria, June 1998), Eurographics Association, pp. 281–292. 41

[MS13] MUÑOZ-SALINAS R.: ArUco: Augmented reality library from the University of Cordoba. Website, 2013. Available online at `http://sourceforge.net/projects/aruco/`; visited on November 7th 2014. 104

[MSB12] MENDES C. M., SILVA L., BELLON O. R.: IMAGO visualization system: an interactive web-based 3D visualization system for cultural heritage applications. *Journal of Multimedia 7*, 2 (April 2012), 205–210. 13

[MSE*10] MUDGE M., SCHROER C., EARL G., MARTINEZ K., PAGI H., TOLER-FRANKLIN C., RUSINKIEWICZ S., PALMA G., WACHOWIAK M., ASHLEY M., MATTHEWS N., NOBLE T., DELLEPIANE M.: Principles and practices of robust, photography-based digital imaging techniques for museums. In *International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage (VAST) - Short and Project Papers* (Paris, France, September 2010), pp. 111–137. 193

[MSK06] MÜLLER G., SARLETTE R., KLEIN R.: Data-driven local coordinate systems for image-based rendering. *Computer Graphics*

Forum (Proceedings of Eurographics) 25, 3 (September 2006). 139, 150, 241, 242

[MSY07] MUKAIGAWA Y., SUMINO K., YAGI Y.: High-speed measurement of BRDF using an ellipsoidal mirror and a projector. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Washington, DC, USA, June 2007), IEEE Computer Society, pp. 1–8. 67

[MTK*10] MUKAIGAWA Y., TAGAWA S., KIM J., RASKAR R., MATSUSHITA Y., YAGI Y.: Hemispherical confocal imaging using turtleback reflector. 336–349. 34, 67, 122, 123, 124, 128, 129, 130, 134, 243

[Mül09] MÜLLER G.: *Data-driven methods for compression and editing of spatially varying appearance*. Dissertation, Universität Bonn, December 2009. 33, 139, 159, 161, 194

[ND06] NGAN A., DURAND F.: Statistical acquisition of texture appearance. In *Eurographics Symposium on Rendering (EGSR)* (Nicosia, Cyprus, June 2006), pp. 31–40. 220

[New71] NEWTON I.: A letter of Mr. Isaac Newton, professor of the mathematicks in the University of Cambridge; containing his new theory about light and colors: Sent by the author to the publisher from Cambridge, febr. 6. 1671/72; in order to be communicated to the R. Society. *Philosophical Transactions (1665-1678) 6* (1671), pp. 3075–3087. 124

[Nex] NEXTENGINE I.: NextEngine 3D scanner HD – technical specifications. Website. Available online at `http://www.nextengine.com/assets/pdf/scanner-techspecs.pdf`; visited on November 7th 2014. 142, 143

[Nic65] NICODEMUS F. E.: Directional reflectance and emissivity of an opaque surface. *Applied Optics 4*, 7 (July 1965), 767–775. 34, 42

[NJRS13] NÖLL T., J. K., REIS G., STRICKER D.: Faithful, compact and complete digitization of cultural heritage using a full-spherical scanner. In *DigitalHeritage2013 International Congress* (Marseille, France, October 2013), IEEE Computer Society. 38, 39, 68, 97, 126, 127, 185

[NKGR06] NAYAR S. K., KRISHNAN G., GROSSBERG M. D., RASKAR R.: Fast separation of direct and global components of a scene using high frequency illumination. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 25*, 3 (July 2006), 935–944. 120

[NRH*77] NICODEMUS F. E., RICHMOND J. C., HSIA J. J., GINSBERG I. W., LIMPERIS T.: *Geometrical considerations and nomenclature for reflectance*. National Bureau of Standards Monograph #160, U.S. Department of Commerce, August 1977. 32, 66

[NS11] NÖLL T., STRICKER D.: Efficient packing of arbitrarily shaped charts for automatic texture atlas generation. *Computer Graphics Forum (Proceedings of Eurographics Symposium on Rendering) 30*, 4 (jun 2011), 1309–1317. 168, 185

[NZG05] NEUBECK A., ZALESNY A., GOOL L. V.: 3D texture reconstruction from extensive BTF data. In *International Workshop on Texture Analysis and Synthesis (Texture)* (October 2005), pp. 13–19. 68, 126, 127, 130, 132

[NZG06] NEUBECK A., ZALESNY A., GOOL L. V.: Light source calibration for IBR and BTF acquisition setups. In *International Symposium on 3D Data Processing, Visualisation and Transmission (3DPVT)* (June 2006), University of North Carolina, pp. 272–278. 68, 126, 127

[Ope] OpenCV camera calibration and 3D reconstruction. Website. Available online at `http://docs.opencv.org/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html`; visited on November 7th 2014. 49, 52

[OvWS12] OBERT J., VAN WAVEREN J. M. P., SELLERS G.: Virtual texturing in software and hardware. In *ACM SIGGRAPH 2012 Courses* (Los Angeles, California, USA, August 2012), pp. 5:1–5:29. 215

[PB96] PROCTOR J. E., BARNES P. Y.: NIST high accuracy reference reflectometer-spectrophotometer. *Journal of Research of the National Institute of Standards and Technology (NIST) 101*, 5 (September 1996), 619–627. 66, 67

[PB11] PENNER E., BORSHUKOV G.: Pre-integrated skin shading. In *GPU Pro 2*, vol. 2. A K Peters/CRC Press 2011, Boca Raton, FL, USA, 2011, pp. 41–55. 15

[PCC*10] PALMA G., CORSINI M., CIGNONI P., SCOPIGNO R., MUDGE M.: Dynamic shading enhancement for reflectance transformation imaging. *ACM Journal on Computing and Cultural Heritage (JOCCH) 3* (October 2010), 6:1–6:20. 42

[PCDS12] PALMA G., CALLIERI M., DELLEPIANE M., SCOPIGNO R.: A statistical method for SVBRDF approximation from video sequences in general lighting conditions. *Computer Graphics Forum (Proceedings of Eurographics Symposium on Rendering) 31*, 4 (June 2012), 1491–1500. 38, 39, 157

[PDCS13] PALMA G., DESOGUS N., CIGNONI P., SCOPIGNO R.: Surface light field from video acquired in uncontrolled settings. In *DigitalHeritage2013 International Congress* (Marseille, France, October 2013), IEEE Computer Society, pp. 31–38. 41

[PG10] PALMER J., GRANT B.: *The art of radiometry*. SPIE Press Monograph. SPIE Press, 2010. 28

[PSR13] PAJAROLA R., SUTER S. K., RUITERS R.: Tensor approximation in visualization and computer graphics. In *Eurographics 2013 Tutorials* (Girona, Spain, May 2013), pp. t6–. 215

[RBS03] ROBERTSON M. A., BORMAN S., STEVENSON R. L.: Estimation-theoretic approach to dynamic range enhancement using multiple exposures. *Journal of Electronic Imaging 12*, 2 (April 2003), 219–228. 76, 92, 94, 106

[Red] REDDY M.: Wavefront OBJ file format. Website. Available online at `http://www.martinreddy.net/gfx/3d/OBJ.spec`; visited on November 7th 2014. 56

[RK09a] RUITERS R., KLEIN R.: BTF compression via sparse tensor decomposition. 1181–1188. 160, 215

[RK09b] RUITERS R., KLEIN R.: Heightfield and spatially varying BRDF reconstruction for materials with interreflections. *Computer Graphics Forum (Proceedings of Eurographics) 28*, 2 (April 2009), 513–522. 157, 186

[RK10] RUMP M., KLEIN R.: Spectralization: Reconstructing spectra from sparse data. In *Eurographics Symposium on Rendering (EGSR)* (Saarbruecken, Germany, June 2010), Lawrence J., Stamminger M., (Eds.), Eurographics Association, pp. 1347–1354. 242

[RK13] RUITERS R., KLEIN R.: BTF based material representations: Current challenges. In *Eurographics Workshop on Material Appearance Modeling: Issues and Acquisition* (Zaragoza, Spain, June 2013), Eurographics Association, pp. 17–20. 134

[RMS*08] RUMP M., MÜLLER G., SARLETTE R., KOCH D., KLEIN R.: Photo-realistic rendering of metallic car paint from image-based measurements. *Computer Graphics Forum (Proceedings of Eurographics) 27*, 2 (April 2008), 527–536. 68, 81, 89

[RNLL10] RAY N., NIVOLIERS V., LEFEBVRE S., LÉVY B.: Invisible seams. *Computer Graphics Forum (Proceedings of Eurographics Symposium on Rendering) 29*, 4 (June 2010), 1489–1496. 185

[Rob02] ROBBINS C. R.: *Chemical and Physical Behavior of Human Hair*, fourth ed. Springer, New York, NY, USA, 2002. 36

[Row97] ROWEIS S.: EM algorithms for PCA and SPCA. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)* (Denver, Colorado, USA, December 1997), MIT Press, pp. 626–632. 161

[RPWL07] R. PETER WEISTROFFER KRISTEN R. WALCOTT G. H., LAWRENCE J.: Efficient basis decomposition for scattered reflectance data. In *Eurographics Symposium on Rendering (EGSR)* (Grenoble, France, June 2007), Eurographics Association, pp. 207–218. 43

[RRK09] RUITERS R., RUMP M., KLEIN R.: Parallelized matrix factorization for fast BTF compression. In *Proceedings of the Eurographics Conference on Parallel Graphics and Visualization* (Munich, Germany, March 2009), Eurographics Association, pp. 25–32. 161, 177, 229

[RSK10] RUMP M., SARLETTE R., KLEIN R.: Groundtruth data for multispectral bidirectional texture functions. In *Proceedings of the European Conference on Colour in Graphics, Imaging, and Vision (CGIV)* (June 2010), Society for Imaging Science and Technology, pp. 326–331. 67, 70, 72, 75, 79, 112, 118, 121, 130, 242, 271

[RSK12] RUITERS R., SCHWARTZ C., KLEIN R.: Data driven surface reflectance from sparse and irregular samples. *Computer Graphics Forum (Proceedings of Eurographics) 31*, 2 (May 2012), 315–324. 140, 157, 170, 176, 181, 185, 241

[RSK13] RUITERS R., SCHWARTZ C., KLEIN R.: Example-based interpolation and synthesis of bidirectional texture functions. *Computer Graphics Forum (Proceedings of Eurographics) 32*, 2 (May 2013), 361–370. 239, 241

[Rui08] RUITERS R.: View-dependent far-field level of detail rendering for urban models. *Computer Graphics & Geometry 10*, 3 (2008). 217, 241

[Rus98] RUSINKIEWICZ S. M.: A new change of variables for efficient BRDF representation. In *Eurographics Workshop on Rendering* (Vienna, Austria, June 1998), Eurographics Association, pp. 11–22. 24, 150

[RWPD05] REINHARD E., WARD G., PATTANAIK S., DEBEVEC P.: *High dynamic range imaging: acquisition, display, and image-based lighting*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005. 53

[RWS*11] REN P., WANG J., SNYDER J., TONG X., GUO B.: Pocket reflectometry. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 30*, 4 (July 2011), 45:1–45:10. 134

[RY00] RAO K. R., YIP P. C.: *The transform and data compression handbook*. CRC Press, Inc., Boca Raton, FL, USA, September 2000. 201

[SCOL*04] SORKINE O., COHEN-OR D., LIPMAN Y., ALEXA M., RÖSSL C., SEIDEL H.-P.: Laplacian surface editing. In *Proceedings of the Eurographics Symposium on Geometry Processing* (Nice, France, July 2004), Eurographics Association and ACM SIGGRAPH, pp. 175–184. 148

[SG31] SMITH T., GUILD J.: The C.I.E. colorimetric standards and their use. *Transactions of the Optical Society 33*, 3 (1931), 73. 28

[SG88] SHOHAM Y., GERSHO A.: Efficient bit allocation for an arbitrary set of quantizers. *IEEE Transactions on Acoustics, Speech, and Signal Processing 36*, 9 (September 1988), 1445–1453. 202

[Sha49] SHANNON C. E.: Communication in the presence of noise. *Proceedings of the IRE 37*, 1 (January 1949), 10–21. 65

[She68] SHEPARD D.: A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM National Conference* (New York, NY, USA, August 1968), ACM, pp. 517–524. 155

[SI99] SANDMEIER S. R., ITTEN K. I.: A field goniometer system (FIGOS) for acquisition of hyperspectral BRDF data. *IEEE Transactions on Geoscience and Remote Sensing 37*, 2 (March 1999), 978–986. 66, 67

[SK12] SCHWARTZ C., KLEIN R.: Acquisition and presentation of virtual surrogates for cultural heritage artefacts. In *EVA 2012 Berlin Conference on Electronic Media and Visual Arts* (Volmerstraße 3, 12489 Berlin, November 2012), Gesellschaft zur Förderung angewandter Informatik e.V., pp. 50–57. 18, 68, 97, 118, 127, 184

[SL00] SKOCAJ D., LEONARDIS A.: Range image acquisition of objects with non-uniform albedo using structured light range sensor. In *International Conference on Pattern Recognition (ICPR)* (Barcelona, Spain, September 2000), IEEE Computer Society, pp. 1778–1781. 145

[SLMB05] SHEFFER A., LÉVY B., MOGILNITSKY M., BOGOMYAKOV A.: ABF++: fast and robust angle based flattening. *ACM Transactions on Graphics 24*, 2 (April 2005), 311–330. 153

[SOC*13] SELLERS G., OBERT J., COZZI P., RING K., PERSSON E., DE VAHL J., VAN WAVEREN J. M. P.: Rendering massive virtual worlds. In *ACM SIGGRAPH 2013 Courses* (Anaheim, California, USA, July 2013), pp. 23:1–23:88. 215

[SP96] SAID A., PEARLMAN W.: A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Transactions on Circuits and Systems for Video Technology 6*, 3 (June 1996), 243–250. 201

[SPB04] SALVI J., PAGÈS J., BATLLE J.: Pattern codification strategies in structured light systems. *Pattern Recognition 37* (April 2004), 827–849. 144, 145

[Sph] SPHEREOPTICS: Reflectance coatings and materials – technical information. Website. Available online at `http://www.laser2000.fr/fileadmin/Produktgruppenkataloge/SPO_MaterialsCoatingsExtendedGuide2009.pdf`; visited on November 7th 2014. 77, 87

[SRK13] SCHWARTZ C., RUITERS R., KLEIN R.: Level-of-detail streaming and rendering using bidirectional sparse virtual texture functions. *Computer Graphics Forum (Proceedings of Pacific Graphics) 32*, 7 (October 2013), 345–354. 230, 232

[SRWK11] SCHWARTZ C., RUITERS R., WEINMANN M., KLEIN R.: WebGL-based streaming and presentation framework for bidirectional texture functions. In *International Symposium on Virtual*

Reality, Archaeology and Intelligent Cultural Heritage (VAST) (Prato, Italy, October 2011), Eurographics Association, pp. 113–120. 18, 191, 209

[SRWK13] SCHWARTZ C., RUITERS R., WEINMANN M., KLEIN R.: WebGL-based streaming and presentation of objects with bidirectional texture functions. *ACM Journal on Computing and Cultural Heritage (JOCCH) 6*, 3 (August 2013), 11:1–11:21. 18, 209

[SS03] SCHARSTEIN D., SZELISKI R.: High-accuracy stereo depth maps using structured light. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Madison, Wisconsin, USA, June 2003), vol. 1, IEEE Computer Society, pp. I–195–I–202 vol.1. 145

[SSB06] SHEIKH H., SABIR M., BOVIK A.: A statistical evaluation of recent full reference image quality assessment algorithms. *IEEE Transactions on Image Processing 15*, 11 (November 2006), 3440–3451. 58

[SSK03] SATTLER M., SARLETTE R., KLEIN R.: Efficient and realistic visualization of cloth. In *Eurographics Symposium on Rendering (EGSR)* (Leuven, Belgium, June 2003), Eurographics Association, pp. 167–177. 35, 67, 70, 72, 118, 121, 130, 149

[SSKLK13] SHREINER D., SELLERS G., KESSENICH J., LICEA-KANE B.: *OpenGL programming guide: The official guide to learning OpenGL, version 4.3*. Addison Wesley Professional, 2013. 53

[SSW*14] SCHWARTZ C., SARLETTE R., WEINMANN M., RUMP M., KLEIN R.: Design and implementation of practical bidirectional texture function measurement devices focusing on the developments at the University of Bonn. *Sensors 14*, 5 (May 2014), 7753–7819. 131

[SSWK13] SCHWARTZ C., SARLETTE R., WEINMANN M., KLEIN R.: DOME II: A parallelized BTF acquisition system. In *Eurographics Workshop on Material Appearance Modeling: Issues and Acquisition* (Zaragoza, Spain, June 2013), Eurographics Association, pp. 25–31. 68, 97, 127, 130, 131

[STD09] SANSONI G., TREBESCHI M., DOCCHIO F.: State-of-the-art and applications of 3D imaging sensors in industry, cultural heritage, medicine, and criminal investigation. *Sensors 9*, 1 (2009), 568–601. 45

[STOK05] SHINOHARA T., TAKAYAMA J.-Y., OHYAMA S., KOBAYASHI A.: Analysis of woven fabric structure using x-ray ct images. *Transactions of the Society of Instrument and Control Engineers E-4*, 1 (September 2005). 36

[SWI97] SATO Y., WHEELER M. D., IKEUCHI K.: Object shape and reflectance modeling from observation. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (Los Angeles, CA, USA, August 1997), ACM Press/Addison-Wesley Publishing Co., pp. 379–387. 38, 39

[SWR*11] SCHWARTZ C., WEINMANN M., RUITERS R., ZINKE A., SARLETTE R., KLEIN R.: Capturing shape and reflectance of food. In *SIGGRAPH Asia 2011 Sketches* (Hong Kong, China, December 2011), ACM, pp. 28:1–28:2. 18, 184

[SWRK11] SCHWARTZ C., WEINMANN M., RUITERS R., KLEIN R.: Integrated high-quality acquisition of geometry and appearance for cultural heritage. In *International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage (VAST)* (Prato, Italy, October 2011), Eurographics Association, pp. 25–32. 18, 68, 81, 118, 127, 130, 184

[SZZ12] SCHRÖDER K., ZHAO S., ZINKE A.: Recent advances in physically-based appearance modeling of cloth. In *SIGGRAPH Asia 2012 Courses* (Singapore, Singapore, 2012), pp. 12:1–12:52. 32

[TAN*05] TSUCHIDA M., ARAI H., NISHIKO M., SAKAGUCHI Y., UCHIYAMA T., M. Y., HANEISHI H., OHYAMA N.: Development of BRDF and BTF measurement and computer-aided design systems based on multispectral imaging. In *AIC Colour 05 - 10th Congress of the International Colour Association* (2005), pp. 129–132. 67, 119, 120, 121, 134, 242

[Tat05] TATARCHUK N.: Practical dynamic parallax occlusion mapping. In *ACM SIGGRAPH 2005 Sketches* (Los Angeles, CA, USA, August 2005), ACM. 38

[TFG*13] TUNWATTANAPONG B., FYFFE G., GRAHAM P., BUSCH J., YU X., GHOSH A., DEBEVEC P.: Acquiring reflectance and shape from continuous spherical harmonic illumination. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 32*, 4 (July 2013), 109:1–109:12. 134

[TFLS11] TSAI Y.-T., FANG K.-L., LIN W.-C., SHIH Z.-C.: Modeling bidirectional texture functions with multivariate spherical radial basis functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 33*, 7 (July 2011), 1356–1369. 216

[TMJ98] TANNER C. C., MIGDAL C. J., JONES M. T.: The clipmap: a virtual mipmap. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (Orlando, Florida USA, July 1998), pp. 151–158. 213, 215

[TMY12] TAGAWA S., MUKAIGAWA Y., YAGI Y.: 8-D reflectance field for computational photography. In *International Conference on Pattern Recognition (ICPR)* (Tsukuba, Japan, November 2012), pp. 2181–2185. 34, 122, 123, 134

[TSA*05] TSUCHIDA M., SAKAGUCHI Y., ARAI H., NISHIKO M., FUJIKAWA N., YAMAGUCHI M., HANEISHI H., OHYAMA N.: High-fidelity color reproduction based on multi-channel BTF/BRDF acquisition, rendering and display. In *ACM SIGGRAPH 2005 Sketches* (Los Angeles, CA, USA, August 2005), ACM. 67, 119, 120, 121, 128, 130, 242

[TWL*05] TONG X., WANG J., LIN S., GUO B., SHUM H.-Y.: Modeling and rendering of quasi-homogeneous materials. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 24*, 3 (July 2005), 1054–1061. 68, 125, 126, 127, 130, 134

[VG97] VEACH E., GUIBAS L. J.: Metropolis light transport. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (Los Angeles, CA, USA, August 1997), pp. 65–76. 30

[VH74] VENABLE W. H., HSIA J. J.: *Optical radiation measurements: describing spectrophotometric measurements*. National Bureau of Standards Technical Note 594-9, U.S. Department of Commerce, November 1974. 31

[vW09] VAN WAVEREN J. M. P.: Beyond programmable shading I: id tech 5 challenges. In *ACM SIGGRAPH 2009 Courses* (New Orleans, Louisiana, USA, August 2009), pp. 7:1–7:312. 215

[WAA*00] WOOD D. N., AZUMA D. I., ALDINGER K., CURLESS B., DUCHAMP T., SALESIN D. H., STUETZLE W.: Surface light fields for 3D photography. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (New Orleans, Louisiana, USA, 2000), ACM Press/Addison-Wesley Publishing Co., pp. 287–296. 41

[War92] WARD G. J.: Measuring and modeling anisotropic reflection. *ACM SIGGRAPH Computer Graphics 26*, 2 (July 1992), 265–272. 67

[WAT92] WESTIN S. H., ARVO J. R., TORRANCE K. E.: Predicting reflectance functions from complex surfaces. *ACM SIGGRAPH Computer Graphics 26*, 2 (July 1992), 255–264. 36, 37

[WB09] WANG Z., BOVIK A.: Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE Signal Processing Magazine 26*, 1 (January 2009), 98–117. 57

[WBS*04] WANG Z., BOVIK A. C., SHEIKH H. R., MEMBER S., SIMONCELLI E. P., MEMBER S.: Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing 13* (April 2004), 600–612. 57

[WD06] WANG J., DANA K. J.: Relief texture from specularities. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 28*, 3 (March 2006), 446–457. 67, 121, 122, 124, 128, 129, 130

[WDR11] WU H., DORSEY J., RUSHMEIER H.: A sparse parametric mixture model for BTF compression, editing and rendering. *Computer Graphics Forum (Proceedings of Eurographics) 30*, 2 (April 2011), 465–473. 157, 193, 215

[Web13] WebGL specification 1.0.2. Website, March 2013. Available online at `http://www.khronos.org/registry/webgl/specs/latest/1.0/`; visited on November 7th 2014. 191

[WGK14] WEINMANN M., GALL J., KLEIN R.: Material classification based on training data synthesized using a BTF database. In *Proceedings of the European Conference On Computer Vision (ECCV)* (September 2014), Springer International Publishing, pp. 156–171. 240, 271

[WGT*05] WENGER A., GARDNER A., TCHOU C., UNGER J., HAWKINS T., DEBEVEC P.: Performance relighting and reflectance transformation with time-multiplexed illumination. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 24* (July 2005), 756–764. 68, 97, 125, 133

[WHON97] WONG T.-T., HENG P.-A., OR S.-H., NG W.-Y.: Image-based rendering with controllable illumination. In *Eurographics Workshop on Rendering* (St. Etienne, France, June 1997), Eurographics Association, pp. 13–22. 42, 43

[Wik14] WIKIPEDIA: Jpeg – color space transformation. Website, 2014. Available online at `http://en.wikipedia.org/w/index.php?title=JPEG&oldid=601339466`; visited on November 7th 2014. 161

[Wil83] WILLIAMS L.: Pyramidal parametrics. *ACM SIGGRAPH Computer Graphics 17*, 3 (July 1983), 1–11. 215

[WLDW11] WU C., LIU Y., DAI Q., WILBURN B.: Fusing multiview and photometric stereo for 3D reconstruction under uncalibrated illumination. *IEEE Transactions on Visualization and Computer Graphics 17*, 8 (2011), 1082–1095. 68, 97, 125

[WLL*09] WEYRICH T., LAWRENCE J., LENSCH H. P. A., RUSINKIEWICZ S., ZICKLER T.: Principles of appearance acquisition and representation. *Foundations and Trends in Computer Graphics and Vision 4* (February 2009), 75–191. 37, 45, 66

[WLVG07] WEISE T., LEIBE B., VAN GOOL L.: Fast 3D scanning with automatic motion compensation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Minneapolis, Minnesota, USA, June 2007), IEEE Computer Society, pp. 1–8. 145

[WMP*05] WEYRICH T., MATUSIK W., PFISTER H., NGAN A., GROSS M.: *Measuring skin reflectance and subsurface scattering*. Tech. rep., Mitsubishi Electric Research Laboratories, TR2005-046, December 2005. 68, 97, 125, 127, 128, 129, 130, 132, 133, 134

[WMP*06] WEYRICH T., MATUSIK W., PFISTER H., BICKEL B., DONNER C., TU C., MCANDLESS J., LEE J., NGAN A., JENSEN H. W., GROSS M.: Analysis of human faces using a measurement-based skin reflectance model. *ACM Transactions on Graphics (Proceedings of SIGGRAPH) 25*, 3 (July 2006), 1013–1024. 68, 97, 125, 127, 134

[WORK13] WEINMANN M., OSEP A., RUITERS R., KLEIN R.: Multi-view normal field integration for 3D reconstruction of mirroring objects. In *Proceedings of the International Conference on Computer Vision (ICCV)* (Sydney, Australia, December 2013), IEEE Computer Society, pp. 2504–2511. 48

[WRO*12] WEINMANN M., RUITERS R., OSEP A., SCHWARTZ C., KLEIN R.: Fusing structured light consistency and Helmholtz normals for 3D reconstruction. In *Proceedings of the British Machine Vision Conference (BMVC)* (Surrey, UK, September

2012), British Machine Vision Association, pp. 108.1–108.12. 48, 68, 97, 99, 101, 143, 147, 148

[WSB*98] WHITE D. R., SAUNDERS P., BONSEY S. J., VAN DE VEN J., EDGAR H.: Reflectometer for measuring the bidirectional reflectance of rough surfaces. *Applied Optics 37*, 16 (June 1998), 3450–3454. 66

[WSB03] WANG Z., SIMONCELLI E., BOVIK A.: Multiscale structural similarity for image quality assessment. In *Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers* (Pacific Grove, CA, USA, Nov 2003), vol. 2, IEEE, pp. 1398–1402 Vol.2. 58

[WSRK11] WEINMANN M., SCHWARTZ C., RUITERS R., KLEIN R.: A multi-camera, multi-projector super-resolution framework for structured light. In *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIM-PVT)* (Hangzhou, China, May 2011), IEEE Computer Society, pp. 397–404. 86, 91, 101, 141, 143, 144, 146, 147, 178

[WVM*05] WILLEMS G., VERBIEST F., MOREAU W., HAMEEUW H., VAN LERBERGHE K., VAN GOOL L.: Easy and cost-effective cuneiform digitizing. In *International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage (VAST)* (Pisa, Italy, 2005), Eurographics Association, pp. 73–80. 134, 142, 143

[WW11] WILKIE A., WEIDLICH A.: How to write a polarisation ray tracer. In *SIGGRAPH Asia 2011 Courses* (Hong Kong, China, December 2011), ACM, pp. 8:1–8:36. 28

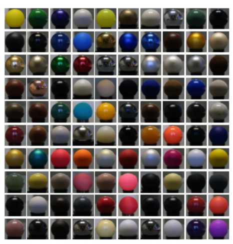
[WXC*08] WU Q., XIA T., CHEN C., LIN H.-Y. S., WANG H., YU Y.: Hierarchical tensor approximation of multi-dimensional visual data. *IEEE Transactions on Visualization and Computer Graphics 14*, 1 (2008), 186–199. 194

[Zha00] ZHANG Z.: A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 22*, 11 (November 2000), 1330–1334. 90, 104, 111

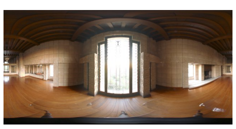
DATA SOURCES

The following data sets used in this thesis have been taken from external sources. We would like to thank the authors for making them available to us.

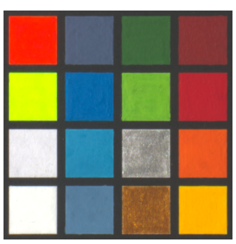
Kitchen light probe
© 1999 Paul Debevec [Deb98]
`http://www.debevec.org/Probes/`



MERL BRDF database
© 2003 Wojciech Matusik [MPBM03]
`http://www.merl.com/brdf/`



Dining room of the Ennis-Brown house light probe
© 2006 Paul Debevec
`http://gl.ict.usc.edu/Data/HighResProbes/`



Colorchecker sample of the SPECTRAL data sets
© 2010 Martin Rump [RSK10]
`http://cg.cs.uni-bonn.de/en/projects/btfdbb/download/spectral/`

UBO2014 BTF data sets
© 2014 Michael Weinmann [WGK14]
`http://cg.cs.uni-bonn.de/en/projects/btfdbb/download/ubo2014/`

List of Figures

LIST OF TABLES