Revealing the Invisible: On the Extraction of Latent Information from Generalized Image Data

Dissertation

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

vorgelegt von
Dipl.-Math. Julian Iseringhausen
aus
Essen

Bonn, Juli 2019

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

Dekan:	Prof. Dr. Johannes Beck
1. Gutachter:	Prof. Dr. Matthias B. Hullin
2. Gutachter:	Prof. Dr. Reinhard Klein
3. Gutachter:	Prof. Dr. Hendrik P.A. Lensch

Tag der Promotion:	17. 12. 2019
Erscheinungsjahr:	2020

Contents

Abstract

The desire to reveal the invisible in order to explain the world around us has been a source of impetus for technological and scientific progress throughout human history. Many of the phenomena that directly affect us cannot be sufficiently explained based on the observations using our primary senses alone. Often this is because their originating cause is either too small, too far away, or in other ways obstructed. To put it in other words: it is invisible to us. Without careful observation and experimentation, our models of the world remain inaccurate and research has to be conducted in order to improve our understanding of even the most basic effects. In this thesis, we[1] are going to present our solutions to three challenging problems in visual computing, where a surprising amount of information is hidden in generalized image data and cannot easily be extracted by human observation or existing methods. We are able to extract the latent information using non-linear and discrete optimization methods based on physically motivated models and computer graphics methodology, such as ray tracing, real-time transient rendering, and image-based rendering.

In Chapter 2, we present our approach on unstructured light field acquisition using water drops as light field imagers. Light fields are a highly influential concept in computational imaging with a wide range of applications. However, thus far light field acquisition has required specialized hardware, a lengthy calibration routine of the imager, or both. Our approach alleviates these limitations by specializing on a particular, but common, scene setup. We utilize water drops on a window as single lenses, each viewing the scene from a different direction. By replacing the calibration routine with non-linear optimization, based on the physi-

[1] I decided to use the word "we" to reference my co-authors and me, the reader and me, and sometimes even me alone. After evaluating multiple variants, I have found that each option has some pros and cons and this is by far the most readable option, especially for readers who frequently read scientific publications.

cally accurate simulation of water drop surfaces, we are able to calibrate and measure light fields from a single photograph of an unknown scene. Using the acquired light field, we render the scene from novel viewpoints and estimate its depth.

In Chapter 3, we reconstruct geometry without a direct line of sight between object, camera, and light source, that is, the object is invisible to the observing camera in the most literal sense. In this setup, there is no way of directly observing the object and the shortest path connecting laser light source, object, and camera contains at least three diffuse reflections. By utilizing ultra-fast transient imaging hardware, we capture a video of light in flight that forms an optical "echo" of the object, analogous to the recording of an acoustic echo. We solve the non-line-of-sight reconstruction problem using a novel analysis-by-synthesis approach that is based on our highly efficient and physically accurate transient renderer as the forward model. Afterwards, we validate our approach on synthetic and measured scenes.

Finally, in Chapter 4 we translate our search for the hidden to an artistic domain. We show that sheets of wooden veneer contain stylized versions of almost arbitrary target images and demonstrate how to reveal them by cutting and rearranging the resulting pieces. Inspired by parquetry, i.e. the mosaic-like, regular placement of pieces of wood, we have developed a discrete optimization method to fully automatically generate pieces of computational art. By embracing the intricate high-frequency structures present in wood and by employing structurally-aware filters, we are able to reconstruct target images at a high resolution using only a small number of wood patches. We demonstrate the effectiveness of our approach by physically fabricating computational parquetry art using a laser cutter.

Zusammenfassung

Der Traum das Unsichtbare sichtbar zu machen, um ein besseres Verständnis unserer Welt zu erlangen, war während der gesamten Menschheitsgeschichte ein unverzichtbarer Antrieb für technologischen und wissenschaftlichen Fortschritt. Viele der Phänomene, die uns jeden Tag direkt beeinflussen, können nicht allein mit Beobachtungen unserer primären Sinnesorgane erklärt werden. Ein häufiger Grund hierfür ist, dass die Ursache entweder zu klein, zu weit entfernt oder anderweitig verdeckt ist. In anderen Worten: Die Ursache ist unsichtbar für uns. Ohne gewissenhafte Beobachtung und Untersuchung verbleiben die Modelle unserer Welt ungenau und wir benötigen weitere Forschung um selbst die grundlegensten Effekte zu verstehen. In dieser Arbeit werden unsere Lösungen zu drei herausfordernden Problemen innerhalb des Forschungsgebietes *Visual Computing* vorgestellt. Generalisierte Bilddaten können eine überraschende Menge an Information enthalten, die weder von Menschen, noch von bestehenden Bildverarbeitungsmethoden extrahiert werden können. Wir zeigen, dass sich diese Information mittels nichtlinearer und diskreter Optimierungsverfahren, welche auf physikalisch motivierten Vorwärtsmodellen basieren, extrahieren lassen. Dabei verwenden wir Methoden der Computergrafik: Raytracing, transientes Echtzeitrendern und bildbasiertes Rendern.

In Kapitel 2 präsentiere ich unseren Ansatz zur unstrukturierten Lichtfeldmessung mittels Wassertropfen. Lichtfelder sind ein weit verbreitetes Konzept innerhalb rechnergestützter Bildgebungsverfahren und sie weisen eine große Anzahl an Anwendungen auf. Jedoch wurden bisher für ihre Messung entweder spezielle Hardware, aufwändige Kalibrationsverfahren oder beides benötigt. Unser Ansatz behebt diese Limitierung durch die Spezialisierung auf ein besonderes (aber häufig vorkommendes) Szenario. Wir verwenden Wassertropfen auf einer Fensterscheibe als einfache, optische Linsen, durch welche wir die Szene aus unterschiedlichen Richtungen betrachten. Wir zeigen, dass sich Lichtfelder mittels eines einzigen

Fotos einer unbekannten Szene in einem Schritt kalibrieren und aufzeichnen lassen. Dazu ersetzen wir den Kalibrierungsschritt durch ein nichtlineares Optimierungsverfahren, welches auf der physikalisch korrekten Simulation der Tropfenoberflächen basiert. Anschließend verwenden wir die gemessenen Lichtfelder zur Bildsynthese und zur Schätzung von Tiefenkarten.

In Kapitel 3 rekonstruieren wir eine unbekannte Geometrie ohne eine direkte Sichtlinie zwischen Objekt, Kamera und Lichtquelle. In dieser Versuchsanordnung besteht keine Möglichkeit das Objekt direkt zu beobachten und der kürzeste Pfad zwischen Laserlichtquelle und Kamera enthält mindestens drei diffuse Reflektionen. Durch die Verwendung von ultraschnellen transienten Kameras zeichnen wir ein Video der Lichtausbreitung innerhalb der Szene auf. Die diffusen Reflektionen formen ein optisches "Echo", analog zu den bekannten akustischen Echos. Wir lösen das Geometrierekonstruktionsproblem mittels eines Optimierungsansatzes, welcher auf unserem hocheffizienten, physikalisch motivierten transienten Renderer basiert. Abschließend validieren wir unseren Ansatz mittels synthetischer und gemessener Datensätze.

Zu guter Letzt übertragen wir in Kapitel 4 unsere Suche nach versteckten Bilddaten auf ein künstlerisches Gebiet. Wir zeigen, dass Echtholzfurnier stilisierte Versionen fast beliebiger Eingabebilder enthält und demonstrieren, wie sich diese durch Schneiden und Neuanordnung des Holzes offenbaren lassen. Inspiriert durch künstlerische Parkett- und Einlegearbeiten, haben wir ein diskretes Optimierungsverfahren entwickelt, welches vollautomatisch unsere computergestützte Kunst erzeugt. Durch Einbeziehung der komplexen Holzmaserungen und die Verwendung strukturerhaltender Bildfilter rekonstruieren wir die Eingabebilder mit einer hohen Auflösung und benötigen dazu lediglich eine kleine Zahl an Mosaikstücken. Wir demonstrieren die Leistungsfähigkeit unseres Verfahrens durch die Herstellung echter computergestützter Kunst mittels eines Lasercutters.

List of Abbreviations

Notation	Description
API	application programming interface
ASIC	application-specific integrated circuit
BRDF	bidirectional reflectance distribution function
CCD	charge-coupled device
CDF	cumulative distribution function
CFA	color filter array
CITES	Convention on International Trade in Endangered Species of Wild Fauna and Flora
CMOS	complementary metal-oxide-semiconductor
CNC	computer numerical control
DNN	deep neural network
DSLR	digital single lens reflex
GPS	global positioning system
GPU	graphics processing unit
JPEG	joint photographic experts group
LED	light emitting diode
MP	megapixel
NLOS	non-line-of-sight
PDF	probability density function
PMMA	polymethyl methacrylate
PNG	portable network graphics
PSNR	peak signal-to-noise ratio
RGB	red green blue
RMS	root mean square
ROS	robot operating system
SfM	structure from motion

Notation	Description
SIFT	scale-invariant feature transform
SPAD	single photon avalanche diode
STEM	science, technology, engineering, and mathematics
TV	total variation
VI-SLAM	visual-inertial simultaneous localization and mapping

Acknowledgments

I would like to thank my advisor Prof. Dr. Matthias B. Hullin for introducing me to the fascinating world of visual computing and for his ongoing support during my PhD studies. I would like to thank my doctoral committee for their efforts in assessing this thesis: Prof. Dr. Matthias B. Hullin, Prof. Dr. Reinhard Klein, Prof. Dr. Thomas Schultz, and Prof. Dr. Jens Schröter. I would like to thank Prof. Dr. Hendrik P.A. Lensch for taking the effort of reviewing my thesis. I would like to thank my coauthors; this thesis would not exist without you (in alphabetical order): Martin Fuchs, Bastian Goldlücke, Weizhen Huang, Matthias B. Hullin, Stanimir Iliev, Nina Pesheva, Michael Weinmann, and Alexander Wender. I would like to thank for their valuable feedback regarding this thesis: Michael Weinmann, Clara Callenberg, Tobias Iseringhausen, and Matthias B. Hullin. I would like to thank for the fruitful and inspiring discussions during the course of my PhD studies (in alphabetical order): Tim Brooks, Clara Callenberg, Robert Cavin, Jiawen Chen, Dennis den Brok, Alexander Dieckmann, Martin Fuchs, Rahul Garg, Tim Golla, Javier Grau, Stefan Hartmann, Max Hermann, Weizhen Huang, Matthias B. Hullin, Jonathan Klein, Reinhard Klein, Tom Kneiphof, Stefan Krumpen, Douglas Lanman, Marc Levoy, Nicholas Trail, Nick Maggio, Rodrigo Martín, Michael Milne, Olivier Mercier, Sebastian Merzbach, Ralf Sarlette, Christopher Schwartz, Heinz-Christian Steinhausen, Julian Straub, Nicholas Trail, Elena Trunz, Zdravko Velinov, Michael Wand, Michael Weinmann, Sebastian Werner, Vitalis Wiens, and Tianfan Xue. I would like to thank X-Rite for partially funding my PhD studies through their scholarship.

CHAPTER 1

Introduction

1.1 Motivation

Revealing the invisible is an exciting prospect that inspires many, scientists and non-scientists alike. It is not surprising that a huge amount of research is being focused on problems that allow us to expand the ability of human vision, resulting in exceptional scientific accomplishments that impact all of humanity. To illustrate, the *four humor theory* dates back to the revolutionary work of Greek physicians Hippocrates of Cos (c. 460 BC to c. 370 BC) and Galen of Pergamon (129 AD to c. 210 AD). It states that all diseases are based on the disorder of the four bodily fluids and thus can be cured by restoring their balance [Gar29, All05]. In western medicine, this theory, together with religious explanations, remained widely accepted for two thousand years. Before the invention of microscopy in the 17th century, there was no instrument to *observe the true cause* for infectious diseases and the world of microbes remained invisible. Using his single-lens microscope, Antonie van Leeuwenhoek was the first to discover bacteria and other microbes (which he called *animalcules, little animals*) in a sample of lake water in 1674 [Pom17]. Following this, scientists have continued to push the boundaries of microscopy and biomedical imaging. Ernst Ruska developed the first electron microscope in 1933 and his brother Helmut Ruska was the first to visualize sub-microscopic pathogens like viruses at the *Laboratorium für Übermikroskopie* in Berlin [KSG00]. Most recently, current generations of transmission electron microscopes have reached an incredible resolution of 0.47 Å, which is about half of the size of a single hydrogen atom [ERKD09]. Medical imaging techniques like x-ray and computed tomography enable us to examine the inside of the human body

non-invasively. On the other end of the scale, in astronomy, gravitational-wave detectors [A+16] and radio telescope arrays [The19] are able to look further and deeper into the universe than the human eye ever could. It is the *curiosity for the invisible* that has led to some of the greatest discoveries in human history and to the emergence of the fields of microbiology, virology, and countless others.

In a photographic context, most digital images contain a layer of information that is directly apparent, usually consisting of the depiction of one or more image subjects. By looking at the image, we can directly answer questions like how many people are shown in the image or whether they are smiling (although semantic image understanding is still an open topic in computer vision). On top of that, images contain a more subtle layer of information, transporting the *mood* of the photograph, which can e.g. be influenced by general image composition or tone mapping. For many real-world scenes, people are naturally good at separating the reflectance of a person or an object from the illumination [FDA01], and to infer cues about the time of day or the weather the photo was taken in. This layer of information is easily understood by human observers, but computer vision algorithms may struggle to analyze it.

In this thesis, we identify that many images contain a deeper, latent layer of information in the sense that it cannot easily be extracted by humans or existing machine vision methods. The extraction of such data is a challenging problem, since it generally leads to highly ill-posed problem formulations. In the following chapters, we will present three challenging visual computing problems that aim to extract hidden content from generalized image data. We approach the problems by combining computer graphics methodology such as ray tracing, transient rendering, and image-based rendering with numerical simulation and non-linear and discrete optimization methods. In Chapter 2, we introduce a method to recover light fields from water drops. Following that, in Chapter 3, we show that we can look around a corner by analyzing a video that captures light in flight, before it reaches the steady state that traditional cameras capture. Finally, in Chapter 4, we demonstrate that a panel of real wood contains almost any target image and we utilize this fact to fabricate pieces of fine art. Figure 1.1 contains an overview of the generalized input image data that serves as the input to our methods, as well as the results that we can infer from the input.

In the following sections, we further motivate the problem, give insights into the different aspects of the problem, and present relevant related work. We have identified a trend in visual computing that we follow in this thesis and also in the following, introductory subsections. Begin-

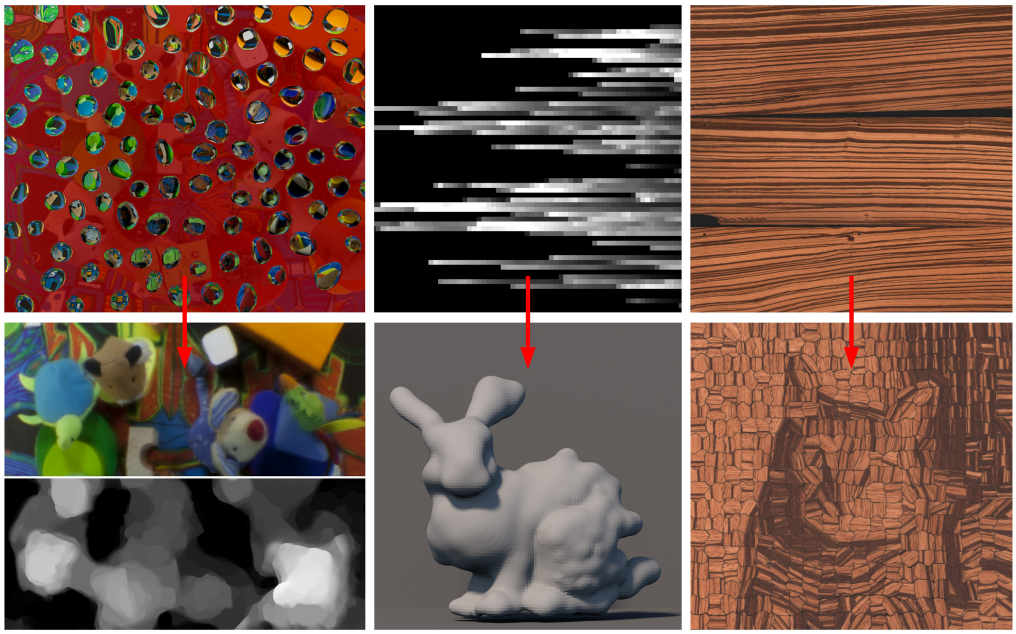


Figure 1.1: Input data and results for the three algorithms presented in this thesis. We are able to extract a surprising amount of information from challenging input data. Depicted here is a range of cute animals, none of which a human viewer could recognize in the input data. Left (Chapter 2): We extract a full, unstructured light field and subsequently synthetic renderings (bottom-left, top half) and depth maps (bottom-left, bottom half) from a single photo of a glass pane with water drops on it. In the input image we have masked out unused pixels (top-left, here marked in red); only water drops are used for light field reconstruction. Each of these drops acts as a little free-form lens, viewing the scene from a different perspective. Center (Chapter 3): We estimate a 3D geometry "around the corner" from a time-resolved transient image. The geometry is hidden from both the light source and the camera, i.e. there is no direct line of sight between camera, light source, and object. Instead, the camera and the laser light source are pointed on a diffuse wall and multi-bounce diffuse reflections are analyzed for geometry estimation. The center top image shows a crop of such a transient image consisting of 64 spatial pixels along the y-axis and 256 time bins along the x-axis. Right (Chapter 4): By cutting and rearranging a panel of wooden veneer, we reconstruct a target image using "wooden pixels", which resembles a fabricated style transfer. Depending on the types of wood used for puzzle generation, different styles can be generated. The output is a fabricated, physical piece of fine art.

ning with the first developments of digital image sensors, many efforts have been taken to improve sensor design in order to capture higher quality image data (Section 1.1.1). Alongside, image sensors have been enhanced in order to directly capture more modalities, like light field data, transient images, or hyperspectral images (Section 1.1.2). However, traditional ways for image quality improvement, like larger sensors or higher-quality optics, are not always applicable. Especially in smartphone photography, the hardware design space is limited by size, thickness, and weight. In order to achieve an image quality comparable to (or even higher than) heavier and larger cameras, computational photography methods have been developed, leading to impressive results (Section 1.1.3). In this thesis, we focus on a class of methods that is built on the same philosophy of (partially) replacing hardware design with computational methods, but goes a significant step further. Many photos and videos of real-world scenes contain features that allow us to extract a surprising amount of hidden information (Section 1.1.4). Some specific scenes even include *computational sensors* that enable us, using scene-specific priors, to infer exciting results from 2D photos, such as light fields or images of what is lying around a corner (Section 1.1.5). In the case of our methods, we are able to achieve this by analyzing scene features that could otherwise be seen as unwanted artifacts (Section 1.1.6). See Figure 1.2 for a graphical overview of the topics covered in this introduction.

1.1.1 Digital image sensors and smartphone photography

The immense ascent of digital image sensors during the last two decades enabled us to capture unprecedented amounts of image data at a consistently decreasing cost. Most common are CCD and CMOS sensors, which can be found in a large number of devices including smartphones, DSLR cameras, mirrorless cameras, action cameras, 360° cameras, and industrial cameras. The first commercially available digital camera was the Cromenco Cyclops in 1975 with a 32×32 pixel count. Nikon's first DSLR was released in 1999 and already had a resolution of 2.7 megapixel (MP). Also in 1999, Kyocera VP-210 Visual Phone with 0.1 MP and a storage capacity of 20 JPEG images was the first camera phone (almost prophetically foreshadowing recent trends in smartphone photography, the Visual Phone featured a single front-facing "selfie camera"). During the following years, the development of digital image sensors accelerated enormously and digital cameras have replaced their analog counterparts in almost all areas. The current generation of high-end DSLR and mirrorless cameras features extreme resolutions of up to 102 MP and Sony's IMX586

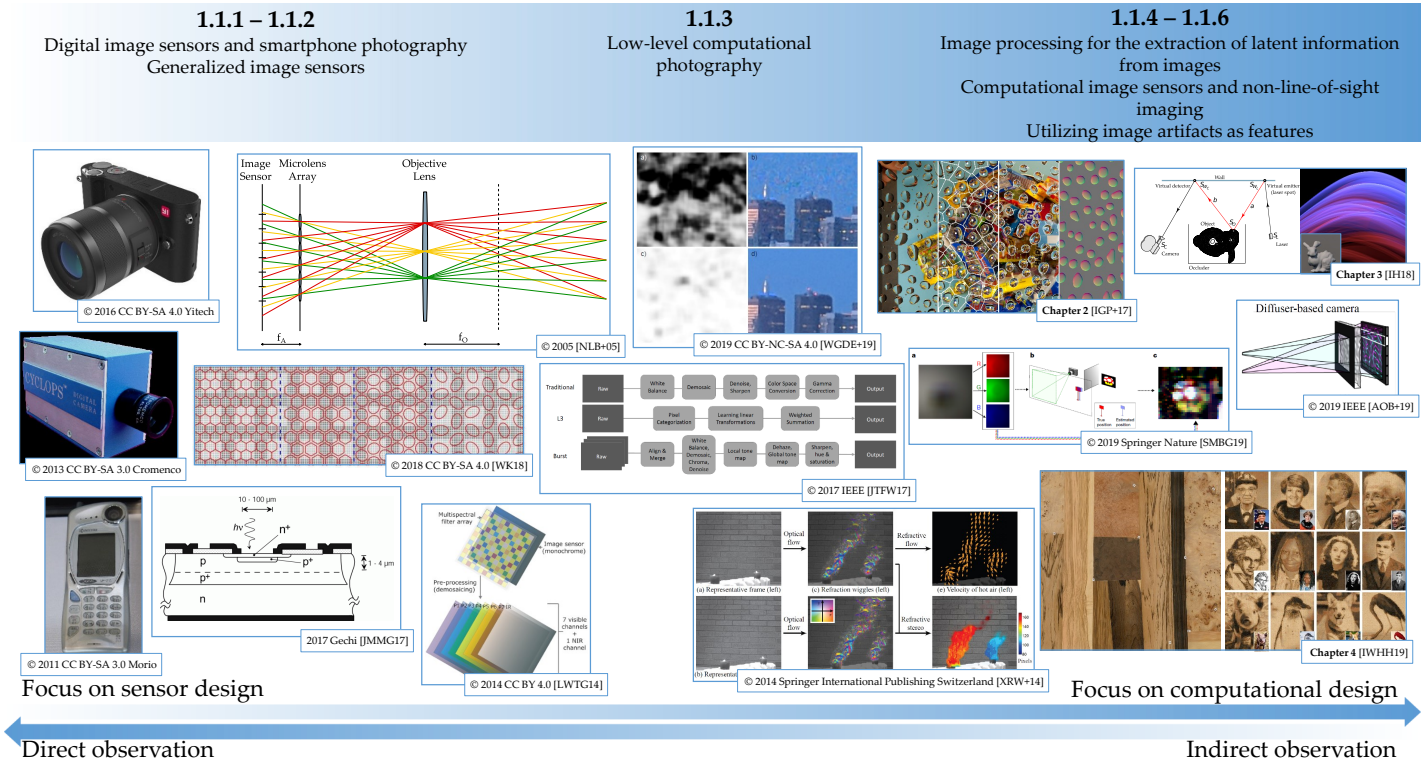


Figure 1.2: We have identified a recent trend that we follow in this thesis and base our introduction on. Beginning with the development of the first digital image sensors, traditionally much emphasis has been put on sensor and optical design (Sections 1.1.1 and 1.1.2). Recently, more and more efforts are taken in the development of computational methods to accompany hardware design, leading to impressive results in computational photography for smartphones (Section 1.1.3). This philosophy is taken even further by the class of visual computing methods our approaches belong to. Rather than relying on direct observation, these methods are designed to extract hidden content from highly uncontrolled image data using computational methods (Sections 1.1.4 to 1.1.6). In this thesis we present our results to extract light fields from water drops (Chapter 2), geometry "around the corner" (Chapter 3), and stylized images from real wood (Chapter 4).

CMOS smartphone image sensor has a resolution of 48 MP formed in a quad Bayer color filter array (CFA).

Today's smartphones are ubiquitous and combine high-resolution digital image sensors, strong processors and wireless connectivity in a compact and portable package. Especially the thin form factor of smartphones has led to new challenges when compared to DSLR photography. The hardware design space for smartphone photography is limited by external factors such as power consumption, cost, weight, and size. This leads to small sensors, small pixels, and small apertures, sometimes paired with fairly inexpensive lenses. The sensor size limits the spatial image resolution, small pixels limit the signal to noise ratio, and small apertures limit the light gathering capability of the whole system in low light scenarios. For comparison, the pixel pitch of a Google Pixel 3 smartphone measures 1.4 µm, while a high-end Sony α9 DSLR features 5.9 µm pixels, which results in an approximately 18 times larger pixel area. Smaller and simpler lens systems, sometimes even made of synthetic materials instead of glass, lead to increased optical aberrations compared to high-quality (but heavy) DSLR lenses. Because of the limited depth of the smartphone body, some of the most recent smartphones employ periscope lenses in order to achieve longer focal lengths [Fau]. Triggered by the rapid developments in sensor technology and in order to alleviate these shortcomings compared to larger camera systems, image processing methodology has been advanced impressively during the last decade.

1.1.2 Generalized image sensors

One way of increasing the amount of information that we are able to infer from image data is by modifying the image sensor to directly acquire the desired modalities. To this end, many efforts have been taken in recent years. Traditional approaches include enhancing the fill factor, improving the dynamic range, speeding up readout times, or reducing the noise in the readout electronics. More fundamentally, imaging sensors have been modified for specific tasks or to capture additional information. Nayar and Mitsunaga [NM00] improve the dynamic range of an image sensor by adding a spatially varying pattern of gray filters, similar to a Bayer color filter array. Sensors can be equipped with multispectral filter arrays to capture more frequencies than the ones with typical RGB color filter arrays [LWTG14], which allows to capture spectral images in a single photo, but complicates the demosaicking process. Traditionally, image sensors have a square pixel layout. For specific applications, different pixel arrangements, such as hexagonal or elliptical pixels, can be advantageous

[WK18]. Generalized image sensors measure additional information compared to the 2D intensity images of a conventional image sensor. Ng et al. [NLB+05] capture 4D light fields using a hand-held camera by adding a microlens array in front of the sensor. Three-dimensional surface geometry can be measured using passive acquisition methods, such as stereo vision or structure-from-motion, or by active acquisition methods, such as time-of-flight imaging [WK15]. Light travels at a fast, but nevertheless finite speed. By coupling ultra-fast sensors and light sources, transient imaging systems capture videos of light in flight, resulting in a three-dimensional (two spatial dimensions, one temporal dimension) data cube [JMMG17]. In medical imaging, computed tomography scans and magnetic resonance imaging are used to form non-invasive 3D images of the human body. Since this generalized data is stored digitally, we are able to explore, analyze, and process it algorithmically.

1.1.3 Low-level computational photography

In the following, we will focus on methods that enhance image data by switching from a hardware-dominated design principle to algorithmic developments. Following this path allows us to extract high-quality results, while at the same time reducing the hardware dependency. By utilizing the superior computing capabilities of smartphones compared to DSLR cameras, current smartphones exhibit an image quality that in many cases matches, and sometimes even surpasses, the one of DSLR cameras. One area where smartphones have particular benefits is the area of usability-centered features. In order to obtain high-quality photos from a DSLR or mirrorless camera that transport the intended emotions, raw image editing is often inevitable and can be a time-consuming task that requires a deep understanding of the photo editing software. Recently, there is a trend to automate many of the tools contained in such software suites and to directly build them into the smartphone's camera app. This enables even non-professional users to capture high-quality photos without the burden of excessive manual image editing. Some of the tools contained in modern computational photography pipelines have previously not been available in traditional pipelines at all. Gharbi et al. [GCB+17] trained a network to automatically apply tone mapping to a high dynamic range image that results in a professionally-looking output image which faithfully depicts reality. Hasinoff et al. [HSG+16] are able to effectively reduce the sensor noise in low-light scenarios and to expand the dynamic range of the resulting image by aligning and fusing bursts of image frames. Instead of demosaicking each camera frame individually, Wronski et al.

[WGDE+19] developed a hand-held multi-frame super-resolution method that generates an RGB image directly from a burst of raw images, using the natural hand tremor as a source for estimating subpixel displacements. Because of the small apertures, smartphones typically have a large depth of field, which can be beneficial e.g. for landscape photography. In portrait photography on the other hand, often a shallow depth of field is desired in order to separate the foreground subject from the background. To simulate a shallow depth of field and to render a realistic blur effect, Wadhwa et al. [WGJ+18] estimate the scene depth from a single, monocular image. Further challenges include the limited zoom range, which is solved by fusing images from multiple (typically two or three) camera modules with different focal lengths. In all these cases, algorithmic developments complement the highly restricted hardware design in order to achieve an image quality, which is similar to or even exceeds the one of DSLR cameras. Low-level computational photography methodology is used to synthesize new, high-quality images from limited, noisy input data.

1.1.4 Image processing for the extraction of latent information from images

The aforementioned methods belong to the class of low-level computational photography methods that, given one or more 2D camera images, render a new 2D image that faithfully depicts reality and is of high quality. Naturally, one is not restricted to infer 2D renderings from 2D image data. Branching from these classical computational photography methods, we have identified a broader range of image processing methods that draw from a similar philosophy and that are highly related to our line of research. By analyzing a given image, we might be able to draw more information from it. In addition to the obvious content, like subject, scene, and environment, image data can hold additional collateral information. Some of this content is easy to parse for a human observer, but difficult to interpret for machines. Questions about image semantics might easily be answered by a person, but pose a considerable machine vision challenge. For example, by answering the question whether everyone in the frame is smiling and looking into the camera, a smartphone camera app is able to take a picture just in the right moment [SA19].

The extraction of latent information from image data is an important, recent research topic and relates to basic research in image processing and image understanding. Often these problems require creative approaches that have the potential to also expand the algorithmic toolbox in other dis-

ciplines of visual computing. Xue et al. [XRW+14] estimate the movement of hot air in a video by analyzing small distortions of the background. Similarly, such almost inconceivable motions were used to turn objects, such as a bag of chips, into visual microphones [DRW+14]. Tiny motions and color changes in videos can be magnified in order to make them visible to a human observer [WRS+12, WRDF13, OJK+18]. Xu et al. [XFM14] reveal which video is running on a television by extracting intensity-based features from the flickering lights that can be seen in windows from the outside. In Chapter 4 we show that a panel of fine wood veneer can contain almost any target image. By cutting and rearranging the wooden veneer, we are able to generate a real wood puzzle that transfers the wood's style onto the target image.

1.1.5 Computational image sensors and non-line-of-sight imaging

Many everyday photos contain *computational image sensors* that can reveal an additional understanding of the scene, such as information about the propagation of light in the scene or 3D geometries. The reflection in a person's eye can be used to estimate an environment map used for image relighting [NN04]. This observation essentially turns the eye into such a computational image sensor. The reflections on both eyes of a person can form a stereo corneal imaging system and by analyzing the epipolar geometry, a 3D model of the scene behind the camera can be extracted [NN06]. In a more general setting, Georgoulis et al. [GRR+17] trained a model to estimate an environment map from a single photograph of an arbitrary, non-Lambertian object. On a much larger scale, Hasinoff et al. [HLGF11] reconstruct an image of the earth from diffuse reflections off the moon's rim. One of the most stunning recent computational imaging results is the imaging of the black hole at the center of galaxy M87, where an array of eight radio telescopes around the earth and a wide frequency bandwidth was utilized [The19]. The key ingredient to any of these image formation procedures is computational, as none of these problems could have been solved in a purely optical way. In Chapter 2, we will present a method to reconstruct a dense, unstructured light field from a single photograph of a window with water drops on it. The calibrated, unstructured light field data is then further processed to generate synthetic renderings and depth maps of the unknown scene.

One particularly intriguing question to ask is what is lying outside the camera's field of view. Since there is no direct line of sight, there is con-

sequently no immediate way for a direct observation. By careful examination of the scene structure, even reflections from diffuse objects can be utilized to recover hidden scene features. One of the first examples of an accidental image sensor for non-line-of-sight imaging was provided by Torralba and Freeman [TF14]. They show that a window can act as a pinhole, turning the room into a camera obscura, and visualize what is lying outside of the room. Similarly, an occluder in the light path can form a pinspeck camera. Bouman et al. [BYY+17] reconstruct one-dimensional non-line-of-sight videos by analyzing the penumbra of a corner and use it to track people outside the camera's line of sight. In a similar setting, Baradad et al. [BYY+18] recover 4D light fields from the shadows cast from an a priori known occluder. Most recently, Yedida et al. [YBT+19] lifted this restriction by jointly estimating the occluder and a 2D image of the occluded scene. Saunders et al. [SMBG19] bring this to a pinspeck setting by utilizing an occluder with known shape, but unknown position. The data used for image formation is not necessarily restricted to electromagnetic waves of the visible spectrum. It has been shown that WiFi signals can be utilized to infer human poses behind a wall [ZLAA+18]. Kirmani et al. [KHDR09] were the first to use femtosecond transient imaging to solve the non-line-of-sight geometry reconstruction problem. In this setting, the shortest path from light source to camera contains at least three diffuse reflections which introduces ambiguities and makes the problem highly ill-posed. In Chapter 3, we develop a novel analysis-by-synthesis approach to the problem which is based on a highly efficient transient renderer.

1.1.6 Utilizing image artifacts as features

A strong connection between our methods is the type of features we utilize for extracting the latent information. We have identified that image content which is traditionally considered as unwanted artifacts or noise can indeed serve as a valuable source of information and all of our methods build on this observation. Antipa et al. [AOB+19] attach a random, diffuse optic to a bare image sensor in order to spatially compress information from the whole scene on each sensor row. By utilizing the rolling shutter that is inherent to many CMOS sensors, they are able to recover a high-speed video of the scene from a single exposure. Traditionally, diffuse optics and rolling shutter are both seen as undesired parts of an imaging system. When capturing a photo on a rainy day, rain drops on windows are often considered as unwanted artifacts, obstructing the view on the actual scene. There is a number of publications that deals with the au-

tomatic or semi-automatic removal of water drops [EKF13, LWYS13] and rain [ZP18, YTF+17] from photographs. Instead of trying to remove these "artifacts", we exploit them as free-form lenses in Chapter 2. In an analogous manner, diffuse, multi-bounce reflections are utilized as a feature for non-line-of-sight geometry reconstruction in Chapter 3. In common structured light setups, such reflections would act as contributions to an unwanted global illumination term that has to be corrected for [GAVN11]. Similarly, in Chapter 4 we use characteristics in wood veneers, that could otherwise be seen as unwanted imperfections, to fabricate fine art. Our approach demonstrates that knotholes and irregularly structured pieces of wood often turn out to be high-quality features for reconstructing a stylized target image.

1.2 Contributions and publications

In Sections 1.2.1 to 1.2.3, we will describe the individual technical contributions that form this cumulative thesis. The corresponding publications, along with other related publications of the author, are listed in Section 1.2.4.

1.2.1 4D imaging through spray-on optics

Light fields form a compelling theory in visual computing, with applications in image-based rendering [LH96, GGSC96, OEED18], medical biology [BSH+17], microscopy [LNA+06], material recognition [WZH+16], face reconstruction [FGWM18], and many other areas. Based on geometric optics, light fields measure the radiance along rays in space and form a subset of the plenoptic function [AB91]. While conventional 2D cameras only measure light intensity, 4D light field cameras additionally sample the direction of rays. This extra information enables certain image operations, such as refocusing or view point changes, to be conducted post-capture. Typical approaches for hand-held plenoptic cameras include placing a lenslet array either in front of the sensor plane [NLB+05] or in front of the main lens [GZC+06]. Other approaches include camera arrays [WJV+05] and robot gantries [LH96]. These imagers typically sample the light field in a structured manner, which simplifies further processing of the measured data, but requires specialized hardware.

Closely related to our work are casual, random, and accidental light field cameras, which shift the design efforts from optics to algorithms. It has been shown that light fields can be sampled using random optics,

like glitter [ZIA14, SP16], randomized lenses [FTF06], or diffuse optics [ANNW16, AKH+18, AOB+19]. In our own previous work [WIG+15], we show that even a wide range of household optics can be used as light field imagers. These kinds of imagers typically generate unstructured light field measurements in the sense that the transformed light rays are incoherent. This requires additional attention for traditional applications such as image synthesis or depth estimation. On the other hand, for many applications, methods based on deep neural networks (DNNs) do not rely on a semantic coherence of the input data and can benefit from the more diverse and less redundant input data compared to structured light fields.

Instead of custom and possibly expensive acquisition hardware, our approach uses a conventional camera and lens system. In our case, the spatio-angular light field sampling is not achieved by specialized lens systems, but by the captured scene itself, which contains one or more *light field transformers*. Light field imaging using arbitrary light field transformers generally consists of two steps: a geometric calibration and a measurement step. Before entering the camera, light rays pass through the light field transformer and get refracted. During calibration, we generate a mapping from camera pixels to the transformed rays in space. This can be achieved by displaying structured light patterns on a display with known position [KPL08] and can be, depending on the number of patterns, a time-consuming task. Afterwards, we sample the light field leaving a scene by capturing another image using the (now calibrated) light field imaging system.

During our initial research using everyday items as light field transformers [WIG+15], we found a number of downsides using this approach, which limit the practical applicability. First, the light field transformer itself has to remain unaltered between calibration and measurement. This means that we are not able to use volatile media as light field imagers using this approach. Second, in order to maintain the validity of the calibration, the position and orientation of primary camera and light field transformer have to remain fixed with respect to each other. This restriction essentially prohibits any hand-held applications. In Chapter 2, we alleviate these restrictions and tackle a much harder problem. By specializing on a particular but common setup, we are able to combine light field calibration and acquisition into one step, using only a single image of an unknown scene. Individual water drops on a window form the light field transformer by acting as lenses, viewing the scene from different directions. Water drops form excellent single lens systems (their surface is almost perfectly smooth), but due to evaporation they are highly volatile as well. In order to be able to generate a pixel-to-ray mapping using ray

tracing, we need to recover the unknown water drop surfaces. Since the water drop surfaces are energy-minimizing, they are uniquely determined by the drop's outline and volume. We detect the water drop outlines using a semi-automatic image segmentation approach and develop a novel, non-linear optimization scheme to estimate the volume. Each water drop captures a partial view of the underlying scene that overlaps with neighboring views that contain common scene features. By observing that the rays corresponding to the same scene feature but different water drops have to meet at the same (unknown) point in space, we are able to formulate the optimization as a bundle adjustment problem, jointly optimizing a cloud of 3D features and volume parameters. We validate the accuracy of the ray-space calibration and water drop surface geometries numerically on synthetic scenes. For a variety of measured static and dynamic scenes, we are able to demonstrate the effectiveness of our method by generating consistent all-in-focus renderings and depth maps from the calibrated light field data. On a higher level, we expand the space of casual light field imaging methods significantly, by showing that light fields of highly uncontrolled (but specific) scenes can be measured in a single shot using commodity hardware.

1.2.2 Non-line-of-sight reconstruction using efficient transient imaging

One of the most basic prerequisites for virtually any optical geometry reconstruction method, such as photometric stereo, structured light, laser triangulation, time of flight, or multi-view stereo, is a direct line of sight between object, sensors and light sources [WK15]. However, recent advances in transient imaging enabled an exciting alternative. Current ultrafast imaging techniques allow us to record videos of light in motion with temporal resolutions down to the order of femtoseconds [JMMG17]. Non-line-of-sight (NLOS) geometry reconstruction treats the case where the object is hidden from both camera and light source. Instead, the (typically Lambertian) object can only be seen "around the corner", i.e. via diffuse reflections off a planar surface that is mutually visible from object, camera, and light source. In a typical measurement setup, a laser is pointed at a diffuse wall where its reflection acts as a cosine lobe light source. The shortest optical path from light source to camera consists of three diffuse reflections from the wall, to the object, and back to the wall, where an optical "echo" is formed that is picked up by the camera. Due to the diffuse reflections, the geometry reconstruction problem is highly ill-posed and

we exploit the time-resolved transient image to draw sufficient information for reconstruction.

Building on our insights from Chapter 2, we again approach this problem in an analysis-by-synthesis manner based on a physically motivated forward model. Our core contribution in this publication is the non-linear, non-convex global optimization scheme that is used to extract the latent geometry information from this challenging input data. The optimizer is built around a novel global refinement scheme that is based on the implicit surface of sums of Gaussian radial basis functions and uses the Levenberg-Marquardt method [Lev44, Mar63] as the non-linear least squares solver in each refinement step. A geometry that is reconstructed using our method usually consists of 50 to 200 Gaussian blobs with four unknowns (position and size) each. It is prohibitive to solve the problem directly due to local minima caused by the non-convexity of the problem. Instead, in order to greatly improve the probability for global convergence and effectively reduce the number of simultaneously optimized variables, our global refinement scheme employs a heuristic that optimizes only a subset of the Gaussian blob parameters at a time.

At the heart of our approach is the forward model, which consists of an extremely efficient, GPU-based transient renderer based on radiative transfer. Real-time rendering performance is achieved by specializing the renderer to the aforementioned, most common scene setup with three light bounces from light source to camera. One of our main contributions regarding the transient renderer is a new linear temporal filter that allows a single triangle to be smeared over several time bins. Using this filter, we are able to generate smooth renderings which are suitable to be used in our optimization pipeline at a substantially lower run time than the naïve approach without temporal filtering. Second, we employ an efficient shadow test in order to avoid light intensity overestimation for a near-physical handling of occlusion effects. Previous real-time transient renderers only supported flat or convex geometries without occlusion effects. By comparing our real-time renderer against an offline ray tracer, we show that each of our augmentations is vital for achieving physical realism. Regarding our overall method, we are able to show that our approach beats the performance of the state-of-the-art [AGJ17] on synthetic data and produces comparable results on measured data.

On a higher level, our contributions are as follows. We are the first to solve the non-line-of-sight reconstruction problem using a purely physically motivated scene representation consisting of a surface-oriented scattering model. Therefore we avoid the systematic bias imposed by approaches that are not based on a physically accurate light transport model.

We pose the NLOS reconstruction problem as a non-linear optimization problem and solve it using our custom global optimization approach. By approaching this problem in an analysis-by-synthesis manner, we can expect our results and reconstruction times to improve whenever the state-of-the-art in transient rendering evolves, e.g. using neural rendering methods.

1.2.3 Computational parquetry: fabricated style transfer with wood pixels

In Chapter 4, we transfer our ideas from the previous chapters to a new domain. Given a target image, our goal is to fabricate a stylized version of the image using real, physical materials. For this purpose, we utilize sheets of wooden veneer containing one or more kinds of real wood to translate the input image into the real world. The appearance profiles of different wood types include low-frequency features (color) as well as high-frequency features (grain structures) and form the basis for the image stylization. Starting with scans of the wooden veneer, we apply a novel, discrete optimization scheme to calculate an optimal way of cutting and shuffling the panels in order to generate a mosaic-like puzzle as a fine-art rendition of the target image. Afterwards, we apply the computed cut patterns using a laser cutter in order to fabricate the parquetry puzzle. The cut pieces are assembled in the correct order and orientation, fixed on a substrate, and a finish is applied.

One of our core technical contributions lies in the combinatorial optimization scheme that operates with a minimum amount of input data (a target image and one or more source textures) in order to generate fabricable cut patterns. In our previous publications, the results had to be reconstructed from the input data by solving ill-posed inverse problems. This time, the stylized target image is almost "hidden in plain sight", as all of its features are directly contained in the wooden veneers. Yet it is completely concealed and revealing it requires cutting and rearranging the veneer, guided by optimization. This problem is related to style transfer and texture synthesis in the sense that we try to translate the appearance of a source texture (wooden veneer) onto the target image. One of the main difficulties in our method is to produce faithful renditions of the target image while still enforcing fabricability, which has a number of consequences. While methods that are purely concerned with the reproduction of digital images can draw from a wide range of image operations, our set of algorithmic tools is limited to cutting the source texture and ap-

plying rigid transformations to these cut-out patches. Additionally, we have to enforce that each piece of wood is only used once, which deviates from patch-based or pixel-based texture synthesis. Furthermore, the physical fabricability also poses a restriction on the applicable optimization methods. While there exists an abundant amount of well-performing style transfer methods based on deep neural networks, to our knowledge there is no learning-based method that is able to solve the kinds of combinatorial optimization problems that our approach requires.

We consider our highly scalable end-to-end pipeline to be our second core contribution. We have demonstrated that the whole parquetry generation pipeline can be implemented entirely using commonly available, hobby-grade hardware. The scans can be conducted using a common flat bed scanner or a calibrated camera and a basic laser cutter is sufficient for cutting the optimized pieces. Thus the whole system could be employed by enthusiast amateurs or hacker spaces. On the other end of the scale, our method could also be implemented on an industrial level in a "parquetry as a service" model, where users upload their target image to a web service. After the system computes a cut pattern, a preview is generated using image-based rendering. The result is presented to the user and the puzzle can be ordered. Then, the puzzle pieces are cut using an industrial-grade laser cutter and are delivered to the customer, together with material required for assembly and instructions. One of the most labor-intensive steps in our pipeline is the final assembly. This process is done by the the user, which helps to reduce the price of the product. The final assembly resembles classical puzzling paired with basic wood working, which are both activities that many enjoy. Finally, experts are able to produce high-quality pieces of computational parquetry using our pipeline, which could be displayed in fine art, automotive, or furniture environments.

1.2.4 List of publications

The following publications are the core contributions of this thesis and form Chapters 2, 3 and 4 respectively:

- **Chapter 2:** J. Iseringhausen, B. Goldlücke, N. Pesheva, S. Iliev, A. Wender, M. Fuchs and M. B. Hullin: 4D Imaging through Spray-On Optics. In *ACM Transactions on Graphics 36(4) (**Proc. SIGGRAPH 2017**)*, July 2017.
- **Chapter 3:** J. Iseringhausen and M. B. Hullin: Non-Line-of-Sight Reconstruction using Efficient Transient Rendering. *arXiv:1809:08044 [cs.GR], ACM Transactions on Graphics (to appear)*, September 2018.
- **Chapter 4:** J. Iseringhausen, M. Weinmann, W. Huang and M. B. Hullin: Computational Parquetry: Fabricated Style Transfer with Wood Pixels. *arXiv:1904.04769 [cs.GR], ACM Transactions on Graphics (to appear)*, April 2019.

In the following, we list other related publications that the author contributed to, sorted in reverse chronological order:

- S. Werner, J. Iseringhausen, C. Callenberg, M. B. Hullin: Trigonometric Moments for Editable Structured Light Range Finding. *Proceedings of Vision, Modeling, and Visualization 2019*, October 2019.
- J. Iseringhausen, R. D. Cavin, N. D. Trail, D. R. Lanman: Eye Tracking System using Dense Structured Light Patterns. *US Patent App. 15/722259*, April 2019.
- A. Wender, J. Iseringhausen, B. Goldlücke, M. Fuchs and M. B. Hullin: Light Field Imaging through Household Optics. In *Proceedings of Vision, Modeling, and Visualization 2015*, October 2015.
- R. Martín, J. Iseringhausen, M. Weinmann and M. B. Hullin: Multimodal Perception of Material Properties. In *Proceedings of ACM SIGGRAPH Symposium on Applied Perception*, September 2015.

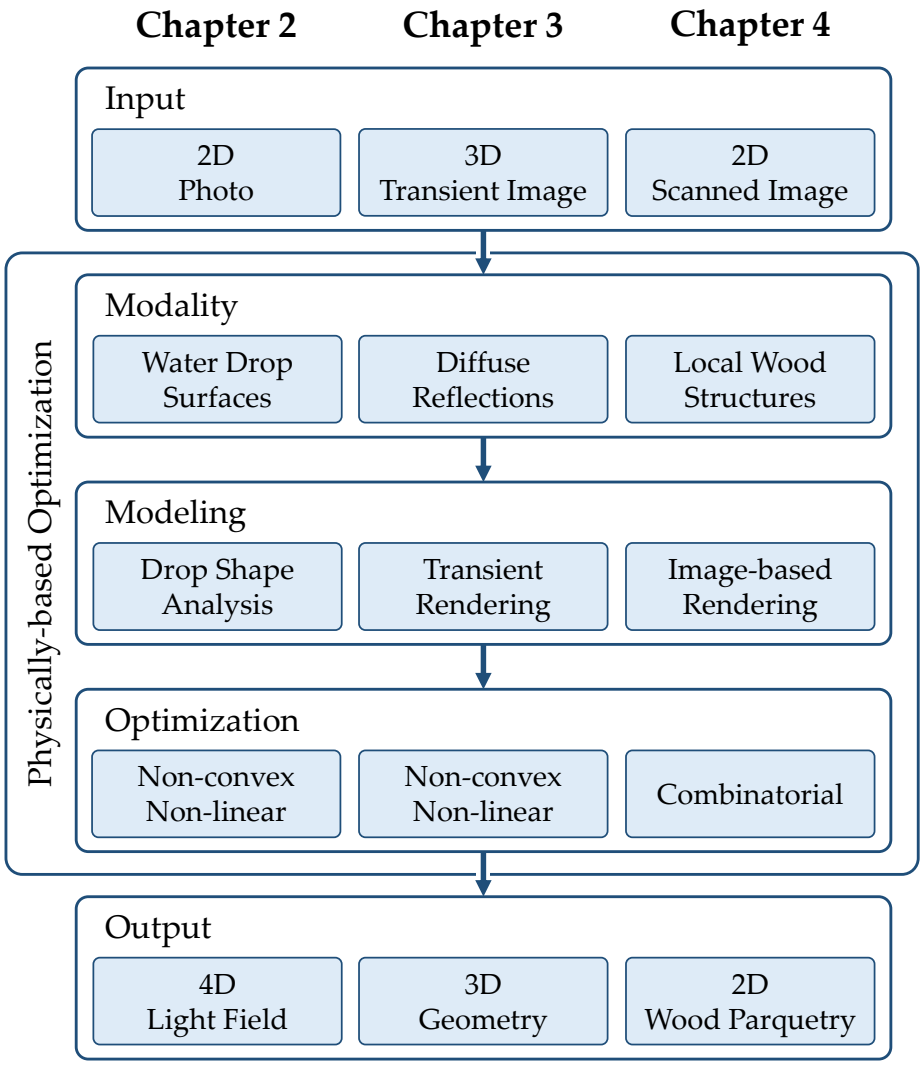


Figure 1.3: Structural overview of the presented methods. In this thesis, we follow a common approach for the extraction of latent information from generalized image data. After analyzing the input data and the underlying scene, we identify a scene modality to exploit and develop a physically-based model. The model is then utilized in a customized optimization scheme that reconstructs the output.

1.3 Outline

The rest of this cumulative thesis is structured as follows. In the following chapters, we present our individual publications that form the thesis. We have built our methods on a common approach that relies on physically based models of the underlying scene and light transport, and on task-specific, novel optimization methods to extract the latent information from the generalized image data. See Figure 1.3 for a structural overview over the methods. In Chapter 2, we present our paper "4D Imaging through Spray-on Optics" [IGP+17], that has been presented at SIGGRAPH 2017. Here we utilize a single image of a glass pane with water drops on it to calibrate and acquire a 4D light field. Using a custom bundle adjustment scheme, we are able to extract a full ray-space calibration even though the captured scene is unknown. Chapter 3 features our approach for "Non-Line-of-Sight Reconstruction using Efficient Transient Rendering" [IH18] in a revised form that has been accepted to ACM Transactions on Graphics. Building on a highly efficient, approximate forward model based on transient rendering, we are able to reconstruct geometries without a direct line of sight from camera and light source in an analysis-by-synthesis scheme. Chapter 4 consists of the final paper presented in this thesis, where we introduce "Computational Parquetry: Fabricated Style Transfer using Wood Pixels" [IWHH19]. This work is currently under review at Transactions on Graphics. We show that scans of wooden veneer panels contain sufficient latent information to act as source textures for style transfer onto a wide range of target images. By employing a novel, discrete optimization scheme, we are able to generate cut patterns that are fabricable using a laser cutter and demonstrate this by producing and assembling a number of fine art wooden parquetry puzzles. Finally, in Chapter 5 we conclude this thesis with a discussion and a future work section.

In this chapter, we present our physically-based optimization approach to the single-shot light field reconstruction from water drops. The method forms the inspirational foundation for the following publications in Chapters 3 and 4.

This chapter was published as [IGP+17]: Julian Iseringhausen, Bastian Goldlücke, Nina Pesheva, Stanimir Iliev, Alexander Wender, Martin Fuchs and Matthias B. Hullin: "4D Imaging through Spray-On Optics". In *ACM Transactions on Graphics 36(4) (Proceedings of SIGGRAPH 2017)*, July 2017.

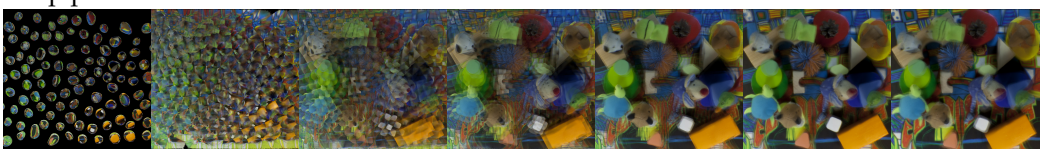
CHAPTER 2

4D Imaging through Spray-on Optics

Abstract Light fields are a powerful concept in computational imaging and a mainstay in image-based rendering; however, so far their acquisition required either carefully designed and calibrated optical systems (micro-lens arrays), or multi-camera/multi-shot settings. Here, we show that fully calibrated light field data can be obtained from a single ordinary photograph taken through a partially wetted window. Each drop of water produces a distorted view on the scene, and the challenge of recovering the unknown mapping from pixel coordinates to refracted rays in space is a severely underconstrained problem. The key idea behind our solution is to combine ray tracing and low-level image analysis techniques (extraction of 2D drop contours and locations of scene features seen through drops) with state-of-the-art drop shape simulation and an iterative refinement scheme to enforce photo-consistency across features that are seen in multiple views. This novel approach not only recovers a dense pixel-to-ray mapping, but also the refractive geometry through which the scene is observed, to high accuracy. We therefore anticipate that our inherently self-calibrating scheme might also find applications in other fields, for instance in materials science where the wetting properties of liquids on surfaces are investigated.

(a) (b) (c)

Drop plane near ⟷ far Back of scene



(d)

Figure 2.1: Using liquids to image light fields ("Animals" example). (a), Our capture setting: the scene is observed by a 2D camera (not in illustration) through a wetted window. Light rays falling through water drops are refracted and sample the scene's light field. (b), Our input is a single image of the scene, as seen by the primary camera. Using drop shape simulation, we establish tentative pixel-to-ray mappings that allow to undistort the individual drop views (c) and, after further refinement, to render a weighted focal stack (d).

2.1 Introduction

Light fields [LH96, GGSC96] describe light leaving a scene on a ray-by-ray basis. They do not only form the foundation of image-based rendering, but have also been shown to facilitate the solution of long-standing vision problems such as depth estimation. For the capture of light fields, few commercial solutions are available; to this day, 2D imagers by far dominate the market. The defining component of a light field imager is an optical and/or mechanical system that maps the 4D space of rays onto the 2D sensor plane. Most such systems are carefully designed to trade between spatial and angular resolution, and to achieve optimal overall imaging performance by maximizing light efficiency and sharpness while avoiding cross-talk and aliasing, all under the given design constraints. On the other end of the scale are "casual" or "random" light field cameras that use every-day reflective or refractive objects [WIG+15] or randomized optical elements [FTF06, ANNW16]. They replace careful optical design by exhaustive calibration of the pixel-to-ray mapping. Here, we take this idea of exploiting low-end optical devices for integral imaging a significant step further. By focusing on a particular, but very common, optical scenario (a window wetted by water drops), we can make extensive use of domain knowledge and physical simulation to greatly facilitate the calibration process. The result is a heterogeneous pipeline that comprises low-level image analysis steps for drop segmentation and feature detection, drop shape simulation to recover the refractive geometry, and a custom bundle adjustment scheme to refine the estimated geometry. With that, our work for the first time enables both the calibration of a dense pixel-to-ray mapping and the acquisition of a light field from a single input image taken through a wetted window.

We consider the following to be our key contributions:

- We propose the use of physical simulation to facilitate the calibration of a-priori unknown imaging systems; in particular, liquid drops as optics for light field imaging.
- We introduce a pipeline for ray-space calibration and the extraction of light field data from a single input image. It combines simple image analysis steps with drop shape simulation, an algorithm for matching and refinement of 2D features, and a custom bundle adjustment scheme to jointly estimate a cloud of sparse 3D features and refine the estimated drop geometry.
- We experimentally validate our pipeline on a selection of static and dynamic scenes.

- Finally, for lack of experimental ground truth data, we evaluate the accuracy of our ray-space calibration and the recovered 3D water drop geometries using synthetic experiments.

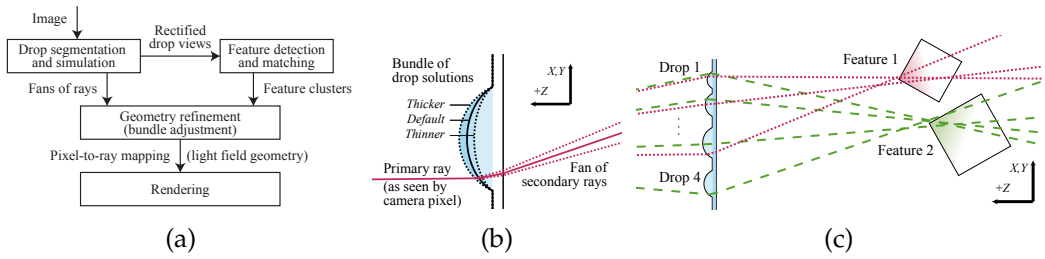
2.2 Related work

Before we explain our method in detail, we will start by discussing existing works that served as a source of inspiration for our work.

Liquid mirrors and lenses. Liquids have been used for optical purposes throughout history, but it was not until the late 19th century that a rapid technical developments and deeper physical understanding enabled astronomers to construct mirror telescopes from liquid mercury, a technology that is still in use today [HBC$^+$98]. In technical optics, today's possibilities include variable lenses controlled e.g. by microfluidic channels [CLJL03] or electrowetting [KH04], and the fabrication of microlens arrays from photoresist through reflow processes [OS02]. The computer graphics community has discovered water not only as a natural phenomenon worthy of digital simulation, but also as a display medium [BNK10, HLR$^+$11]. Just as we propose in this paper, in these works liquids were exposed to weakly controlled conditions, letting them assume a-priori unknown free-form shapes. Only very recently have researchers succeeded in using such settings for multi-view reconstruction [YTK$^+$16]; to our knowledge, our work is the first to perform a full ray-space calibration from a single image taken through water drops.

Light fields. The research history on light fields, while significantly shorter, is nevertheless very rich and diverse [IWLH11]. In this section, we briefly review publications that are the most relevant to our work. They can serve as a starting point for a deeper exploration of the field.

The idea of capturing ray-space radiance measurements can be traced back to Lippmann [Lip08]. Yet, it was not until the computer age that light field data could be used to synthesize novel images [GGSC96, LH96], paving the way for a widespread adoption in the graphics and vision communities. Light fields are not only a mainstay of image-based rendering, but have also proven a valuable tool in a wide range of applications, including post-capture refocusing and parallax [Ng05, LNA$^+$06], depth estimation [KZP$^+$13, THMR13, WG14, WER16], as well as for advanced filtering purposes like glare removal [RAWV08].



(a) (b) (c)

Figure 2.2: Illustrations of the imaging pipeline and the underlying ray geometry at different stages. (a), Flow diagram of the reconstruction scheme, which combines a strong physical model (drop shape simulation) with computer vision elements such as image segmentation, feature detection and matching, and bundle adjustment. (b), Until the drop parameter is uniquely determined, each image location (primary ray) corresponds not to a single secondary ray but a fan of rays. (c), Secondary rays from different drops that have been identified to belong to the same scene-space feature (here illustrated by the red and green ray bundles) should intersect as closely as possible. We express this constraint in a cost function (Equation (2.2)) that sums up, for each feature f, the mutual line-line distances over all pairs of secondary rays belonging to that feature under the given drop volume parameters.

Much theoretical work has been done on light fields, most of it relating back to Adelson and Bergen's definition of the plenoptic function [AB91]. Milestones in light field analysis include the development of a sampling framework for image-based rendering by Chai et al. [CTCS00], Ng's Fourier slice theorem [Ng05] that identifies 2D images with 4D slices of the light field in Fourier domain and Wetzstein et al.'s theory [WIH13] that unifies the multiplexing of light fields with other plenoptic dimensions. Motivated by practical challenges in the construction of light field imagers, Wei et al. [WLM+15] proposed a unified sampling framework that takes into account lens aberrations and misalignment.

Since light fields in their most common definition are a four-dimensional representation of ray space, their capture poses numerous practical challenges as well. Among the setups proposed are robotic gantries [LH96], camera arrays [WJV+05], as well as multiplexing optics like lenslet arrays [Ng05, GZC+06], amplitude masks [VRA+07], elaborate mirror arrangements [MTK+11, TAV+10, FKR13], kaleidoscopes [HP03, MRK+13], random elements [FTF06, ANNW16] and even household items [WIG+15]. We note that calibrating an unknown integral imager's ray geometry is closely linked to capturing the geometry of reflective and transparent objects [IKL+08]. Here, most of literature deals with extensions to structured light scanning [TLGS05, HFI+08, WORK13].

Kutulakos and Steger investigated the conditions and constraints under which reflective and refractive geometry can be recovered [KS08]. In our approach, we constrain ourselves to optical surfaces that follow well-explored physical laws. We integrate this knowledge to estimate the shape of our refractive surface, and hence the geometry of viewing rays, using physical simulation.

Finally, on a higher level, we draw a great deal of inspiration from works on lightweight or free-hand capture techniques, recently culminating in Torralba and Freeman's explorative paper on accidental cameras [TF14]. From the first days of light field acquisition, researchers have aimed to avoid high-precision robotic and opto-mechanical designs, instead augmenting the available hardware by appropriate calibration steps [GGSC96, DLD12]. By replacing optical design with calibration, and calibration with simulation, our work continues in this tradition.

2.3 Experimental setup and procedure

In this section, we describe the experimental setup used to capture light fields through water drops.

Parts. Our camera was a Canon EOS 5D Mark II with the 24–105 mm $f/4$ kit lens set to a fixed 105 mm focal length and $f/22$ aperture. As substrate for our drops, we used 2 mm thick PMMA sheets. The liquid was tap water. Our model can account for slight changes in refractive index or surface energies by adjusting the drop volume parameter (see Section 2.4.1). Four diffused 50 W LED area lights served as the light source.

Setup. An illustration of our setup can be found in Figure 2.1a. Using a checkerboard target at various distances and the Camera Calibration Toolbox for MATLAB [Bou04], we calibrated the intrinsic camera parameters to obtain a pixel-to-ray mapping. The camera was then mounted on a tripod and faced down approximately vertically, which we confirmed by placing a small spirit level on the camera's rear display. The tripod mounting point was located approximately 100 cm above the floor. To obtain stationary drops (a requirement for simulation), we mounted the acrylic sheet horizontally at an approximate distance of 50 cm (measured with tape) below the camera's tripod mounting point, and focused the lens to its surface. The LED lights were mounted immediately underneath the window, facing downward onto the scene. Although our method works in ambient light, reflections in the drop surfaces had to be avoided since

they interfere with the drop segmentation and distort the measured light field. Our coordinate system is oriented such that the X and Y axes lie in the plane of the window, with the Z axis pointing toward the camera. The pixel-per-millimeter scale in the drop plane was obtained by combining the intrinsic camera calibration and the known distance of the substrate.

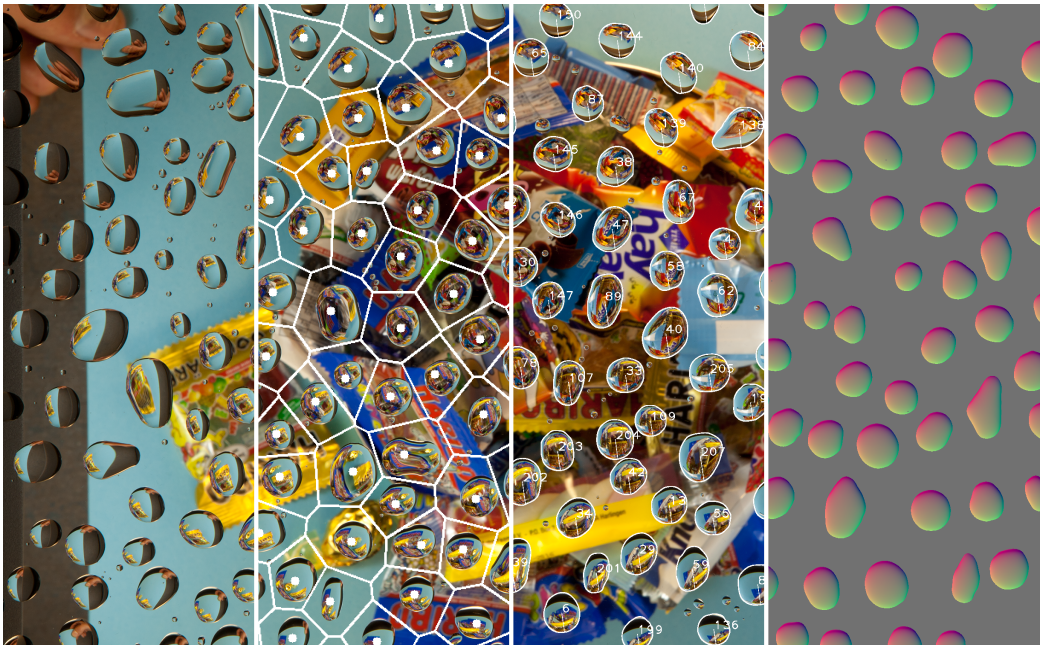
Capturing procedure. To capture a light field, we first arranged the scene and ensured that it was well lit. We then used a spray bottle to apply water drops to the acrylic surface. The drops typically take a few seconds to assume their final shape, a process that can be accelerated by gently tapping on the substrate. We triggered image exposure using a remote control. For the CarStunt scene, we used a microcontroller to simultaneously release four toy cars using a solenoid mechanism, and to time the camera exposure. The resulting raw images were converted to 16-bit PNG format using the Camera Raw importer in Adobe Photoshop CS5. Example input images can be seen in Figures 2.1b and 2.3a.

2.4 Reconstruction pipeline

The input to our reconstruction pipeline consists of a single image, like the one shown in Figure 2.1b, as well as a small number of additional parameters like camera projection, the distance of the window and the physical properties of the materials involved (density, refractive index, surface energy). The desired output is a dense mapping from pixels in the input image to light field rays, 3D drop surface reconstructions, as well as depth estimates and renderings of the scene from new virtual camera positions. To achieve this goal, we propose a reconstruction pipeline (Figure 2.2a) that consists of four major analysis and processing stages:

- extraction and simulation of water drops and ray geometries,
- extraction of scene features that serve as stereo constraints,
- a refinement step (bundle adjustment) to determine the volume parameter for each drop and establish the final pixel-to-ray mapping, and
- post-processing of the resulting light field (depth estimation and rendering).

Here, we motivate and explain these stages.

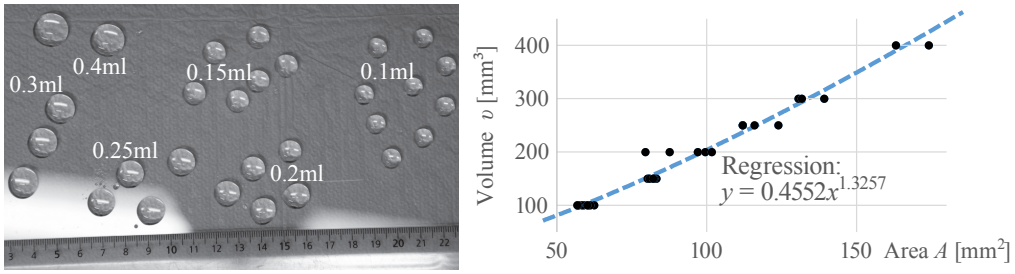


(a) (b) (c) (d)

Figure 2.3: Segmentation of drops in an image, and simulation of their shape. (a), Input image. (b), Result of semi-automatic circle detection visualized as Voronoi diagram. (c), Final drop contours. Both drop segmentation steps were corrected by additional manual input where needed. (d), Visualization of drop surfaces after simulation. Shown is the solution for the default drop volume parameter.

2.4.1 Drop extraction and simulation

Since the surface of a sessile drop is energy minimizing, for known physical parameters, the geometry is determined up to a single scalar parameter by the contact line (where drop surface and substrate meet) [AG97]. So the first step is to find this contour in the input image. Fully automatic segmentation of drops in images is an unsolved computer vision problem; existing approaches to image restoration [EKF13, SCK10] only produce drop contours as a by-product and are not accurate enough to serve as input for drop shape simulation. We approach this problem in a semi-automatic fashion. Since all drops are more or less round, we initialize a map of coarse drop locations with a circle detector (Figure 2.3b), drop centers serve as foreground constraints and their Voronoi diagram as background constraints. A state-of-the-art image segmentation algorithm [GRC$^+$10] is then used to determine accurate drop contours. To aid the automatic segmentation in ambiguous or otherwise challenging regions,



(a) (b)

Figure 2.4: Without the influence of gravity, area A and volume v of a spherical drop would relate as $v \sim A^{3/2}$. (a), In a pilot experiment, we placed drops of known volume on the substrate and measured their contact area. (b), Regression of a power function reveals that the actual exponent is slightly lower ($v \sim A^{1.33}$). We use this result to initialize the default drop volume.

the user can provide additional constraints by annotating additional drop and background regions. The result is a contour line for each drop (Figure 2.3c) which serves as input for a physical simulation that computes the drop geometries [IP06]. A detailed description of this simulation step can be found in Appendix 2.9.1. Although we experimentally established a rough relation between a drop's contact area and its expected *default volume* by using a small syringe to place drops of roughly known volume on an acrylic window and fitting a power function to the observations (Figure 2.4), the exact volume parameter is not yet known at this stage. For each drop, we simulate a bundle of surfaces that sample a range of values around the default parameter value. One such default solution is visualized in Figure 2.3d.

Under a geometric optics model, each pixel samples a *primary ray* entering the camera, which in turn corresponds to a *secondary ray* of light leaving the scene. Given the refractive geometry of glass pane and water drops, the relation between primary and secondary rays can now be computed via ray tracing. For each primary ray, we thus obtain a *fan* of secondary rays, one ray for each value of the (yet unknown) drop volume parameter (Figure 2.2b).

2.4.2 Feature extraction and matching

To further constrain the solution, we use SIFT [Low99] to extract keypoints from the image and identify scene features that are visible in multiple neighboring drops. The main challenge in this stage is that the drop views in the input image are strongly distorted, making scene features appear

Figure 2.5: Two examples of feature clusters found in different scenes, projected back into the original images. The total number of such clusters and the number of keypoints in each cluster depend on the visual complexity of the scene, as well as the drop arrangement.

quite differently in different views (Figure 2.1c). Prior to keypoint extraction, we therefore undistort the drop views using the default pixel-to-ray mapping from the previous stage. In particular, we perform a simple projection of each drop view to a plane located roughly at the distance of the scene. This effectively rectifies the view (Figure 2.1d), allowing SIFT to perform well despite the fact that the default drop volume estimate (used for undistortion) may not be the final one. The next step is to match keypoints found in neighboring views that correspond to the same scene feature. Using the algorithm from Appendix 2.9.2, we obtain a set of scene features that are visible in more than one drop, and for each of the features a set of keypoints in the input image that show the feature (the *feature cluster*, Figure 2.5). We define the matching matrix G to reflect the relation between scene-space features and image-space keypoints,

$$G(f,k) = \begin{cases} 1, & \text{if keypoint } k \text{ belongs to feature } f, \\ 0, & \text{else.} \end{cases} \tag{2.1}$$

In combination with the results from the previous stage, we further know the fan of secondary rays that belongs to each keypoint as a function of the drop volume parameter.

2.4.3 Geometry refinement

The features found in the previous stage now become the stereo constraints in our reconstruction: all secondary rays belonging to the *same feature* should intersect in the same point in space (Figure 2.2c). At the same time, the secondary rays belonging to features in the *same drop* are all controlled jointly by that drop's volume parameter. The purpose of this stage is to determine the vector of volume parameters $\mathbf{v} = (v_1, \ldots, v_m)$ (one parameter per drop) that produces the best global agreement between secondary rays. To this end, we define a cost function $F(\mathbf{v})$ that sums up, across all features f and all pairs of image keypoints (k_i, k_j) that represent a given feature in drops i and j, the line-line distance dist_{ray} between the corresponding secondary rays,

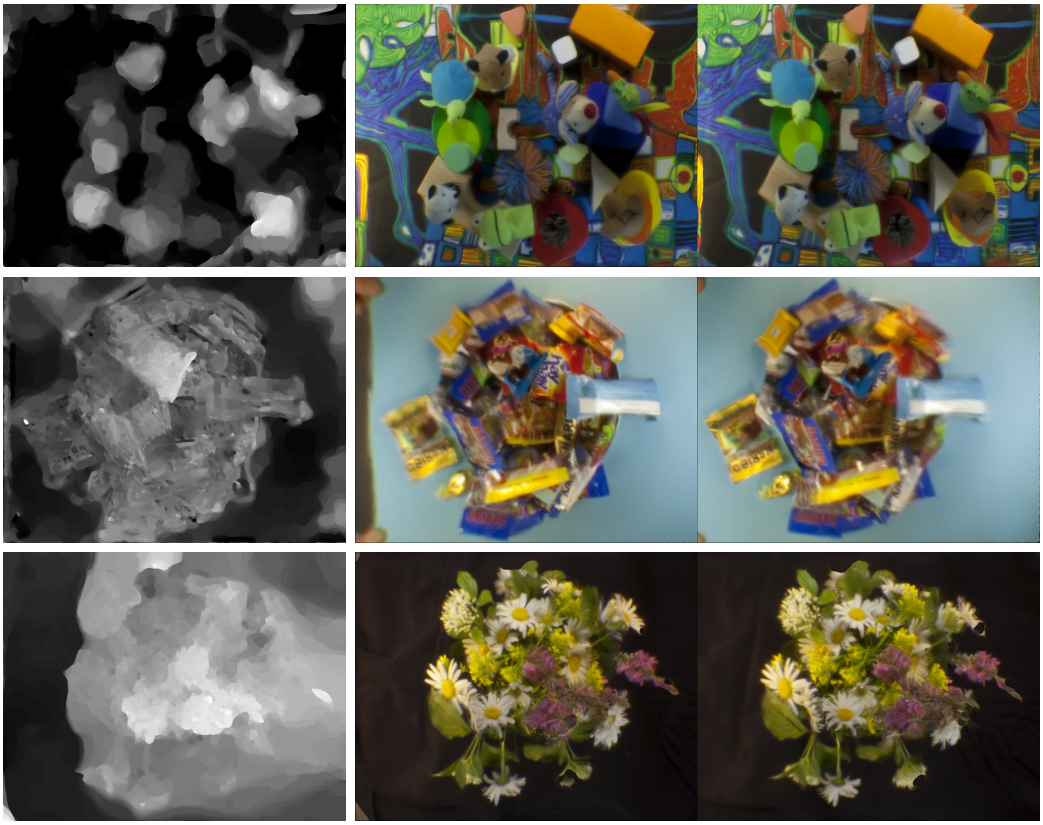
$$F(\mathbf{v}) = \sum_{f} \sum_{k_i \neq k_j} G(f, k_i)\, G(f, k_j)\, \text{dist}_{\text{ray}}^{(v_i, v_j)}(k_i, k_j). \tag{2.2}$$

This formulation is closely related to bundle adjustment, or the joint estimation of viewing parameters and scene geometry from multi-view stereo images [HZ04]. Rather than the usual reprojection error of features in image space, our cost function measures the distance between rays in scene space. To approach the high-dimensional non-linear problem of minimizing $F(\mathbf{v})$, we use an iterative coordinate descent scheme. We simultaneously perform line searches along all coordinate axes (volume parameters) and choose the solution with the lowest cost. This updating step is iterated until a local minimum of F is reached. To increase the chance of obtaining a good solution close to the global optimum, we restart the optimization process $n_{\text{iterations}} = 3$ times with perturbed solution vectors.

The outcome of the refinement stage is a vector of drop volumes $\mathbf{v}$ that is locally optimal under Equation (2.2). This results in a dense and uniquely defined mapping from input pixels to secondary rays, which concludes the geometric calibration of the light field. To validate the outcome, we also compute the root mean square (RMS) scene feature localization error. We obtain it from the pairwise line-line distances across all pairs of matched keypoints, a value that will increase when either drop or scene geometries are inconsistent.

2.4.4 Rendering

For the further assessment of the resulting light fields, we implemented a specialized renderer. Unlike light fields captured using properly designed



(a) (b)

Figure 2.6: Depth estimation and rendering for the "Animals" (top), "Candy" (middle) and "Flowers" (bottom) light fields. From a weighted focal stack (Figure 2.1d), we estimate a depth map (a) and use it to render all-in-focus images (b). The cross-eye stereogram shown here was obtained by performing all rendering steps twice under different camera settings. Animated versions of these results are available in the supplemental video.

optical systems, the ones reconstructed from liquid drops using the described method are irregularly and sparsely sampled. In addition, the estimated ray geometry is affected by residual inaccuracies.

To obtain high-quality 2D images from these liquid light fields, we use a rendering scheme that is guided by a per-pixel depth estimate. First, we set the parameters of a synthetic camera. For the desired viewpoint, we define a stack of planes of sufficient extent and resolution to fully contain the scene. By propagating all rays to the plane and integrating them there, a focused image is obtained from a light field [LH96]; all focused images together form a focal stack (Figure 2.1d). The sparsity of views necessitates

careful selection of rays and a specific weighting scheme. At any given location in a given plane, we retrieve a set of rays that intersect in this location. From these rays and the corresponding pixel values in the input image, we compute a weighted average color value, and the uncertainty as the weighted standard deviation of radiance samples. The underlying assumption is that if all samples have the same color, they probably originate from the same point in the scene. Hence, a low standard deviation indicates a likely depth value. We use this relation to extract a per-pixel depth assignment from the focal stack (Figure 2.6a).

As the final step, we follow the standard practice [WG14] of using the depth map to extract an all-in-focus image from the focal stack (Figure 2.6b). To render the scene under a different synthetic view, all steps including the focal stack computation are repeated. We provide implementation details and parameters in Appendix 2.9.3.

2.5 Results

To demonstrate our method, we acquired liquid light fields of six scenes, three of which are shown in Section 2.4.3. All input images as well as the recovered ray mappings are available as supplemental datasets to this paper. We further provide a collection of animated results in the supplemental video. All reconstructions rely exclusively on "wet" rays that passed through drops, except Figure 2.7 where some of the artifacts introduced by "dry" rays can be seen.

The colorful "Animals" scene consists of plush animals and wooden building blocks in front of a richly textured Hundertwasser pattern. All surfaces are of mostly Lambertian (diffuse) reflectance. After undistorting the drop views using the initial drop estimate, the algorithm produces a large number of plausible clusters that reach even into the peripheral parts of some drops (Figure 2.5), proving the good quality of the rectification step. After the light field calibration, the alignment of the drop views and the depth estimates are of sufficient quality (Figure 2.6a) to produce all-in-focus renderings that are rich in detail (Figure 2.6b) and convey a good depth impression. In the drop estimation step, the 3D localization errors for the sparse feature clusters are on the order of 4.5 mm and hence relatively high compared to the other datasets. We notice that features located around depth discontinuities tend to produce the highest errors. A possible explanation is that in regions with prominent occlusion effects, detected features may not correspond to real points in space and can therefore be stereo-inconsistent.

Figure 2.7: Rendering of the "Animals" data set using both "wet" and "dry" rays. The usage of "dry" rays increases the resolution (see e.g. the furry texture at the mouse's nose) but also introduces artifacts due to unsegmented drops and incomplete coverage.

Using the same scene, we also experimented with the usage of "dry" rays for rendering (Figure 2.7). We observed a noticeable increase in detail for projections close to the primary camera projection, but also heavy artifacts caused by the numerous unsegmented small drops and the "Swiss cheese" topology of the direct view. To our knowledge, there is no fully automatic, pixel-precise and robust segmentation method that would enable the use of "dry" rays in the geometry refinement step as well. Here, mislabeled pixels would not only produce visual artifacts but also add an uncontrollable error source to the drop volume estimation.

The "Candy" scene is an arrangement of different kinds of candy (chocolate bars, gummy bears, etc.) in small plastic packages. It exhibits strongly non-Lambertian reflectance, since many of the packages are made of of high-gloss material or even partly transparent. The scene has a relatively shallow depth range (7 cm) which, despite the challenging materials, allows the feature optimization to achieve sub-millimeter localization errors. As expected from the view-dependent nature of glossy and transparent materials, the reconstructed depth maps are not as smooth as in the other scenes. Still, the recovered depth estimates coarsely reflect the overall scene structure and are sufficient to produce output renderings of relatively high resolution (Section 2.4.3). In fact, the stereo pair conveys a

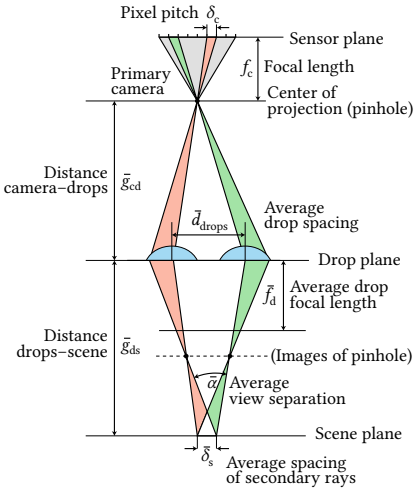
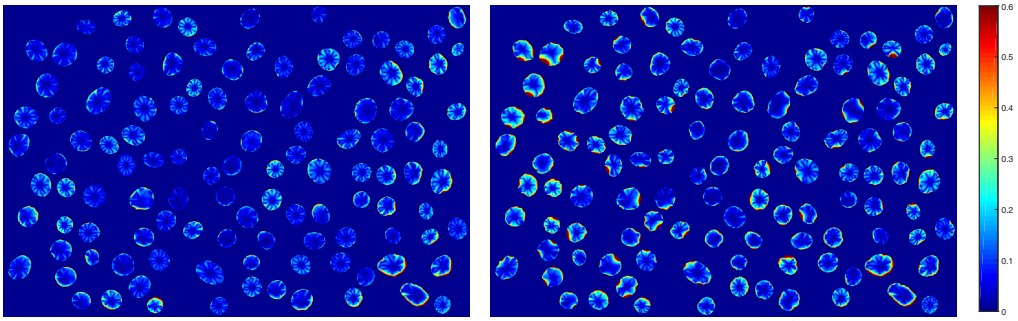


Figure 2.8: Description of geometric parameters used in Section 2.6.1.

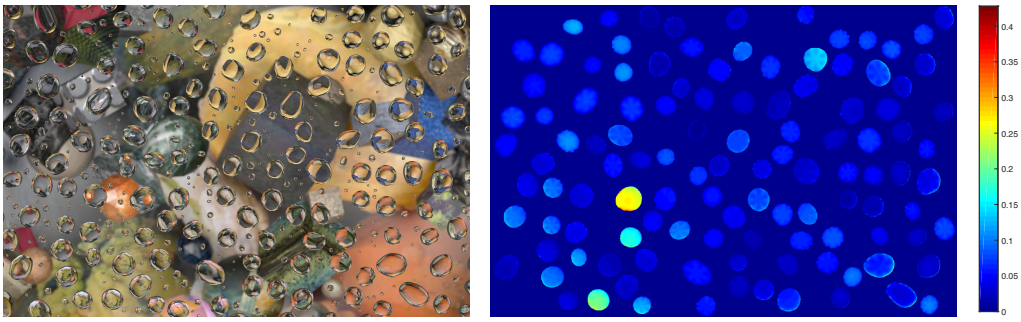
decent stereo impression of the scene, including view-dependent specular highlights. We note that in regions of constant color, small errors in the depth estimate may have little or no effect on the rendered outcome.

The "Flowers" scene consists of an arrangement of meadow flowers that are of mostly diffuse reflectance. The recovery of ray geometry works robustly, as evidenced by a small feature reconstruction error. Nevertheless, this light field proves to be extremely challenging to render: the recovered depth maps and, consequently, the renderings, contain numerous artifacts (Section 2.4.3). We identify several factors that may contribute to this problem. They include the total scene depth (measured with a ruler at 25 cm), the presence of repetitive features (daisy petals and small yellow flowers), and overall high spatial and angular frequencies which are not adequately sampled by the sparse and low-resolution drop views.



(a) Error of secondary rays [°], ground truth segmentation

(b) Secondary ray error [°], semiautomatic segmentation (Section 2.4.1)



(c) Synthetic input image, f/22

(d) Drop surface error [mm], ground truth segmentation

Figure 2.9: False color error plots for our light field calibration on a synthetic scene.

Scene	n_{drops}	# Clusters	RMS error	ET	$\bar{A}_{\text{drops}}$	$\bar{f}_{\text{drops}}$	n_{sec}	$\bar{\delta}_{\text{s}}$	$\bar{\alpha}$
Animals	126	1924	4.46 mm	250 ms	113.72 mm^2	89.49 mm	6 457 957	0.10 mm	3.03°
Candy	210	5454	0.79 mm	40 ms	85.01 mm^2	109.04 mm	5 064 711	0.10 mm	2.68°
Flowers	123	1868	1.31 mm	500 ms	112.50 mm^2	89.59 mm	6 236 003	0.10 mm	2.98°
CarStunts*	226	3424	2.66 mm	5 ms	84.09 mm^2	103.44 mm	5 389 975	0.10 mm	2.65°
Dwarfs*	143	2188	4.02 mm	250 ms	93.01 mm^2	84.17 mm	6 214 855	0.09 mm	2.72°
Firework*	205	489	1.33 mm	125 ms	85.39 mm^2	106.02 mm	5 036 900	0.10 mm	2.72°

Table 2.1: Our example scenes in numbers: count of drops n_{drops} used for reconstruction, number of feature clusters, RMS localization error of 3D features, exposure time, average drop footprint $\bar{A}_{\text{drops}}$, average drop focal length $\bar{f}_{\text{drops}}$, number of secondary rays n_{sec} in final light field, average spacing $\bar{\delta}_{\text{s}}$between secondary rays at a typical scene distance, average angle $\bar{\alpha}$ between drop views at scene depth (view separation). Results for the scenes marked with * are presented and discussed in the supplemental document.

2.6 System performance and quantitative evaluation

Spray-on optical systems are highly volatile and therefore hard to impossible to fully characterize "in the wild". Here, we list basic geometric relations for scattered arrangements of lens-like elements, and discuss the factors that affect the ray-optical system resolution under a pinhole model for the primary camera. We further use a synthetic replica of our experimental setup to measure the reconstruction accuracy of our pipeline under realistic conditions.

2.6.1 Resolution

Since light field imagers commonly trade spatial resolution against angular resolution, we used the following three measures to characterize our system: The average spacing between secondary rays when intersecting a plane at a typical scene depth ($\bar{g}_{\mathrm{ds}} = 300\,\mathrm{mm}$), the average angular separation $\bar{\alpha}$ between different drop views at that depth, and the total number n_{sec} of secondary rays. Assuming the drops to behave like thin lenses and taking into account the geometric parameters introduced in Figure 2.8, we can estimate the spatial resolution $\bar{\delta}_{\mathrm{s}}$ of a setup in the paraxial limit as

$$\bar{\delta}_s = \frac{\left(\bar{g}_{\mathrm{ds}} - \bar{f}_{\mathrm{d}}\right) \cdot \bar{g}_{\mathrm{cd}}}{\bar{f}_{\mathrm{d}} \cdot f_{\mathrm{c}}} \delta_{\mathrm{c}} \tag{2.3}$$

and its average view separation $\bar{\alpha}$ as

$$\bar{\alpha} = 2\tan^{-1}(\bar{d}_{\mathrm{drops}}/2\bar{g}_{\mathrm{ds}}). \tag{2.4}$$

Example values from our experimental datasets can be found in Table 2.1. Supposing uniformly distributed drops, the total number of secondary rays n_{sec} can be estimated as

$$n_{\mathrm{sec}} = n_{\mathrm{pr}} \frac{\bar{A}_{\mathrm{drops}} \cdot n_{\mathrm{drops}}}{A_{\mathrm{sensor}} \cdot \left(g_{\mathrm{cd}}/f_{\mathrm{c}}\right)^2}, \tag{2.5}$$

where, in addition to the symbols introduced in Figure 2.8, $\bar{A}_{\mathrm{drops}}$ is the average drop footprint (area), n_{drops} the total number of segmented drops, and A_{sensor} the sensor area.

2.6.2 Synthetic experiment

Since we are not aware of any solutions for 3D scanning water drop surfaces, we assessed the accuracy of our algorithm using a synthetic experiment. Using the Mitsuba renderer [Jak10], we modeled our imaging setup, procedurally generated and rendered a scene with random clutter under different aperture settings ($f/2$, $f/4$, $f/8$, $f/22$, pinhole), and extracted ground-truth primary and secondary ray geometries. An example rendering can be found in Figure 2.9c. The textures were randomly sampled from the Describable Textures Dataset [CMK+14], and for the 116 virtual water drops we re-used meshes from previous simulations, which fulfill the Young-Laplace equation and can therefore be assumed to be physically plausible under the given constraints.

We then performed a full ray-space calibration (starting with drop simulation) using our pipeline, and computed the RMS angular error in secondary rays and the RMS error in the intersection point between primary ray and drop. For both measures, the perfectly known "dry" rays were of course excluded. By randomly removing drops from the set, we varied the density of views fed into the bundle adjustment step. As the error plot in Figure 2.10 shows, the typical ray-space calibration error thus obtained was 0.1° to 0.2° with a typical RMS drop surface error of 0.06 mm. Notably, up to $f/8$ the calibration quality was mostly independent of the aperture and even across a wide range of drop numbers. The pipeline only started to break down when neighboring views stopped to share the same scene features due to the increased distance between them. Example error maps for the full set of drops are shown in Figures 2.9a and 2.9b. We observe that a few drops show significantly higher errors than the rest, which we attribute to mismatched keypoints.

These results were obtained using ground-truth segmentation of drop contours, also obtained from the renderer. To evaluate the influence of errors in the segmentation, we also performed the semi-automatic segmentation step as described in Section 2.4.1. For the full set of drops at $f/22$, this change increased the RMS angular error from 0.136° to 0.234°.

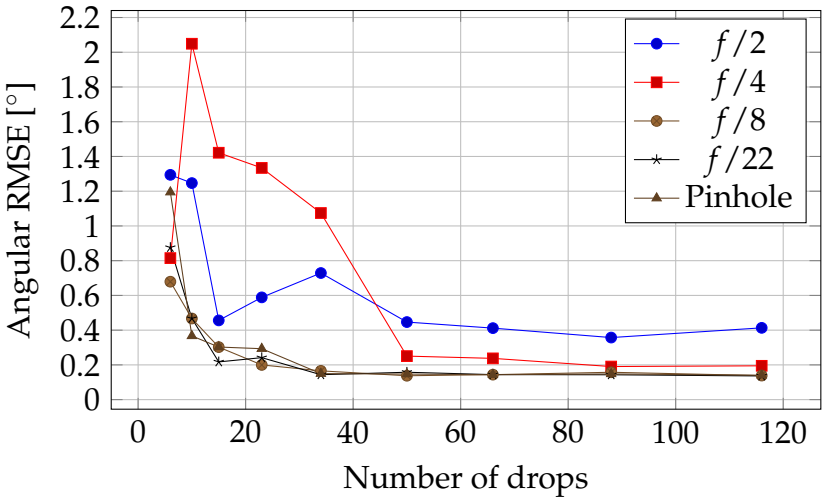


Figure 2.10: The RMS angular error for secondary rays, plotted as function of the number of drops used for the reconstruction. Only as about 80% of the drops are removed from the set, the error starts increasing significantly. For large apertures ($f/2$, $f/4$) this effect can be observed earlier.

2.7 Discussion and outlook

What is possible? We were able to show, to our knowledge for the first time, that capturing a light field through weakly controlled liquid optics, like water drops on a window, is an ambitious but realistic goal. From a single input image, our pipeline successfully recovers drop geometries, pixel-to-ray mappings and depth maps. Since water drops are minimal surfaces and hence smooth, the resulting image quality is at least comparable to what has been achieved using randomly structured reflective or refractive materials [ANNW16, FTF06, WIG+15], even though our approach does not rely on exhaustive calibration. The recovered drop geometries, while technically a by-product, are of high quality, so depending on one's viewpoint one might also interpret our method as a 3D scanner for water drops that exploits stereo cues from the surrounding light field.

What are the limiting factors? The main limitations of our method are the restriction to a horizontal plane and the need for manual interaction during the drop segmentation step. For non-trivial scenes, like CarStunts, using colored water can reduce the amount of manual intervention required. We captured our experimental data in a conservative, near-pinhole setting ($f/22$) to achieve good focus in the plane and in the scene. This

limitation is not exclusive to our method; in fact, a large part of light field research relies on synthetic or experimental reference data obtained under pinhole [HJKG16] or near-pinhole [KZP+13, V+08] settings. On the other hand, our evaluation on synthetic data suggests that the ray-space recovery and the drop surface estimation work reliably for much wider apertures as well. Finally, we note that rendering new views from sparsely and irregularly sampled light fields (especially with some residual ray-space uncertainty) remains a major challenge that even state-of-the-art techniques are still not quite up to. In fact, most image-based techniques do not generalize to our setting, so significant work will have to be done on depth estimation and filtering techniques to obtain the highest possible output quality under the given constraints.

What might become possible, and how? Water drops on slanted substrates constitute a dynamic phenomenon that is currently not covered by our model. The simulation of such scenarios is of great interest in various application fields (like architectural and automotive design) and the subject of ongoing research. It is therefore our hope that a solution could become possible in the not-too-distant future. While we demonstrate our high-level approach on the recovery of light fields, the quality obtained in a setting as uncontrolled as ours will obviously never rival that from a properly designed optical system. However, we can imagine many imaging situations under unfavorable conditions that could benefit from restoration techniques based on similar ideas. The recovered drop geometries can be of interest in materials science where the wetting behavior of liquids on surfaces is an important area of investigation. Estimating a handful additional material parameters (like the surface tensions, currently assumed to be known) seems like a plausible leap regarding the hundreds of degrees of freedom we are already recovering.

2.8 Conclusion

In this work, we set out to explore the challenge of capturing light fields through drops sitting on a clear window. To this end, we introduced a novel approach for establishing the ray geometries in this scenario, and crafted a reconstruction pipeline from it. Starting from a 2D input image, our algorithm segments drop outlines, simulates drop shapes, traces rays through the drops to undistort the image, and uses image features to refine the parameters. A key feature of our pipeline is its transparency, modularity and robustness regarding the choice of the individual components.

The resulting light fields typically contain 100 to 250 scattered views (one per drop), which can then be combined to render the scene from novel viewpoints.

Our research is motivated by a line of work that aims to replace carefully designed and highly specialized capture setups with a combination of casually captured data, careful calibration and computational reconstruction. By contributing a novel take on integral imaging, and by showcasing the use of physical simulation to regularize severely underconstrained imaging tasks, we hope that this paper will serve as a source of inspiration for future work.

2.9 Appendix

2.9.1 Drop shape analysis

The drop shape is approximated by a triangle mesh (we use $n_{\text{vertices}} = 12781$ vertices), which we initialize as a spherical cap that fulfills the given volume and the contact angle that follows from the material-specific wetting parameters under Young's Law [GBWQ04]. An iterative procedure then gradually transforms the initial circular contact line until the desired contact line L is obtained [IP06]. During this transition, the drop surface is gradually updated to fulfill the Young-Laplace equation while preserving the drop volume. The core numerical method employed is an iterative minimization procedure, first developed for homogeneous surfaces [Ili95] and then extended to treating heterogeneous surfaces and line tension effects [Ili97, IP03]. Equivalent tools are available in the public domain, for example *Surface Evolver* [Bra92, Bra13]. Physical constants used in the simulation are: $g = 9.81\,\mathrm{m/s^2}$ for the gravity acceleration, and the respective material values to model the wetting behavior of water on acrylic glass (the mass density $\rho_{\text{water}} = 1000\,\mathrm{kg/m^3}$ of the liquid and the surface tensions $\gamma_{\text{water}} = 72.8\,\mathrm{mN/m}$, $\gamma_{\text{PMMA}} = 41.0\,\mathrm{mN/m}$).

2.9.2 Feature clustering

Keypoints k_1 and k_2 that form a correspondence match should not only be visually similar but also geometrically plausible. We therefore define the distance measure

$$\begin{aligned}\operatorname{dist}^{(v_1,v_2)}(k_1,k_2) = \alpha \operatorname{dist}_{\text{ray}}^{(v_1,v_2)}(k_1,k_2) \\ + (1-\alpha)\operatorname{dist}_{\text{SIFT}}(k_1,k_2)\,,\end{aligned} \tag{2.6}$$

where $\text{dist}_{\text{ray}}^{(v_1,v_2)}(k_1,k_2)$ is the line-line distance between the two corresponding secondary rays predicted under the drop volume parameters v_1 and v_2, and $\text{dist}_{\text{SIFT}}(k_1,k_2)$ the Euclidean distance between SIFT feature vectors. To achieve compatibility, both distance functions are normalized to the interval $[0,1]$ by dividing by the maximum respective distance across all pairs of keypoints. The parameter $\alpha \in [0,1]$ controls the relative weighting of the two terms. We keep it constant at $\alpha = 0.2$.

Using this distance measure, we construct a sparse graph of feature correspondences by adding clusters of scene-space features. We start with the pair of keypoints that are closest to each other with a distance d_{min}, and proceed by adding keypoints from adjacent drops with a distance no greater than $\beta \cdot d_{\text{min}}$ to the existing ones. This procedure is iterated until every drop belongs to at least $n_{\text{clusters}} = 15$ clusters. In all our experiments, we set $\beta = 2$; keypoints that already belong to a cluster will no longer be considered in following iterations.

2.9.3 Rendering

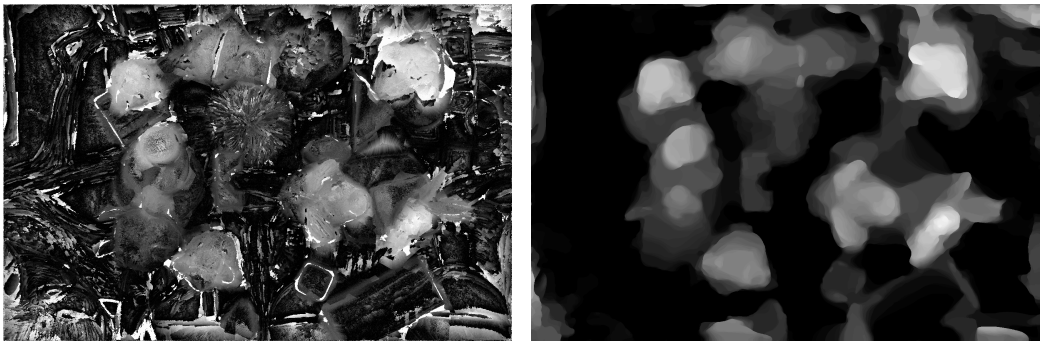
For depth estimation, we use a variant of a plane sweep algorithm in order to deal with the irregular set of rays. The depth map is viewpoint-dependent. For a given camera and target resolution, we initialize a range of 75–100 depth layers at discrete distances $z \in \{z_1,\ldots,z_N\}$ from the camera. For each depth layer z and pixel x, we compute the color vector $I_z^{\{r,g,b\}}(x)$ as a weighted average of the radiances L over the set of rays $R_{x,z}$, a subset of all rays intersecting the plane within the footprint of the pixel,

$$I_z^{\{r,g,b\}}(x) = \frac{1}{\sum w_{\mathbf{r}}} \sum_{\mathbf{r}\in R_{x,z}} w_{\mathbf{r}} L^{\{r,g,b\}}(\mathbf{r})\,. \tag{2.7}$$

For the set $R_{x,z}$ we choose the five intersecting rays that have the smallest angular distance $\alpha_{\mathbf{r}}$ to the query ray that belongs to pixel x in the virtual camera, i.e., that are most representative for the desired synthetic view. The weights $w_{\mathbf{r}}$ are given by

$$w_{\mathbf{r}} = 1 - \frac{\alpha_{\mathbf{r}}}{\max_{\mathbf{q}\in R_{x,z}}(\alpha_{\mathbf{q}})}. \tag{2.8}$$

An example of such a weighted focal stack $I_z(x)$ is shown in Figure 2.1d. We use it to compute the cost $\rho(x,z)$ for assigning depth z to x using the



(a) Naïve (unregularized) depth map (b) TV-regularized depth map

Figure 2.11: Effect of regularization on the depth estimate.

root-mean-square-deviation

$$\rho(x,z) = \frac{1}{3} \sum_{c \in \{r,g,b\}} \left(\sum_{\mathbf{r} \in R_{x,z}} \frac{\left(L^c(\mathbf{r}) - I_z^c(x)\right)^2}{|R_{x,z}|} \right)^{1/2} \tag{2.9}$$

over the radiances $L(\mathbf{r})$. Minimizing this cost for each pixel independently results in a noisy depth estimate with significant errors around depth discontinuities (Figure 2.11). Therefore, we formulate the cost of the full depth map d on the image plane Ω as

$$E(d) = \int_\Omega \|\nabla d(x)\| + \lambda \rho\left(x, d(x)\right) \mathrm{d}x. \tag{2.10}$$

The total variation (TV) penalty of the gradient of the depth map encourages piecewise smooth solutions and can be optimized using the technique of functional lifting [PCBC10]. We use the implementation provided by `cocolib` [GSC12]. Given the depth map d, we obtain the all-in-focus image I_{all} from the chosen view point by extracting the color from the layer corresponding to the correct depth label, i.e. setting $I_{\text{all}}(x) = I_{d(x)}(x)$.

This chapter features our non-linear, non-convex optimization method for non-line-of-sight geometry reconstruction using transient imaging. Building on our findings from Chapter 2, we have designed the algorithm in an analysis-by-synthesis scheme around our novel, physically-based forward model for three-bounce diffuse light transport.

This chapter was published as [IH18]: Julian Iseringhausen, Matthias B. Hullin: "Non-Line-of-Sight Reconstruction using Efficient Transient Rendering". *arXiv:1809:08044 [cs.GR]*, September 2018. Here we present a revised version, as accepted to ACM Transactions on Graphics.

CHAPTER 3

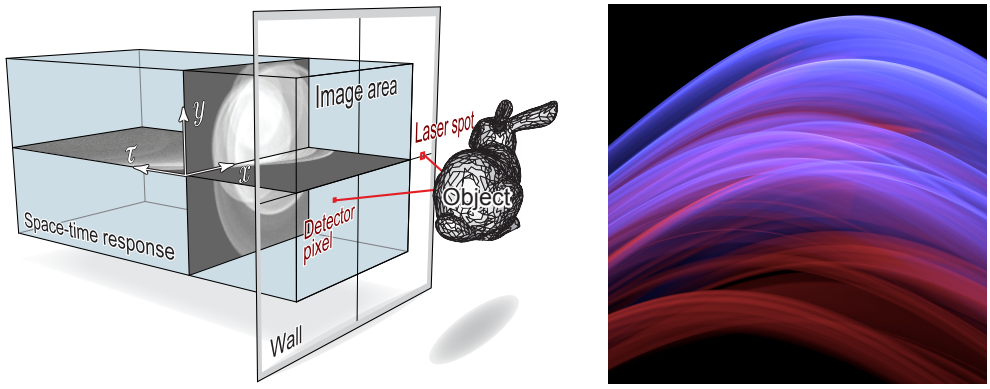
Non-Line-of-Sight Reconstruction using Efficient Transient Imaging

Abstract Being able to see beyond the direct line of sight is an intriguing prospective and could benefit a wide variety of important applications. Recent work has demonstrated that time-resolved measurements of indirect diffuse light contain valuable information for reconstructing shape and reflectance properties of objects located around a corner. In this paper, we introduce a novel reconstruction scheme that, by design, produces solutions that are consistent with state-of-the-art physically-based rendering. Our method combines an efficient forward model (a custom renderer for time-resolved three-bounce indirect light transport) with an optimization framework to reconstruct object geometry in an analysis-by-synthesis sense. We evaluate our algorithm on a variety of synthetic and experimental input data, and show that it gracefully handles uncooperative scenes with high levels of noise or non-diffuse material reflectance.

3.1 Motivation

Every imaging modality from ultrasound to x-ray knows situations where the target is partially or entirely occluded by other objects and therefore cannot be directly observed. In a recent strand of work, researchers have aimed to overcome this limitation, developing a variety of approaches to extend the line of sight of imaging systems, for instance using wave optics [KHFG14, BLK18] or by using the occluder itself as an accidental imager [BYY+17]. Among all the techniques proposed, a class of methods has received particular attention within the computer vision and imaging communities. The main source of information for these methods are indirect reflections of light within the scene, represented by time-resolved impulse responses. From such responses, it has been shown that the presence and position of objects "around a corner" [KHDR09], or even their shape [VWG+12] and/or reflectance [NZV+11] can be reconstructed. In this paper, we focus on the archetypal challenge of reconstructing the shape of an unknown object from 3-bounce indirect and (more or less) diffuse reflections off a planar wall (Figure 3.1) [KHDR09]. The overwhelming majority of approaches to this class of problem rely on ellipsoidal *backprojection*, where intensity measurements are smeared out over the loci in space (ellipsoidal shells) that correspond to plausible scattering locations under the given geometric constraints [VWG+12, BZT+15, GTH+16, KZSR16, AGJ17]. Ellipsoidal backprojection implicitly assumes that the object is a volumetric scatterer, and it does not take into account surface orientation and self-occlusion of the object. More importantly, unlike linear backprojection used in standard emission or absorption tomography, ellipsoidal backprojection is not the adjoint of a physically plausible forward light transport operator. Where such operators have been identified [LKB+18], they are typically constrained to rudimentary volumetric, non-opaque, isotropic scattering models. This necessitates heavy heuristic filtering, and the reconstructed shapes are typically flat and low in detail. On the other hand, algorithms based on ellipsoidal backprojection generally have much shorter runtimes than our approach, since they do not require a global optimization scheme.

Here, we propose an alternative approach that mitigates some of the problems of backprojection by formulating the non-line-of-sight sensing problem in an analysis-by-synthesis sense. In other words, we develop a physically plausible and efficient forward simulation of light transport (transient renderer) and combine it with a nonlinear optimizer to determine the scene hypothesis that best agrees with the observed data. The



(a) (b)

Figure 3.1: (a) The challenge of looking around the corner deals with the recovery of information about objects beyond the direct line of sight. In this illustration of a setting proposed by Velten et al. [VWG+12], an unknown object is located in front of a wall, but additional obstacles occlude the object from any optical devices like light sources or cameras. Our only source of information are therefore indirect reflections off other surfaces (here, a planar "wall"). A point on the wall that is illuminated by an ultrashort laser pulse turns into an omnidirectional source of indirect light ("laser spot"). After scattering off the unknown object, some of that light arrives back at the wall, where it forms an optical "echo" or space-time response (shown are 2D slices) that can be picked up by a suitable camera. Locations on the wall can be interpreted as omnidirectional detector pixels that receive different mixtures of backscattered light contributions at different times. We assume that neither camera nor laser can directly illuminate or observe the object, leaving us with the indirect optical space-time response as the only source of information. Note that for the sake of clarity, laser source, camera, and occluder are not shown here. The complete setup is illustrated in Figure 3.3. (b) We propose a novel transient renderer to simulate such indirectly scattered light transport efficiently enough for use as a forward model in inverse problems. In this artistic visualization, light contributions removed by the shadow test are marked in red, and the net intensity in blue. Together with an optimization algorithm, the renderer can be used to reconstruct the geometry of objects outside the line of sight.

method is enabled by a number of technical innovations, which we consider the key contributions of this work:

- a scene representation based on level sets and a surface-oriented scattering model for time-resolved light transport around a corner (wall to object to wall) based on time-resolved radiative transfer,
- an extremely efficient GPU-based custom renderer for three-bounce

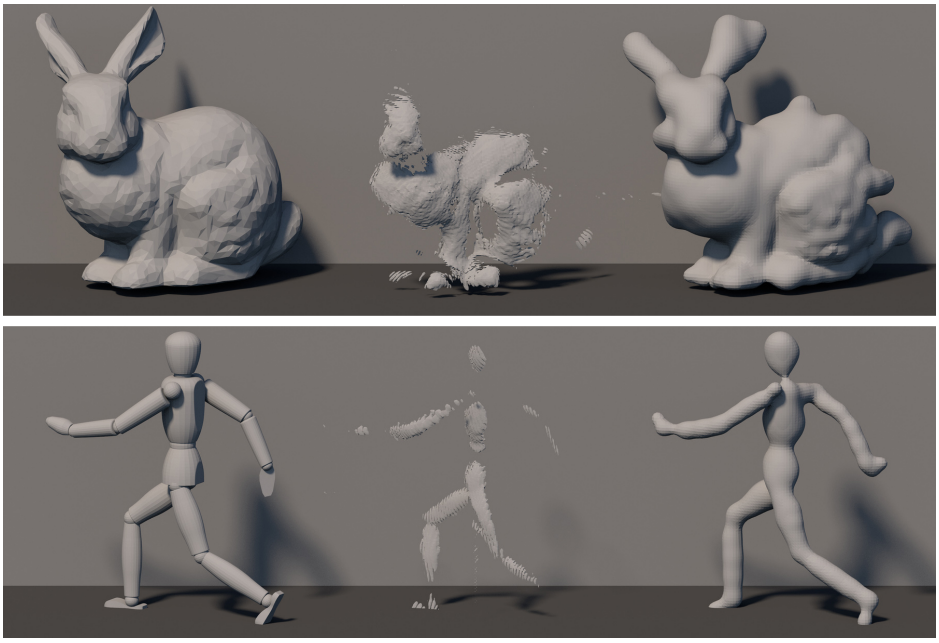


Figure 3.2: Left to right: ground-truth object geometry; reconstruction using a state-of-the-art method (ellipsoidal backprojection); reconstruction using the technique presented in this paper. Top row: `BunnyGI` dataset; bottom row: `Mannequin1Laser` dataset. Our method relies on highly efficient and near-physical forward simulation, and it exemplifies the use of computer graphics as a technical tool to solve inverse problems in other fields.

backscatter that features near-physical handling of occlusion effects and a novel temporal filtering scheme for triangular surfaces, and

- a global, self-refining optimization strategy to minimize the reconstruction error.

We evaluate our method on a number of synthetic and experimental datasets and find that it is capable of achieving significantly higher object coverage and detail than ellipsoidal backprojection, even on greatly reduced and degraded input data. Our renderer not only naturally accommodates surface BRDFs, but is also open to extensions like higher-order light bounces or advanced background models that will be needed in order to tackle future non-line-of-sight sensing problems. The method, as proposed here, is not capable of delivering high reconstruction rates in this first implementation. However, we believe that being able to generate transient renderings for the around-the-corner setting very efficiently will enable novel approaches to the problem, for instance based on machine learning.

3.2 Related work

The research areas of transient imaging and non-line-of-sight reconstruction have recently received tremendous attention from the computer vision, graphics, imaging and optics communities. For a structured overview on the state of the art, we refer the interested reader to a recent survey [JMMG17].

3.2.1 Transient imaging

Imaging light itself as it propagates through space and time poses the ultimate challenge to any imaging system. To obtain an idea of the frame rate required, consider that in vacuum, light only takes about 3 picoseconds ($3 \cdot 10^{-12}\,s$) per millimeter of distance traversed. The typical transient imaging system consists of an ultrashort (typically, sub-picosecond) light source and an ultrafast detector. Oddly, three of the highest-performing detection technologies are over 40 years old: streak tubes [VRB11] wherein a single image scanline is "smeared out" over time on a phosphor screen; holography using ultrashort pulses [Abr78], and gated image intensifiers [LV14]. More common nowadays, however, are semiconductor devices that achieve comparable temporal resolution without the need for extreme light intensities or voltages. Among the technologies reported in literature are regular reverse-biased photodiodes [KHDR09], as well as time-correlated single-photon counters which conveniently map to standard CMOS technology [GKH$^+$15]. On the low end, it has also been shown that transient images can be computationally reconstructed from multi-frequency correlation time-of-flight measurements [HHGH13], although data thus obtained typically suffers from the low temporal bandwidth of these devices, which necessitates heavy regularization.

3.2.2 Transient rendering

The simulation of transient light transport, when done naïvely, is no different from regular physically-based rendering, except that for each light path that contributes to the image, its optical length must be calculated and its contribution stored in a time-of-flight histogram [SSD08]. A number of offline transient renderers have been made available to the public [SC14, JMM$^+$14]. Even with advanced temporal sampling [JMM$^+$14] and efficiency-increasing filtering strategies such as photon beams [MJGJ17], such renderers still take on the order of hours to days to produce converged results. In contrast, the special-purpose renderer introduced in

this paper is capable of producing close-to-physical renderings of around-the-corner settings in a matter of milliseconds. Finally, there have been efforts to simulate the particular characteristics of single-photon counters [HGJ17], an emerging type of sensor that can be expected to assume a major role in transient imaging.

3.2.3 Analysis of transient light transport and looking around corners

The information carried by transient images has been the subject of several investigations. Wu et al. laid out the geometry of space-time streak images for lensless imaging [WWB$^+$12], and discussed the influence of light transport phenomena such as subsurface scattering on the shape of the temporal response [WVO$^+$14]. Economically, the most important use of transient light transport analysis today is likely in multi-path backscatter removal for correlation-based time-of-flight ranging [Fuc10, and many others].

In this paper, we direct our main attention to the idea of exploiting time-resolved measurements of indirect reflections for the purpose of extending the direct line of sight and, in effect, looking around corners [KHDR09, VWG$^+$12]. While a variety of geometric settings have been investigated, the bulk of work in this area relies on the arrangement illustrated in Figures 3.1 and 3.3 and further introduced in the following Section 3.3.

The reconstruction strategies can be roughly grouped in two classes. One major group is formed by backprojection approaches where each input measurement casts votes on those locations in the scene where the light could have been scattered [VWG$^+$12, LV14, BZT$^+$15, GTH$^+$16, KZSR16, AGJ17]. A smaller but more diverse group of work relies on the use of forward models to arrive at a scene hypothesis that best agrees with the measured data. Here, reported approaches fall into several categories. A combinatorial labeling scheme was developed by Kirmani et al. [KHDR09]. If the capture geometry is sufficiently constrained, frequency-domain inverse filtering [OLW18a] can be employed. Variational methods using simple linearized light transport tensors [NZV$^+$11, HXHH14] and simplistic models based on radiative transfer [KPM$^+$16, PBT$^+$17] are (in principle) capable of expressing opacity effects like shadowing and occlusion, and physically plausible shading. These approaches are closest to our proposed method. In concurrent work, Heide et al. [HOZ$^+$17] added such extra factors as additional weights into their least-squares data term,

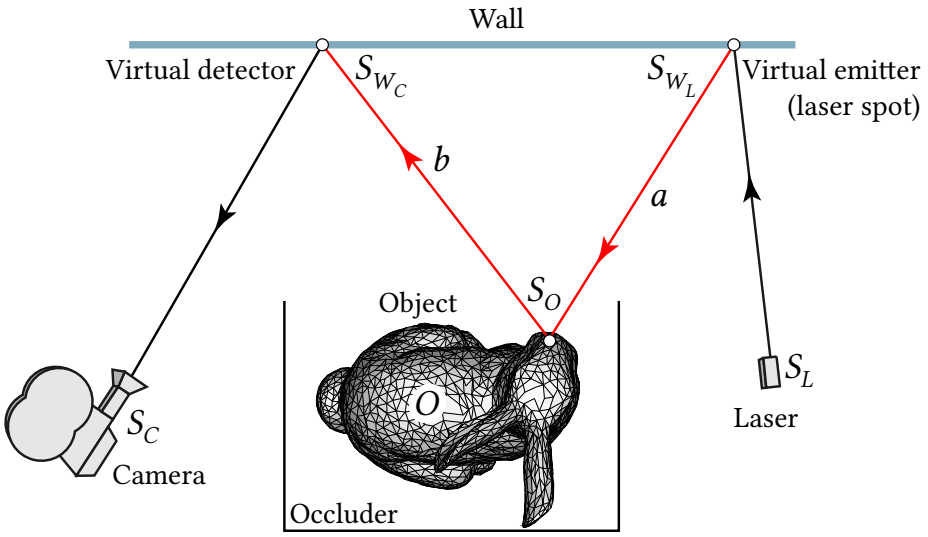


Figure 3.3: Schematic top view of the scene arrangement, where the unknown object is occluded from direct observation. We assume that the temporal response has been "unwarped" (e.g., [KZSR16]), so only the occluded segments a and b contribute to the total time of flight and to the shading in Equation (3.4).

achieving non-line-of-sight reconstructions of significantly improved robustness. Thrampoulidis et al. [TSX+17] applied a similar idea on the reconstruction of 2D albedo maps on known geometry that are further obscured by known occluders between object and wall. For homogeneous volumetric media in direct sight, Gkioulekas et al. [GZB+13] extensively relied on physically-based rendering to recover their scattering parameters and phase function. With the proposed method, we demonstrate what we believe is the first reconstruction scheme for non-line-of-sight object geometry that is based on a near-physical yet extremely efficient special-purpose renderer and, by design, produces solutions that are self-consistent. We believe that our work can serve as an example for other uses of computer graphics methodology as a technical tool for solving inverse problems in imaging and vision.

3.3 Problem statement

Here we introduce the geometry of the non-line-of-sight reconstruction problem as used in the remainder of the paper. For simplicity, we neglect the constant factor c (the speed of light) connecting *time* and *(optical) path length*. Thus, time and distance can be used synonymously and all discus-

sions become independent of the absolute scale.

3.3.1 Problem geometry and transient images

We model our setting after the most common scenario from literature (Figure 3.3), where the unknown object is observed indirectly by illuminating a wall with a laser beam and measuring light reflected back to the wall. Following Kadambi et al. [KZSR16], the laser spot on the wall acts as an area light source, and observed locations on the wall are equivalent to omnidirectional detectors that produce an "unwarped" transient image [VWJ+13] (Figure 3.1). The extent of the observed wall, the size of the object and its distance to the wall are usually on the same order of magnitude. The *transient image* or *space-time response* $\mathbf{I} \in \mathbb{R}^{n_x \times n_\tau}$ is the entirety of measurements taken using this setup, n_x being the number of combinations of detector pixels and illuminated spots and n_τ the number of bins in a time-of-flight histogram recorded per location. For a two-dimensional array of observed locations (for instance, when using a time-gated imager), the space-time response can be interpreted as a three-dimensional data cube similar to a video.

3.3.2 Problem formulation

The idea underlying ellipsoidal backprojection is that any entry in the transient image, or the response of a pair of emitter and detector positions for a given travel time, corresponds to an ellipsoidal locus of possible candidate scattering locations. If no further information is available, any measured quantity of light therefore "votes" for all locations on its ellipsoid. Finally, the sum or product of all such votes is interpreted as occupancy measure, or probability of there being an object at any point in space. We refer to a recent study [LKB+18] that discusses the design options for such algorithms in great detail.

In contrast, we formulate the reconstruction task as a non-linear least-squares minimization problem

$$\min_{\mathbf{P}} \| \mathbf{I}_{\text{ref}} - I\left(G\left(\mathbf{P}\right)\right) \|_2^2, \tag{3.1}$$

where $\mathbf{P}$ is a parameter vector describing the scene geometry, $G(\cdot)$ is a function that generates explicit scene geometry (a triangle mesh), $\mathbf{I}_{\text{ref}}$ is the measured space-time scene response, and $I(\cdot)$ is a forward model (renderer) that predicts the response under the scene hypothesis passed as

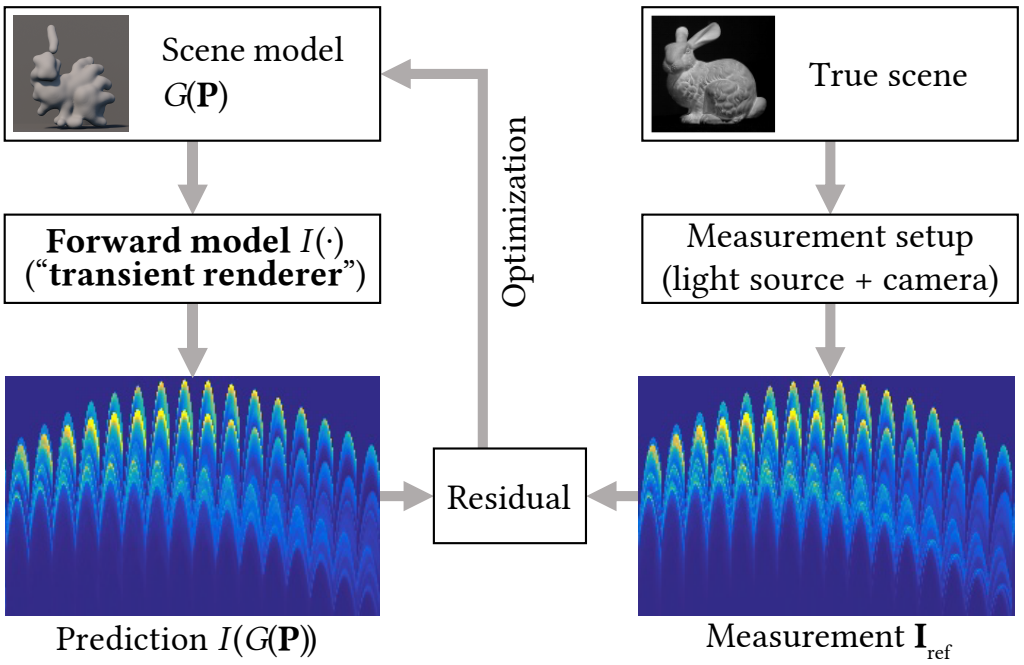


Figure 3.4: Overview of our analysis-by-synthesis scheme for looking around a corner. Our pipeline heavily relies on custom-made components (scene representation, renderer, residual function, optimizer) to make this approach viable.

argument. The purpose of the optimization is to find the scene geometry $G(\mathbf{P})$ that minimizes the sum of squared pixel differences between the predicted and the observed responses. Figure 3.4 illustrates this principle.

A key feature of this formulation is that the solution by its very definition is optimally consistent with the chosen physical model of light transport, and that ongoing improvements in forward modeling will also benefit the reconstruction. Furthermore, our approach naturally handles opaque, oriented surfaces, whereas in backprojection, surface geometry is implicitly defined and needs to be derived using additional filtering steps . Furthermore, our method is able to handle arbitrary surface BRDFs, where current backprojection methods implicitly assume diffuse cloud-like scattering [LKB+18]. A downside of our approach is that it requires a full model of the scene, and that any unknowns (such as background or noise) can distort the solution in ways that are hard to predict. On the other hand, we believe that our approach lends itself for future extensions like higher-order light bounces.

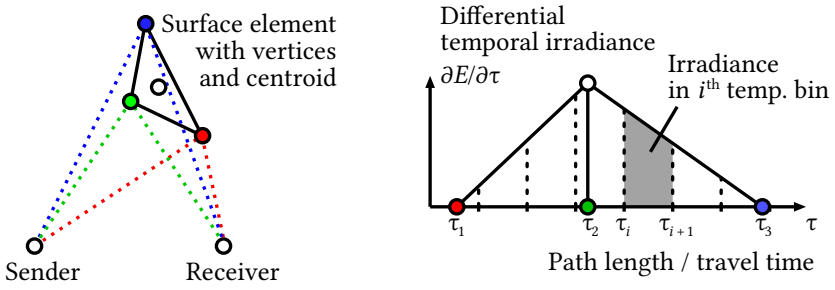


Figure 3.5: To compute the total irradiance α_t contributed by a surface triangle to a given detector pixel, we evaluate the radiative transfer using the element's centroid. We then use a first-order filter to distribute this irradiance over the temporal bins that are affected by the triangle. To this end, we compute the three optical path lengths, or travel times, $\tau_{1\ldots3}$ belonging to the triangle's three vertices. The irradiance ending up in any temporal bin is then obtained by constructing a triangular function of total area α_t using the three arrival times as illustrated, then geometrically integrating over the time interval that corresponds to the bin. The true temporal distribution depends on the position and orientation of the triangle. However, the effectiveness of the temporal filter can be seen in Table 3.1 and Figures 3.6 and 3.9.

3.4 Method

In the following, we introduce the components of our reconstruction algorithm in detail.

3.4.1 Geometry representation

We seek to parameterize the scene geometry in terms of a vector $\mathbf{P}$ that has a small number of degrees of freedom to make the optimization in Equation (3.1) tractable. Rather than using $\mathbf{P}$ to directly store a mesh representation with vertices and faces, we express the geometry implicitly as an isosurface of a scalar field $M_{\mathbf{P}}(\mathbf{x})$ composed of globally supported basis functions. This approach is also common in surface reconstruction from point clouds [CBC+01]. In our case, the vector $\mathbf{P}$,

$$\begin{aligned}\mathbf{P} &= (\mathbf{p}_1, \ldots, \mathbf{p}_m) \\ &= ((\mathbf{x}_1, \sigma_1), \ldots, (\mathbf{x}_m, \sigma_m)),\end{aligned} \tag{3.2}$$

lists the centers $\mathbf{x}_i$ and standard deviations σ_i of m isotropic Gaussian blobs. From the scalar field

$$M_{\mathbf{P}}(\mathbf{x}) = \sum_{i=1}^{m} e^{-\|\mathbf{x}-\mathbf{x}_i\|_2^2/(2\sigma_i^2)} \tag{3.3}$$

we extract the triangle mesh $G(\mathbf{P})$ using a GPU implementation of Marching Cubes [LC87]. For all our reconstructions, we used a fixed resolution of 128^3 voxels for the reconstruction volume, and a fixed threshold of $\frac{3}{4}$ for the isosurface. The extension to other implicit functions, such as anisotropic Gaussians or general radial basis functions, is trivial.

3.4.2 Rendering (synthesis)

We propose a custom renderer that is suitable for use as forward model $I(\cdot)$ inside the objective function, Equation (3.1). In order to be suited for this purpose, the renderer must be sufficiently close to physical reality. At the same time, it has to be very efficient because hundreds of thousands of renderings may be required over the course of the optimization run. We achieve this efficiency by restricting the renderer to a single type of light path and rendering only light bounces from the wall to the object and back to the wall. Following the notation of [PH10] and by dropping any constant terms, we can write the incoming radiance for each camera pixel as

$$L = \int_O f\left(S_{W_L} \to S_O \to S_{W_C}\right) \eta\left(S_O \leftrightarrow S_{W_C}\right) \eta\left(S_{W_L} \leftrightarrow S_O\right) \mathrm{d}S_O, \tag{3.4}$$

where $O = G(\mathbf{P})$ denotes the object, f the object's BRDF and $S_{\_}$ surface points as shown in Figure 3.3. The geometric coupling term η is defined as

$$\eta\left(S_1 \leftrightarrow S_2\right) = V\left(S_1 \leftrightarrow S_2\right) \frac{|\cos(\theta_1)|\,|\cos(\theta_2)|}{\|S_1 - S_2\|_2^2}, \tag{3.5}$$

with V being the binary visibility function and θ_i the angle of the ray connecting S_1 and S_2 to the respective surface normal. Since our object is already represented as a triangle mesh, we are able to approximate Equation (3.4) by assuming a constant radiance over each triangles' surface,

$$\begin{aligned} L &\approx \sum_{t \in T} f\left(S_{W_L} \to S_t \to S_{W_C}\right) \eta\left(S_t \leftrightarrow S_{W_C}\right) \eta\left(S_{W_L} \leftrightarrow S_t\right) A_t \\ &=: \sum_{t \in T} \alpha_t. \end{aligned} \tag{3.6}$$

Here, T is the set of all triangles of our object, P_t is the centroid, and A_t the area. We denote the total irradiance contributed by triangle t as α_t. In our experiments, we use Lambertian and metal BRDFs, but other reflectance functions can be used as well. This approximation can be seen as an extension of the one found in [KPM+16]. We further add two important features to increase physical realism and generate a smooth transient image.

Our first addition are visibility tests (V) for both segments of the light path, which is necessary for handling non-convex objects. We first connect the laser point and the triangle centroid by a straight line, and test whether this segment intersects with any of the other triangles of the object mesh. For all visible triangles for which no intersection is found, we test the visibility of the second path segment (return of scattered light to the wall) in the same way. This shadow test avoids overestimation of backscatter from self-occluding object surfaces. We note, however, that our way of performing the test only for the triangle centroid leads to a binary decision (triangle entirely visible or entirely shadowed) and therefore potentially makes the objective non-continuous. This can be reduced by using a triangle grid of sufficiently high resolution.

To render a transient image, we extend the pixels of the steady-state renderer to record time-of-flight histograms. The light contribution α_t enters into this histogram according to the geometric length of the corresponding light path; this length is simply the sum of the two Euclidean distances from laser point to point on triangle and back to the receiving point on the wall (see Figure 3.3). We found that the temporal response is prone to artifacts if only the centroid of the triangle is taken into account for the path length. Instead, we use the path lengths for the triangle's three corner vertices to determine the temporal footprint of the surface element. Using a linear filter, we then distribute the contribution α_t over the temporal domain (Figure 3.5). This procedure ensures that the rendered outcome is smooth in the temporal and spatial domains even when a single surface element covers dozens of temporal bins (Figure 3.6).

3.4.3 Optimization (analysis)

The optimization problem in Equation (3.1) is non-convex and non-linear, so special care has to be taken to find a solution (a set of blobs) that, when rendered, minimizes the cost function globally. While it would be desirable to optimize over the whole parameter vector $\mathbf{P}$ simultaneously, this is computationally prohibitive. To address this problem, we developed the iterative optimization scheme summarized in Algorithm 1, with subroutines provided in Algorithms 2 and 3. Figure 3.7 shows several intermedi-

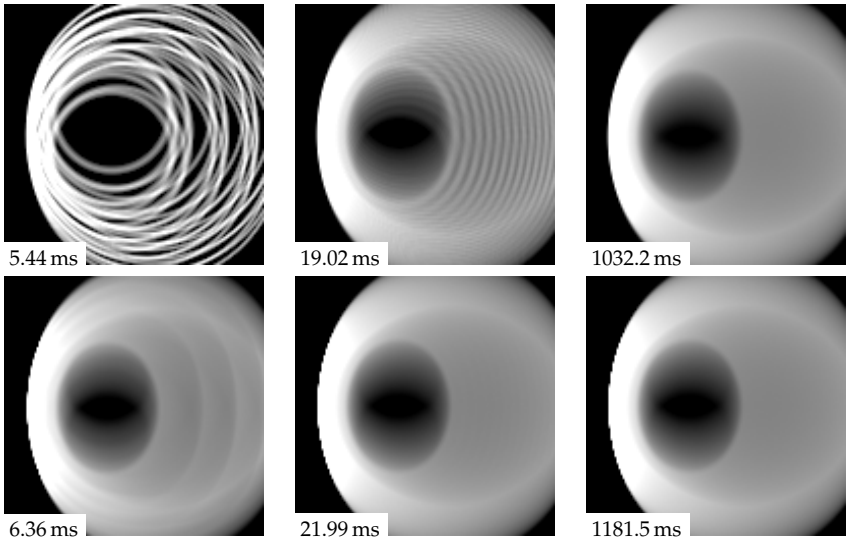


Figure 3.6: The temporal filter also results in overall smoother spatial slices of the space-time response. Here we verify the performance of the filter by rendering the response generated by a planar square using different levels of detail. Shown is a single time slice without (top row) and with temporal filtering (bottom row). From left to right: coarse tesselation (4×4 quads), medium tesselation (16×16 quads), fine tesselation (128×128 quads). Numbers indicate the rendering time for the entire transient data cube (128×128 pixels, 192 time bins) on an NVIDIA GTX 980. Note the significant quality improvement at only 14–17% increased computational cost.

ate results during execution of the optimization scheme.

The heart of our optimization algorithm is the inner optimization loop ITERATE($\mathbf{p}$, $\mathbf{P}$), which determines the $k = 10$ nearest neighbors of a given pivot blob $\mathbf{p}$ using the routine FIND_NEIGHBORS($\mathbf{p}$, $\mathbf{P}$). It then optimizes the *positions* of those blobs using the Levenberg-Marquardt algorithm, LEVENBERG_MARQUARDT($\mathbf{P}$) [Lev44, Mar63]. The function SET_VARIABLE($\mathbf{x}$) is used to label these parameters as variable to the solver, while all other blobs are kept fixed during the optimization run using SET_FIXED($\mathbf{x}$). Derivatives for the Jacobian matrix are computed numerically using finite differences (by repeatedly executing our forward renderer with the perturbed parameter vector). In a subsequent step, the sizes of the selected blobs are also included in a second optimization run, with a parameter $\sigma_{\max}$ defining an upper limit for the blob size. We found that this two-stage approach is necessitated by the strong non-convexity of the objective function. By optimizing over multiple blobs simultaneously, we allow the optimizer to recover complex geometry features that

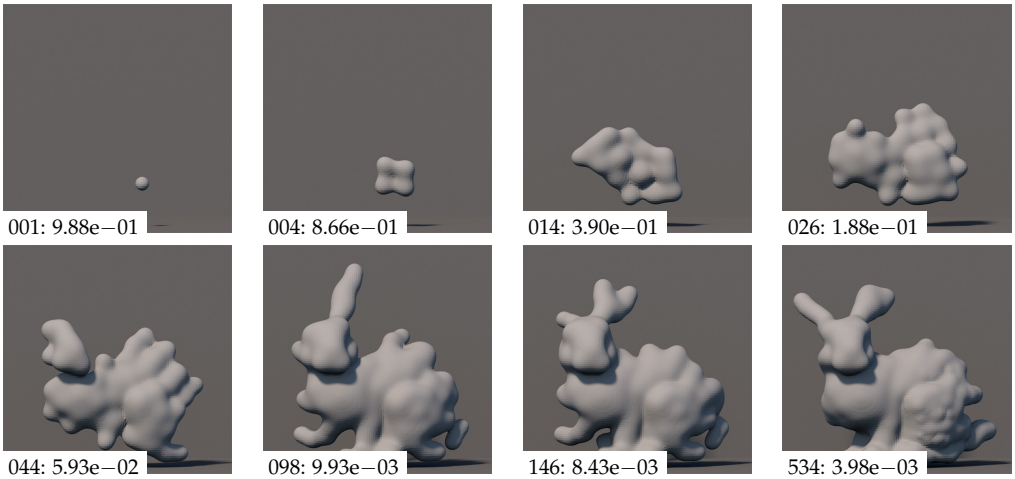


Figure 3.7: Convergence of reconstructed geometry for the Bunny dataset over the course of the optimization. Number pairs denote iteration number and value of cost function (relative to start value).

are influenced by more than a single blob.

The algorithm starts with a single blob as initial solution, then performs an outer loop over four phases: sampling, mutation, reiteration, and regularization. In the following, we provide a full description of the individual phases and explain our design choices. The parameters used in our reconstructions are shown in Table 3.2.

Sampling. Our algorithm pivots around locations in the reconstruction volume that are randomly chosen according to a distribution (PDF) that aims to give problematic regions a higher probability of being sampled. We obtain the PDF by backprojecting the absolute value of the current residual image into the working volume. For locations $\mathbf{x}$ that are sampled by the function SAMPLE(), our working hypothesis is that *something* about the solution should change there; we address this by selecting the nearest blob to this location (FIND_NEAREST($\mathbf{P}$, $\mathbf{x}$)) and applying and testing our three mutation strategies on it. Since each mutation probably increases the cost function, it is followed by a relaxation of the neighborhood of the pivot blob.

Mutation. We employ three mutation strategies to generate variations of the current solution. ADD_BLOB($\mathbf{P}$, $\mathbf{x}$) adds a new blob $(\mathbf{x}, \sigma_0)$ to $\mathbf{P}$. DELETE_BLOB($\mathbf{P}$, $\mathbf{x}$) deletes the blob $\mathbf{p} \in \mathbf{P}$ that is closest to $\mathbf{x}$. DUPLICATE_BLOB($\mathbf{P}$, $\mathbf{x}$) replaces the blob $\mathbf{p} \in \mathbf{P}$ by two new blobs that are dis-

Algorithm 1 Global optimization scheme

Require: Reference image $\mathbf{I}_{\text{ref}}$, Threshold c_{thresh}
Ensure: Parameter vector $\mathbf{P}$, Cost c

$\mathbf{x} \leftarrow \text{SAMPLE}(\varnothing)$
$\mathbf{P}, c \leftarrow \text{ADD\_BLOB}(\varnothing, \mathbf{x})$
while $c > c_{\text{thresh}}$ **do**
    $\mathbf{x} \leftarrow \text{SAMPLE}(\mathbf{P})$
    $\mathbf{P}_1, c_1 \leftarrow \text{ADD\_BLOB}(\mathbf{P}, \mathbf{x})$
    $\mathbf{P}_2, c_2 \leftarrow \text{DUPLICATE\_BLOB}(\mathbf{P}, \mathbf{x})$
    $\mathbf{P}_3, c_3 \leftarrow \text{DELETE\_BLOB}(\mathbf{P}, \mathbf{x})$
    $i \leftarrow \arg\min_x c_x$
    if $c_i < c$ **then**
        $\mathbf{P}, c \leftarrow \mathbf{P}_i, c_i$
    $\mathbf{P}_r, c_r \leftarrow \text{REITERATE}(\mathbf{P})$
    if $c_r < c$ **then**
        $\mathbf{P}, c \leftarrow \mathbf{P}_r, c_r$
    $\mathbf{P}, c \leftarrow \text{CHECK\_DELETE}(\mathbf{P}, c)$

Algorithm 2 Inner optimization scheme

function ITERATE($\mathbf{p}$, $\mathbf{P}$)
    $\mathbf{P}_{\text{opt}} \leftarrow \text{FIND\_NEIGHBORS}(\mathbf{p}, \mathbf{P}, 10)$
    SET_FIXED($\mathbf{P}$)
    for all $(\tilde{\mathbf{p}}, \tilde{\sigma}) \in \mathbf{P}_{\text{opt}}$ **do**
        SET_VARIABLE($\tilde{\mathbf{p}}$)
    $\mathbf{P} \leftarrow \text{LEVENBERG\_MARQUARDT}(\mathbf{P})$
    for all $(\tilde{\mathbf{p}}, \tilde{\sigma}) \in \mathbf{P}_{\text{opt}}$ **do**
        SET_VARIABLE($\tilde{\mathbf{p}}$)
        SET_VARIABLE($\tilde{\sigma}$)
    $\mathbf{P} \leftarrow \text{LEVENBERG\_MARQUARDT}(\mathbf{P})$
    $c \leftarrow \text{COMPUTE\_COST}(\mathbf{P})$
    return $\mathbf{P}$, c

placed by a vector $\pm\mathbf{d}$ from the original position so they can be separated by the optimizer. Out of the three solutions (each one after performing an inner optimization ITERATE($\mathbf{p}$, $\mathbf{P}$) on the neighborhood), the one with the lowest cost c_i is chosen to be the new solution. A call to ITERATE consists of two non-linear optimizations, one solely over the blob positions, followed by an optimization over both blob positions and sizes. This procedure is

Algorithm 3 Subroutines to Algorithm 1.

```
1: function ADD_BLOB(P, x)
2:     p ← (x, σ_0)
3:     return ITERATE(p, P ∪ p)

1: function CHECK_DELETE(P)
2:     for all p ∈ P do
3:         if COMPUTE_COST(P \ p) < η · c then
4:             P ← P \ p
5:     c ← COMPUTE_COST(P)
6:     return P, c

1: function DUPLICATE_BLOB(P, x)
2:     p ← FIND_NEAREST(P, x)
3:     p_1, p_2 ← SPLIT(p)
4:     return ITERATE(p, P \ p ∪ p_1 ∪ p_2)

1: function REITERATE(P)
2:     p ← CHOOSE_RANDOM(P)
3:     return ITERATE(p, P)

1: function REMOVE_BLOB(P, x)
2:     p ← FIND_NEAREST(P, x)
3:     return ITERATE(p, P \ p)
```

essential due to the non-convexity of the cost function, initial experiments have shown that skipping the first optimization generally results in unwanted, strong local minima, where a single blob spans large parts of the reconstruction volume.

Reiteration. As the next step, another call to ITERATE is performed on a random group of neighboring blobs. This re-evaluation of previously relaxed blobs is necessary to avoid being stuck in local minima during early iterations, when the hypothesis does not yet contain enough blobs to properly describe the transient response.

Regularization. Finally, the algorithm first checks each blob for its significance to the solution (CHECK_DELETE), and deletes it if doing so does not worsen the total cost by more than a small factor η. This regularizing step prevents the build-up of excess geometry in hidden regions that is not supported by the data. It is the only step that can lead to an increase in the cost c; all other heuristics ensure that the cost falls monotonically.

3.4.4 Implementation details

Our reconstruction software is written in C++. Geometry generation and rendering are implemented on the GPU, using NVIDIA CUDA and the Thrust parallel template library for the bulk of the tasks and the NVIDIA OptiX prime ray-tracing engine for the shadow tests. The optimization algorithm is implemented using the Ceres solver [AMO15]. Intermediate results are visualized on-the-fly using the VTK library [SML06]. We used various workstations in our experiments, with Intel Core i7 CPUs and NVIDIA GeForce GPUs ranging from GTX 780 to Titan Xp.

3.5 Evaluation

In this section, we verify the correctness of our renderer, and use it to reconstruct geometry from simulations and experimental measurements of around-the-corner scattered light. Input data, as well as output volumes and meshes of our proposed method and the state-of-the-art ellipsoidal backprojection method of [AGJ17] can be found in the supplemental material.

3.5.1 Correctness of renderer

Before we evaluate the performance of our overall reconstruction system, we test correctness and performance of the forward model that is at its heart, our custom renderer. To this end, we prepare test scenes and render reference images using Microsoft's Time of Flight Tracer [SC14], a transient renderer based on `pbrt` version 2 [PH10].

All our synthetic models use the same arbitrary unit for length and time. The standard temporal resolution (size of histogram bin) of our virtual detectors is 0.4 units. Typical time resolutions of real-world devices are 10 ps for streak cameras or 100 ps for SPAD detectors. Equating the bin size with these time constants results in a conversion factor to real-world distances of 8.3 mm and 83 mm per world unit, respectively. We arranged the scene such that the wall is a diffuse plane at $z = 0$ with normal in positive z direction. The object, with a typical size of 50 units, was located on the z axis at $z = 45$. The laser spot was modeled as a cosine-lobe light source pointing in positive z direction at one of four wall locations $(45, 0, 0)$, $(-45, 0, 0)$, $(0, 45, 0)$ and $(0, -45, 0)$. The range of observed points on the wall was represented by an area of 80×80 units^2 which was observed by an orthographic camera centered at $(x, y) = (0, 0)$.

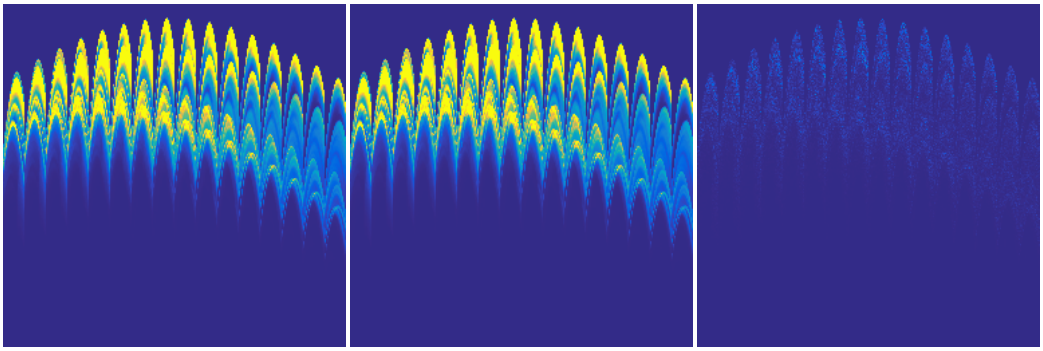


Figure 3.8: The physically-based renderings with and without global illumination are virtually indistingishuable. From left to right: Rendering with global illumination; rendering without global illumination; difference of the two renderings.

Comparison		PSNR [dB]	Rel. $L2$ error [%]
RTFull	/ RTTrunc	69.809	0.486
RTTrunc	/ OursFull	69.796	0.489
RTTrunc	/ OursNoFilter	53.379	3.237
RTTrunc	/ OursNoShadow	45.638	7.892
RTTrunc	/ OursNoShadowNoFilter	44.942	8.550

Table 3.1: Using the Stanford Bunny as test object, we compare our renderer to ray-traced renderings with maximum path lengths of 2 (RTTrunc) and ∞ (RTFull). With all the features enabled (OursFull), our renderer matches the ray-traced solution for the 3-bounce setting (wall-object-wall) to 0.49 %, which is on the same order as the influence of global illumination (RTFull) on this scene. Omission of shadow tests and temporal filtering result in significantly higher error values.

Using a 30 % reflective triangle mesh model of the Stanford Bunny, we generated two reference renderings of $16 \times 16 \times 256$ spatio-temporal resolution using the physically-based renderer, one with full global illumination and one with a maximum path length of 2 reflections. With the cosine light source representing the spot lit by the laser, a path length of 2 includes light scattering from the wall to the object and back to the wall, but not light that has been interreflected at the object or that has bounced between object and wall multiple times. In Figure 3.8, both versions are shown along with the difference. At least for our around-the-corner setting, we found that the error caused by truncating the path length to 2 is not very significant, with 69.809 dB peak signal-to-noise ratio (PSNR) or a relative $L2$ difference of 0.486 %.

We then used the truncated rendering as reference for our own ren-

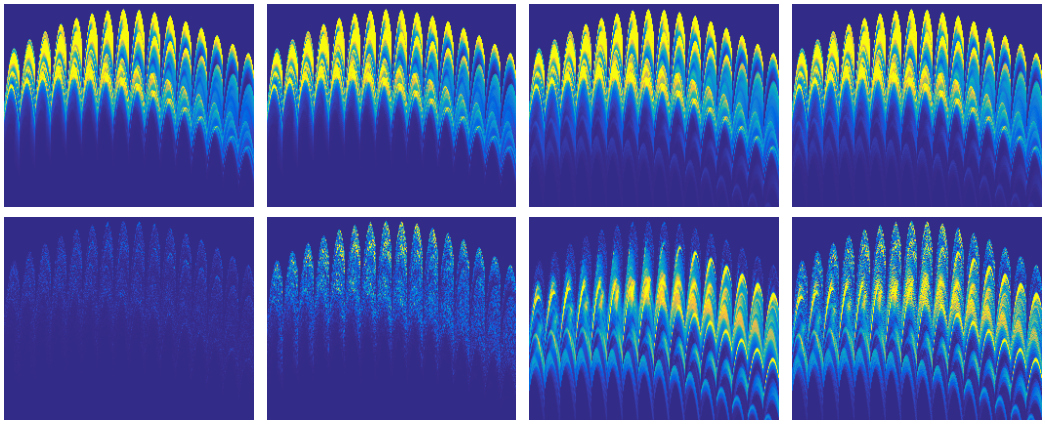


Figure 3.9: The effect of our augmentations on the rendering error. The top row shows transient renderings made with our renderer, the bottom row shows the respective difference to the ground truth `toftracer` rendering (range scaled for print). From left to right: Our renderer with all features turned on; temporal filtering turned off; shadow tests turned off; temporal filtering and shadow tests turned off. Error metrics for these renderings are provided in Table 3.1.

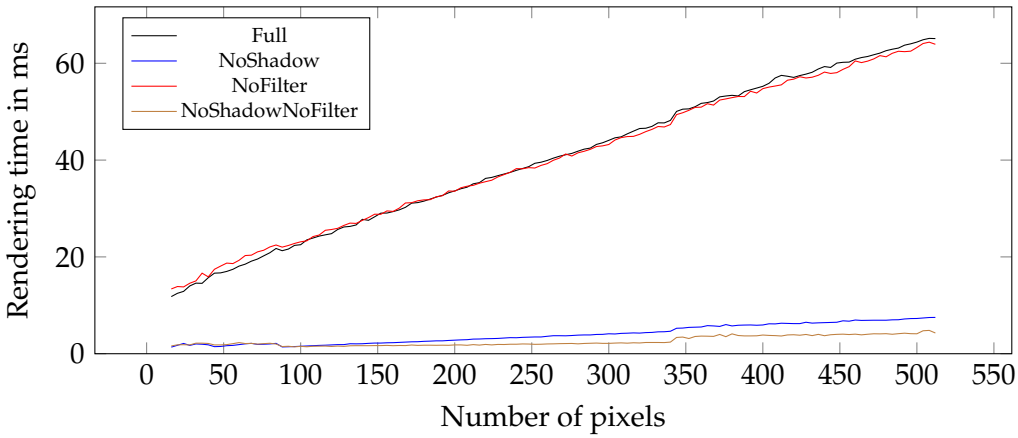


Figure 3.10: Rendering performance of four versions of our algorithm (with/without filtering, with/without shadow test) as a function of output pixel count.

derer, and tested the effect of temporal filtering and shadow testing on the difference (Figure 3.9). A naïve version of our renderer, with all refinements disabled, reached the reference up to an error of a little under 10 %. After activating the temporal filtering and the shadow tests, our fast renderer delivered a close approximation to the ray-traced reference with with 69.796 dB peak signal-to-noise ratio (PSNR) or a relative $L2$ difference

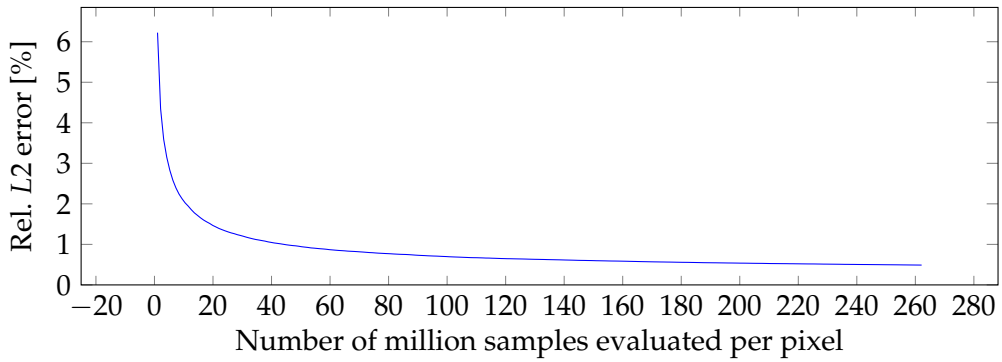


Figure 3.11: Difference between our fast renderer and the ray-traced reference solution with a varying number of samples per pixels.

of 0.489 %. All error values are provided at a glance in Table 3.1. The main result from this investigation is that both features are essential to our renderer. The gain in accuracy comes at the expense of significantly increased runtime when using the shadow test (Figure 3.10). For small numbers of pixels, a significant part of that runtime is caused by the construction of acceleration structures—here, about 10 ms for an object with approximately 55,000 triangles. Another noteworthy observation is that the Monte-Carlo rendering used as reference was likely not fully converged (Figure 3.11) even after evaluating 250 million samples per pixel. We expect that more exhaustive sampling would likely have further reduced the error.

3.5.2 Geometry reconstruction

We used various types of input data to test our algorithm: synthetic data generated using a path tracer or our own fast renderer, as well as experimental data obtained from other sources. The results from these reconstructions are scattered throughout the paper, referencing the datasets from Table 3.2 by their respective names. Meshes are rendered in a daylight environment using Mitsuba [Jak10], with a back wall and ground plane added as shadow receivers for better visualization of the 3D shapes. Note that these planes are not part of the experimental setup.

Synthetic datasets After establishing in Section 3.5.1 that our fast renderer produces outcomes that are almost identical to the ray-traced reference, we used both the path tracer and our fast renderer to generate a variety of around-the-corner input data. In particular, we prepared several variations of the `Mannequin` scene, reducing the number of pixels, the

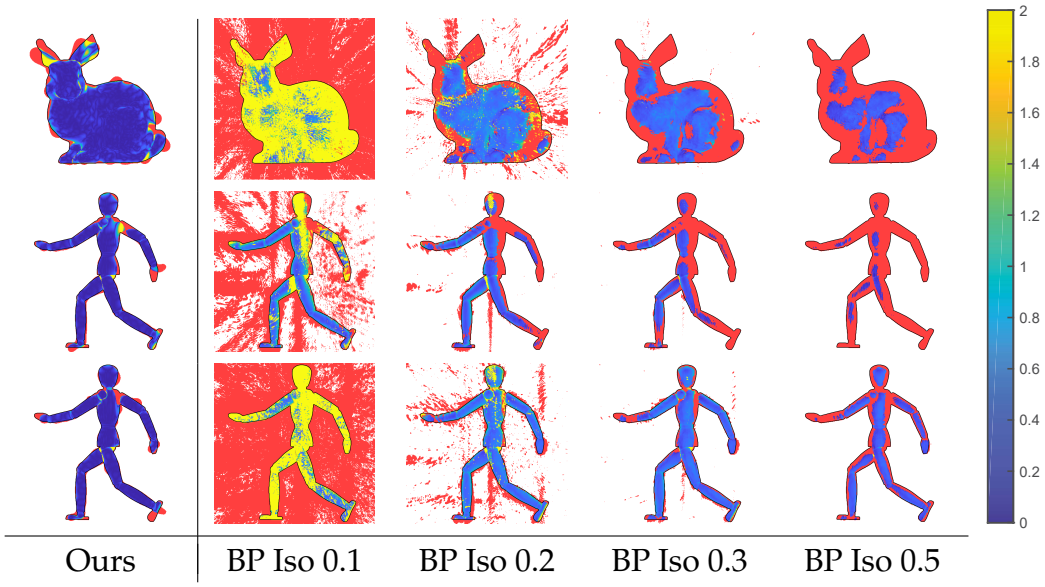


Figure 3.12: Absolute depth error (in world units) in the reconstructions obtained from the synthetic `Bunny` (top), `Mannequin1Laser` (middle), and `Mannequin` (bottom) datasets. The left column shows the result obtained using our method, the right four columns show depth errors for backprojection (BP) with varying isovalues. The black line indicates the ground-truth object silhouette. Red color inside the silhouette indicates a missing (false-negative) surface and outside a silhouette it indicates excess (false-positive) geometry. Note that the range is clamped to $[0, 2]$ for visualization; values plotted in yellow can be significantly higher. See Figure 3.13 for a quantitative analysis.

number of laser spots, as well as the temporal resolution. An overview of all our datasets, as well as the parameters used for reconstructing them, can be found in Table 3.2. Like the backprojection method, ours too has a small number of parameters: the upper bound for the blob size σ_0 and the regularization parameter η.

We show renderings of the reconstructed meshes alongside the backprojected solutions, obtained using the Fast Backprojection code provided by Arellano et al. [AGJ17], and ground truth (Figure 3.2). They show that the quality delivered by our algorithm, in general, outperforms the state-of-the-art method on the synthetic datasets examined in this study. The meshes produced by our method tend to be more complete, smoother, and overall closer to the true surface. We also performed more quantitative evaluations. Figures 3.12 and 3.13 show the error of the recovered surface in z-direction for three datasets. In general, meshes generated using the backprojection method tend to lie in front of the true surface. This

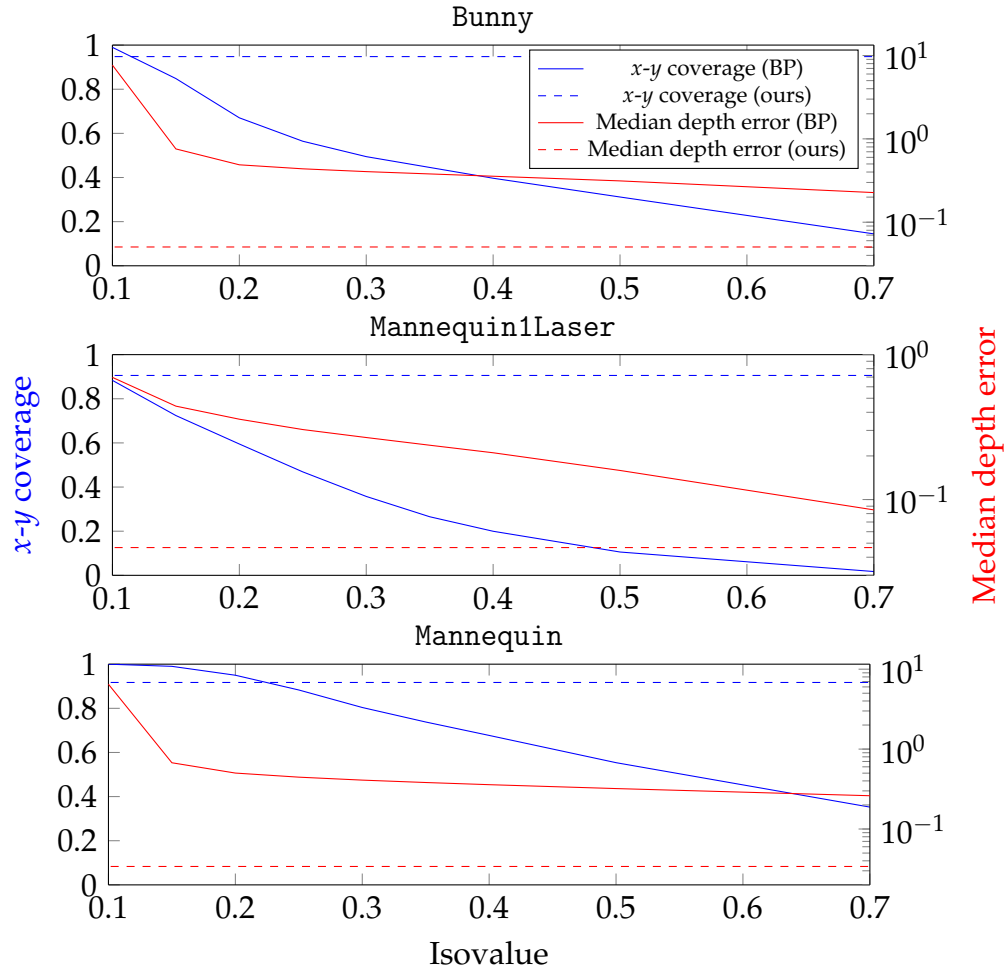


Figure 3.13: Evaluation of the depth map coverage in the x-y plane (higher is better) and the median absolute depth error in z direction (lower is better) for the `Bunny`, `Mannequin1Laser`, and `Mannequin` datasets. The proposed method achieves coverage values above 90 % with a median depth error as low as 0.03 to 0.05 world units. For the state-of-the-art method, no isovalue is capable of simultaneously achieving high coverage and low depth error. A qualitative visualization of this study can be found in Figure 3.12.

is due to the way surface geometry is reconstructed from the density volumes obtained by the backprojection algorithm. Even if the peak of the density distribution lies exactly on the object geometry, extracting an isosurface will displace it by a certain distance. Our reconstructions, which are based on a surface scattering model, do not suffer from this effect.

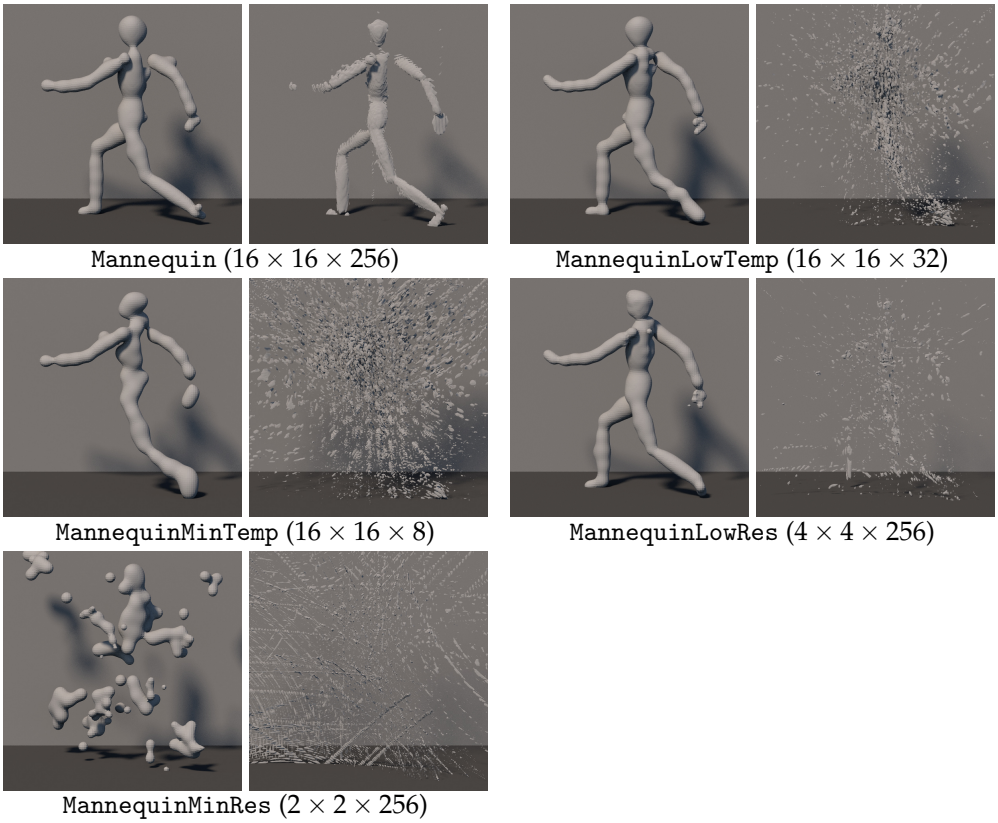


Figure 3.14: Reconstruction of the `Mannequin*` dataset using different levels of degradation. From left to right: `Mannequin`, `MannequinLowTemp`, `MannequinMinTemp`, `MannequinLowRes`, `MannequinMinRes`. Left: Our reconstruction, right: backprojection. Unlike backprojection, our reconstruction method handles degradations in the input data quite gracefully. Even an extremely low spatial resolution of 2×2 pixels or a temporal resolution of only 8 bins still produces roughly identifiable results.

Degradation experiments To put the robustness of our method to the test, we performed a series of experiments that deliberately deviate from an idealized, noise-free, Lambertian and global-illumination-free light transport model, or reduce the amount of input data used for the reconstruction. In a first series of experiments, we sub-sampled the `Mannequin` dataset both spatially and temporally, and observed the degradation in reconstructed outcome (Figure 3.14). In a second series, we added increasing amounts of Poisson noise (Figure 3.15). Next, we replaced the diffuse reflectance of the `BunnyGI` model by a metal BRDF (Blinn model as implemented by `pbrt`) and decreased the roughness value (Figure 3.16). Our fast renderer used during reconstruction was set to the same BRDF param-

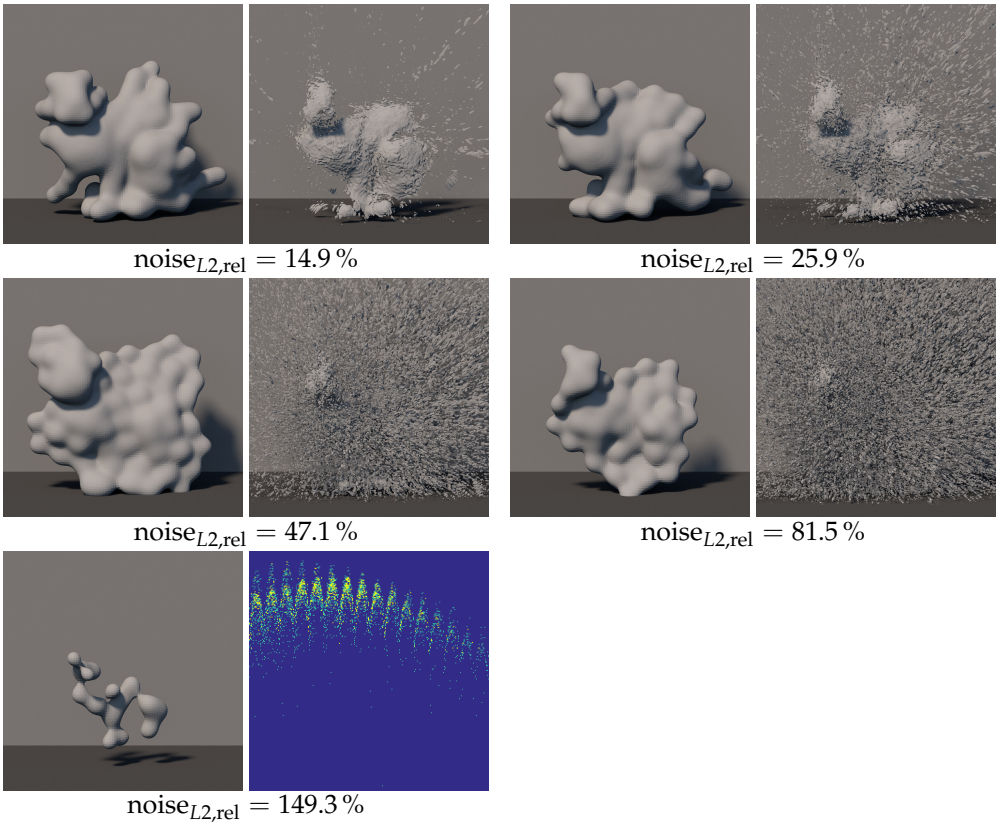


Figure 3.15: Reconstruction of the `BunnyGI` dataset with different levels of Poisson noise applied to the input data. Relative $L2$ errors: 14.9 %, 25.9 %, 47.1 %, 81.5 %, 149.3 %. Left: Our reconstruction, right: backprojection. Our algorithm is based on a noise-free forward model. It therefore manages to localize the object reliably even under very noisy conditions (albeit at reduced reconstruction quality). In the rightmost example (streak plot), at most two photons have been counted per pixel, resulting in data that contains 50 % more noise than signal.

eters that were used to generate the input data. Finally, we constructed a strongly concave synthetic scene (`Bowl`) and used high albedo values in order to test the influence of unaccounted-for global illumination on the reconstructed geometry (Figure 3.17).

As expected, in all these examples, the further the data deviates from the ideal case, the more the reconstruction quality decreases. While backprojection tends to be more robust with respect to low-frequency bias (`Bowl` experiment), our method quite gracefully deals with high-frequency noise by fitting a low-frequent rendering to it. For highly specular materials, the discretization of the surface mesh and the sensing locations on the wall may lead to sampling issues: specular glints that are missed by the

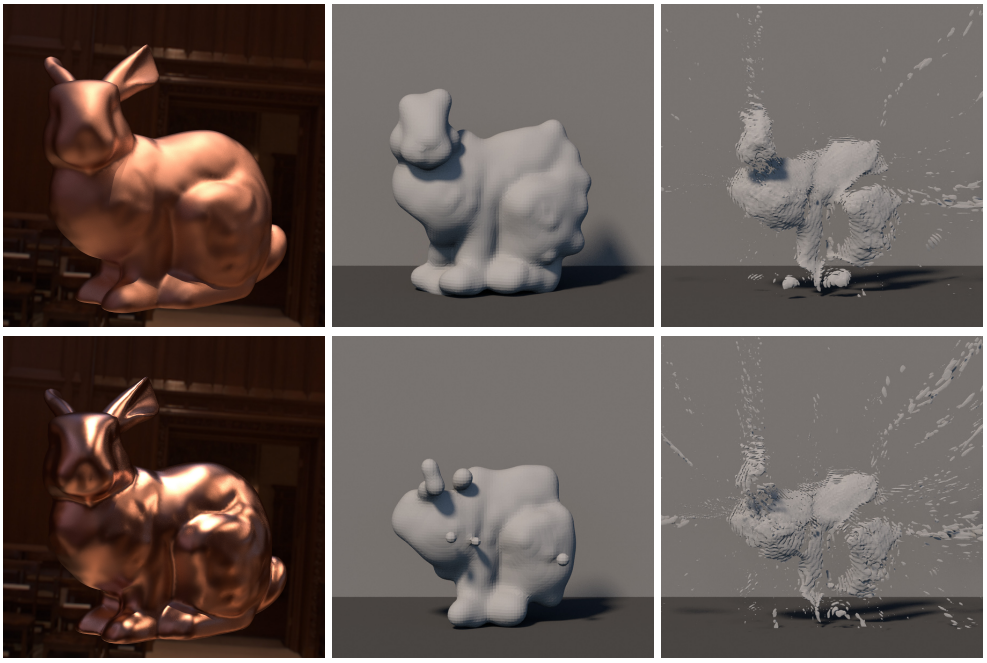


Figure 3.16: Reconstruction of the `BunnyMetal*` scenes with `pbrt`'s `metal` BRDF applied to the object (top row: Blinn roughness 0.05; bottom row: Blinn roughness 0.01). From left to right: reference rendering in Grace Cathedral environment [Deb98]; our proposed method; backprojection.

forward simulation cannot contribute to the solution.

Experimental datasets We show reconstructions of two experimental datasets obtained using SPAD sensors.

The first dataset (`SPADScene`) was measured by Buttafava et al. [BZT+15], by observing a single location on the wall with a SPAD detector, and scanning a pulsed laser to a rectangular grid of locations. We note that this setup is dual, and hence equivalent for our purpose, to illuminating the single spot and scanning the detector to the grid of different locations. The dataset came included with the Fast Backprojection code provided by Arellano et al. [AGJ17]. To apply our algorithm on the `SPADScene` dataset, we first subtracted a lowpass-filtered version (with $\sigma = 1000$ bins) of the signal to reduce noise and background, then downsampled the dataset from its original temporal resolution by a factor of 25.

Like in the original work, the reconstruction remains vague and precise details are hard to make out (Figure 3.18). The reconstructed blobby objects appear to be in roughly the right places, but their shapes are poorly defined. We note that our method quite clearly carves out the letter "T"

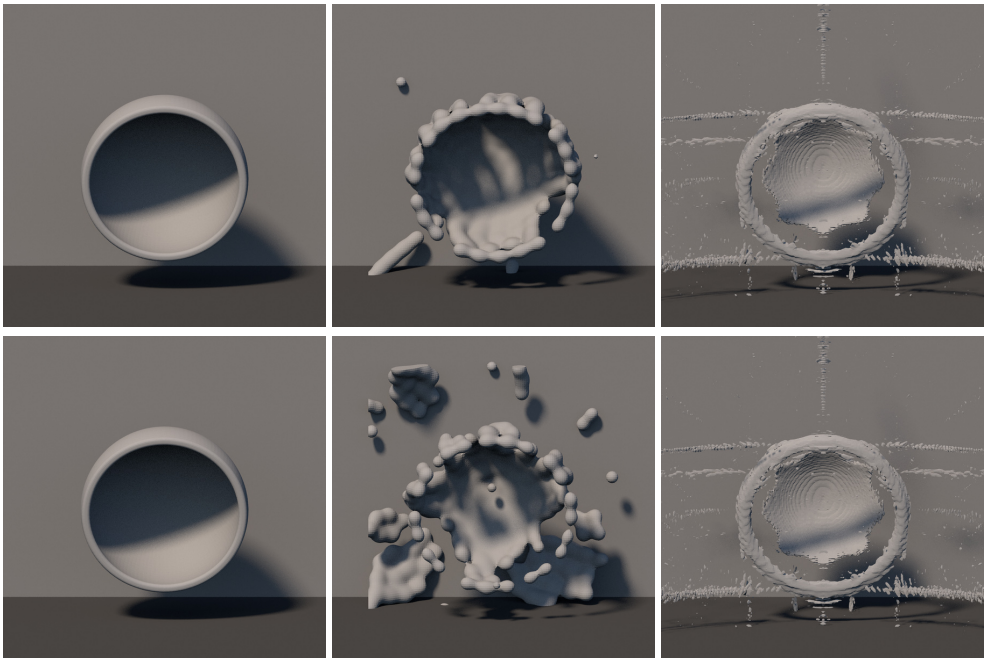


Figure 3.17: `Bowl` scene. A strongly concave shape with high albedo (top row: 30 %; bottom row: 100 %) features large amounts of interreflected light in the input data, which leads to spurious features in the reconstructed geometry. From left to right: reference geometry; our proposed method; backprojection.

where backprojection delivers a less clearly defined shape (Figure 3.19).

The second dataset (`OTooleDiffuseS`) is a measurement of a letter "S" cut from white cardboard, which O'Toole et al. measured via a diffuse wall using their confocal setup [OLW18a]. In this setup, illumination and observation share the same optical path and are scanned across the surface. We downsampled the input data by a factor of $4 \times 4 \times 4$ in the spatial and temporal domains. Although the inclusion of the direct reflection in the data allowed for a better background subtraction and white point correction than in the case of the previous dataset, it becomes clear that there must be more sources of bias. In particular, we identified a temporal blur of roughly 3 time bins. Adding a similar blur to our renderer (a box filter of width 3 bins), made the reconstructed "S" shape much more clearly recognizable as such (Figure 3.20).

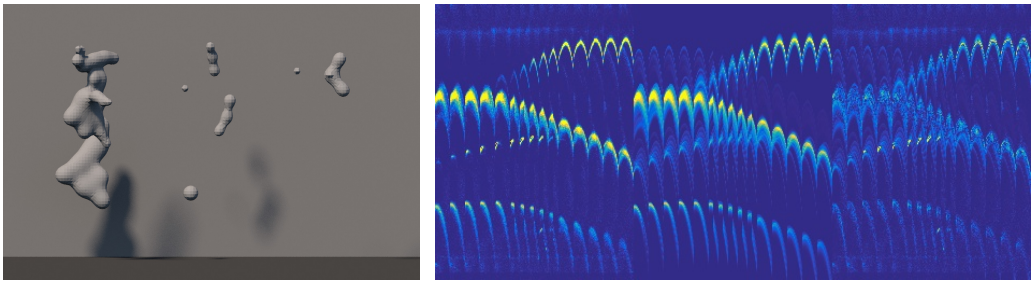
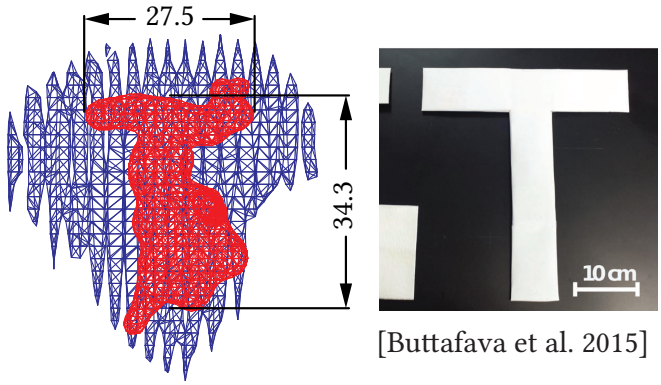


Figure 3.18: Reconstruction of the experimental `SPADScene` dataset [BZT+15]. Shown is the output mesh and the transient data (from left to right: observation, prediction, residual).



[Buttafava et al. 2015]

Figure 3.19: The "T" object from the experimental `SPADScene` dataset published by Buttafava et al. [BZT+15]. Shown are reconstructions obtained using backprojection (blue) and the proposed method (red), along with approximate dimensions using the scale provided in the original work (right).

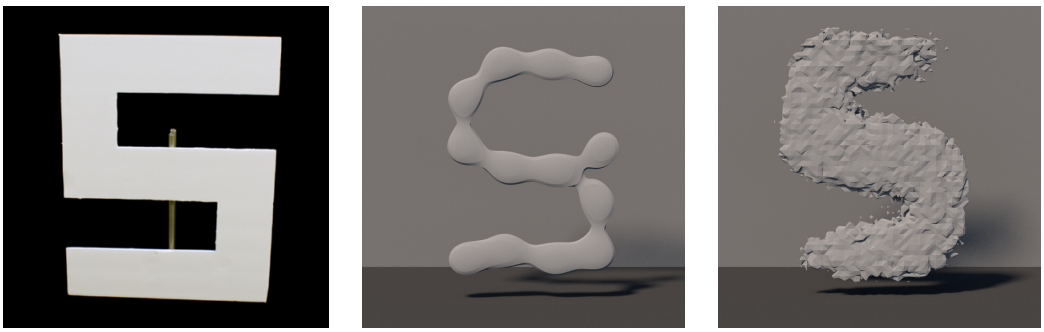


Figure 3.20: `OTooleDiffuseS` dataset [OLW18a]. From left to right: photo of diffuse "S"-shaped cutout; surface mesh reconstructed using our method; mesh reconstructed using method described in [OLW18a].

3.6 Discussion

In the proposed approach, we develop computer graphics methodology (a near-physical, extremely efficient rendering scheme) to reconstruct occluded 3D shape from three-bounce indirect reflections. To our knowledge, this marks the first instance of a non-line-of-sight reconstruction algorithm that is consistent with a physical forward model. This solid theoretical foundation leads to results that, under favorable conditions, show higher object coverage and detail than the de-facto state of the art, error backprojection. In extreme situations, like very low spatial / temporal resolutions or high noise levels, we have shown that our method breaks down significantly later than the current state of the art (Figures 3.14 and 3.15). Under conditions that are not covered by the forward model (noise, bias / background, global illumination) the results are on par or slightly inferior to existing methods. In terms of runtime, our method typically takes several hours or even days for a reconstruction run (Table 3.2) and therefore cannot compete with recent optimized versions of error backprojection [AGJ17] or GPU-based deconvolvers [OLW18b], which are typically on the order of 10 s to 100 s and 1 s respectively. However, we consider this a soft hindrance that has to be considered together with the fact that the capture of suitable input data, too, is far from being instantaneous. This latter factor is governed by the physics of light and therefore may turn out, in the long run, to impose more severe limitations to the practicality of non-line-of-sight sensing solutions.

We noted that the reconstruction quality of the SPAD datasets stays behind the quality of the synthetic datasets (whether path-traced or using our own renderer). Our image formation model approximates the physical light transport up to very high accuracy (as shown in Section 3.5.1), but does not explicitly model the SPAD sensor response to the incoming light. The SPAD data is biased due to background noise and dark counts, and the temporal impulse response is asymmetric and smeared out due to time jitter and afterpulsing [GRA$^+$11, HGJ17]. While these effects could easily be incorporated into our forward model, doing so would require either a careful calibration of the imaging setup (which was not provided with the public datasets) or an estimation of the noise parameters from input data. In this light, we find the presented results very promising for this line of research, and consider the explicit application of measured noise profiles and the modeling of additional imaging setups as future work.

A key feature of our method is that, within the limitations of the forward model (opaque, but not necessarily diffuse, light transport without

further interreflections) good solutions can be immediately identified by a low residual error. However, the non-convex objective and possibly unknown noise and background terms may make it challenging to reach this point. Our optimization scheme, while delivering good results in the provided examples, offers no guarantee of global convergence. As of today, it is unclear which of the two factors will prove more important in practice, the physical correctness of the forward model or the minimizability of the objective derived from it.

3.7 Future work

We imagine that extended versions of our method could be used to jointly estimate geometry and material. Advanced global optimization heuristics could further improve the convergence behavior and the overall quality of the outcome. We imagine that hierarchical approaches or hybrid solutions might bring further improvement, for instance by using the (physically inaccurate but global) solution of one reconstruction scheme to warm-start another local optimization run using a more accurate model like ours.

The extrinsic and intrinsic calibration of traditional 2D imaging setups is well understood [Zha00]. However, this problem has not been satisfyingly solved by the NLOS reconstruction community so far. The current best practice is to manually estimate the positions and normals of the projected camera pixels, potentially leading to a systematic bias in the (typically non-metric) reconstructions. Our proposed method presents not only an alternative solution to NLOS reconstruction, but also lays out a foundation for solving related problems. Here, we presented a method for recovering the scene geometry, where the acquisition geometry was assumed to be known. In future work, we would like to study the *dual problem*, where the scene geometry is known (a calibration target), but the acquisition geometry is unknown. We conducted initial experiments with our synthetic `Bunny` dataset and were able to recover the positions and normals of four projected pixels up to a very high precision, regardless of an overly imprecise initial guess, see Figure 3.21. Again, we utilize Equation (3.1) as the objective function, but the parameter vector $\mathbf{P}$ consists of the positions and normals of the projected pixels. Challenges will include the generalization to real-world data, the design of an optimum calibration target, and the validation against measured data. We could also imagine utilizing our forward model to estimate the parameters of a SPAD sensor response model [HGJ17].

Finally, our renderer is not constrained to use in a costly iterative

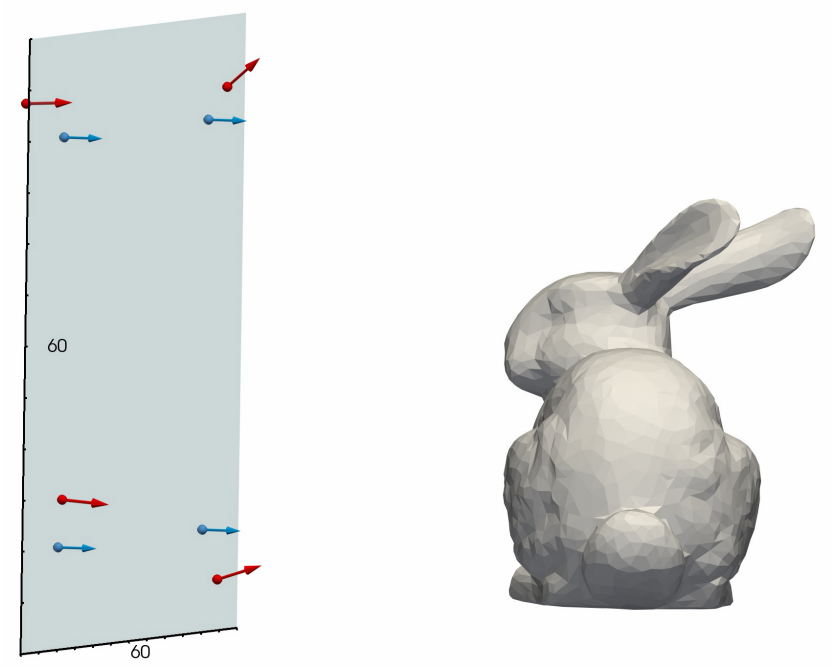


Figure 3.21: An illustration of our preliminary NLOS camera calibration experiment. A transient image of the `Bunny` scene has been acquired using our transient renderer. The blue arrows denote the ground truth positions and normals of the projected camera pixels. The red arrows show an initial guess before optimization with a positional RMSE of 2.7 units and an average angular error of 16.6°. After optimization using Equation (3.1), the optimized pixel positions and normals coincide with the ground truth up to floating point precision. Due to the greatly reduced number of variables compared to the geometry reconstruction problem, the optimization concluded in less than one minute.

solver. Just as well, we can imagine using it to enable new machine learning approaches to the problem. A suitably trained feedforward neural network, for example, would deliver instant results. Whereas existing renderers are too slow for generating large amounts of training data, our renderer would be fast enough to obtain millions of datasets in a single day. Together with a suitable signal degradation model [HGJ17], we expect that it will be possible to closely approximate the most relevant real-world scenarios.

Name	Reference	Resolution	# Lasers	s_{geom}	s_{camera}	η	σ_0	t_0	δ_t	c_n/c_0 [%]	T [min]	n_{iters}	n_{blobs}
`Bunny`	Ours	$16 \times 16 \times 256$	4	40×40	80×80	1.01	1.5	80	0.4	0.32	5096	660	156
`BunnyGI`	pbrt	$16 \times 16 \times 256$	4	40×40	80×80	1.01	1.5	80	0.4	0.59	5611	181	109
`BunnyMetal0.05`	pbrt	$16 \times 16 \times 256$	4	40×40	80×80	1.01	1.5	80	0.4	2.02	3419	259	101
`BunnyMetal0.01`	pbrt	$16 \times 16 \times 256$	4	40×40	80×80	1.01	1.5	80	0.4	11.75	3005	361	87
`BowlAlbedo0.3`	pbrt	$16 \times 16 \times 256$	4	26×26	80×80	1.001	0.4	80	0.4	4.36	5579	167	155
`BowlAlbedo1`	pbrt	$16 \times 16 \times 256$	4	26×26	80×80	1.001	0.4	80	0.4	31.53	4280	267	197
`Mannequin`	Ours	$16 \times 16 \times 256$	4	40×49	80×80	1.005	1.5	90	0.4	1.46	2326	505	69
`MannequinLowRes`	Ours	$4 \times 4 \times 256$	4	40×49	80×80	1.005	1.5	90	0.4	1.41	1251	252	76
`MannequinMinRes`	Ours	$2 \times 2 \times 256$	4	40×49	80×80	1.005	1.5	90	0.4	4.44	931	350	101
`MannequinLowTemp`	Ours	$16 \times 16 \times 32$	4	40×49	80×80	1.005	1.5	90	3.2	1.21	1322	166	67
`MannequinMinTemp`	Ours	$16 \times 16 \times 8$	4	40×49	80×80	1.005	1.5	90	12.8	7.95	420	102	23
`Mannequin1Laser`	Ours	$16 \times 16 \times 256$	1	40×49	80×80	1.005	1.5	90	0.4	0.59	1419	243	57
`SPADScene`	Measured	$185 \times 1 \times 256$	1	—	—	1.005	4.5	373	0.748	20.31	1280	328	43
`OTooleDiffuseS`	Measured	$64 \times 64 \times 2048$	1	—	—	1.01	0.015	0.756	0.0012	33.43	67	13	13

Table 3.2: Parameters of our reconstructed scenes, where s_{geom} is the size of the ground truth object projected onto the diffuse camera wall in world units, s_{camera} is the area covered by the camera in world units, η is the drop deletion factor in Algorithm 3, σ_0 is the initial blob standard deviation, t_0 is the time stamp of the first time bin, δ_t is the size of a time bin, and c_n/c_0 is the residual cost after optimization (relative to the initial cost). The total reconstruction times T are taken from file timestamps and vary due to manual termination of the reconstruction procedure, execution on different GPU models, overhead through parallel execution of multiple jobs, as well as debugging output. The optimizations terminated after n_{iters} iterations and consist of n_{blobs} Gaussian blobs. Please note that the exact scene geometry is only known for the synthetic experiments.

This chapter consists of our work on fabricated style transfer using real wood veneer. Following our approaches in Chapters 2 and 3, we extract hidden information from image data. In this case, we show that arbitrary target images are latently present in the physical materials and develop a discrete optimization method to generate fabricable cut patterns.

This chapter was published as [IWHH19]: Julian Iseringhausen, M. Weinmann, W. Huang, Matthias B. Hullin: "Computational Parquetry: Fabricated Style Transfer with Wood Pixels". *arXiv:1904.04769 [cs.GR]*, April 2019. Currently under review at ACM Transactions on Graphics.

CHAPTER 4

Computational Parquetry: Fabricated Style Transfer with Wood Pixels

Abstract Parquetry is the art and craft of decorating a surface with a pattern of differently colored veneers of wood, stone or other materials. Traditionally, the process of designing and making parquetry has been driven by color, using the texture found in real wood only for stylization or as a decorative effect. Here, we introduce a computational pipeline that draws from the rich natural structure of strongly textured real-world veneers as a source of detail in order to approximate a target image as faithfully as possible using a manageable number of parts. This challenge is closely related to the established problems of patch-based image synthesis and stylization in some ways, but fundamentally different in others. Most importantly, the limited availability of resources (any piece of wood can only be used once) turns the relatively simple problem of finding the right piece for the target location into the combinatorial problem of finding optimal parts while avoiding resource collisions. We introduce an algorithm that allows to efficiently solve an approximation to the problem. It further addresses challenges like gamut mapping, feature characterization and the search for fabricable cuts. We demonstrate the effectiveness of the system by fabricating a selection of pieces of parquetry from different kinds of unstained wood veneer.

4.1 Motivation

The use of differently colored and structured woods and other materials to form inlay and intarsia has been known at least since ancient Roman and Greek times. In the modern interpretation of this principle, pieces of veneer form a continuous thin layer that covers the surface of an object (*marquetry* or *parquetry*) [JDJ96]. The techniques denoted by these two terms share many similarities but are not identical. Marquetry usually refers to a process similar to "painting by numbers", where a target image is segmented into mostly homogeneous pieces which are then cut from more or less uniformly colored veneer and assembled to form the final ornament or picture. Parquetry, on the other hand, denotes the (ornamental) covering of a surface using a regular geometric arrangement of differently colored pieces. While most artists in their work embrace the grain and texture found in their source materials, they mostly use it as a decorative effect. Nevertheless, the resulting artworks can attain high levels of detail, depending on the amount of labor and care devoted to the task (Figure 4.2).

To overcome the "posterized" look of existing woodworking techniques, make use of fine-grained wood structures, and obtain results that are properly shaded, we introduce *computational parquetry*. Our technique can be considered a novel hybrid of both methods and is vitally driven by a computational design process. The goal of computational parquetry is to make deliberate use of the rich structure present in real woods, using heterogeneities such as knots, grain or other texture as a source of detail for recreating more faithful renditions of target images in wood, using a moderate number of pieces, see Figure 4.1. Since this goal can only be achieved by exhaustively searching suitable pieces of source material to represent small regions of the target image, the task is absolutely intractable to solve by hand. In the computer graphics world, our technique is closely related to patch-based image synthesis [BZ17] or texture synthesis [WLKT09], a well-explored family of problems for which a multitude of very elaborate and advanced solutions exist today.

Our end-to-end system for fabricated style transfer only uses commonly available real-world materials and can be implemented on hobby-grade hardware (laser cutter and flatbed scanner).

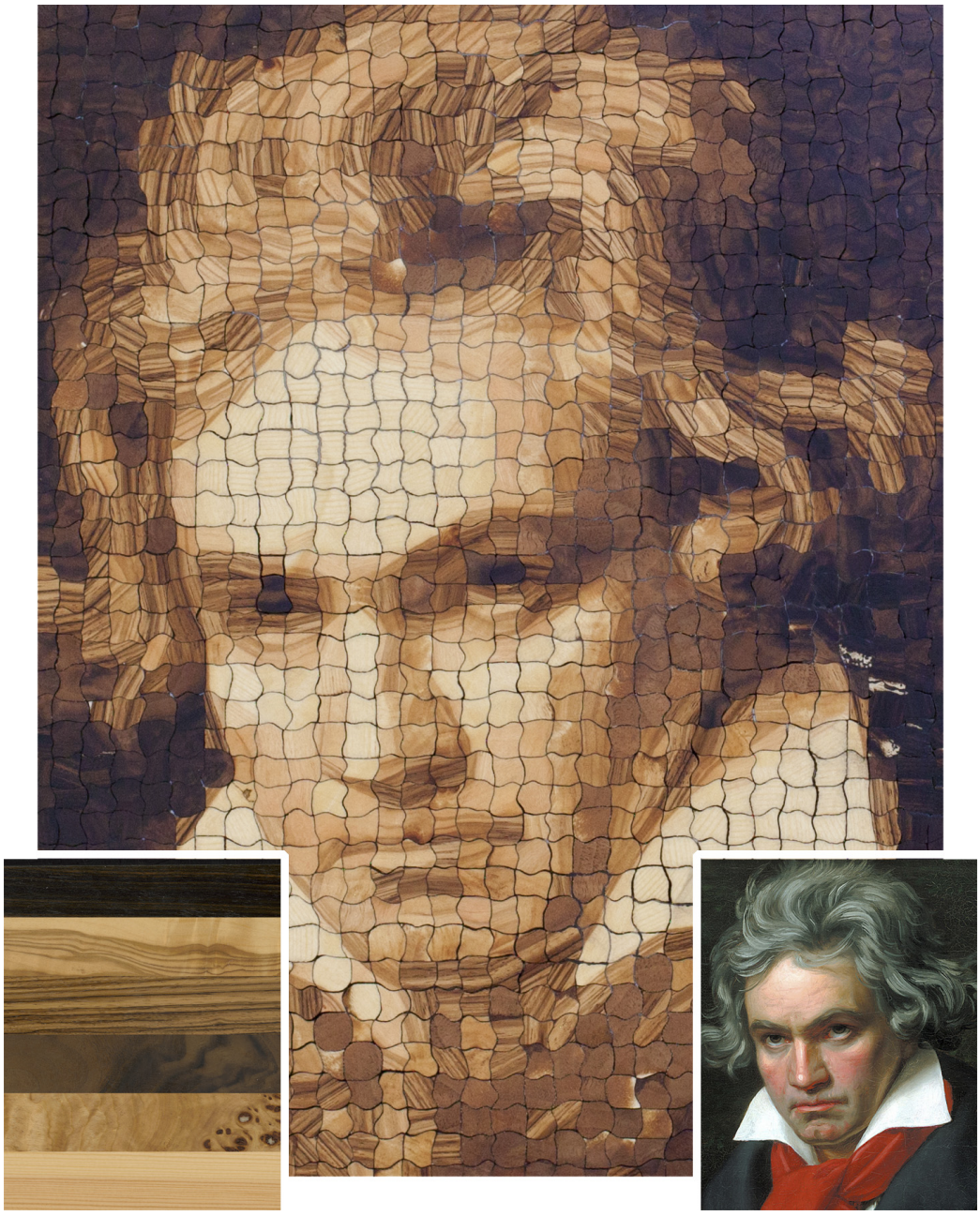


Figure 4.1: A fabricated piece of wood parquetry, produced using our pipeline. The inputs are a set of six different wood veneers (bottom left corner: poplar burl, walnut burl, santos rosewood, quartersawn zebrawood, olive, fir), and a target image (bottom right corner). The total size of the parquetry is approx. 27 cm $\times$ 34 cm. By combining the different appearance profiles (including color and grain structures) of multiple wood types, we are able to produce results with high contrast and fine structural details.

4.2 Related work

History of the craft. History knows a rich tradition of techniques that use patches of material for the purpose of composing images. Ancient Roman and Greek mosaics are probably the best-known early instances of this idea. An exemplary mosaic from the second century AD is shown in Figure 4.3. Often, such mosaics consist of largely uniformly shaped primitive shapes (e.g., square tiles) that are aligned with important structures, such as object boundaries, found in the target image. A modern counterpart of mosaics is pixel artwork, which has played a similarly ubiquitous role predominantly through video games in the 80s and 90s. Here, the design pattern is generally aligned with a Cartesian grid.

Marquetry can be considered a generalization of mosaic. This art of forming decorative images by covering object surfaces with fragments of materials such as wood, bone, ivory, mother of pearl or metal, has also been known at least since Roman times [Ulr07], see Figure 4.3. The appearance of the resulting image, however, is mostly dominated by the choice of materials and the shape of fragments. The closely related term parquetry refers to the assembly of wooden pieces to obtain decorative floor coverings. Either technique can be implemented either by carving and filling a wood surface (inlay) or covering the entire surface with a continuous layer of thin veneer pieces. The materials can be altered in appearance, for instance by staining, painting or carving.

In this paper, we use the term parquetry more restrictively to refer to two-dimensional arrangements of wood veneer that are unaltered in color (except for a final layer of clear varnish that is applied to the entire design). While some artists use computational tools, such as posterization, to find image segmentations (Figure 4.2), we believe that our method marks the first time that a measured texture of the source material has been used to drive the design process, explicitly making use of features present in the wood.

Stylization. With the goal of non-photorealistic rendering, numerous techniques have been proposed to transform 2D inputs into artistically stylized renderings [KCWI13]. This includes approaches for the simulation of different painting media such as paints, charcoal and watercolor [CKIW15, LBDF13, PPW18]. In recent years, the potential of deep learning has been revealed for rendering a given content image in different artistic styles [JYF+17]. Inspired by the ancient mosaics and the application of mosaics for arts (see e.g. Salvador Dalí's lithograph *Lin-*

Figure 4.2: Two modern examples of marquetry portraits of different complexity. Left: Self-portrait by Laszlo Sandor (using two maple specimens, brown and black walnut, beech, Indian rosewood, okoume and sapele; original size approx. 10 cm $\times$ 10 cm). Right: Portrait of a girl by Rob Milam (using wenge, Carpathian elm burl, Honduran rosewood, lauan, pear, plaintree, maple and ash; original size approx. 53 cm $\times$ 53 cm).

Figure 4.3: Examples for intarsia and ancient mosaics: The intarsia from the year 1776 depicts the adoration of St. Theodulf of Trier and a landscape with plowing farmers and St. Theodulf (left). The mosaic from the 2nd century AD depicts a scene from the Odyssey (right).

coln in Dalivision [Dal91] or *Self Portrait I* by Chuck Close [CY95]), a lot of effort has been spent on non-photorealistic rendering in mosaic-style. The original photo mosaic approach [Sil97] creates a mosaic by matching and stitching images from a database. Further work focused on the application to non-rectangular grids and color correction [FR98] and tiles of arbitrary shape (jigsaw image mosaics or puzzle image mosaics) [KP02,

BGP05, PCK09], the adjustment of the tiles in order to emphasize image features within the resulting mosaic [Hau01, EW03, LVJ10, BMP12] as well as speed-ups of the involved search process [BP05, BGP05, KSRY11]. More recently, texture mosaics have also been generated with the aid of deep learning techniques (e.g. [JBS17]). We refer to respective surveys [BBFG06, BBFG07] for a more detailed discussion of the underlying principles. Furthermore, panoramic image mosaics [SS97] have been introduced where photos taken from different views are stitched based on correspondences within the individual images and a final image blending.

Example-based synthesis. *Pixel-based* synthesis techniques [PL98, EL99, WL00, HJO+01] rely on copying single pixels from an exemplar to the desired output image while matching neighborhood constraints. In contrast, *patch-based or stitching-based* texture synthesis approaches [PFH00, EF01, KSE+03] involve copying entire patches from given exemplars. One major challenge of these approaches is the generation of correspondences between locations in the exemplar image and locations in the generated output image to copy the locally most suitable patches from the exemplar to the output image. For this purpose, common strategies include arranging patches in raster scan order and subsequently selecting several patch candidates that best fit to the already copied patches. As this matching process becomes computationally challenging for larger images, several investigations focused on improving matching efficiency [Ash01, TZL+02, BSFG09, BSGF10, BGSF11, DIIM04, LLX+01, SCSI08, HS12, OA12, WL00]. In addition, finding an adequate composition and blending of the copied patches has been addressed based on simple compositions of irregularly shaped patches [PFH00], the blending of overlapping patches within the overlap region [LLX+01], the specification of seams within the overlap region using dynamic programming or graph cuts [EF01, KSE+03], or the application of a weighted averaging for several overlapping regions [WSI07, SCSI08, BSFG09].

Furthermore, *optimization-based* techniques [PS00, HZW+06, KEBK05, KFCO+07, DSB+12, KNL+15] are based on the formulation of texture synthesis in terms of an optimization problem which is solved by minimizing an energy function and combines pixel-based and patch-based techniques. Recently, the potential of deep learning has also been demonstrated in the context of optimization-based texture synthesis (see e.g. [GEB15, GEB16, LW16a, LW16b]). For a more detailed review, we refer to the surveys provided by Wei et al. [WLKT09] and Barnes and Zhang [BZ17].

Patch-based synthesis in the real world, as described and performed

in this work, is characterized by fundamental constraints that are inherent to the task of parquetry and other forms of real-world collage. Any piece of input material can only be used once without being stained, scaled, stretched, copied, blended or filtered. Our synthesis algorithm therefore restricts itself to cutting operations and rigid transformations. More importantly, it must keep track of resource use in order to prevent source patches from colliding with each other. On the output side, the cuts must be fabricable, i.e., the individual fragments must be connected (no isolated pixels) and they may not expose too thin protruding structures. We are not aware of prior work that addressed these specific challenges.

Computational fabrication. Developments in the context of stylized fabrication [BCMP18] took benefit from the rapid progress in fabrication technology. In the context of 2D arts, the computational fabrication of paintings has been approached based on robotic arms to paint strokes for a given input image (e.g. [DLPT12, LPD13, TL12]). The fabrication of artistic renditions of images has been approached based on a computational non-photorealistic rendering pipeline, the generation of respective woodblocks and a final woodblock printing process [PPW18]. Further work addressed mosaic rendering using colored paper [GPSY06], where computational approaches have been used for tile generation and tile arrangement. This is followed by the respective generation of colored paper tiles and their arrangement according to the energy optimization.

Most works in computational fabrication aim at obtaining constant results despite possible variations in the material used. In contrast, we embrace the "personality" of the input and use it to create artworks that are inherently unique.

4.3 Method

The main objective of this work is the development of a computational pipeline for creating faithful renditions of a target image $\mathbf{I}_T$ from wood samples by exploiting the rich structures in wood as a source of detail. The pipeline devised in this work takes n_{samples} physical, wooden material samples and a target image as inputs and consists of three major steps: data acquisition, data analysis and cut pattern generation (i.e. tile generation, arrangement, and boundary shape optimization), and the final fabrication of the real-world counterpart (Figure 4.4).

In the first step, the wooden samples are prepared before they can be scanned with a flatbed scanner. This is followed by extracting local fea-

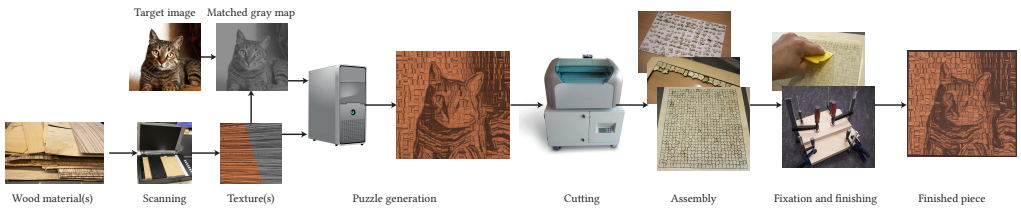


Figure 4.4: The proposed end-to-end pipeline for creating faithful renditions of target images based on exploiting the rich structure present in input wood samples as a source of detail. The involved major steps are data acquisition, cut pattern optimization and the final fabrication of the real-world rendition of the target image.

tures in the input images and by detecting corresponding patches between the source textures and the target image, yielding a stylized, digital wood parquetry of the target image. Finally, the patches are converted to cut instructions (taking into account that the cuts have to be fabricable by a laser cutter), specified pieces are cut with a laser cutter, and assembled to a physical sample of parquetry. We discuss details in the following sections.

4.3.1 Data acquisition

Before the scanning can be conducted, we first prepare the wood samples. Whereas thicker veneers can be utilized directly, standard veneers (0.6 mm to 0.8 mm thick) are glued to a substrate of 1.5 mm birch plywood in order to improve stability and minimize waviness. Especially burl veneers tend to be very brittle and assume strongly warped shapes; in contrast, the bending of the substrate is relatively easy to counter by screwing it to a rigid substrate. We enhance the contrast of the wood veneers (and consequently the contrast of the final parquetry) by sanding and applying a thin layer of clear coat or oil finish. After letting the finishing layer dry, the specimens are placed on a flatbed scanner and scanned at 300 dpi. The scans are aligned in order to get a common coordinate frame and a mask is generated to separate usable veneer from empty background and screw holes. We note that for larger scale productions, this step could easily be automated using machine vision techniques. The output of this step is a set of source textures $\mathbf{I}_S = \{\mathbf{I}_{S,i} : i \in \{1, \ldots, n_{\text{samples}}\}\}$, one for each physical wood sample.

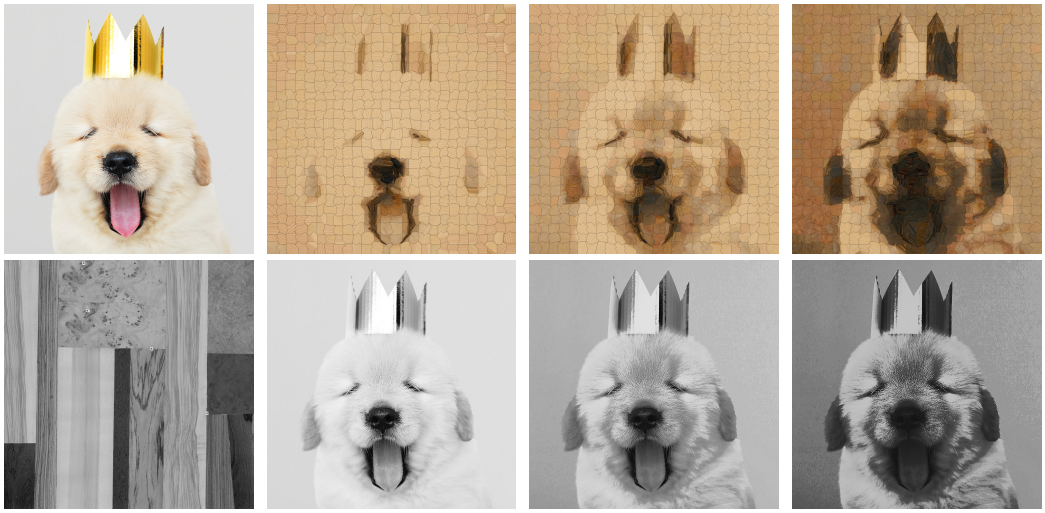


Figure 4.5: Top row: For a target image exhibiting low contrast and a bad foreground separation (left) the generated wood puzzle shows the same, undesirable effects when discarding histogram matching (middle left). In contrast, applying histogram matching (right) allows to exploit the whole wood texture gamut, which yields a high contrast at the cost of a strong change in appearance compared to the target. By interpolating between the intensity filter responses obtained with and without histogram matching, we generate a parquetry with medium contrast (middle right). Bottom row: The images depict the intensity filter response of a wooden veneer panel (left) and the respective responses obtained for the target image without histogram matching (middle left) and with histogram matching (right), as well as their interpolation (middle right).

4.3.2 Feature extraction

In order to find patch correspondences between the target image and the source images, we define a suitable representation for textural structures within the individual patches. We densely evaluate texture features using a filter bank consisting of 2 image filters, an intensity filter and a Sobel edge filter. We have experimented with higher-dimensional filter banks similar to the Leung–Malik filter bank [LM01] and found that the potential increase in reconstruction quality does not offset the additional computational cost induced by the higher-dimensional feature space. Applying the filter bank to an image $\mathbf{I}$ results in the 2-dimensional feature response maps

$$\mathbf{F}(\mathbf{I}) = \left(w_{\text{intens}} \cdot \mathbf{F}_{\text{intens}}(\mathbf{I}),\ w_{\text{edge}} \cdot \mathbf{F}_{\text{edge}}(\mathbf{I})\right)^{\top}, \tag{4.1}$$

where $\mathbf{F}_x$ are the particular image filters, $w_x \in [0,1]$ are the feature weights, $x \in \{\text{intens}, \text{edge}\}$. The weights allow artistic control over the

emphasis on overall intensity matching (w_{intens}) and fine scale gradient features (w_{edge}). Please note that this approach can easily be expanded to different feature vectors, allowing additional artistic control. We increase the probability of finding good matches by taking n_{rot} rotated versions of the wood source textures into account.

Source and target textures may exhibit highly different gamuts and filter response distributions, so we apply a histogram matching step in order to achieve a meaningful matching between target and source patches. We use a CDF-based histogram equalization [Rus02, ch.4] to transform the intensity distribution of the target image to that of the available source textures. As the wood samples generally span a smaller gamut than the target image, gamut mapping is of great importance to allow for the representation of the target image based on sampling the whole range of available wood patch intensities so that characteristic image structures can be emphasized. For challenging target images with low foreground contrast or bad foreground separation (Figure 4.5) we found that the histogram equalization tends to overshoot. We alleviate this by interpolating between equalized and original target intensity,

$$\mathbf{F}'_{\text{intens}}(\mathbf{I}_T) = (1 - w_{\text{hist}})\mathbf{F}_{\text{intens}}(\mathbf{I}_T) + w_{\text{hist}}\mathbf{F}_{\text{equalize}}(\mathbf{I}_T, \mathbf{I}_S), \tag{4.2}$$

where w_{hist} denotes the interpolation weight, $\mathbf{F}_{\text{equalize}}$ the histogram equalization operator, and $\mathbf{I}_S$ the set of all wood textures.

The output of this step is a set of $n_{\text{samples}} \cdot n_{\text{rot}}$ filter responses

$$\mathbf{F}(\mathbf{I}_S) = \left\{ \mathbf{F}(\mathbf{I}_{S,i,\phi_j}) : i \in \{1, \ldots, n_{\text{samples}}\}, j \in \{1, \ldots, n_{\text{rot}}\} \right\} \tag{4.3}$$

for the source textures, and one filter response $\mathbf{F}(\mathbf{I}_T)$ for the target image. We typically used $n_{\text{rot}} = 15$ source texture rotations for our experiments.

4.3.3 Cut pattern optimization

After evaluating the filter responses, the next step is to find corresponding patches between target image and source textures. To this end we divide the target image into a regular, axis-aligned grid of square patches that overlap by $1/4$ of their size with their respective neighbors. By choosing overlapping patches, we are able to align the cut pattern to a data term (Section 4.3.4), which in turn allows the cuts to follow image features. The patch size depends on the desired size and appearance of the fabricated output.

Given a target patch $\mathbf{P}_T \subset \mathbf{I}_T$ containing $n_P \times n_P$ pixels, we determine a corresponding source patch $\mathbf{P}_S$ by a dense template matching using the sum of squared differences,

$$\begin{aligned} D_{i,j}(x,y) &= \sum_{u,v=1}^{n_P} \Big(\mathbf{F}(\mathbf{P}_T(u,v)) - \mathbf{F}(\mathbf{I}_{S,i,\phi_j}(x+u,y+v))\Big)^2, \\ \mathbf{P}_S &= \arg\min_{i,j,x,y} D_{i,j}(x,y). \end{aligned} \tag{4.4}$$

Due to the decreasing number of available wood patches, the probability of finding good patch correspondences also decreases as the algorithm advances. For many classes of photos, e.g. portraits or pictures of animals, salient regions usually occur in the image center. To take this into account, we store the target patches $\mathbf{P}_T$ in a priority queue, sorted by their distance to the center of the photo. We avoid the multiple usage of already matched veneer sample regions by carrying along a binary mask for each source texture.

Target image regions with less salient features can be represented by larger patches. To exploit this, we implemented an adaptive patch matching step, where we subdivide a patch into four smaller patches if their associated cost is lower than the cost of the larger patch multiplied by a factor w_{adaptive}. The factor w_{adaptive} can be used to control the artistic balance between larger and smaller patches. We apply this step n_{adaptive} (typ. 0 to 2) times.

At the end of this step, we have covered the reconstructed image plane with partially overlapping square patches.

4.3.4 Dynamic programming

Arranging the previously matched patches according to the target image results in overlapping regions. We resolve these (non-fabricable) overlaps by finding optimal cuts according to the target image reproduction cost

$$\sum_{x,y} \left(\mathbf{F}(\mathbf{R}(x,y)) - \mathbf{F}(\mathbf{I}_T(x,y))\right)^2, \tag{4.5}$$

where $\mathbf{R}$ denotes the reconstructed wood parquetry image. For image regions with only two overlapping source patches, we obtain an optimum solution using dynamic programming. For details regarding the implementation of axis-aligned patch merging using dynamic programming see e.g. [EF01]. As we enforce our cuts to be guided by features in the target

image, the corresponding, local cost $c(x, y)$ for merging two horizontally neighboring patches $\mathbf{P}_{S,\{1,2\}}$ along pixel x is given by

$$c(x,y) = \sum_{x'=0}^{x-1} \left(\mathbf{F}(\mathbf{P}_{S,1}(x',y)) - \mathbf{F}(\mathbf{P}_T(x',y))\right)^2 + \sum_{x'=x}^{n-1} \left(\mathbf{F}(\mathbf{P}_{S,2}(x',y)) - \mathbf{F}(\mathbf{P}_T(x',y))\right)^2, \tag{4.6}$$

where $\mathbf{P}_T \subset \mathbf{I}_T$ and n is the size of the overlap. We assign patch $\mathbf{P}_{S,1}$ to the region left of the cut and $\mathbf{P}_{S,2}$ to the remaining region. By approaching this problem using dynamic programming, we enforce 6-connectivity of the cut and in turn physical fabricability. Vertically neighboring patches can be aligned in an analogous manner.

In regions where four patches overlap, we have to find two intersecting cuts, one for the horizontal and one for the vertical direction. This prevents cut optimization via dynamic programming. Instead, we find an approximate solution by alternating optimizations for one cut direction while keeping the other direction fixed. We experimentally observed two repetitions of this process to be sufficient.

In order to generate a representation that is laser-cuttable, we fit cubic Bézier curve segments to the cuts. The user can choose between G0 continuous and G1 continuous curve segments, or to skip this process entirely and generate axis-aligned cuts. Finally, the output of this step is a vector graphics file containing cut instructions which can be directly executed by the laser cutter.

4.3.5 Fabrication

In the next step, the optimized, still digital piece of parquetry is physically fabricated. To this end, we use a laser cutter for cutting the veneer boards from the back side and for engraving IDs which facilitate the identification of individual patches during their assembly. For other materials, this step could also be conducted using a CNC mill or a water jet cutter. The patches are separated from the rest of the veneer and laid out in a frame. To fix the patches, we attach a back plate using wood putty. After the putty has dried, we sand the veneers and finish them with clear coat or hard wax oil.

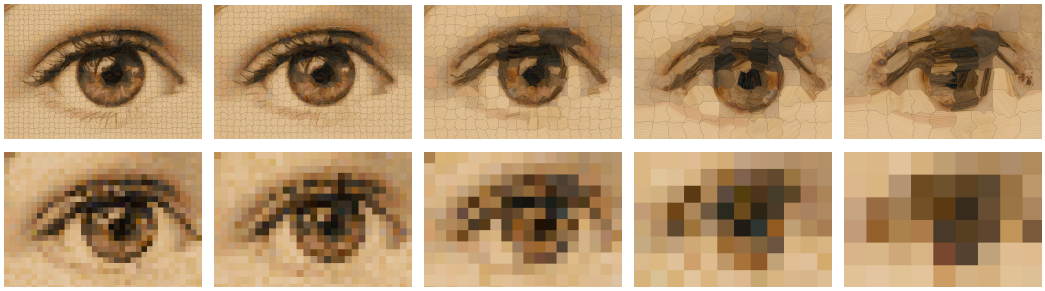


Figure 4.6: Effect of different resolutions on the reconstruction quality. We show reconstructions obtained with our framework (top row) and a "baseline" where high frequency features are removed and each patch is replaced by its mean color (bottom row). With decreasing resolution (from left to right), we observe that the structurally aware filters are important for reconstruction quality. The reconstruction quality obtained with our proposed technique gracefully declines with patch resolution and still produces visually pleasing results for very coarse patches.

4.3.6 Implementation details

The method was implemented in C++ using the OpenCV library [Bra00] and parallelized with OpenMP. Fitting a single patch typically takes 0.5 s to 3 s on an Intel Core i7-5820K CPU, where the runtime is dominated by template matching. Thus, the runtime primarily depends on the number of pixels per patch, and on the size of the wood samples tested.

During our experiments, we used a Plustek OpticPro A360 Plus flatbed scanner for A3-sized veneer boards, and a Cruse Synchron Table Scanner 4.0 for scanning larger panels. The fabrication (cutting) was performed on a Trotec Rayjet with a 12 W CO_2 laser and an Epilog Fusion 40 M2 engraver with a 75 W CO_2 laser.

4.4 Results

We begin our evaluation with the analysis of user-controllable design choices in the optimization, such as the effect of different energy terms, different sizes and shapes of the individual patches. This is followed by an ablation study, where we investigate the gradual decline in quality that occurs when repeatedly producing the same target image from the same wood veneer panel. We further demonstrate a few examples of fabricated parquetry obtained from different woods and under different conditions. Finally, we show the robustness of our method with respect to different

target images by presenting synthetic results for different targets, each optimized using the default parameter set.

Symbol	Parameter	Default
w_{intens}	Intensity priority weight	0.5
w_{edge}	Edge priority weight	0.5
w_{hist}	Histogram matching weight	0.5
s_{image}	Reconstructed image size (shorter axis)	360 mm
s_{patch}	Patch size	14 mm
n_{adaptive}	Adaptive patch levels	0
w_{adaptive}	Adaptive patch quality factor	1.2

Table 4.1: User-controllable stylization parameters and their default values.

4.4.1 User-controlled stylization

Our method allows the stylization of the generated renditions of target images based on user guidance. Before discussing the effect of individual user-controllable parameter choices on the style of the generated renditions, we first provide insights regarding the involved physical materials. We found an image of a human eye (Figure 4.10) to be a good target for quality assessment, because it contains features with different frequencies, as well as rounded structures. An overview over the user-controllable parameters related to stylization can be found in Table 4.1 and a more detailed description in Section 4.3.

Materials For the purpose of a better comparability, we generated synthetic renderings using the same scan of a wooden veneer panel as input for all results in this section (unless otherwise noted). The panel has a size of 1500 mm $\times$ 1000 mm and contains veneer samples from various wood types. The woods used in our experiments are not protected under CITES. They include maple burl, ash burl, poplar burl, buckeye burl, elm burl, birch burl, walnut burl, pine, wenge, santos rosewood, olive tree, makassar ebony, apple tree, and zebrawood. We sanded the panel and applied a layer of clear coat to enhance the contrast of the individual fiber strands. The physical sample was scanned at 300 dpi using a Cruse Synchron Table Scanner 4.0. A downscaled version of the scan can be found in Figure 4.7.

Histogram matching The target image gamut is generally larger than the gamut of the wood textures. Without taking this into account, the tem-

Figure 4.7: Scan of the wooden veneer panel used for the results in Section 4.4. The panel has physical dimensions of 1500 mm × 1000 mm and contains veneer samples from various wood types. The fiducial markers facilitate optical calibration on suitably equipped cutting systems.

plate matching step will generally draw patches from the gamut boundaries, which results in reproductions with high contrasts, but flat shading. By matching the target image histogram to the wood texture histogram, we compress the target image gamut to match the wood textures. This reduces the overall contrast, but puts more emphasis on shading nuances, see Figures 4.5 and 4.8. We found a simple interpolation between the matched and the unmatched input image to effectively improve contrast while preserving the original style of the image (Figure 4.5).

Patch size We evaluated the influence of the patch size on the style of the resulting target image renditions. Figure 4.6 shows rendered results for different patch sizes ranging from 7.7 mm to 31.0 mm. Our experiments suggest that patches with 5 mm edge length are the lower bound for physical producibility using our pipeline. Smaller patches could easily get lost and would be difficult to assemble. The reconstruction quality improves as the patch size decreases and approaches an almost photorealistic appearance for very small patches. In contrast, reconstructions with coarse

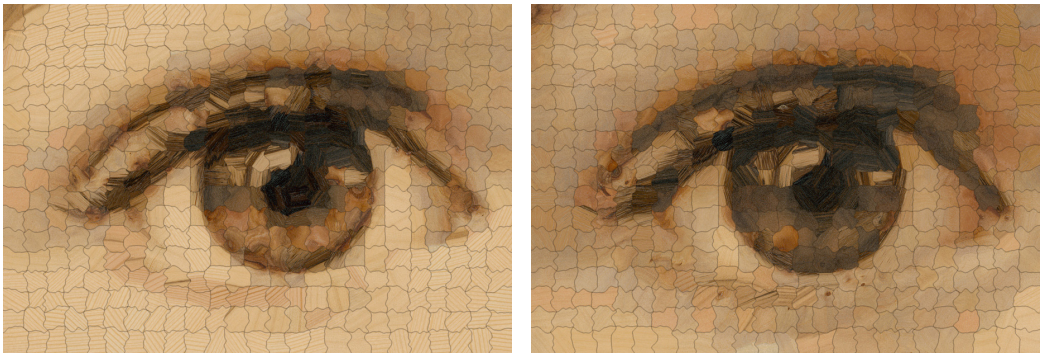


Figure 4.8: The effect of histogram matching. Without histogram matching ($w_{\text{hist}} = 0$, left), we obtain a higher contrast. With histogram matching ($w_{\text{hist}} = 1$, right), the contrast is reduced, but the shading appears less flat.

Figure 4.9: Effect of different adaptive reconstruction parameters. From left to right: $(n_{\text{adaptive}} = 1, w_{\text{adaptive}} = 1.2)$, $(n_{\text{adaptive}} = 2, w_{\text{adaptive}} = 1.2)$, $(n_{\text{adaptive}} = 1, w_{\text{adaptive}} = 1.5)$. As expected, high-frequency image structures are only touched for large values of w_{quality} (e.g., we accept a large decline in reconstruction quality). Nonetheless, we find the effect to be visually pleasing in all images and subject to personal preferences.

patch sizes exhibit a different, more sketch-like style.

As demonstrated in Figure 4.6, exploiting the structures inherent to the wooden materials greatly enhances the visual quality on all resolutions, thereby providing evidence for the effectiveness of our structurally aware template matching step. The perceived resolution of any image depends on the image size, resolution, and viewing distance. In order to give the reader an impression about the amount of additional perceived resolution introduced by the wood pixels, we include a comparison to a "baseline" that discards the wood structure and instead replaces each patch by its mean color.

Finally, we evaluate the effect of adaptive patch sizes in Figure 4.9. Analogous to adaptive grid methods, this allows us to reduce the total number of wood patches without sacrificing reconstruction quality. Regarding stylization, the larger patches result in an overall smoother ap-

Figure 4.10: Effect of intensity vs. edge filter. The highlighted zoom-ins depict the respective reconstructed regions for weights $(w_{\text{intens}}, w_{\text{edge}})$: $(1.0, 0.0)$, $(0.5, 0.5)$, and $(0.2, 0.8)$ from left to right. Using only intensity penalty enforces the stylization to match intensity. Structural details become increasingly well preserved with an increasing weight of the edge term.

pearance with fewer cuts.

Feature vector weights To analyze the effect of differently weighted feature vectors in the template matching step (Equation (4.4)) on the wood puzzle appearance, we show results obtained for various parameter choices in Figure 4.10. The obtained renditions for the highlighted regions of the eyelid (top row) and the iris (lower row) show that high weights for the intensity penalty w_{intens} enforce the matching regarding the intensity features. Finer structures, such as eyelashes, become better preserved by increasing the penalty w_{edge} on the edge filter responses.

Boundary shape optimization We also show the respective results before and after cut optimization. As demonstrated in Figure 4.11, the use of square patches on a regular grid results in a pixel-like rendition of the target image. Merging neighboring patches according to the data term

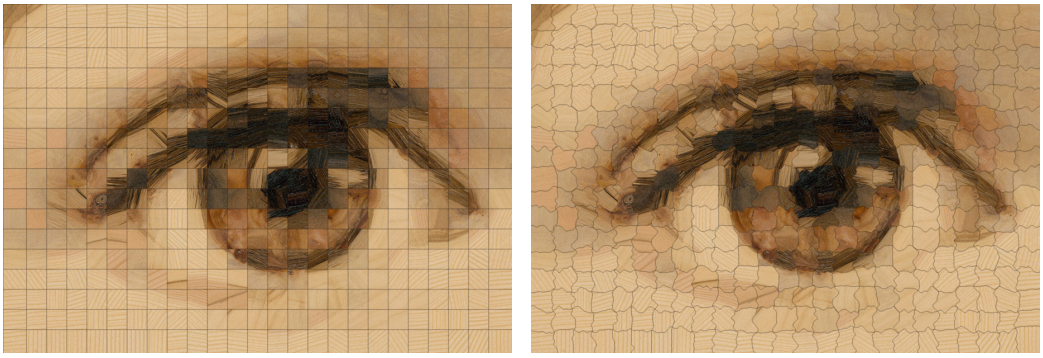


Figure 4.11: The effect of the boundary shape optimization using dynamic programming. Without dynamic programming (left), the generated rendition of the target image has a pixelized style. With dynamic programming (right), the cuts are optimized according to the underlying data term and the rendition exhibits a smoother, more organic style.

reduces the pixelation effect, thereby putting more emphasis onto the underlying image structures. We found that the representation of rounded, high-contrast image features specifically benefits from the dynamic programming step.

4.4.2 Ablation study

Our approach is inherently resource constrained. Thus we expect the reconstruction quality to scale with the area of available wood samples. To evaluate this effect, we applied our pipeline several times to generate renditions of the same target image under a decreasing availability (and quality) of source patches. The respective results are shown in Figure 4.12. We observe that the reconstruction quality decreases gracefully and the target image stays recognizable until the very last reconstruction. After the last reconstruction (partially) finished, there was no space left on the veneer panel that was large enough for another patch.

We noticed two types of degradation: intensity and high-frequency detail degradation. Most noticeable is the degradation in overall intensity matching after the panel runs out of dark patches (iteration 5). Less noticeable is the degradation of high-frequency content, e.g. around the eyes after iteration 3. These types of degradations could be alleviated by reconstructing target images with different intensity distributions or by "interlacing" the reconstruction runs.

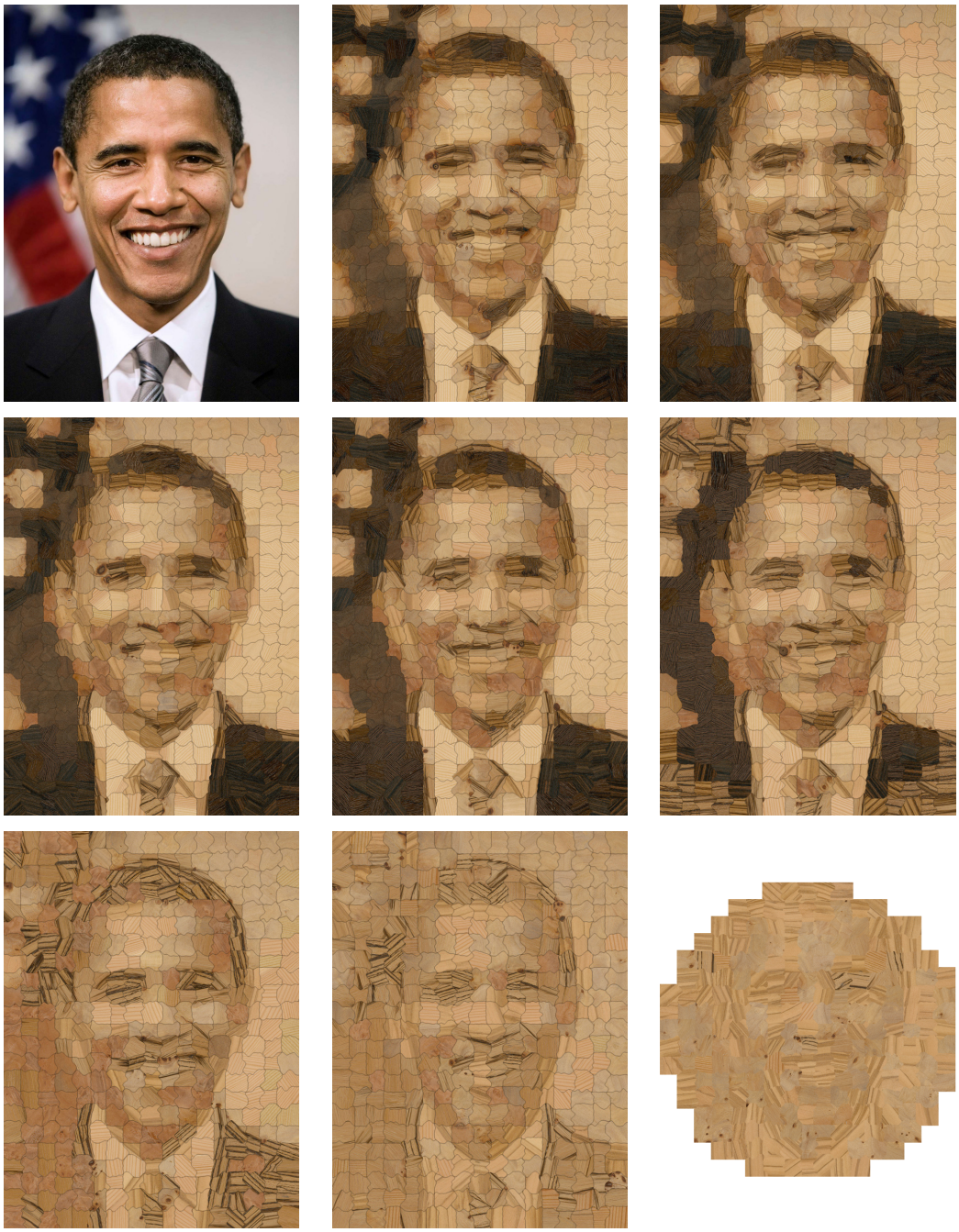


Figure 4.12: From left to right, top to bottom: target image, renditions of a target image generated under a decreasing amount, and quality, of available patches from a single wood sample. The last reconstruction did not complete because there were no patches left on the wood sample. Please zoom in to see image details.

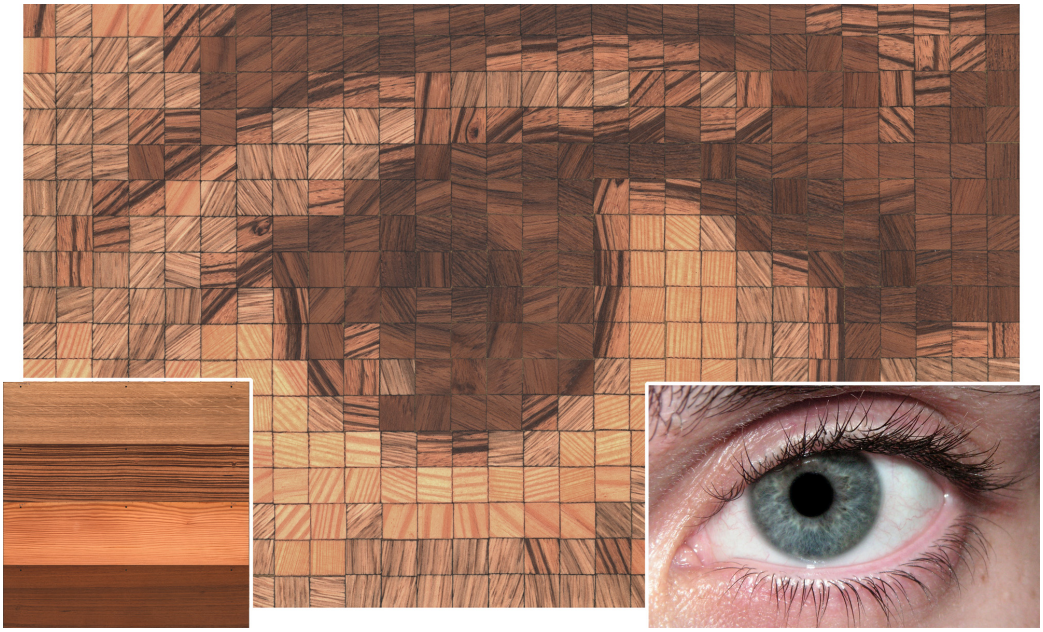


Figure 4.13: A fabricated piece of wood parquetry made from four different quarter-cut thick veneers (bottom left corner, from top to bottom: oak, zebrawood, fir, American walnut). The target image is a human eye (bottom right corner). The veneer puzzle consists of 28×17 wooden pixels and has a total size of approx. 28 cm $\times$ 17 cm.

4.4.3 Fabricated results

We present exemplary results of physically produced veneer puzzles in Figures 4.1, 4.13 and 4.14. The veneer puzzles in Figures 4.1 and 4.13 have been fabricated using multiple wood types. Since different wood types can differ vastly in color and grain structure, these results show a high contrast and perceived resolution. Fine details, such as hair, eyebrows, or eyelashes are faithfully reproduced.

The results in Figure 4.14 have each been produced using a different single wood type. The amount and quality of detail within a pixel is inherently limited to the features present in the original material. Woods with a limited feature gamut thus lead to a strongly stylized outcome, which we imagine could also be utilized as an artistic tool.

We decided to finish most of the pieces using hard wax oil in order to accomplish a natural look. A clear coat finish (Figure 4.14, right) results in a highly specular appearance.

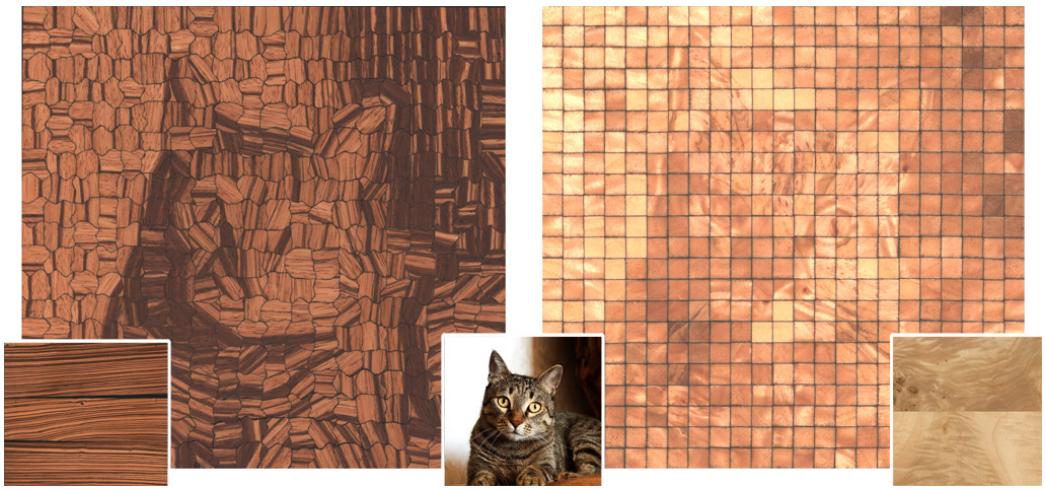


Figure 4.14: Exemplary results of fabricated parquetries using the same target image (bottom center), but different wood types and finish. The left image was fabricated using zebrawood with an oil finish. The right image was produced using poplar burl veneer with clear coating, resulting in a highly specular appearance with limited contrast. The samples consist of 20×19 and 23×22 wooden pixels respectively and their physical dimensions are about $15\,\text{cm} \times 15\,\text{cm}$. The left puzzle has optimized patch boundaries, the right puzzle consists of square patches.

With row/column labels engraved on the back side, it takes about 1 h to 2 h for a single person to assemble a 500-piece parquetry inside a suitably dimensioned frame. Although somewhat repetitive, the authors found this activity to be satisfying and relaxing. For thin veneers that are laminated onto a plywood substrate, the final image remains hidden until the finished composition is turned around.

4.4.4 Synthetic results

In addition to the evaluation of different parameter choices, we show renditions for several target images depicting portraits and animals in Figure 4.15. To demonstrate the robustness of our approach with respect to different target images, each of these results has been produced using the default parameters shown in Table 4.1. The depicted results demonstrate the potential of computational parquetry for fine arts. Portraits and animal pictures can be easily recognized as their characteristic appearance is preserved in the stylized result. Please see the supplemental material for additional results.

Figure 4.15: Exemplary synthetic renditions of portraits and animals. Each of these results has been composed using the veneer sample panel shown in Figure 4.7 and the default parameters listed in Table 4.1. Our algorithm is able to handle a wide range of input including color photographs, black and white photographs, drawings, and paintings. The images show, from left to right, top to bottom: Grace Hopper, Eileen Collins, Felix Hausdorff, Katherine Johnson, Ludwig van Beethoven, Whoopi Goldberg, Hedy Lamarr, Alan Turing, a piglet, a penguin, a Corgi, and a flamingo.

4.5 Discussion and future work

A practical drawback of our method is that it requires a surface finish to be applied to the wood two times, once before scanning and then again after the final assembly of the finished puzzle. The first application is important, since this step changes the appearance of the wood samples significantly. For the algorithm, it is crucial to choose suitable patches based on their final appearance. We apply the sanding/finishing procedure a second time in order to flatten out small height variations, which are inevitable after puzzling. For a large-scale, automatic production of custom, wooden parquetry puzzles, we would like to minimize the amount of manual interaction. Thus, we conducted initial experiments on training a model to predict the change of appearance from unfinished to finished veneers. Using these predictions, it might become possible to defer the application of surface finish until after the final puzzle has been assembled. To this end, we trained a U-Net [RFB15] on image pairs before and after applying the finish. Based on the preliminary results in Figure 4.16, we believe that this would be a good direction for future work.

Our approach allowed us to produce visually pleasing pieces of wood parquetry, even without having a professional wood-working background. However, we expect that certain technical imprecisions (such as sub-perfectly applied clear coating) would be mitigated with more experience. Also, we expect that cut clearances and discolorations will be improved with further fine tuning of the cutting equipment.

Here, we treat wood as being a diffuse reflector and ignore any directional effects. Real wood exhibits anisotropic BRDF characteristics, which means that rotation of a part could be used to modulate its intensity. This might also enable the generation of new types of puzzles, where a hidden pattern is revealed by the right permutation and rotation of some parts.

In our experiments, we restricted ourselves to fabricating parquetry based on wood veneers, since they are commonly available and can be cut using a laser cutter. Generally, our pipeline is not restricted to this type of material. Using a water jet cutter, other materials like marble or brushed metal could be processed as well. The process could also be extended to multi-material parquetry.

Parquetry generation is inherently resource-constrained and in the scope of our work, the amount of available source samples was limited. Having access to a larger database of veneers (either by increasing the number of samples per wood type, or by introducing new wood types) would certainly improve the reconstruction quality. However, since this

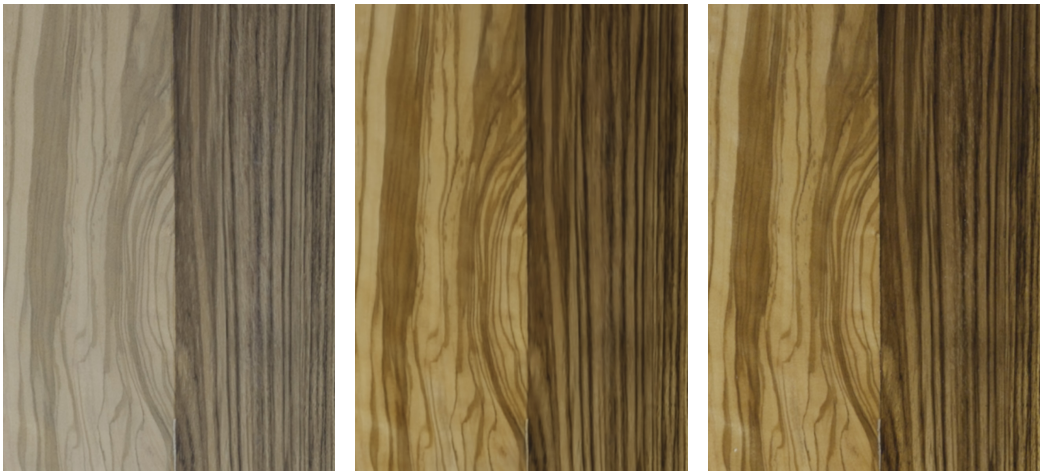


Figure 4.16: We envision using deep learning to predict the change in surface finish induced by a layer of oil or clear coat. Being able to do so would alleviate the need for a pre-finishing step prior to texture acquisition. From left to right: input image, surface finish appearance predicted by our preliminary model, ground truth image.

is an artistic process reaching the highest reconstruction quality might not always be the goal. Using only a single type of wood, or a selection of wood samples with a particular structure, can lead to equally interesting and fascinating results, see e.g. Figure 4.14.

When preparing our puzzle for assembly as a game, various degrees of difficulty could be imagined. As all pieces are made from wood, semantic labels are not immediately accessible as they sometimes are in regular puzzles (water, buildings, skin, foliage, sky/clouds, etc.). Given a bag of identically-shaped (square) pieces, it would seem extremely challenging to arrive at the one "correct" solution; at the same time, there would be numerous mechanically valid "approximate" solutions, or permutations between sets of similar-looking parts. Here, the cuts generated by the dynamic programming step offer a welcome cue for assembly, as they cause adjacent pieces to snap into place.

4.6 Conclusions

We approached the fabrication of structure-aware parquetry based on a novel end-to-end pipeline that takes wood samples and a target image as inputs and generates a cut pattern for parquetry puzzles. To the best of our knowledge, there is no prior work that addresses the challenges inherent

to the task of producing a physical sample of wood parquetry using commodity hardware from minimal input (a target image). The challenges include the single use of individual pieces of input material without being deformed, scaled, blended, or filtered, as well as keeping track of resource use in order to prevent source patches from colliding with each other, while still faithfully reproducing the target image. Practical aspects regarding the fabricability have also been taken into account. The varying structural details within the wood samples lead to unique and fascinating artworks, and the design of the overall process allows even users without a particular woodworking background to experience producing pieces of this new type of art.

CHAPTER 5

Conclusion

This final chapter provides an outlook on future work, as well as a discussion of the results presented in the scope of this thesis. Here, we present a general discussion of the cumulative thesis as a whole. For an individual discussion of each publication, please see Sections 2.7, 3.6 and 4.5 respectively.

5.1 Limitations and future work

The methods presented in Chapters 2 and 3 are analysis-by-synthesis methods and their forward models rely on a deep understanding of the underlying scene structure and light propagation. Only by carefully optimizing the forward models to their respective scene setups, we were able to achieve evaluation runtimes that were fast enough to utilize these models in the inner loop of our optimization schemes. Naturally, this restricts our methods to their specific scene setups. The method presented in Chapter 2 is restricted to a horizontal window with sessile water drops on it and the algorithm in Chapter 3 is specialized to scenes with exactly three light bounces. However, these methods essentially consist of two parts: a forward model and a global optimization scheme. Therefore, it is possible to apply our optimization approaches to different scenes by modifying the forward model. For future work we see two different directions. First, we would like to improve the quality of the forward models in order to improve the overall accuracy. In the context of non-line-of-sight reconstruction, in addition to the simulation of the physical light transport, we would like to also account for the response of the acquisition hardware (e.g. SPAD sensor). The restriction to three-bounce light transport could be

alleviated as soon as real-time rendering methods for global light transport improve, e.g. by neural rendering techniques. Second, we would like to apply our method to new settings. By changing the direction of the gravitational term in the water drop simulation, we would be able to capture light fields from inclined surfaces. Our optimization approach could also be adapted to much larger scales. By modeling the windows of skyscrapers, they could be turned into extremely wide-baseline light field transformers, where each window forms a separate reflective lens. Similarly, for the computational parquetry method from Chapter 4, we would like to investigate the eligibility of other materials, like stone, marble, metal, or cardboard for the generation of puzzles. This might pose further restrictions to the space of available image operations for optimization and require different hardware for cutting.

Current generations of smart phones are packed with sensors, containing up to four cameras (future generations might contain even more [DPR]), time of flight ranging sensors, inertial sensors, GPS, and more. The extensive sensory equipment is complemented by strong processors and high connectivity. This feature set and their ever-growing ubiquity would make current and future-generation smartphones a canonical platform for multiple directions of future research.

First, it would be exciting to prepare our approaches to be fully *casual* and hand-held, which involves multiple challenges. Especially the light field imaging and computational parquetry problems involve a calibrated camera with known pose. In a casual setting and exploiting the inertial measurement unit, this could be solved using visual-inertial simultaneous localization and mapping (VI-SLAM) [MT17, vSUC18], structure from motion (SfM) [SF16], or marker-based pose estimation. Regarding non-line-of-sight geometry reconstruction, current smartphone generations already contain time of flight ranging sensors based on SPADs, which could be utilized for transient imaging once an API is offered that allows access to the raw intensity histograms. Most likely, due to the low resolution and baseline, multiple measurements from different positions would have to be acquired, either using a single or multiple phones.

Second, the computational complexity of the non-linear, non-convex optimization methods from Chapters 2 and 3 is rather high and needs to be reduced. Even though the transient rendering in Chapter 3 is highly efficient and allows real-time frame rates, over the whole course of an optimization, hundreds of thousands or even millions of candidate renderings are required in order to evaluate the cost function and the corresponding Jacobian. This high number originates from the global optimization scheme which requires solving multiple non-linear least squares

sub-problems per iteration and is necessitated by the non-convexity of the problem. This leads to reconstruction times ranging from several minutes to more than a day. Even though the computation could be conducted offline on a server, the long waiting time is still undesired. There are multiple ways of optimizing the performance of the methods. The forward models could be further optimized, the gradients could be evaluated using automatic differentiation on the GPU, and the underlying non-linear solvers could be further tuned. While these measures would undoubtedly improve the overall performance, it is unlikely that they would suffice to reach near-real-time performance. The performance gains are limited by the high-dimensional nature of the problems, which induces the requirement for a large number of evaluations during a complete optimization run. As an alternative approach for further performance optimization, we have noted that current and future smartphone generations do, and will likely continue to employ specialized application-specific integrated circuits (ASICs) for efficient, real-time neural network inference, like the Apple Neural Engine [App] or the Google Edge Tensor Processing Unit [Goo]. By design, our forward models are consistent with the underlying physical processes and it would be interesting to investigate their applicability for training DNNs using synthetic data. Even though our problems currently prohibit the acquisition of real, ground-truth data for supervised training, our models could still enable us to apply learning-based methods. Thus we would shift the computational complexity from inference to training and could enable up to real-time inference on smartphones. The optimization method used for generating the computational parquetry puzzles does not suffer from the aforementioned performance problems. Here, the runtime is dominated by the dense feature matching, which could be accelerated using methods based on sparse patch matching [PTSF19] or pyramid matching.

Third, it would be highly exciting to invest in a common *casual computational imaging framework*, that runs on smartphones, combines the many methods targeted on revealing invisible information from images, and utilizes the smartphone's additional sensor equipment. Combining a diverse set of inspiring casual computational imaging methods that utilize scene features like water drops, shadows, eyes, and other reflecting surfaces in a single app could prove useful for education, entertainment, and benchmarking. By providing a common set of low-level image processing algorithms, such a framework could also facilitate future research on casual computational imaging, similar to the role of OpenCV [Bra00] for computer vision or the robot operating system (ROS) [QGC+09] for robotics. Light fields could be acquired using methods based on our own work on

translucent, accidental optics [WIG+15, IGP+17], or based on shadows in the scene [BYY+18]. Other methods could amplify unnoticeable motions and color changes in videos [XRW+14, WRS+12, WRDF13, OJK+18]. Non-line-of-sight imaging and geometry reconstruction could be conducted based on transient imaging [IH18, AGJ17], occluders [YBT+19], pinspecks [SMBG19], or even eyes [NN06]. Environment maps could be extracted by analyzing reflections from eyes [NN04] and other non-Lambertian objects [GRR+17]. In order to combine this wide range of algorithms in a meaningful way, it would be beneficial to train a classifier that automatically detects exploitable scene features and presents a list of algorithms that could be applied. Furthermore, it would be interesting to combine multiple scene features and methods to refine the results, or even to create completely new results.

5.2 Discussion and Outlook

In this thesis, we presented three methods to extract invisible information from multiple types of generalized, challenging image data. By applying scene-specific domain knowledge, we were able to augment, or even completely replace hardware and optical design with simulation and optimization in order to recover three different modalities. In Chapter 2, we reconstructed light fields from water drops, in Chapter 3, we extracted geometries without a direct line-of-sight, and in Chapter 4, we fabricated physical renditions of arbitrary target images using wooden veneer. Guided by the physics of the underlying light transport, we were able to formulate the respective reconstructions as optimization problems with physically-based forward models. In order to describe the light transport and to develop the models, we have utilized results from a wide range of disciplines, including computer graphics, optical physics, fluid dynamics, and numerics. Since each of our approaches targets a highly different scene, we have developed three specialized global optimization schemes based on non-linear and discrete optimization. With each of our publications, we were able to either solve previously open challenges, or to improve the existing state-of-the-art. On a higher level, we have exemplified the viability of optimization methods based on physically motivated forward models and computer graphics for processing challenging input data. We consider our methods to be basic research on image processing of incidental and uncontrolled data and hope that our work inspires future inter-disciplinary research on searching, finding, and extracting hidden information from images. We believe that our physically-based optimiza-

tion approach, translated to further fields of research, such as life sciences, environmental sciences, chemistry, or other fields of physics, could lead to more useful and surprising results.

We have deliberately developed our approaches to make them available to the widest possible user base. To this end, we have implemented our methods on commonly available and inexpensive hardware. Instead of using camera arrays or specialized optics, we have shown that a light field can be acquired using a single photograph of a window with water drops on it. This approach uses a conventional digital camera and does not require any specific lab equipment at all, making it well-suited for future casual and hand-held applications. Our approach to non-line-of-sight reconstruction requires additional transient imaging hardware. However, it has been shown that fairly inexpensive time of flight cameras based on photonic mixer devices can be used for transient imaging [HHGH13, KWB+13]. Also, recent miniature time of flight ranging sensors like the STMicroelectronics VL53L family have been implemented using SPAD technology and development boards are widely available at low-cost. Our method for generating computational parquetry puzzles naturally requires a laser cutter for fabricating the puzzles. However, there are no special requirements to the laser cutter and even basic models are sufficient. For the acquisition of the source textures, no special hardware other than a digital camera or a flatbed scanner is required. Together, this makes the approach feasible for enthusiast home users and hacker spaces. Implementing the parquetry puzzle generation as a web service would give virtually anybody access to this new fine art experience, since only a target image has to be provided. The cutting step would be carried out by the service provider.

The amount of invisible information that we are able to draw from images is quite surprising and could be used to illustrate basic principles in optical physics, fluid dynamics, optimization, engineering, and the nature of light propagation. We would be thrilled to see comprehensible basic research like ours, that is reproducible by a large group of users, to be used as an educational tool to excite young people (or any people for that matter) for the STEM fields of science, technology, engineering and mathematics.

Bibliography

[A+16] B. P. Abbott et al. Observation of gravitational waves from a binary black hole merger. *Physical Review Letters*, 116:061102, Feb 2016.

[AB91] E. H. Adelson and J. R. Bergen. The plenoptic function and the elements of early vision. *Computational Models of Visual Processing*, 1(2), 1991.

[Abr78] N. Abramson. Light-in-flight recording by holography. *Optics Letters*, 3(4):121–123, October 1978.

[AG97] A. W. Adamson and A. P. Gast. *Physical Chemistry of Surfaces*. Wiley, 1997.

[AGJ17] V. Arellano, D. Gutierrez, and A. Jarabo. Fast back-projection for non-line of sight reconstruction. *Optics Express*, 25(10), 2017.

[AKH+18] N. Antipa, G. Kuo, R. Heckel, B. Mildenhall, E. Bostan, R. Ng, and L. Waller. Diffusercam: lensless single-exposure 3D imaging. *Optica*, 5(1):1–9, Jan 2018.

[All05] T. C. Allbutt. The historical relations of medicine and surgery to the end of the sixteenth century. In *Congress of Arts and Science*, St. Louis, 1905.

[AMO15] S. Agarwal, K. Mierle, and Others. Ceres solver. `http://ceres-solver.org`, 2015.

[ANNW16] N. Antipa, S. Necula, R. Ng, and L. Waller. Single-shot diffuser-encoded light field imaging. In *IEEE International Conference on Computational Photography (ICCP)*, 2016.

[AOB+19] N. Antipa, P. Oare, E. Bostan, R. Ng, and L. Waller. Video from stills: Lensless imaging with rolling shutter. In *IEEE International Conference on Computational Photography (ICCP)*, 2019.

[App] Apple Inc. Apple a12 bionic. `https://www.apple.com/iphone-xs/a12-bionic/`. Accessed on 18 July 2019.

[Ash01] M. Ashikhmin. Synthesizing natural textures. In *ACM SIGGRAPH Symposium on Interactive 3D Graphics (I3D)*, pages 217–226, New York, NY, USA, 2001. ACM.

[BBFG06] S. Battiato, G. D. Blasi, G. M. Farinella, and G. Gallo. A Survey of Digital Mosaic Techniques. In *Eurographics Italian Chapter Conference*. The Eurographics Association, 2006.

[BBFG07] S. Battiato, G. D. Blasi, G. M. Farinella, and G. Gallo. Digital mosaic frameworks - an overview. *Computer Graphics Forum*, 26(4):794–812, 2007.

[BCMP18] B. Bickel, P. Cignoni, L. Malomo, and N. Pietroni. State of the art on stylized fabrication. *Computer Graphics Forum*, 37(6):325–342, 2018.

[BGP05] G. D. Blasi, G. Gallo, and M. Petralia. Puzzle image mosaic. *Proc. IASTED/VIIP2005*, 2005.

[BGSF11] C. Barnes, D. B. Goldman, E. Shechtman, and A. Finkelstein. The patchmatch randomized matching algorithm for image manipulation. *Communications of the ACM*, 54(11):103–110, 2011.

[BLK18] J. Boger-Lombard and O. Katz. Non line-of-sight localization by passive optical time-of-flight. arXiv:1808.01000v1, 2018.

[BMP12] S. Battiato, A. Milone, and G. Puglisi. Artificial mosaics with irregular tiles based on gradient vector flow. In *ECCV 2012 Workshops and Demonstrations*, pages 581–588, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[BNK10] P. C. Barnum, S. G. Narasimhan, and T. Kanade. A multi-layered display with water drops. *ACM Transactions on Graphics (TOG)*, 29(4):76:1–76:7, 2010.

[Bou04] J.-Y. Bouguet. Camera calibration toolbox for MATLAB, 2004.

[BP05] G. D. Blasi and M. Petralia. Fast Photomosaic. In *Poster Proc. of WSCG*, 2005.

[Bra92] K. A. Brakke. The surface evolver. *Experimental Mathematics*, 1(2):141–165, 1992.

[Bra00] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[Bra13] K. A. Brakke. Surface Evolver 2.70, 2013. `http://facstaff.susqu.edu/brakke/evolver/evolver.html`.

[BSFG09] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (TOG)*, 28(3):24:1–24:11, 2009.

[BSGF10] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein. The generalized patchmatch correspondence algorithm. In *European Conference on Computer Vision (ECCV)*, pages 29–43, Berlin, Heidelberg, 2010. Springer-Verlag.

[BSH+17] N. Bedard, T. Shope, A. Hoberman, M. A. Haralam, N. Shaikh, J. Kovačević, N. Balram, and I. Tošić. Light field otoscope design for 3D in vivo imaging of the middle ear. *Biomedical Optics Express*, 8(1):260–272, Jan 2017.

[BYY+17] K. L. Bouman, V. Ye, A. B. Yedidia, F. Durand, G. W. Wornell, A. Torralba, and W. T. Freeman. Turning corners into cameras: Principles and methods. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2270–2278, 2017.

[BYY+18] M. Baradad, V. Ye, A. B. Yedidia, F. Durand, W. T. Freeman, G. W. Wornell, and A. Torralba. Inferring light fields from shadows. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[BZ17] C. Barnes and F.-L. Zhang. A survey of the state-of-the-art in patch-based synthesis. *Computational Visual Media*, 3(1):3–20, 2017.

[BZT+15] M. Buttafava, J. Zeman, A. Tosi, K. Eliceiri, and A. Velten. Non-line-of-sight imaging using a time-gated single photon avalanche diode. *Optics Express*, 23(16):20997–21011, 2015.

[CBC+01] J. Carr, R. Beatson, J. Cherrie, T. Mitchell, W. Fright, B. McCallum, and T. Evans. Reconstruction and representation of 3D objects with radial basis functions. In *Proc. 28th Annual Conf. on Computer Graphics and Interactive Techniques*, pages 67–76. ACM, 2001.

[CKIW15] Z. Chen, B. Kim, D. Ito, and H. Wang. Wetbrush: Gpu-based 3D painting simulation at the bristle level. *ACM Transactions on Graphics (TOG)*, 34(6):200:1–200:11, 2015.

[CLJL03] N. Chronis, G. L. Liu, K.-H. Jeong, and L. P. Lee. Tunable liquid-filled microlens array integrated with microfluidic network. *Optics Express*, 11(19):2370–2378, September 2003.

[CMK+14] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[CTCS00] J.-X. Chai, X. Tong, S.-C. Chan, and H.-Y. Shum. Plenoptic sampling. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 307–318, 2000.

[CY95] C. Close and J. Yau. *Chuck Close: Recent Paintings*. Pace Wildenstein, New York, 1995.

[Dal91] S. Dalí. *The Salvador Dalí Museum Collection*. Bulfinch Press, Boston, 1991.

[Deb98] P. Debevec. Light probe image gallery, 1998. `http://www.pauldebevec.com/Probes/`.

[DIIM04] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the Twentieth Annual Symposium on Computational Geometry*, pages 253–262, New York, NY, USA, 2004. ACM.

[DLD12] A. Davis, M. Levoy, and F. Durand. Unstructured light fields. *Computer Graphics Forum*, 31(2):305–314, May 2012.

[DLPT12] O. Deussen, T. Lindemeier, S. Pirk, and M. Tautzenberger. Feedback-guided stroke placement for a painting machine. In *Proceedings of the Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging*, pages 25–33, Goslar Germany, Germany, 2012. Eurographics Association.

[DPR] DPReview. Light and sony team up to make the next generation of multi-camera smartphones. `https://www.dpreview.com/news/1460201356/light-announces-partnership-with-sony-to-pair-its-computational-tech-designs-with-sony-s-sensors`. Accessed on 17 July 2019.

[DRW+14] A. Davis, M. Rubinstein, N. Wadhwa, G. Mysore, F. Durand, and W. T. Freeman. The visual microphone: Passive recovery of sound from video. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 33(4):79:1–79:10, 2014.

[DSB+12] S. Darabi, E. Shechtman, C. Barnes, D. B. Goldman, and P. Sen. Image melding: Combining inconsistent images using patch-based synthesis. *ACM Transactions on Graphics (TOG)*, 31(4):82:1–82:10, 2012.

[EF01] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, pages 341–346, 2001.

[EKF13] D. Eigen, D. Krishnan, and R. Fergus. Restoring an image taken through a window covered with dirt or rain. In *IEEE International Conference on Computer Vision (ICCV)*, pages 633–640. IEEE, 2013.

[EL99] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1033–1038, 1999.

[ERKD09] R. Erni, M. D. Rossell, C. Kisielowski, and U. Dahmen. Atomic-resolution imaging with a sub-50-pm electron probe. *Physical Review Letters*, 102:096101, Mar 2009.

[EW03] G. Elber and G. Wolberg. Rendering traditional mosaics. *The Visual Computer*, 19(1):67–78, 2003.

[Fau] C. Faulkner. A peek inside the huawei p30 pro's periscope lens shows off its clever zoom. `https://www.theverge.com/2019/4/22/18511229/huawei-p30-periscope-lens-teardown-clever-zoom-camera`. Accessed on 24 July 2019.

[FDA01] R. W. Fleming, R. O. Dror, and E. H. Adelson. How do humans determine reflectance properties under unknown illumination? In *Proceedings of the IEEE Workshop on Identifying Objects Across Variations in Lighting: Psychophysics & Computation. Colocated with CVPR 2001*, 2001.

[FGWM18] M. Feng, S. Z. Gilani, Y. Wang, and A. Mian. 3D face reconstruction from light field images: A model-free approach. In *European Conference on Computer Vision (ECCV)*, pages 508–526, 2018.

[FKR13] M. Fuchs, M. Kächele, and S. Rusinkiewicz. Design and fabrication of faceted mirror arrays for light field capture. *Computer Graphics Forum*, 32(8):246–257, 2013.

[FR98] A. Finkelstein and M. Range. Image mosaics. In *Proceedings of the International Conference on Electronic Publishing*, pages 11–22, London, UK, UK, 1998. Springer-Verlag.

[FTF06] R. Fergus, A. Torralba, and W. Freeman. *Random lens imaging*. MIT CSAIL Technical Report 2006-058, 2006.

[Fuc10] S. Fuchs. Multipath interference compensation in time-of-flight camera images. In *20th International Conference on Pattern Recognition (ICPR)*, pages 3583–3586. IEEE, 2010.

[Gar29] F. H. Garrison. *An Introduction to the History of Medicine with Medical Chronology, Suggestions for Study and Bibliographic Data*. W. B. Saunders Company, Philadelphia, 4th edition, 1929.

[GAVN11] M. Gupta, A. Agrawal, A. Veeraraghavan, and S. G. Narasimhan. Structured light 3D scanning in the presence of global illumination. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[GBWQ04] P.-G. D. Gennes, F. Brochard-Wyart, and D. Quéré. *Capillarity and wetting phenomena: drops, bubbles, pearls, waves*. Springer Science & Business Media, 2004.

[GCB+17] M. Gharbi, J. Chen, J. T. Barron, S. W. Hasinoff, and F. Durand. Deep bilateral learning for real-time image enhancement. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 36(4):118, 2017.

[GEB15] L. A. Gatys, A. S. Ecker, and M. Bethge. Texture synthesis using convolutional neural networks. In *Proc. of the International Conference on Neural Information Processing Systems*, pages 262–270, 2015.

[GEB16] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, 2016.

[GGSC96] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *Proc. 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 43–54, New York, NY, USA, 1996. ACM.

[GKH+15] G. Gariepy, N. Krstajic, R. Henderson, C. Li, R. R. Thomson, G. S. Buller, B. Heshmat, R. Raskar, J. Leach, and D. Faccio. Single-photon sensitive light-in-flight imaging. *Nature Communications*, 6, 2015.

[Goo] Google LLC. Google edge tpu. `https://cloud.google.com/edge-tpu/`. Accessed on 18 July 2019.

[GPSY06] Y. J. Gi, Y. S. Park, S. H. Seo, and K. H. Yoon. Mosaic rendering using colored paper. In *Proceedings of the International Conference on Virtual Reality, Archaeology and Intelligent Cultural Heritage (VAST)*, pages 25–30, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.

[GRA+11] A. Gulinatti, I. Rech, M. Assanelli, M. Ghioni, and S. Cova. A physically based model for evaluating the photon detection efficiency and the temporal response of SPAD detectors. *Journal of Modern Optics*, 58(3-4):210–224, 2011.

[GRC+10] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman. Geodesic star convexity for interactive image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3129–3136. IEEE, 2010.

[GRR+17] S. Georgoulis, K. Rematas, T. Ritschel, M. Fritz, T. Tuytelaars, and L. Van Gool. What is around the camera? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5170–5178, 2017.

[GSC12] B. Goldlücke, E. Strekalovskiy, and D. Cremers. The natural vectorial total variation which arises from geometric measure theory. *SIAM Journal on Imaging Sciences*, 5(2):537–563, 2012.

[GTH+16] G. Gariepy, F. Tonolini, R. Henderson, J. Leach, and D. Faccio. Detection and tracking of moving objects hidden from view. *Nature Photonics*, 10(1), 2016.

[GZB+13] I. Gkioulekas, S. Zhao, K. Bala, T. Zickler, and A. Levin. Inverse volume rendering with material dictionaries. *ACM Transactions on Graphics (TOG)*, 32(6):162, 2013.

[GZC+06] T. Georgiev, K. C. Zheng, B. Curless, D. Salesin, S. Nayar, and C. Intwala. Spatio-angular resolution tradeoffs in integral photography. In *Eurographics Symposium on Rendering (EGSR)*, pages 263–272. Eurographics Association, 2006.

[Hau01] A. Hausner. Simulating decorative mosaics. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques*, pages 573–580, New York, NY, USA, 2001. ACM.

[HBC+98] P. Hickson, E. F. Borra, R. Cabanac, S. C. Chapman, V. D. Lapparent, M. Mulrooney, and G. A. H. Walker. Large zenith telescope project: a 6-m mercury-mirror telescope. In *Astronomical Telescopes & Instrumentation*, pages 226–232. International Society for Optics and Photonics, 1998.

[HFI+08] M. B. Hullin, M. Fuchs, I. Ihrke, H.-P. Seidel, and H. P. A. Lensch. Fluorescent immersion range scanning. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 27(3):87:1–87:10, August 2008.

[HGJ17] Q. Hernandez, D. Gutierrez, and A. Jarabo. A computational model of a single-photon avalanche diode sensor for transient imaging. arXiv:1703.02635, 2017.

[HHGH13] F. Heide, M. B. Hullin, J. Gregson, and W. Heidrich. Low-budget transient imaging using photonic mixer devices. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 32(4):45:1–45:10, 2013.

[HJKG16] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke. A dataset and evaluation methodology for depth estimation on 4D light fields. In *Asian Conference on Computer Vision (ACCV)*. Springer, 2016.

[HJO+01] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques*, pages 327–340, New York, NY, USA, 2001. ACM.

[HLGF11] S. W. Hasinoff, A. Levin, P. R. Goode, and W. T. Freeman. Diffuse reflectance imaging with astronomical applications. In *2011 International Conference on Computer Vision*, pages 185–192. IEEE, 2011.

[HLR+11] M. B. Hullin, H. P. A. Lensch, R. Raskar, H.-P. Seidel, and I. Ihrke. Dynamic display of BRDFs. In O. Deussen and M. Chen, editors, *Computer Graphics Forum (Proc. EUROGRAPHICS)*, pages 475–483, Llandudno, UK, 2011. Eurographics, Blackwell.

[HOZ+17] F. Heide, M. O'Toole, K. Zhang, D. B. Lindell, S. Diamond, and G. Wetzstein. Robust non-line-of-sight imaging with single photon detectors. arXiv:1711.07134, 2017.

[HP03] J. Y. Han and K. Perlin. Measuring bidirectional texture reflectance with a kaleidoscope. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, pages 741–748, 2003.

[HS12] K. He and J. Sun. Computing nearest-neighbor fields via propagation-assisted kd-trees. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 111–118, 2012.

[HSG+16] S. W. Hasinoff, D. Sharlet, R. Geiss, A. Adams, J. T. Barron, F. Kainz, J. Chen, and M. Levoy. Burst photography for high dynamic range and low-light imaging on mobile cameras. *ACM Trans. Graph.*, 35(6):192:1–192:12, November 2016.

[HXHH14] F. Heide, L. Xiao, W. Heidrich, and M. B. Hullin. Diffuse mirrors: 3D reconstruction from diffuse indirect illumination using inexpensive time-of-flight sensors. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[HZ04] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2004.

[HZW+06] J. Han, K. Zhou, L.-Y. Wei, M. Gong, H. Bao, X. Zhang, and B. Guo. Fast example-based surface texture synthesis via discrete optimization. *The Visual Computer*, 22(9):918–925, 2006.

[IGP+17] J. Iseringhausen, B. Goldlücke, N. Pesheva, S. Iliev, A. Wender, M. Fuchs, and M. B. Hullin. 4D imaging through spray-on optics. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 36(4), 2017.

[IH18] J. Iseringhausen and M. B. Hullin. Non-line-of-sight reconstruction using efficient transient rendering. arXiv:1809.08044 [cs.GR], ACM Transactions on Graphics (to appear), 2018.

[IKL+08] I. Ihrke, K. Kutulakos, H. Lensch, M. Magnor, and W. Heidrich. State of the art in transparent and specular object reconstruction. In *EUROGRAPHICS 2008 STAR*, 2008.

[Ili95] S. Iliev. Iterative method for the shape of static drops. *Computer Methods in Applied Mechanics and Engineering*, 126(3):251–265, 1995.

[Ili97] S. Iliev. Static drops on an inclined plane: equilibrium modeling and numerical analysis. *Journal of Colloid and Interface Science*, 194(2):287–300, 1997.

[IP03] S. Iliev and N. Pesheva. Wetting properties of well-structured heterogeneous substrates. *Langmuir*, 19(23):9923–9931, 2003.

[IP06] S. Iliev and N. Pesheva. Nonaxisymmetric drop shape analysis and its application for determination of the local contact angles. *Journal of Colloid and Interface Science*, 301(2):677–684, 2006.

[IWHH19] J. Iseringhausen, M. Weinmann, W. Huang, and M. B. Hullin. Computational parquetry: Fabricated style transfer with wood pixels. arXiv:1904.04769 [cs.GR], ACM Transactions on Graphics (to appear), 2019.

[IWLH11] I. Ihrke, G. Wetzstein, D. Lanman, and W. Heidrich. State of the art in computational plenoptic imaging. In *EUROGRAPHICS 2011 STAR*, 2011.

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

[JBS17] N. Jetchev, U. Bergmann, and C. Seward. Ganosaic: Mosaic creation with generative texture manifolds. arXiv:1712.00269, 2017.

[JDJ96] A. Jackson, D. Day, and S. Jennings. *The Complete Manual of Woodworking*. Knopf, 1996.

[JMM+14] A. Jarabo, J. Marco, A. Muñoz, R. Buisan, W. Jarosz, and D. Gutierrez. A framework for transient rendering. *ACM Transactions on Graphics (TOG)*, 33(6):177, 2014.

[JMMG17] A. Jarabo, B. Masia, J. Marco, and D. Gutierrez. Recent advances in transient imaging: A computer graphics and vision perspective. *Visual Informatics*, 1(1):65–79, 2017.

[JTFW17] H. Jiang, Q. Tian, J. Farrell, and B. A. Wandell. Learning the image processing pipeline. *IEEE Transactions on Image Processing*, 26(10):5032–5042, 2017.

[JYF+17] Y. Jing, Y. Yang, Z. Feng, J. Ye, and M. Song. Neural style transfer: A review. arXiv:1705.04058, 2017.

[KCWI13] J. E. Kyprianidis, J. Collomosse, T. Wang, and T. Isenberg. State of the "art": A taxonomy of artistic stylization techniques for images and video. *IEEE Transactions on Visualization and Computer Graphics*, 19(5):866–885, 2013.

[KEBK05] V. Kwatra, I. Essa, A. Bobick, and N. Kwatra. Texture optimization for example-based synthesis. *ACM Transactions on Graphics (TOG)*, 24(3):795–802, 2005.

[KFCO+07] J. Kopf, C.-W. Fu, D. Cohen-Or, O. Deussen, D. Lischinski, and T.-T. Wong. Solid texture synthesis from 2D exemplars. *ACM Transactions on Graphics (TOG)*, 26(3), 2007.

[KH04] S. Kuiper and B. H. W. Hendriks. Variable-focus liquid lens for miniature cameras. *Applied Physics Petters*, 85(7):1128–1130, 2004.

[KHDR09] A. Kirmani, T. Hutchison, J. Davis, and R. Raskar. Looking around the corner using transient imaging. In *IEEE International Conference on Computer Vision (ICCV)*, pages 159–166, 2009.

[KHFG14] O. Katz, P. Heidmann, M. Fink, and S. Gigan. Non-invasive single-shot imaging through scattering layers and around corners via speckle correlations. *Nature Photonics*, 8(10):784–790, 2014.

[KNL+15] A. Kaspar, B. Neubert, D. Lischinski, M. Pauly, and J. Kopf. Self tuning texture optimization. *Computer Graphics Forum*, 34(2):349–359, 2015.

[KP02] J. Kim and F. Pellacini. Jigsaw image mosaics. *ACM Transactions on Graphics (TOG)*, 21(3):657–664, 2002.

[KPL08] S. Kammel and F. Puente Leon. Deflectometric measurement of specular surfaces. *IEEE Transactions on Instrumentation and Measurement*, 57(4):763–769, April 2008.

[KPM+16] J. Klein, C. Peters, J. Martín, M. Laurenzis, and M. B. Hullin. Tracking objects outside the line of sight using 2D intensity images. *Scientific Reports*, 6(32491), 2016.

[KS08] K. N. Kutulakos and E. Steger. A theory of refractive and specular 3D shape by light-path triangulation. *International Journal of Computer Vision*, 76(1):13–29, 2008.

[KSE+03] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: Image and video synthesis using graph cuts. *ACM Transactions on Graphics (TOG)*, 22(3):277–286, 2003.

[KSG00] D. Kruger, P. Schneck, and H. Gelderblom. Helmut ruska and the visualisation of viruses. *The Lancet*, 355(9216):1713–1717, 2000.

[KSRY11] D. Kang, S. Seo, S. Ryoo, and K. Yoon. A parallel framework for fast photomosaics. *IEICE Transactions on Information and Systems*, 94-D(10):2036–2042, 2011.

[KWB+13] A. Kadambi, R. Whyte, A. Bhandari, L. Streeter, C. Barsi, A. Dorrington, and R. Raskar. Coded time of flight cameras: Sparse deconvolution to address multipath interference and recover time profiles. *ACM Transactions on Graphics (TOG)*, 32(6), November 2013.

[KZP+13] C. Kim, H. Zimmer, Y. Pritch, A. Sorkine-Hornung, and M. Gross. Scene reconstruction from high spatio-angular resolution light fields. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 32(4):73:1–73:12, 2013.

[KZSR16] A. Kadambi, H. Zhao, B. Shi, and R. Raskar. Occluded imaging with time-of-flight sensors. *ACM Transactions on Graphics (TOG)*, 35(2):15:1–15:12, March 2016.

[LBDF13] J. Lu, C. Barnes, S. DiVerdi, and A. Finkelstein. Realbrush: Painting with examples of physical media. *ACM Transactions on Graphics (TOG)*, 32(4):117:1–117:12, July 2013.

[LC87] W. Lorensen and H. Cline. Marching Cubes: A high resolution 3D surface construction algorithm. In *Proc. 14th Annual Conf. on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, pages 163–169. ACM, 1987.

[Lev44] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2(2):164–168, 1944.

[LH96] M. Levoy and P. Hanrahan. Light field rendering. In *Proc. 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 31–42, New York, NY, USA, 1996. ACM.

[Lip08] G. Lippmann. La photographie intégrale. *CR Acad. Sci.*, 146:446–451, 1908.

[LKB+18] M. La Manna, F. Kine, E. Breitbach, J. Jackson, T. Sultan, and A. Velten. Error backprojection algorithms for non-line-of-sight imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 1–1, 2018.

[LLX+01] L. Liang, C. Liu, Y.-Q. Xu, B. Guo, and H.-Y. Shum. Real-time texture synthesis by patch-based sampling. *ACM Transactions on Graphics (TOG)*, 20(3):127–150, 2001.

[LM01] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, June 2001.

[LNA+06] M. Levoy, R. Ng, A. Adams, M. Footer, and M. Horowitz. Light field microscopy. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, pages 924–934, 2006.

[Low99] D. G. Lowe. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1150–1157, 1999.

[LPD13] T. Lindemeier, S. Pirk, and O. Deussen. Image stylization with a painting machine using semantic hints. *Computers & Graphics*, 37(5):293–301, 2013.

[LV14] M. Laurenzis and A. Velten. Nonline-of-sight laser gated viewing of scattered photons. *Optical Engineering*, 53(2):023102–023102, 2014.

[LVJ10] Y. Liu, O. Veksler, and O. Juan. Generating classic mosaics with graph cuts. *Computer Graphics Forum*, 29(8):2387–2399, 2010.

[LW16a] C. Li and M. Wand. Combining markov random fields and convolutional neural networks for image synthesis. arXiv:1601.04589, 2016.

[LW16b] C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European Conference on Computer Vision (ECCV)*, pages 702–716, 2016.

[LWTG14] P.-J. Lapray, X. Wang, J.-B. Thomas, and P. Gouton. Multispectral filter arrays: Recent advances and practical implementation. *Sensors*, 14:21626–59, 11 2014.

[LWYS13] H.-C. Liao, D.-Y. Wang, C.-L. Yang, and J. Shin. Video-based water drop detection and removal method for a moving vehicle. *Information Technology Journal*, 12:569–583, 04 2013.

[Mar63] D. W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441, 1963.

[MJGJ17] J. Marco, W. Jarosz, D. Gutierrez, and A. Jarabo. Transient photon beams. In *Spanish Computer Graphics Conference (CEIG)*. The Eurographics Association, June 2017.

[MRK+13] A. Manakov, J. F. Restrepo, O. Klehm, R. Hegedüs, E. Eisemann, H.-P. Seidel, and I. Ihrke. A reconfigurable camera add-on for high dynamic range, multi-spectral, polarization, and light-field imaging. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 32(4):47:1–47:14, July 2013.

[MT17] R. Mur-Artal and J. D. Tardós. Visual-inertial monocular slam with map reuse. *IEEE Robotics and Automation Letters*, 2(2):796–803, April 2017.

[MTK+11] Y. Mukaigawa, S. Tagawa, J. Kim, R. Raskar, Y. Matsushita, and Y. Yagi. Hemispherical confocal imaging using turtleback reflector. In *Asian Conference on Computer Vision (ACCV)*, Lecture Notes in Computer Science, pages 336–349. Springer, 2011.

[Ng05] R. Ng. Fourier slice photography. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, pages 735–744, 2005.

[NLB+05] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan. Light field photography with a hand-held plenoptic camera. Technical Report CTSR 2005-02, Stanford University, 2005.

[NM00] S. K. Nayar and T. Mitsunaga. High dynamic range imaging: spatially varying pixel exposures. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, volume 1, pages 472–479 vol.1, June 2000.

[NN04] K. Nishino and S. K. Nayar. Eyes for relighting. *ACM Trans. Graph.*, 23(3):704–711, August 2004.

[NN06] K. Nishino and S. K. Nayar. Corneal imaging system: Environment from eyes. *International Journal of Computer Vision*, 70(1):23–40, Oct 2006.

[NZV+11] N. Naik, S. Zhao, A. Velten, R. Raskar, and K. Bala. Single view reflectance capture using multiplexed scattering and time-of-flight imaging. *ACM Transactions on Graphics (TOG)*, 30(6):171, 2011.

[OA12] I. Olonetsky and S. Avidan. Treecann - k-d tree coherence approximate nearest neighbor algorithm. In *European Conference on Computer Vision (ECCV)*, pages 602–615, Berlin, Heidelberg, 2012. Springer-Verlag.

[OEED18] R. S. Overbeck, D. Erickson, D. Evangelakos, and P. Debevec. Welcome to light fields. In *ACM SIGGRAPH 2018 Virtual, Augmented, and Mixed Reality*, 2018.

[OJK+18] T.-H. Oh, R. Jaroensri, C. Kim, M. Elgharib, F. Durand, W. T. Freeman, and W. Matusik. Learning-based video motion magnification. In *European Conference on Computer Vision (ECCV)*, September 2018.

[OLW18a] M. O'Toole, D. B. Lindell, and G. Wetzstein. Confocal non-line-of-sight imaging based on the light-cone transform. *Nature*, 555(25489):338–341, 2018.

[OLW18b] M. O'Toole, D. B. Lindell, and G. Wetzstein. Real-time non-line-of-sight imaging. In *ACM SIGGRAPH 2018 Emerging Technologies*, SIGGRAPH '18, pages 14:1–14:2, New York, NY, USA, 2018. ACM.

[OS02] F. T. O'Neill and J. T. Sheridan. Photoresist reflow method of microlens production part i: Background and experiments. *Optik – International Journal for Light and Electron Optics*, 113(9):391–404, 2002.

[PBT+17] A. K. Pediredla, M. Buttafava, A. Tosi, O. Cossairt, and A. Veeraraghavan. Reconstructing rooms using photon echoes: A plane based model and reconstruction algorithm

for looking around the corner. In *IEEE International Conference on Computational Photography (ICCP)*, pages 1–12. IEEE, 2017.

[PCBC10] T. Pock, D. Cremers, H. Bischof, and A. Chambolle. Global Solutions of Variational Models with Convex Regularization. *SIAM Journal on Imaging Sciences*, 2010.

[PCK09] D. Pavić, U. Ceumern, and L. Kobbelt. Gizmos: Genuine image mosaics with adaptive tiling. *Computer Graphics Forum*, 28(8):2244–2254, 2009.

[PFH00] E. Praun, A. Finkelstein, and H. Hoppe. Lapped textures. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques*, pages 465–470, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[PH10] M. Pharr and G. Humphreys. *Physically Based Rendering, Second Edition: From Theory To Implementation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2nd edition, 2010.

[PL98] R. Paget and I. D. Longstaff. Texture synthesis via a noncausal nonparametric multiscale markov random field. *IEEE Transactions on Image Processing*, 7(6):925–931, 1998.

[Pom17] J. C. Pommerville. *Fundamentals of Microbiology*. Jones & Bartlett Learning, 11th edition, 2017.

[PPW18] A. Panotopoulou, S. Paris, and E. Whiting. Watercolor woodblock printing with image analysis. *Computer Graphics Forum*, 37(2):275–286, 2018.

[PS00] J. Portilla and E. P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *IEEE International Conference on Computer Vision (ICCV)*, 40(1):49–70, 2000.

[PTSF19] A. Pemasiri, K. N. Thanh, S. Sridharan, and C. Fookes. Sparse over-complete patch matching. *Pattern Recognition Letters*, 122:1–6, 2019.

[QGC+09] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng. Ros: an open-source robot

operating system. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) Workshop on Open Source Robotics*, Kobe, Japan, May 2009.

[RAWV08] R. Raskar, A. Agrawal, C. A. Wilson, and A. Veeraraghavan. Glare aware photography: 4D ray sampling for reducing glare effects of camera lenses. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 27(3):56:1–56:10, August 2008.

[RFB15] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. arXiv:1505.04597, 2015.

[Rus02] J. C. Russ. *Image Processing Handbook, Fourth Edition*. CRC Press, Inc., Boca Raton, FL, USA, 4th edition, 2002.

[SA19] N. Shiee and A. Agarwala. Photobooth on Pixel. `https://ai.googleblog.com/2019/04/take-your-best-selfie-automatically.html`, 2019. Accessed on 27 June 2019.

[SC14] M. Slaney and P. A. Chou. Time of flight tracer. Technical report, Microsoft Research, November 2014.

[SCK10] Q. Shan, B. Curless, and T. Kohno. Seeing through obscure glass. In *European Conference on Computer Vision (ECCV)*, pages 364–378, Berlin, Heidelberg, 2010. Springer-Verlag.

[SCSI08] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani. Summarizing visual data using bidirectional similarity. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[SF16] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[Sil97] R. Silvers. *Photomosaics*. Henry Holt and Co., Inc., New York, NY, USA, 1997.

[SMBG19] C. Saunders, J. Murray-Bruce, and V. K. Goyal. Computational periscopy with an ordinary digital camera. *Nature*, 565, 1 2019.

[SML06] W. Schroeder, K. Martin, and B. Lorensen. *The Visualization Toolkit–An Object-Oriented Approach To 3D Graphics*. Kitware, Inc., fourth edition, 2006.

[SP16] A. Stylianou and R. Pless. Sparklegeometry: Glitter imaging for 3D point tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 919–926, 2016.

[SS97] R. Szeliski and H.-Y. Shum. Creating full view panoramic image mosaics and environment maps. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques*, pages 251–258, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[SSD08] A. Smith, J. Skorupski, and J. Davis. Transient rendering. Technical Report UCSC-SOE-08-26, School of Engineering, University of California, Santa Cruz, 2008.

[TAV+10] Y. Taguchi, A. Agrawal, A. Veeraraghavan, S. Ramalingam, and R. Raskar. Axial-cones: Modeling spherical catadioptric cameras for wide-angle light field rendering. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 29(6):172:1–172:8, December 2010.

[TF14] A. Torralba and W. T. Freeman. Accidental pinhole and pinspeck cameras. *International Journal of Computer Vision*, 110(2):92–112, 2014.

[The19] The Event Horizon Telescope Collaboration. First m87 event horizon telescope results. iv. imaging the central supermassive black hole. *The Astrophysical Journal Letters*, 875(1):L4, 2019.

[THMR13] M. W. Tao, S. Hadap, J. Malik, and R. Ramamoorthi. Depth from combining defocus and correspondence using light-field cameras. In *IEEE International Conference on Computer Vision (ICCV)*, pages 673–680, 2013.

[TL12] P. A. Tresset and F. F. Leymarie. Sketches by paul the robot. In *Proceedings of the Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging*, pages 17–24, Goslar Germany, Germany, 2012. Eurographics Association.

[TLGS05] M. Tarini, H. P. A. Lensch, M. Goesele, and H.-P. Seidel. 3D acquisition of mirroring objects using striped patterns. *Graphical Models*, 67(4):233–259, 2005.

[TSX+17] C. Thrampoulidis, G. Shulkind, F. Xu, W. T. Freeman, J. H. Shapiro, A. Torralba, F. N. C. Wong, and G. W. Wornell. Exploiting occlusion in non-line-of-sight active imaging. arXiv:1711.06297, 2017.

[TZL+02] X. Tong, J. Zhang, L. Liu, X. Wang, B. Guo, and H.-Y. Shum. Synthesis of bidirectional texture functions on arbitrary surfaces. *ACM Transactions on Graphics (TOG)*, 21(3):665–672, July 2002.

[Ulr07] R. B. Ulrich. *Roman Woodworking*. Yale University Press, 2007.

[V+08] V. Vaish et al. The (New) Stanford Light Field Archive, 2008. `http://lightfield.stanford.edu/lfs.html`.

[VRA+07] A. Veeraraghavan, R. Raskar, A. Agrawal, A. Mohan, and J. Tumblin. Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 26(3), 2007.

[VRB11] A. Velten, R. Raskar, and M. Bawendi. Picosecond camera for time-of-flight imaging. In *Imaging and Applied Optics*, page IMB4. Optical Society of America, 2011.

[vSUC18] L. von Stumberg, V. Usenko, and D. Cremers. Direct sparse visual-inertial odometry using dynamic marginalization. In *International Conference on Robotics and Automation (ICRA)*, May 2018.

[VWG+12] A. Velten, T. Willwacher, O. Gupta, A. Veeraraghavan, M. G. Bawendi, and R. Raskar. Recovering three-dimensional shape around a corner using ultrafast time-of-flight imaging. *Nature Communications*, 3:745, 2012.

[VWJ+13] A. Velten, D. Wu, A. Jarabo, B. Masia, C. Barsi, C. Joshi, E. Lawson, M. Bawendi, D. Gutierrez, and R. Raskar. Femto-photography: Capturing and visualizing the propagation of light. *ACM Transactions on Graphics (TOG)*, 32(4):44:1–44:8, July 2013.

[WER16] T.-C. Wang, A. Efros, and R. Ramamoorthi. Depth estimation with occlusion modeling using light-field cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2016.

[WG14] S. Wanner and B. Goldlücke. Variational light field analysis for disparity estimation and super-resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(3):606–619, 2014.

[WGDE+19] B. Wronski, I. Garcia-Dorado, M. Ernst, D. Kelly, M. Krainin, C.-K. Liang, M. Levoy, and P. Milanfar. Handheld multiframe super-resolution. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 38(4), 7 2019.

[WGJ+18] N. Wadhwa, R. Garg, D. E. Jacobs, B. E. Feldman, N. Kanazawa, R. Carroll, Y. Movshovitz-Attias, J. T. Barron, Y. Pritch, and M. Levoy. Synthetic depth-of-field with a single-camera mobile phone. *ACM Trans. Graph.*, 37(4):64:1–64:13, July 2018.

[WIG+15] A. Wender, J. Iseringhausen, B. Goldlücke, M. Fuchs, and M. B. Hullin. Light field imaging through household optics. In D. Bommes, T. Ritschel, and T. Schultz, editors, *Vision, Modeling & Visualization*, pages 159–166. Eurographics Association, 2015.

[WIH13] G. Wetzstein, I. Ihrke, and W. Heidrich. On plenoptic multiplexing and reconstruction. *International Journal of Computer Vision*, 101(2):384–400, 2013.

[WJV+05] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy. High performance imaging using large camera arrays. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, pages 765–776, 2005.

[WK15] M. Weinmann and R. Klein. Advances in geometry and reflectance acquisition. In *SIGGRAPH Asia 2015 Courses*, New York, NY, USA, 2015. ACM.

[WK18] W. Wen and S. Khatibi. Virtual deformable image sensors: Towards a general framework for image sensors with flexible grids and forms. *Sensors*, 18(6), 6 2018.

[WL00] L.-Y. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques*, pages 479–488, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[WLKT09] L.-Y. Wei, S. Lefebvre, V. Kwatra, and G. Turk. State of the art in example-based texture synthesis. In *Eurographics 2009, State of the Art Reports (STAR)*, pages 93–117. Eurographics Association, 2009.

[WLM+15] L.-Y. Wei, C.-K. Liang, G. Myhre, C. Pitts, and K. Akeley. Improving light field camera sample design with irregularity and aberration. *ACM Transactions on Graphics (TOG)*, 34(4):152:1–152:11, 2015.

[WORK13] M. Weinmann, A. Osep, R. Ruiters, and R. Klein. Multi-view normal field integration for 3D reconstruction of mirroring objects. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2504–2511, December 2013.

[WRDF13] N. Wadhwa, M. Rubinstein, F. Durand, and W. T. Freeman. Phase-based video motion processing. *ACM Transactions on Graphics (TOG)*, 32(4):80:1–80:10, July 2013.

[WRS+12] H.-Y. Wu, M. Rubinstein, E. Shih, J. Guttag, F. Durand, and W. T. Freeman. Eulerian video magnification for revealing subtle changes in the world. *ACM Trans. Graph. (Proceedings SIGGRAPH 2012)*, 31(4), 2012.

[WSI07] Y. Wexler, E. Shechtman, and M. Irani. Space-time completion of video. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 29(3):463–476, 2007.

[WVO+14] D. Wu, A. Velten, M. O'Toole, B. Masia, A. Agrawal, Q. Dai, and R. Raskar. Decomposing global light transport using time of flight imaging. *International Journal of Computer Vision*, 107(2):123–138, April 2014.

[WWB+12] D. Wu, G. Wetzstein, C. Barsi, T. Willwacher, M. O'Toole, N. Naik, Q. Dai, K. Kutulakos, and R. Raskar. Frequency analysis of transient light transport with applications in bare sensor imaging. In *European Conference on Computer Vision (ECCV)*, pages 542–555. Springer, 2012.

[WZH+16] T.-C. Wang, J.-Y. Zhu, E. Hiroaki, M. Chandraker, A. A. Efros, and R. Ramamoorthi. A 4D light-field dataset and cnn architectures for material recognition. In *European Conference on Computer Vision (ECCV)*, pages 121–138, 2016.

[XFM14] Y. Xu, J.-M. Frahm, and F. Monrose. Watching the watchers: Automatically inferring TV content from outdoor light effusions. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 418–428. ACM, 2014.

[XRW+14] T. Xue, M. Rubinstein, N. Wadhwa, A. Levin, F. Durand, and W. T. Freeman. Refraction wiggles for measuring fluid depth and velocity from video. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 767–782, Cham, 2014. Springer International Publishing.

[YBT+19] A. B. Yedidia, M. Baradad, C. Thrampoulidis, W. T. Freeman, and G. W. Wornell. Using unknown occluders to recover hidden scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12231–12239, 2019.

[YTF+17] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan. Deep joint rain detection and removal from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1357–1366, 2017.

[YTK+16] S. You, R. T. Tan, R. Kawakami, Y. Mukaigawa, and K. Ikeuchi. Waterdrop stereo. arXiv:1604.00730v1 [cs.CV], 2016.

[Zha00] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 22:1330–1334, December 2000. MSR-TR-98-71, Updated March 25, 1999.

[ZIA14] Z. Zhang, P. Isola, and E. H. Adelson. Sparkle vision: Seeing the world through random specular microfacets. arXiv:1412.7884, 2014.

[ZLAA+18] M. Zhao, T. Li, M. Abu Alsheikh, Y. Tian, H. Zhao, A. Torralba, and D. Katabi. Through-wall human pose estimation using radio signals. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[ZP18] H. Zhang and V. M. Patel. Density-aware single image deraining using a multi-stream dense network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 695–704, 2018.

List of Figures

List of Tables

Attribution of Source Materials

- Mirrorless Camera (Figure 1.1): © 2016 Yitech, CC BY-SA 4.0.
- Cromenco Cyclops (Figure 1.1): © 2013 Cromenco, CC BY-SA 3.0.
- Kyocera VP-210 (Figure 1.1): © 2011 Morio, CC BY-SA 3.0.
- Virtual Deformable Image Sensors: Towards to a General Framework for Image Sensors with Flexible Grids and Forms (Figure 1.1): © 2018 Wei Wen, Siamak Khatibi, CC BY-SA 4.0.
- Thin SPAD cross-section (Figure 1.1): 2007 Gechi, public domain.
- Multispectral Filter Arrays: Recent Advances and Practical Implementation (Figure 1.1): © 2014 Pierre-Jean Lapray, Xingbo Wang, Jean-Baptiste Thomas, Pierre Gouton, CC BY 4.0.
- arxiv.org/abs/1905.03277: Handheld Multi-frame Super-resolution (Figure 1.1): © 2019 Bartlomiej Wronski, Ignacio Garcia-Dorado, Manfred Ernst, Damien Kelly, Michael Krainin, Chia-Kai Liang, Marc Levoy, Peyman Milanfar CC BY-NC-SA 4.0.
- Reprinted by permission from Springer International Publishing Switzerland: Springer Nature SPRINGER EBOOK Refraction Wiggles for Measuring Fluid Depth and Velocity from Video, Tianfan Xue, Michael Rubinstein, Neal Wadhwa, Anat Levin, Fredo Durand, William T. Freeman, COPYRIGHT 2014 (Figure 1.1).
- Reprinted by permission from Springer Nature: Nature NATURE Computational Periscopy with an Ordinary Digital Camera, Charles Saunders, John Murray-Bruce, Vivek K. Goyal, COPYRIGHT 2019. (Figure 1.1)
- Left Blue Eye (Figure 4.13): public domain.
- Marquetry Self Portrait (Figure 4.2): © 2008 Laszlo Sandor, CC BY 4.0.
- Marquetry portrait "Girl 1" (Figure 4.2): © 2015 Rob Milam, included with permission of the artist.
- Intarsia image, Workshop David Roentgen (Figure 4.3): 2011, public domain.
- Mosaïque d'Ulysse et les sirènes, Bardo Museum in Tunis (Figure 4.3): public domain.
- Adult brown tabby cat (Figures 4.4 and 4.14): © Tomas Andreopoulos, Pexels license.
- Close-up Photo of Dog Wearing Golden Crown (Figure 4.5): © rawpixel.com, Pexels license.
- Closeup Photo of Human Eye (Figure 4.10): © Skitterphoto, CC0 1.0.
- Official presidential transitional photo of then-President-elect Barack Obama (Figure 4.12): © 2008 The Obama-Biden Transition Project, CC BY 3.0.
- Ludwig van Beethoven, oil on canvas (Figure 4.1): 1820 Joseph Karl Stieler, public domain.
- Commodore Grace M. Hopper, USNR Official portrait photograph (Figure 4.15): 1984 Naval History and Heritage Command, public domain.
- STS-93 Commander Eileen M. Collins (Figure 4.15): 1998 NASA, Robert Markowitz, public domain.
- Felix Hausdorff (Figure 4.15): 1913-1921 Universitätsbibliothek Bonn, public domain.
- Katherine G. Johnson (Figure 4.15): 2018 NASA, public domain.
- Ludwig van Beethoven (Figure 4.15): 1854 Emil Eugen Sachse, public domain.
- Whoopi Goldberg in New York City, protesting California Proposition 8 (Figure 4.15):

© 2008 David Shankbone, CC BY 3.0.

- Hedy Lamarr in "The Heavenly Body" (Figure 4.15): 1944 Employees of MGM, public domain.
- Passport photo of Alan Turing at age 16 (Figure 4.15): 1928-1929 unknown author, public domain.
- Tiny cute piglet looking at the photographer (Figure 4.15): 2012 Petr Kratochvil, public domain.
- Penguin (Figure 4.15): © 2016 Pexels, Pixabay license.
- Adult Brown and White Pembroke Welsh Corgi Near the Body of Water (Figure 4.15): © Muhannad Alatawi, Pexels license.
- Phoenicopterus ruber, the Greater Flamingo (Figure 4.15): 1827-1838 John James Audubon, public domain.