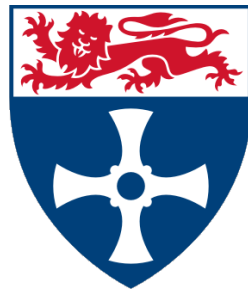


# Automated inverse-rendering techniques for realistic 3D artefact compositing in 2D photographs



**Ana Maria Mihut**

School of Computing Science  
Newcastle University

This dissertation is submitted for the degree of  
*Doctor of Philosophy*

February 2020

## **Declaration**

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Ana Maria Mihut  
February 2020



## Acknowledgements

Throughout the time dedicated towards my formal postgraduate education, a number of terrific individuals have shown their support and involvement, and I would like express my gratitude towards them.

First, I shall thank my graduate advisor, Dr. Graham Morgan for his guidance and academic insight. Moreover, I want to thank him for always having an open door and taking the time to discuss research during off-work hours on a large number of occasions. I thank Dr. Gary Ushaw for his involvement with my work. His dedication for producing high quality research, his active encouragements and expert advice are aspects that I admire and plan to emulate in the future.

I want to acknowledge my family: Otilia, Adrian, Marta, and Rich for the unconditional love and encouragement they have shown. My parents, both engineers, have inspired me to follow a career in Computer Science, and have sacrificed a great deal so I could be exposed to countless opportunities as I grew up, and for this I am deeply grateful. My grandmother is, to this day, my inspiration.

# Abstract

The process of acquiring images of a scene and modifying the defining structural features of the scene through the insertion of artefacts is known in literature as *compositing*. The process can take effect in the 2D domain (where the artefact originates from a 2D image and is inserted into a 2D image), or in the 3D domain (the artefact is defined as a dense 3D triangulated mesh, with textures describing its material properties).

Compositing originated as a solution to enhancing, repairing, and more broadly editing photographs and video data alike in the film industry as part of the post-production stage. This is generally thought of as carrying out operations in a 2D domain (a single image with a known width, height, and colour data). The operations involved are sequential and entail separating the foreground from the background (matting), or identifying features from contour (feature matching and segmentation) with the purpose of introducing new data in the original. Since then, compositing techniques have gained more traction in the emerging fields of Mixed Reality (MR), Augmented Reality (AR), robotics and machine vision (scene understanding, scene reconstruction, autonomous navigation). When focusing on the 3D domain, compositing can be translated into a pipeline <sup>1</sup> - the incipient stage acquires the scene data, which then undergoes a number of processing steps aimed at inferring structural properties that ultimately allow for the placement of 3D artefacts anywhere within the scene, rendering a plausible and consistent result with regard to the physical properties of the initial input.

This generic approach becomes challenging in the absence of user annotation and labelling of scene geometry, light sources and their respective magnitude and orientation, as well as a clear object segmentation and knowledge of surface properties. A single image, a stereo pair, or even a short image stream may not hold enough information regarding the shape or illumination of the scene, however, increasing the input data will only incur an extensive time penalty which is an established challenge in the field.

Recent state-of-the-art methods address the difficulty of inference in the absence of

---

<sup>1</sup>In the present document, the term *pipeline* refers to a software solution formed of stand-alone modules or stages. It implies that the flow of execution runs in a single direction, and that each module has the potential to be used on its own as part of other solutions. Moreover, each module is assumed to take an input set and output data for the following stage, where each module addresses a single type of problem only.

data, nonetheless, they do not attempt to solve the challenge of compositing artefacts between existing scene geometry, or cater for the inclusion of new geometry behind complex surface materials such as translucent glass or in front of reflective surfaces.

The present work focuses on the compositing in the 3D domain and brings forth a software framework <sup>2</sup> that contributes solutions to a number of challenges encountered in the field, including the ability to render physically-accurate soft shadows in the absence of user annotated scene properties or RGB-D data. Another contribution consists in the timely manner in which the framework achieves a believable result compared to the other compositing methods which rely on offline rendering. The availability of proprietary hardware and user expertise are two of the main factors that are not required in order to achieve a fast and reliable results within the current framework.

---

<sup>2</sup>In the present document, the term *framework* refers to software that was designed to address a specific problem and is assumed to be used as a library or plugin in conjunction with other software. It implies portability.

# Table of contents

List of figures	ix
List of tables	xi
<b>1 Introduction</b>	<b>1</b>
1.1 Overview . . . . .	1
1.2 Contribution . . . . .	3
1.2.1 Chapter outline . . . . .	4
1.3 Publications . . . . .	5
<b>2 Background</b>	<b>7</b>
2.1 Classification of object compositing techniques . . . . .	7
2.1.1 2D techniques - matting and compositing . . . . .	8
2.1.1.1 Environment matting . . . . .	8
2.1.2 3D hybrid techniques . . . . .	9
2.1.2.1 Measured light transport properties . . . . .	10
2.1.2.2 Approximated light models . . . . .	13
2.2 Radiometry and the physically based rendering model . . . . .	16
2.2.1 Physical characteristics of electromagnetic radiation . . . . .	16
2.2.1.1 Transverse nature of light and the plane wave model . . . . .	16
2.2.1.2 Polarization and the index of refraction . . . . .	18
2.2.2 Physical interaction of light with media . . . . .	20
2.2.2.1 Reflection and transmission . . . . .	20
2.2.3 Mathematical model of light-material interaction . . . . .	25
2.2.3.1 The micro-facet model for surface representation . . . . .	25
2.2.3.2 The bidirectional reflective distribution function . . . . .	28
2.2.3.3 Surface reflectance - the specular term of the BRDF . . . . .	30
2.2.3.4 Subsurface reflectance - the diffuse term of the BRDF . . . . .	34
2.3 Inverse rendering and image decomposition . . . . .	35
2.3.1 Estimating surface geometry through Shape-from-Shading . . . . .	37
2.3.2 Estimating surface materials through image decomposition . . . . .	40

2.4	Related work . . . . .	43
2.4.1	Scene geometry estimation and structural reconstruction . . . . .	43
2.4.2	Illumination estimation models . . . . .	47
2.4.3	Material property representation and inverse rendering . . . . .	52
2.4.4	PBR techniques in the wild . . . . .	58
2.4.4.1	Production PBR . . . . .	58
2.4.4.2	Real-time PBR . . . . .	62
2.5	Background Summary . . . . .	66
<b>3</b>	<b>Methodology</b>	<b>67</b>
3.1	Chapter Overview . . . . .	67
3.2	Scene depth reconstruction . . . . .	69
3.2.1	Initial depth acquisition and refinement . . . . .	70
3.2.2	Creation of the counterpart 3D scene projection . . . . .	78
3.2.2.1	Preliminaries . . . . .	78
3.2.3	Creation of the counterpart topology . . . . .	79
3.3	Shadow mapping and relighting . . . . .	82
3.3.1	Illumination model . . . . .	82
3.3.2	Relighting step . . . . .	85
3.3.3	Overview of the shadow casting process . . . . .	87
3.3.4	Initial lighting setup . . . . .	92
3.3.5	Relighting during the composition step . . . . .	93
3.3.6	Geometry reprojection . . . . .	96
<b>4</b>	<b>Results and evaluation</b>	<b>97</b>
4.1	Evaluation overview . . . . .	97
4.1.1	Dataset criteria . . . . .	98
4.2	Evaluation of the scene acquisition consistency . . . . .	99
4.2.1	Evaluation based on a real-world stereo dataset . . . . .	99
4.2.2	Evaluation based on synthetic scenes and near-stereo . . . . .	101
4.3	Evaluation of the relighting consistency . . . . .	106
4.4	Discussion . . . . .	110
<b>5</b>	<b>Suitable domains of application</b>	<b>116</b>
<b>6</b>	<b>Conclusion</b>	<b>119</b>
6.1	Contributions overview . . . . .	119
6.2	Limitations . . . . .	121
6.3	Further work . . . . .	122

<b>References</b>	<b>125</b>
<b>Appendix A Full listing of supplemental materials</b>	<b>137</b>
A.1 Website resources . . . . .	137
A.2 Hardware specification . . . . .	138
A.3 Libraries, APIs, 3rd-party software . . . . .	138
A.3.1 Compatibility . . . . .	138
A.3.2 Development tools . . . . .	138
<b>Appendix B Supplemental code</b>	<b>140</b>
B.1 Intrinsic scene representation . . . . .	141
B.2 Shadow masking . . . . .	143
B.3 Additional depth reconstruction . . . . .	145
<b>Appendix C Supplemental results</b>	<b>146</b>
C.1 Illumination setup across datasets . . . . .	146

# List of figures

1.1	AR technology . . . . .	1
2.1	ARM extraction and compositing . . . . .	9
2.2	IBL Probes . . . . .	11
2.3	Outdoor illumination estimation and compositing . . . . .	15
2.4	Electromagnetic spectrum . . . . .	16
2.5	Plane wave equation . . . . .	17
2.6	Polarization phases . . . . .	19
2.7	Material types: flat, metallic, dielectrics . . . . .	21
2.8	Roughness of surface . . . . .	22
2.9	BRDF specular and diffuse terms . . . . .	23
2.10	Microfacet overview . . . . .	25
2.11	Visualising incident light ray interaction with medium . . . . .	27
2.12	Hemisphere representation . . . . .	29
2.13	Fresnel colour table . . . . .	32
2.14	Image decomposition . . . . .	41
2.15	Estimation through the Box model . . . . .	44
2.16	Single Image Depth Estiamtion . . . . .	45
2.17	Tango Indoo Scene . . . . .	46
2.18	Light Estimation . . . . .	48
2.19	Relighting . . . . .	50
2.20	Predicting Illumination . . . . .	52
2.21	Photometric Stereo . . . . .	53
2.22	Illumination estimation . . . . .	55
2.23	2D Compositing example . . . . .	58
2.24	DWAS fabric . . . . .	60
2.25	Disney MERL slices . . . . .	61
2.26	UE4 material representation . . . . .	63
2.27	Surface decomposition . . . . .	65
3.1	Diagram Overview . . . . .	68

3.2	Epipolar Geometry and Point Correspondence . . . . .	69
3.3	Stereo matching window . . . . .	71
3.4	Computing a disparity map . . . . .	71
3.5	Dynamic programming for stereo matching. . . . .	73
3.6	Cascaded Convolution . . . . .	74
3.7	Diagram pipeline for matching . . . . .	76
3.8	Dense mesh reconstruction . . . . .	81
3.9	Image-based lighting in our framework . . . . .	84
3.10	Indirect lighting . . . . .	85
3.11	Common shadow errors in compositing - scenario I . . . . .	86
3.12	Common shadow errors in compositing - scenario II . . . . .	86
3.13	Common shadow errors in compositing - scenario III . . . . .	87
3.14	Visualisation of umbra and penumbra effects . . . . .	89
3.15	Visualisation of the Rendering Equation . . . . .	90
3.16	Exponential Shadow Mapping . . . . .	93
3.17	Visualisation of the statistical shadow mapping function . . . . .	95
4.1	Evaluating scene consistency - stereo dataset . . . . .	100
4.2	Evaluation Chart for stereo depth reconstruction . . . . .	101
4.3	Chart of the stereo dataset overall time and input size . . . . .	102
4.4	Evaluating scene consistency - synthetic dataset . . . . .	103
4.5	Evaluating fault-tolerance . . . . .	105
4.6	Evaluating relighting consistency - stereo dataset . . . . .	107
4.7	Evaluating relighting consistency - synthetic scene . . . . .	108
4.8	Evaluating relighting consistency - near-stereo dataset . . . . .	109
4.9	Comparison of reconstruction . . . . .	113
4.10	Visualisation of surface orientation extraction . . . . .	113
4.11	Shadow comparison . . . . .	114
4.12	Statistical shadow maps . . . . .	115
5.1	Unity3D - HoloLens Spatial Mapping . . . . .	117
6.1	Compositing light sources . . . . .	123
6.2	Compositing in the presence of translucent media . . . . .	124
B.1	The modern rendering pipeline . . . . .	140
B.2	Result of dense scene reconstruction . . . . .	141
B.3	Result for shadow masking . . . . .	143
B.4	Additional result . . . . .	145
C.1	Illumination setup preview . . . . .	147



# List of tables

4.1	Performance overview - reconstruction of stereo dataset . . . . .	99
4.2	Performance overview - reconstruction of synthetic dataset . . . . .	102
4.3	Performance overview - reconstruction of near-stereo dataset . . . . .	105
A.1	Specification for the hardware used in this framework. . . . .	138
A.2	APIs and external library dependencies in our framework . . . . .	138
A.3	Listing of development tools . . . . .	139

# Chapter 1

## Introduction

### 1.1 Overview

Research advances made in the previous decades in the areas of machine vision, robotics, and computer graphics have led to the rise of new multidisciplinary research fields. Such fields are concerned with the study of illumination source estimation from flat 2D images, reconstructing full environments in 3D from a number of images (a process known as *scene reconstruction*), automatic annotation of objects in a movie, or a photograph.

The field of computer vision emerged from the need to simulate human perception of the surrounding environment in order to aid the field of robotics. Nowadays, computer vision plays an important role in more than just robotics – it lies at the core of powerful photo and video editing software widely used in the game and movie industries, and it will continue to be an important part of shaping the future of Augmented Reality (A.R.) technology.



Fig. 1.1 Examples of A.R. technology. Left-most image [104] is captured from the See Signal project achieved through hardware developed by Motion Leap which allows the user to visualize signal strength in a room using a bespoke head-mounted display (HMD). Centre-image [108] illustrates Microsoft’s HoloLens hardware featuring gesture interaction for positioning 3D artefacts in the real-world. Right-most image [4] showcases Apple’s ARKit technology made available on the phone and tablet, enabling the user to visualize composited 3D objects into a real environment. In all three cases, either a head-mounted display or hand-held tablet represent the medium where the surrounding environment and 3D artefacts are brought together.

Automatic inference of lighting conditions, depth, and material properties from an image or video is still an open question in computer vision – most of these are identified with the help of user input for accuracy. By striving to develop algorithms able to create accurate estimations of these properties, it is possible to create more intelligent software solutions, not only for robotics, but in a wide range of applications. One such application is realistic and seamless synthetic object insertion into photographs and videos.

Many applications dealing with data inference from an image do not benefit from a full automatic process – they rely on user annotation. This is due to the fact that the contents of an image may be influenced by a light source external to the image, but which is illuminating parts of the image, or casting shadows across it. Algorithms cannot correctly and efficiently infer light sources if they are not physically present in the image, and therefore resort to approximating lighting globally [9]. More than one illumination setup can render the same scene, which means there could potentially be a very large number of valid scenarios, and algorithms cannot reach those in real-time. In addition to this, depth cannot be annotated in an image, or specified by an inexperienced user, and the algorithms rely on the information from the hardware (i.e. depth of field data from a Kinect sensor). Even though advances have been made towards a general data-driven automatic illumination estimation algorithm, these techniques are not suitable for outdoor images which lack in structural information, and they do not account for the light that bounces off the different objects already present in the image [79]. This is why 3D scene reconstruction techniques are required in order to translate 2D data into accurate 3D geometry. Moreover, due to the rudimentary geometry extracted from 2D images, synthetic object insertion does not work correctly if the newly added object is obstructed partially by pre-existing geometry. In addition to this, material properties (density, specularity, reflectance) play an important role in creating accurate scene lighting, and current 3D scene reconstruction and synthetic object insertion algorithms do not handle transparent and reflective materials correctly (glass, water, etc.).

The area of 3D scene reconstruction for facilitating synthetic object insertion into 2D images or video presents a large number of challenges ranging from accurate and efficient illumination inference, to correctly determining the final lighting conditions after the object was added to the scene. This opens a number of opportunities for developing novel techniques, and improving already known ones. Current methods perform well if the scene illumination is diffuse, and as a consequence the object addition yields better results in indoor scenes, as opposed to outdoor scenes. This allows the research to be targeted towards developing novel techniques to deal with outdoor images (perhaps even taken with unconventional lenses that produce dramatic field of view distortion – fish eye lens). Current methods cannot deal with automatic material property inference or recreation of the material properties when it comes to 3D scene reconstruction, which means that there is room for contribution in the area of physically-based 3D scene reconstruction for object

insertion. Lastly, none of the current methods deal with synthetic object insertion into videos. A video is composed by frames (2D images) which means that it is possible to adapt the algorithm for object insertion into single 2D image to a sequence of single 2D images (video).

3D scene reconstruction for synthetic object insertion into images is a recent field that emerged due to demand from the advancements made within the creative industries. With the help of user input light sources were detected, and so was a simplistic reconstruction of the scene. Recent advancements strive to diminish and possibly eliminate user input. They set this foundation by providing automatic light estimation and depth estimation techniques in order to reconstruct robust 3D scenes. In the future these techniques will be perfected for carrying out real-time realistic synthetic object insertion in a single 2D image as well as throughout a video with no help from the user.

## 1.2 Contribution

Synthetic 3D object compositing into a scene (image or video) represents an operation required in various applications ranging from augmented reality to scene editing. These applications rely on the user’s knowledge and expertise in order to composite synthetic 3D meshes into a scene. In a vast amount of cases the user will generally rely on commercial solutions in order to annotate the images to specify minimum geometry, define the illumination conditions (light source location and shape or range), and even modify the 3D mesh topology in order to match the scene’s perspective. This process is cumbersome and often times inefficient on a large set of input data.

Recent solutions have emerged to address the challenges that come with the process of automating 3D object compositing, including techniques that rely on accurate depth mapping using the Kinect sensors [67, 59] in order to aid in the geometry acquisition step. Incipient techniques relied on coarse representation of indoor scenes through simplistic geometry (i.e. a box, a set of planes) [26]. These techniques rely on hardware generated data, or on pre-existing depth models, and are not computed for each individual set of inputs, therefore in some cases yielding incorrect results.

Furthermore, recent solutions are not equipped to handle the case of compositing of 3D artefacts behind the initial scene geometry [77, 1], as they focus more on modelling the general correctness of material representation. In most cases the scene illumination conditions are estimated globally or through an environment map, however these approaches cannot obtain the correct shadows in the newly created scene unless coarse geometry is specified at the very beginning of the composition step [34].

The framework presented in this thesis is a concise, light-weight, staged process that takes as input a stereo image pair, uses a matching and filtering kernel to provide a high quality disparity map which is specifically calculated to be compatible with a depth buffer during the 3D rendering step. In addition to this step, the algorithm approximates the scene illumination, and computes a realistic shading model that relates to the modifications made within the scene.

Generally, it was believed that inferring, annotating, or extracting scene geometry was a required step in object compositing solutions. The key insight of our approach is that 3D object compositing at the correct perspective, shading and illumination model is possible in the absence of user annotated scene information.

In order for the scene’s properties (illumination) to be consistent with the alteration created by inserting a synthetic object, a shading model inferred from existing geometry was traditionally required. In contrast, our approach recreates the shading model based on a global approximation in conjunction with point and directional light calculation from the input stereo image pair. Due to our choice of implementation for the illumination model, we bring another novel contribution to reverse-rendering and compositing, namely, the ability to composite light sources that contribute to the scene and produce convincing results, not just unlit static meshes.

Finally, we establish a hybrid shadow-mapping approach based on statistical techniques for the initial scene in order to account for physically-accurate soft shadows.

### 1.2.1 Chapter outline

Each chapter in the present document builds the context required around the developed framework, and aims to clarify the challenges in the field and demonstrate the contribution as a response to those challenges.

**Introduction.** This section offers insight into the multidisciplinary field of compositing and its applications to a wide range of industries that are continuously evolving, facing challenges with every technological advancement. Furthermore, the chapter briefly places our contributions in contrast with similar, established methods in the field.

**Background.** The chapter is split into two parts - a theoretical part aimed at explaining how the physical laws that govern light-surface material interaction are converted from a mathematical model to a computer simulation. The second part is the related work. This particular subsection discusses related work in relation to each stage of the framework developed as part of this project. Generally, these stages involve reconstruction of a 3D scene from images, approximating light and materials, and compositing.

**Methodology.** Three subsections form the body of the Methodology chapter, each subsection offering a detailed description of the inner workings relevant to each stage of the pipeline.

**Evaluation and discussion of results.** The first part of the chapter describes the motivation behind the design of each test, followed by results (figures and charts) that illustrate how the work performs in best as well as in worst-case scenarios. In this way, it becomes clear what the future direction for reasearch would be in order to address the weaker points of the work. The second part is a discussion that puts into contrast, where possible, intermediate results from our pipeline with intermediate results from similar published compositing techniques.

**Conclusion and further work.** This part reiterates the contributions, and lists limitations identified from the results and analysis section. Here, a number of directions for future research are proposed that can address the limitations (i.e. that are viable and worth-wile candidates).

**Supplementary material.** There are three pieces of supplementary material, each addressing a specific area related to the rest of the thesis. These materials include: additional figures to accompany the results, pseudo-code to explain implementations that are not as important as the ones previously defined in the Methodology section, and a full listing of all the (software and hardware) tools involved in the creation of the work.

## 1.3 Publications

Throughout the postgraduate studies, the author has published the following work related to the thesis:

1. "*Towards a general solution for compositing 3D artefacts in 2D photographs*", A.M. Mihut, G. Morgan, Doctoral Consortium, Eurographics April 2018
2. "*Lighting and Shadow Techniques for Realistic 3D Synthetic Object Compositing in Images*", A.M. Mihut, R. Davison, G. Ushaw, G. Morgan, Techical Paper and ACM journal, ICDPS ACM February 2018
3. In preparation: "*Inverse rendering techniques for representing complex material appearance in artefact compositing*" ACM Transaction on Graphics (TOG).

The following publications were achieved in collaborations with other research groups, where the author's knowledge and experience contributed to the projects' outcome:

1. "*Interactive Storytelling Using 360-degree Video and Light Fields*", F. chweiger (BBC), A. Sheikh (BBC), A. Brown (BBC), M. Brooks (BBC), P. Golds (BBC), B. Weir (BBC), G. Thomas (BBC), A. Cherbetji (Foundry), N. Redmond (Foundry), A. Mihut (Foundry), L. Hastings (Foundry). In The 16th ACM SIGGRAPH European Conference on Visual Media Production 2019 (CVMP).
2. "*Accurate real-time complex cutting in finite element modeling*", T. Xin, P. Marris, A. M. Mihut, G. Ushaw, G. Morgan. In: Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017).
3. "*Real-time ATO reconfiguration for operational stability*." A. Iliasov, I. Lopatkin, A. M. Mihut, A. Romanovsky. Software Engineering for Resilient Systems: 6th International Workshop 2015.
4. "*Graphic Processing Unit Simulation of Axon Growth and Guidance through Cue Diffusion on Massively Parallel Processors*", A. Mihut, G. Morgan, M. Kaiser. In: Dynamic Connectome Lab, Technical Report 2, 2014.
5. Project materials will be made available at <https://anadevmihut.xyz/Research>

# Chapter 2

## Background

### 2.1 Classification of object compositing techniques

Photo-realistic rendering of virtual artefacts (2D image areas, or 3D meshes) into real world scenes has been a topic of research that brought together various fields of study from computer graphics (i.e. for simulating illumination - matter interaction), to machine learning, and computer vision. Depending on the desired result and its application, a number of compositing techniques have emerged over the past decade, and can be split, at a top level, into two main categories: matting and compositing, and mixed reality compositing.

Matting and compositing techniques take as input a part of an image and attempt to *paste* it into another image, such that the result is seamless and photo-realistic. This type of technique operates only in the 2D domain (i.e. an image defined by a pixel-coloured values across its width and height). One of its immediate uses include image editing software or post-production operations. Another common characteristic for this type of technique is a reliance upon user expertise or guidance, through the specifying of annotations of the images; these annotations can take the form of individual points, or whole areas of an image, that are marked as being of particular interest to the composition process, such as light sources, reflective surfaces, or areas displaying a large amount of depth disparity.

Mixed reality techniques are broader in terms of input and implementation compared to their 2D counterpart algorithms, but their aim is ultimately the same - synthesize a plausible, realistic result. Mixed reality compositing operates in both the 2D domain as well as the 3D one. This means that the input could be one of a variety of options including a static single image, a stereo image pair, a video, or a synthetic 3D scene. The primary difference is that the artefact to be composited is not a part of a flat image, but a 3D mesh (static or animated). As these techniques can composite real-world objects into existing scenes, they are popular in the areas of product visualisation, augmented reality, mixed reality, and post-production compositing in film. Mixed-reality techniques,



however, still rely on annotation and user guidance, but their specialized hardware can gather various spatial data.

Our work is concerned with the 3D compositing approaches, and as such, specific related work will be discussed into further detail throughout the following sections of this chapter. In the present section, the intention is to acknowledge and provide references to the 2D techniques that have immediate application to their 3D counterpart.

### 2.1.1 2D techniques - matting and compositing

The *matte* extracted from the foreground describes the opacity characterizing a specific area. Digital matting and compositing techniques rely on a two-staged approach. Firstly, during the matting operation, foreground elements undergo extraction (masking). During the second step - compositing, these extracted elements are overlapped to a new background image. Traditionally, the foreground extraction step would rely on the use of a blue-screen (*blue screen matting*) [141], and later on *rotoscoping* [166]. Both of these techniques involve the user and expensive hardware (i.e. an optical printer). Moreover, they cannot account for modeling reflections, refractions, and shadows.

#### 2.1.1.1 Environment matting

Environment matting is the term describing compositing in the 2D domain. This process entails segmentation of the foreground and background elements in images, and using the matte (mask-like layer) to store parts of the new background that the foreground would obscure. This process is pioneered by Zongker et al. [174] and focuses on defining the properties characterising light-object interaction during light transport. In the incipient stages, the background of the objects is known(measured) [121], or estimated based on a collection of images which used a set of priors in order to extract light and material properties [163].

In more recent work, the compositing of areas presenting transparency and shadowing has been refined. Gutierrez et al. [55] simulate caustics in their compositing stage by firstly recovering a depth map where pixels are ordered based on gradient (darker values correlate to higher order depth). The depth map is then involved in calculating symmetry of the light transport process. For areas presenting transparency, a different approach had to be taken in order to achieve both plausible caustic shadows and the light at a grazing angle modelled commonly with the Fresnel equations. Yeung et al. [170] approach the matting and compositing of translucent objects from an efficiency aspect - their model allows for such areas to be composited without relying on 3D models to simulate the environment, rendering both caustics and reflected light. This approach comes at a cost, namely, heavy user annotation of the image based on the understanding of the scene's

properties encoder (as seen in Figure 2.1).

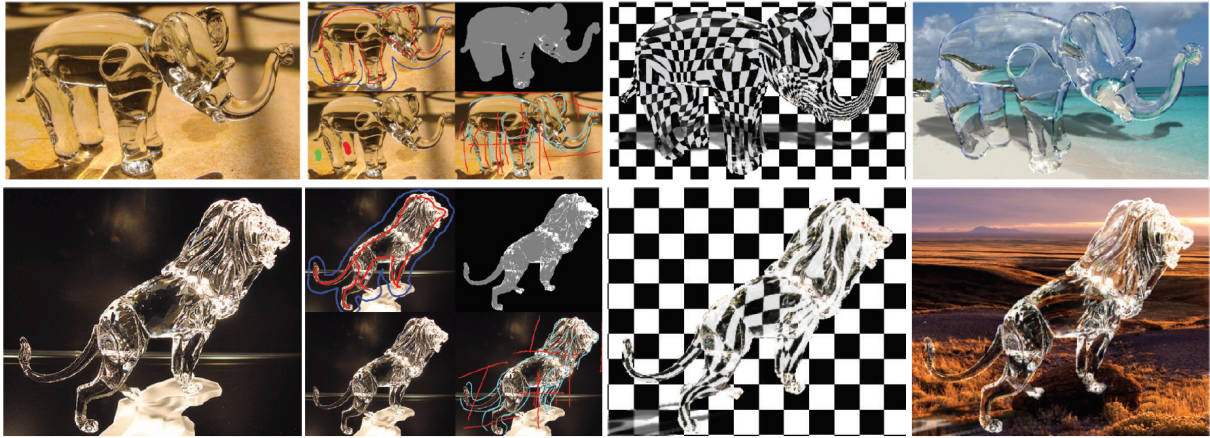


Fig. 2.1 Matting and compositing of transparent and refractive foreground into photographs achieved with the pipeline of Yeung et al. [170] that uses an *Attenuation-Refractive Matte*(ARM) responsible for encoding characteristic properties of transparent objects from observation. The user markup for extracting specularities comes in the form of short paths or strokes along the affected area. First column contains the input images, the second column illustrates the user annotation strokes and enclosed paths. The penultimate column contains the ARM, and the last column represents the composition result with Fresnel reflections and superimposed umbra region.

In the past, matting and compositing have been solved as an independent sequence of steps, however, the work of Wang and Cohen [159] illustrates that it is possible to handle both into a single optimization process without extensive input from the user. This was one of the first efforts towards automating the compositing pipeline in a 2D domain. Their approach assumes that both foreground and background are known, whereas, traditional methods treated the background as an unknown. This has primarily application in landscape photography retargetting, when objects appear small or far away in the vastness of the surrounding scenery, and the desired result is to enlarge those elements or bring them in focus.

### 2.1.2 3D hybrid techniques

This type of technique uses a broad set of inputs and vastly different approaches depending on the desired result as well as the domain of application - most approaches are offline, however, swift advancements in Augmented Reality (AR), Mixed Reality (MR), Virtual Reality (VR) and mobile technology have recently raised the interest for real-time compositing. Depending on the framework's end-goal and the chosen illumination model (grounded or estimated), we can broadly classify the hybrid techniques into two categories: those using measured light conditions, and those using approximated light models..

### 2.1.2.1 Measured light transport properties

The techniques described in this section have the common aim of measuring and accurately simulating light transport and light-material interactions that take place in various real-world scenes. This approach is investigated in the context of rendering artefacts (virtual objects in either the 3D or 2D domain) that can then be introduced into an artificial scene representation (backdrop) producing a result that is consistent with the initial illumination conditions. Although different methods of measurement and calibration rely on estimated parameters, the goal remains the same - to measure different parameters that are sufficient in describing information required to generate physically accurate renderings.

### Image based lighting and high definition formats

There is a minimum number of factors that play an important role in recreating a seamless 3D composite of an object into an image. These factors concern: illumination conditions, correct perspective, accurate light-geometry interaction, and accurate representation of the light-material interaction. These factors are often interdependent - the effects of light-material interaction are also dependent on geometry - a newly composited artefact could occlude a light source, or could itself be a light source, therefore changing the overall scene illumination. Capturing the scene's lighting conditions is paramount and a number of techniques involving dedicated hardware have been extensively used in motion picture production.

It is a common practice across image based lighting (IBL) techniques to represent the illumination through a single omnidirectional (or panoramic) image, known as either an *environment map* or *light probe*. This resulting image is characterised by its panoramic or 360° layout, and the high dynamic range pixel format (HDR) used to encode it. Once obtained, the image is applied as a texture onto a spherical or dome-like object, and used as a base for transferring the real-world light condition on to the 3D artefact soon to-be-composited. This transfer happens by sampling the texture - each pixel in the texture is representative of an incident light ray over the solid angle defined by that specific pixel. This approach is known as *image-based lighting* (IBL), and it originates from the reflection-mapping technique developed by Blinn and Newell [13], and further generalised based on a lookup table (LUT) approach by Miller and Hoffman [109].

Previous methods did not account for capturing the full range of light in a scene, and that was an impediment especially for modelling outdoor illumination conditions, which exhibit a high dynamic range. Debevec [34] developed a pipeline to address this issue by placing a highly reflective sphere inside the scene intended to capture the incident light, and record the scene in an HDR format. A few drawbacks of this approach concern the lack of information in the resulting texture related to the area in front of the sphere (which

would be obstructed by the viewing volume), and the opposite region position behind the sphere which forms a reflected area around the rim of the sphere's surface. This issue could be addressed by taking more than one photograph, and stitching the results, however, that could introduce further errors due to the different directions that each photograph used [143].

In the early years of IBL, HDR-capable hardware came at a considerable cost, and solutions emerged to address this problem. The idea was to use common low dynamic range (LDR) images and apply various post-processing effects for inverse tone mapping [91]. However, the results could sometimes suffer from overexposure which meant information was lacking in these areas and the environment map was not suitable for sampling.

The light probe design was further improved by Debevec et al. [35] as seen in Figure 2.2. Their novel design augmented the reflective sphere prop with a number of diffuse strips between the quadrants of the sphere. This was used to capture the incident light with only a single shot of the scene and relying on single exposure. The intensity of the multiple lights present in the scene could then be estimated by solving a number of linear equations. Most techniques that were based on this approach required further manual calibration and editing of either the input of the computed output [122].

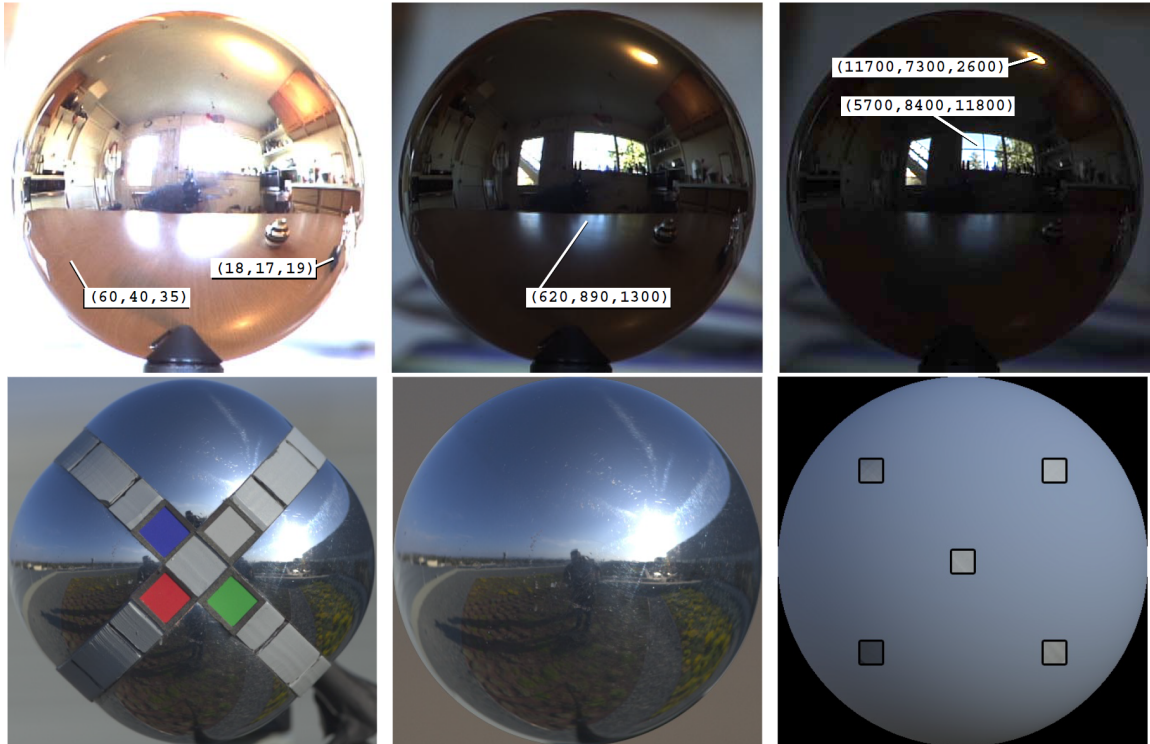


Fig. 2.2 Light probes for IBL. The first row illustrates the mirror sphere used by Debevec et al. in their initial set of studies [34] at different exposures. The second row contains the augmented light probe further developed by the authors [35] along with its digital representation.

## Spatially varying image based lighting

Traditional IBL techniques that rely on a single light probe to generate an HDR environment map present a limitation when it comes to offering insight into the interaction between the geometry present in the real scene and the composited 3D artefact. This becomes particularly difficult when dealing with multiple soft shadows, inter-reflections, or translucent objects that were already part of the original scene.

Illumination in a real-world environment is not uniformly distributed throughout the captured scene - this is due to the surfaces present in the scene as well as the nature of the light sources. In order to measure and reuse spatially varying illumination properties, it is necessary to capture the scene from more than one angle, in order to build an overall angular distribution. It is also possible to acquire this variation through a geometric model describing the structure of the scene, where light sources are clearly defined (parallax, position, direction). Depending on the desired result, a number of spatially varying IBL methods make different assumptions about the scene structure, and can be classified into three categories.

**Radiance sampling in the absence of geometry.** Techniques that belong to this class use a large variety of angular radiance sampling, and do not account for the geometry representation of the scene. In some cases, the geometry representation is minimal. Gortler et al. [51] pioneered the notion of incident light fields (ILF) which is used in this class of methods to represent a subset of the light field that concentrates on the region in the scene where the virtual object will be placed during rendering (compositing). ILF techniques aim to accurately capture illumination through interpolating adjacent sample points from within the HDR angular variation maps [156].

**Radiance sampling with rough geometry.** This particular type of technique is balanced compared to the other two classes - it uses a moderate amount of scene geometry representation or annotation (horizontal surfaces), and an HDR environment map. However, this approach is limiting due to its assumption that all the surfaces in the original scene are Lambertian (refer to Section 2.2.3.4 for more details). Compared to the previous technique, which relied on an exhaustive capture of spatially varying illumination measurement, the present method is applicable to dynamic environments such as film sets. In such a scenario, the capture has to be acquired quickly and then exploited through computer vision techniques for geometry inference [131]. The results produced by such techniques are reliable only in the cases of diffuse materials, and cannot be used as a generic application for modelling a variety of material properties.

**Radiance sampling with explicitly specified geometry.** Unlike the other two classes discussed, this method relies on a full geometry annotation and reconstruction

using both RGB-D hardware and user scene mark-up of surfaces that are likely to be used for compositing, light sources, and possible occluders. In addition to this, the light sampling comes from a larger number of HDR maps, or light fields (voxelized) correlated to a model of the scene. One of the first approaches was developed by Debevec et al. [36] and involved 3D scanning of the scene using calibrated lasers to capture outdoor environments. This system would capture both geometry and textures and would superimpose the later on the model. Furthermore, [171] expanded the technique for material inference using inverse global illumination. Some of the drawbacks of this technique include the requirement for user expertise when it comes to scene annotation, and dependence on designated hardware such as 3D scanners, RGB-D sensors, and HDR cameras.

### 2.1.2.2 Approximated light models

This category of illumination modelling has the same goal as the measured model category, and it aims to achieve that through estimation of the interactions between light and the various surfaces present in a scene, and not through calibration of hardware that measures this interaction accurately. The various estimation methods are built on exploiting shortcomings and features of the human visual system (amount of clutter in a room, shadowed regions which present little information, etc.). A note-worthy advantage for this type of techniques is the usability, as little to no user expertise is required for setting up proprietary hardware. Furthermore, this category of methods eliminates the requirement of light probe placement in the scene to capture variance, and generally requires less setup, which is time-consuming.

Approximated light models can be created from images, however the calculation of light transport properties from a single regular low dynamic range image is an ill-posed problem, with an infinite number of solutions that could generate the same observed scene. When assumptions are made with regards to defining structural properties of the scene and the illumination, the process of compositing artefacts becomes possible. Such assumptions are divided into material reflectance models (assume the surface is Lambertian), or in illumination distribution (enforcing specific priors from a collection of observed scenarios). Depending on how the scene illumination conditions are inferred from the scene, we can classify the techniques into three categories:

#### **Indoor light models based on known geometry and material properties.**

Methods of this category rely on the presence of objects with known material properties and known or simplified geometry representation. Rammamorthi and Hanrahan’s fundamental work [126] proposed to estimate the incoming radiance from images where lighting parameters were measured. The input data is acquired in this case using Microsoft’s

Kinect RGB-D sensor to record the depth and structural variation of the environment, along with texture data. Subsequently, this information is used along with an assumption of a completely diffuse world (the scene is entirely represented by materials that do not exhibit complex light interaction such as specular highlights or translucency).

Recent research looks into further automating aspects of the estimation and compositing pipeline [84]. This approach attempts to fit (by perspective and scale) 3D models to segmented and detected objects in a photograph, but relies on the same assumption of a Lambertian model for surface representation.

### **Indoor light models based on limitation of the human visual system.**

Based on a variety of studies related to human perception, it has been shown that the human visual system cannot recognise a wide range of illumination settings [128, 85]. In the context of 3D compositing of artefacts this means that accuracy of light-matter interaction modelling is not always required in order to trick human perception. Methods of this class exploit these limitations and aim to focus on estimating illumination to create a plausible output, not necessarily a physically correct one.

Recent work suggest that local illumination consistency (at the focus point of compositing) carries more weight compared to globally consistent illumination [117, 86], and this factor is often employed when modelling the artefact's scale, perspective, and reflectivity.

One of the first methods to apply these findings was developed by Reinhard et al. [130] and it involved transferring colour information obtained from the source image, on to the artefact to be inserted. This was a versatile approach since it coped with both static images as well as video streams, and it was an automatic process that did not rely on user annotation. The disadvantage of their approach was the relatively low quality of the result when rendering special materials.

A large number of methods from this category strive for automating parts if not the entire pipeline from geometry estimation to rendering a plausible result without relying on the user's expertise. One of the seminal works developed by Karsch et al. [77] relied on user annotation of both planar surfaces as well as visible light sources in the original photograph. After this information underwent classification, the user could composite multiple artefacts into the scene without any further adjustments.

The authors extended their previous work by automating the pipeline while constraining the domain of application [79]. The objects inserted could not be specular or any other kind of special material, the surface representation of the geometry in the present scene assumed a Lambertian model, and the scene's depth was estimated based on similarity with a large dataset of ground truth images. The method produces plausible results in indoor non-cluttered scenes only.



### Outdoor light estimation models

Indoor scene illumination can be modelled with a very specific set of approaches, which cannot be applied to estimating outdoor illumination. This is due in part to the type of geometry present in a typical outdoor scene, but mostly due to the nature of light and its range. Automated machine learning approaches for identifying in which one of the two categories a photograph fits have been developed as early as [44, 120], and can be employed to make an informed decision about estimating the light conditions.

The early work on outdoor illumination estimation [111] relied on hardware meta-data (time-stamp of the photograph) in order to model a number of essential physical properties (position, orientation, intensity) of the sun light. This information was used to illuminate the artefact which was then composited in the photograph. Lalonde et al. [92, 91] relies on the Perez's sky model [123] in order to estimate illumination conditions through relying on a sequence of input images in a time-lapse order. The assumption made is that throughout the stream, the scene information stays roughly constant while the illumination varies. This allows for extraction of several environment maps which are sampled for a high quality result. This method was further improved by the authors (Figure 2.3) and relies on a single input image, a number of priors from large image datasets, and a statistical approach to sampling the visible sky and cast shadows.

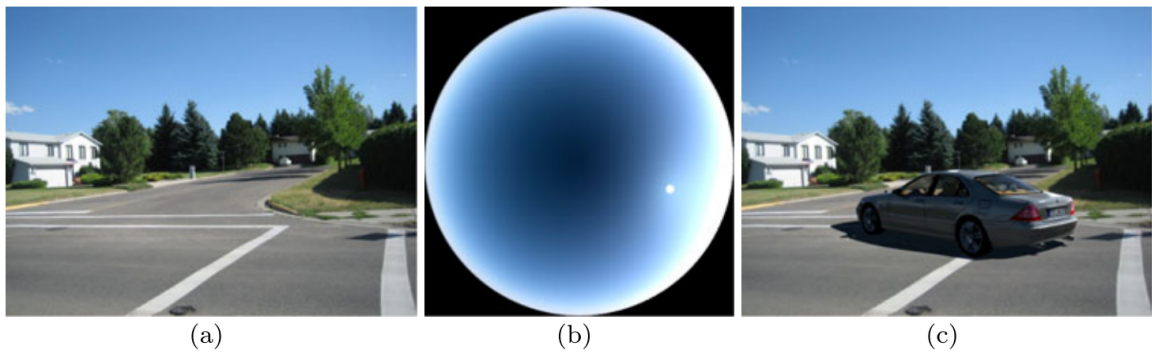


Fig. 2.3 Outdoor illumination estimation and compositing [92]. From left to right: (a) Original image. (b) Estimated sky appearance model. (c) Composition of a stock model.



## 2.2 Radiometry and the physically based rendering model

Physically based rendering encompasses a collection of techniques aimed at using the physical properties of light's interaction with matter in order to reach a visual fidelity comparable to real-world illumination.

### 2.2.1 Physical characteristics of electromagnetic radiation

Light is a type of electromagnetic radiation and the measurement of its properties through a variety of techniques encompasses the field of study known as *radiometry*. Light can be classified into three regions: ultraviolet light, visible light, and infrared light. Together, these three regions fit in the frequency range  $[3 \times 10^{11}, 3 \times 10^{16}] Hz$ . Generally, in Computer Graphics, the focus lies on understanding visible light, which has a wavelength restricted by a significantly smaller range of  $[400, 700] nm$  and is the only range sensed by the human visual perception system Figure 2.4.

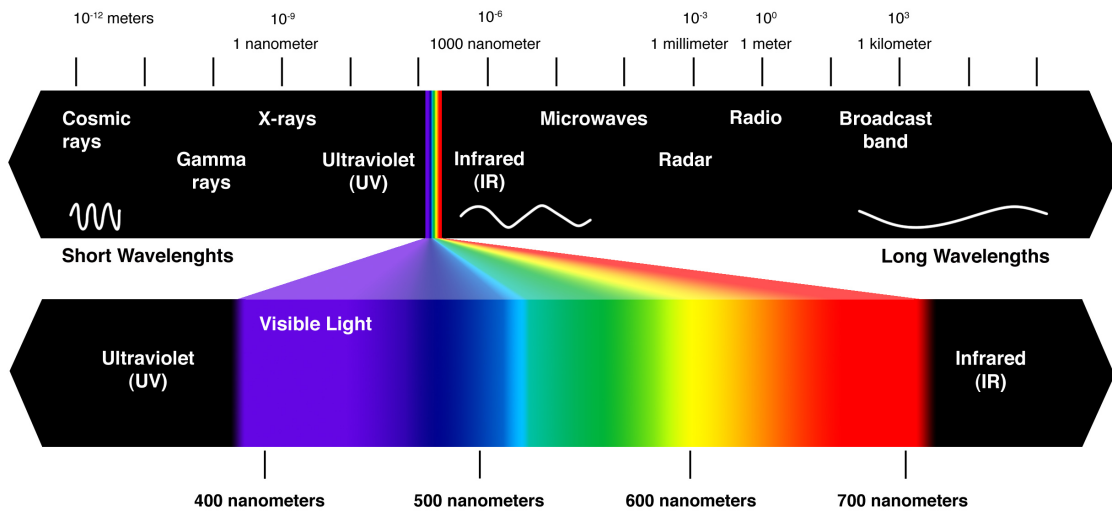


Fig. 2.4 The spectrum covers electromagnetic waves corresponding to a wide frequency range. This range is separated into bands, from radio waves (on the right of the axis) to gamma rays (towards the end of the left side on the axis). These groups can be further classified as long wavelengths (beginning of the band) or short wavelengths (end of the band). Source: Wikimedia Commons [2].

#### 2.2.1.1 Transverse nature of light and the plane wave model

Due to its electromagnetic nature, the radiant flux is transported through coupled fields, known commonly in literature as the electric and magnetic fields respectively (denoted  $E$

and  $H$  in Figure 2.5). They are modelled through periodic functions of space and time known as *time-harmonic fields*, specifically plane wave equations [14].

$$E = E_0 \exp^{-j(k \cdot x + \omega t)} \quad (2.1)$$

$$H = H_0 \exp^{-j(k \cdot x + \omega t)} \quad (2.2)$$

The plane wave equations for the electric and magnetic fields defined in (2.1) describe the sinusoid functions dependent on the wave vector  $k$  that describes the direction of propagation of the wave. The path of this wave can be evaluated through the function at each spatial point  $x$  that projects  $x$  perpendicularly onto the direction vector  $k$ . In other words,  $k$  is the normal to a plane, and all points on that plane have the same value. The scalar offset for the plane is controlled by the second term,  $\omega t$ , which simply says that as the time  $t$  increases, each plane moves away at a speed  $\omega$ . Thus, this complex exponential creates an endless series of moving planes of constant (complex) value (figure 2.5). This is particularly useful in defining the *wavefront* - movement of the electromagnetic field through space at constant phase.

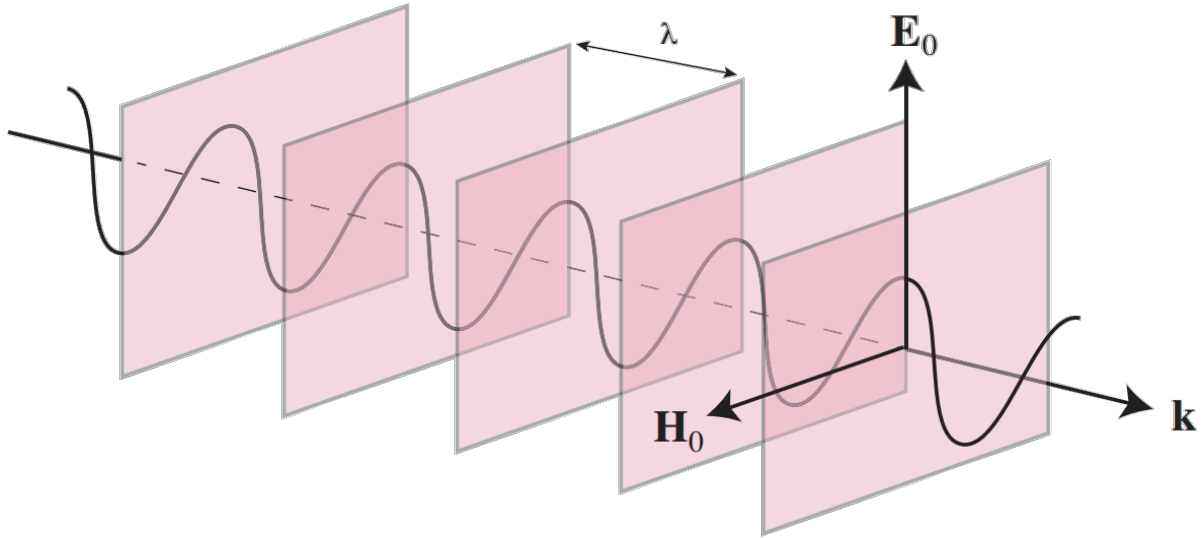


Fig. 2.5 The plane wave propagates in the direction of the 3 dimensional vector  $k$ . The electric field vector  $E_0$ , the magnetic field vector  $H_0$ , and  $k$  form an orthogonal triplet.  $\lambda$  denotes the wavelength, formally defined as  $2\pi/k$  which measures the distance from one peak to another. Source image: Wikimedia Commons [148].

To proceed in understanding the properties of light as electromagnetic energy, we turn to Maxwell's equations in a form specialized for plane waves [48]. These equalities define the relation between the different fields in a compact manner that takes into account the material(medium) through which the wave is propagating.

$$k \cdot E_0 = 0 \quad (2.3)$$

$$k \cdot H_0 = 0 \quad (2.4)$$

$$k \times E_0 = \omega \mu H_0 \quad (2.5)$$

$$k \times H_0 = -\omega \varepsilon E_0 \quad (2.6)$$

Equations 2.3 and 2.4 state that vector  $k$  is orthogonal to both the electric field and the magnetic field, meaning that the wave nature of light is in fact *transverse*. When the wave is homogeneous, then the two fields and the wave vector form a set of mutually perpendicular axes in 3D, as shown in figure 2.5.

Equations 2.5 and 2.6 define the properties of the medium denoted  $\mu$  for permeability and  $\varepsilon$  for permittivity. Based upon these equations, we can infer that the two material properties admit a plane wave, providing they meet a specific condition.

### 2.2.1.2 Polarization and the index of refraction

The equations 2.5 and 2.6 can be re-expressed in order to account for the *refractive* property of light upon interaction with a surface:

$$N = c\sqrt{\varepsilon\mu} = \sqrt{\frac{\varepsilon\mu}{\varepsilon_0\mu_0}} \quad (2.7)$$

where  $N$  represents the complex refractive index, and the constant value  $c$  the speed of light (travelling in a vacuum). In this case  $\varepsilon_0$  and  $\mu_0$  are indicative for the vacuum medium. The refractive index  $N$  is commonly represented in literature as

$$N = \eta + j\kappa, \{\eta, \kappa\} \geq 0 \quad (2.8)$$

in order to make the distinction between the *real index of refraction*  $\eta$  which measures the speed of light being affected by matter as compared to  $c$ , and the imaginary term  $\kappa$ , known as *the extinction coefficient* characteristic for the light's permeability through the medium, notably, how light loses its intensity as it converts to other types of energy during propagation (a process known as *absorption*). The value of  $N$  changes proportionally to the wavelength.

As previously discussed, the electric and magnetic fields are time-varying, however, they need not be radially symmetric along the propagation vector at any moment. Instead, these two fields have two distinct moments of oscillation: they can find themselves *in*

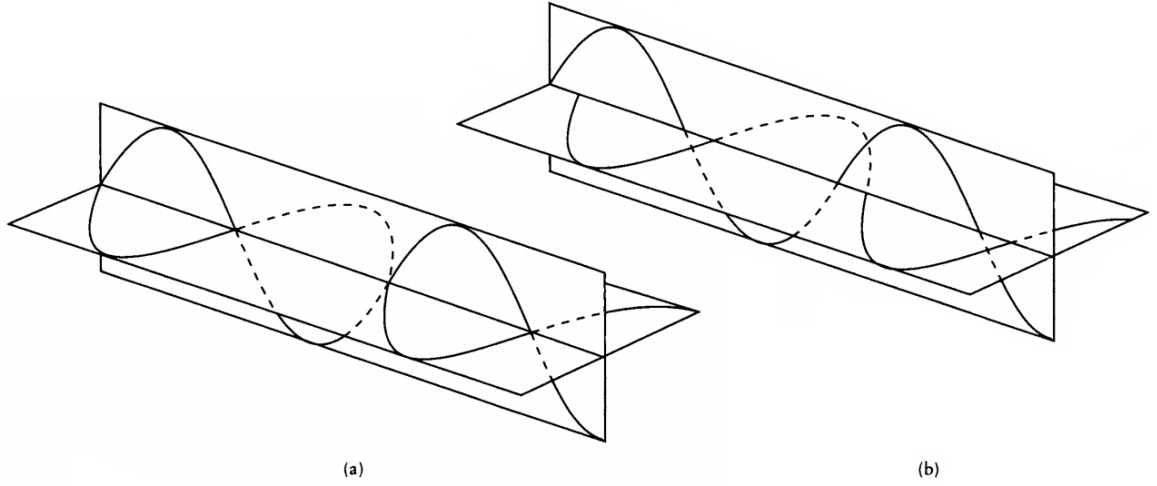


Fig. 2.6 (a) The two fields are in phase. (b) The two fields are out of phase.

*phase* in regards with one another, or *out of phase*. When they are in phase, the peaks and zero-crossing occur at the same location along the  $X$  axis (Figure 2.6.a). By contrast, when they are out of phase, their zero-crossings and peaks find themselves at different locations 2.6.b. This swap occurs in a cycle [52]. If we consider either field ( $E$  or  $H$ ), we could express its wavefront by expanding the complex exponential:

$$F_{E,H} = A \cos(\kappa z - \omega t) - B \sin(\kappa z - \omega t) \quad (2.9)$$

Upon considering the field at the point where plane  $z = 0$ , the equation 2.9 becomes

$$F_{E,H} = A \cos(\omega t) + B \sin(\omega t) \quad (2.10)$$

We can think of the equation 2.10 as the curve swept out by the tip of the electric field in the plane  $z$  as it moves through space. This curve takes the shape of an ellipse, generally characterized by two axes and one *azimuth angle*  $\psi$  measured relative to an arbitrary axis. Depending on the relation between the major and minor axis, the curve described can become a circle ( $A = B \neq 0$ ) or even degenerate into a line ( $A = 0$  or  $B = 0$ ). In these cases we call the light as being elliptically polarized, circularly polarized or linearly polarized according to case. In the case of a stationary plane sweeping the curve regardless of shape, the ellipse is considered to be a *vibration ellipse*, where the value of the 2 defining axes and the azimuth angle are known as *ellipsometric parameters*. The shape and structure of the vibration ellipse reveals the relationship between the phases of different components of the electric field; this relationship is called the *polarization* of the field.

Polarization is important in physically based rendering because particular materials respond differently to light of different polarizations. Examples of such materials include crystals (quartz, tourmaline, etc.), plastic film, clear glass, undisturbed water. If the real part  $\eta$  of the complex index of refraction  $N$  varies with different forms of linearly polarized light, the material is said to be *linearly birefringent* [49]. If the imaginary part  $\kappa$  varies, the material is *linearly dichroic* [107]. Similarly, circularly birefringent and circularly dichroic materials are sensitive to the degree of circular polarization in the incident light. When simulating light-material interaction in computer-based simulations, it is important to treat the scenarios separately, as dichronic surfaces will appear to split the incoming light ray into two distinct coloured rays, while the linearly birefringent surfaces will cause a phenomenon of double refraction (the refractive index is dependent on the polarization of the incoming ray).

### 2.2.2 Physical interaction of light with media

The simplest form of a PBR model takes into account light's physical properties and uses those in solving the interaction of light with the surrounding matter, specifically, the amount of light refracted and reflected (specular and diffuse) dependent on the medium's homogeneity. More complex models take into account phenomena beyond diffuse and specular reflections, such as interference, diffraction, and subsurface scattering. *Interference* accounts for the brilliant colours that we see in thin films, including peacock feathers, oil slicks, and soap bubbles. *Diffraction* is responsible for some (though not all) soft shadows and light bleeding around the edges of objects. *Subsurface scattering* is a special case of interference, that commonly models light - skin interaction.

#### 2.2.2.1 Reflection and transmission

In the previous sections, we have described a number of fundamental characteristic of light. In order to understand its interaction with matter in a way relevant to computer graphics applications, we have to classify the commonly occurring mediums based on their material properties, such as: optically fully transparent, partially transparent, optically opaque, isotropic/anisotropic, smooth/rough, emissive/absorbent.

When we consider the ideal case of capturing light's interaction with matter, we turn to a medium characterized by a constant index of refraction (*homogeneous*). For a medium which appears to be fully transparent (i.e. glass, water, diamond),  $\eta$  holds low values at wavelengths where the light is noticeable to the human eye. In this case, the absorption process is neglected, and as a consequence, the incoming rays transported through the environment continue to traverse it without a direction change (*scattering*). If the homogeneous medium would be significantly more absorbent, light's intensity would decrease noticeably upon interaction. Regardless of scattering or absorption, the

incident light rays can potentially only decrease in intensity whilst their trajectory remains unchanged.

### Light interaction in a heterogeneous environment

In contrast to the homogeneous medium interaction, the index of refraction varies across a heterogeneous environment. When the variation of the index of refraction (denoted  $t$  in Figure 2.7) takes place at constant acceleration and continuously then the observed light behaviour follows a curved path. When  $t$ 's variation peaks abruptly (occurs when the magnitude of the incident ray is shorter than the wavelength), the light ray *scatters* - it splits in different directions but its total distribution does not change [15].

In addition to absorption and scattering, there is a third type interaction between light and surfaces, namely *emission*, where new light is created from other forms of energy (opposite of absorption). This aspect is not commonly accounted for in the PBR model. Real-world surfaces present complex material properties, and therefore both scatter and absorb light concomitantly and the way we perceive their appearance is entirely dependent on the viewing angle as well as the total amount of scattered and absorbed light at the contact points.

**Fresnel equations.** If we consider the ideal case of an infinite, perfectly flat planar boundary between two media with different refractive indices, Maxwell's equation could be illustrated as:

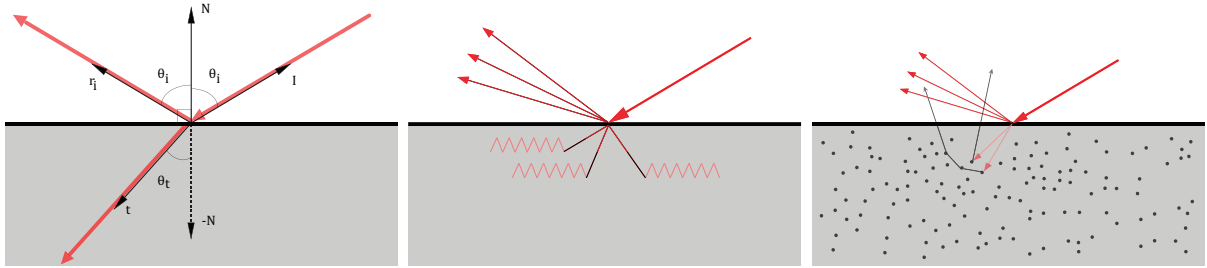


Fig. 2.7 Left-most image illustrates the phenomena of light rays scattering as the value of the refractive index ( $t$ ) varies from one medium to another. Centre image - In mediums exhibiting metal-like appearance, refracted light is completely absorbed. Right-most image - Dielectric materials do not present a uniform surface, and as a consequence, all refracted rays scatter (split) randomly. In some cases, where the energy was not entirely converted, the rays traverse and exit the medium, after undergoing some amount of absorption. This process repeats until intensity reaches zero.

Although analytical solutions do not exist for computing the behaviour of light at the change in value of the refraction index, Maxwell's equations for the ideal case defined above are known as *Fresnel equations* and shading algorithms heavily rely on the ratio described by them.

In the ideal case, the incoming light ray is halved - one ray represents light that is reflected by the surface from the contact point, and the second half represents the refracted ray, that enters the medium through the contact point. The incident angle measured between the entering ray and the surface's normal is equal to the angle of reflection measured between the normal and the exiting light ray. The refraction angle is dependent on the value of refractive index characterising the specific medium. The Fresnel laws describe this proportion of reflected to refracted light taking into account an energy conservation principle.

**Scattering in anisotropic surfaces.** Most surfaces that we encounter in the real-world are not optically-flat (uniform across the entire medium). Irregularities are present throughout the medium, and are responsible for how light reflects, however they are smaller in coverage than the area of a single fragment sample (an area that the shading kernel will colour into the final position rendered on screen). In order to model this behaviour, a surface is represented as a collection of microscopic optically flat surfaces (planes). The overall (macro) surface appearance is calculated as an average of sampled points from each micro surface that has different surface orientation. When the micro surfaces are aligned, the reflected light is focused in that area giving the surface a shiny appearance. When the sampled points belong to surfaces with variance in orientation, the appearance of the material is blurred, making the surface look rougher (seen in Figure 2.8).

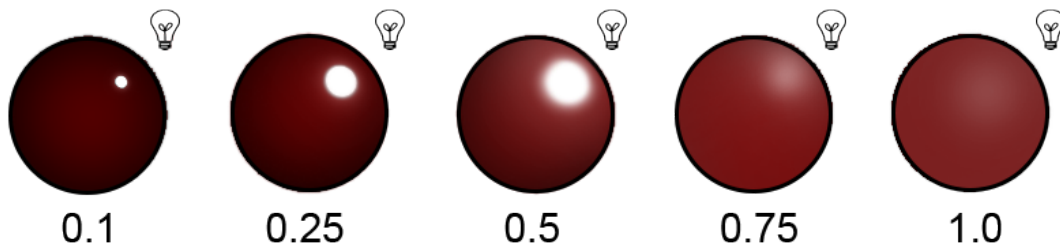


Fig. 2.8 From left to right, each sphere illustrates increased roughness values. A high value (1.0) produces a wider specular coverage, whereas a smaller value results in a well defined, small and crisp area of specular reflection. The blurred surface when the roughness value reaches maximum is due to the variation in micro surface orientation.

When the surface appearance is rough due to the mismatch in alignment between the microscopic and macroscopic surfaces, the shading model adopted to best represent this interaction is through a statistical distribution. The surface is viewed as reflected and refracted light in multiple directions for each point in the sample.

**Subsurface scattering in non-optically flat surfaces.** This part concerns the refracted light. In the case of metallic surfaces (fully opaque), refracted light is fully

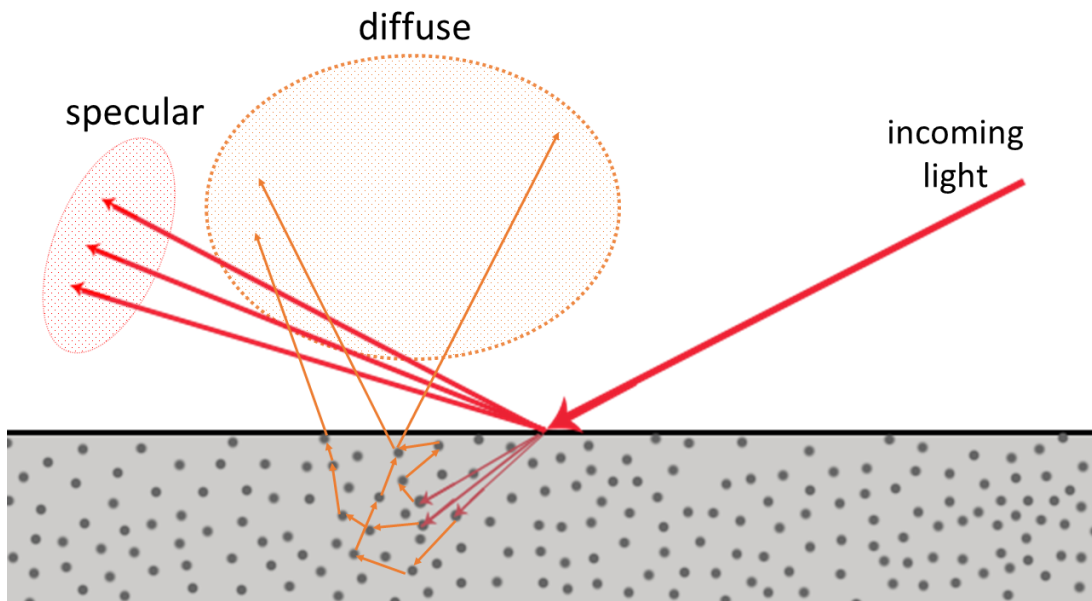


Fig. 2.9 A surface cannot exceed reflectivity of 100% of the incoming light energy for the BRDF to hold. The specular term is used to represent surface reflectance, while the diffuse term is used for scattering.

absorbed due to the values of  $\kappa$  (the imaginary part of the refractive index covered in the previous section) in the visible spectrum. Non-metallic media (known commonly in the related literature as *dielectrics*), interact with light just as regular participating environments once the light is refracted inside causing scattering and absorption of energy, until the light ray exits the medium or loses its intensity (*transmission*). The different behaviours of light transport depending on the surface model can be observed in Figure 2.11

### Energy conservation in the PBR model

The typical BRDF model does not account for interaction with emissive surfaces. This is because the outgoing radiance should not exceed the incoming light energy. In Figure 2.8 the roughness level increases along with the area defined by the specular reflection, however, at the same time, the overall brightness decreases. Smooth (optically flat) surfaces present intense and well defined specular reflections unlike rough surfaces (microfacet alignment varies), which produce dim specular reflections.

BRDF models make a clear distinction between the specular and diffuse light, in order to compute them as two separate terms. From the physically-based approach, for each contact point on the surface, the incident light ray is split into exactly two vectors: the refraction ray and the reflected ray, where the latter one forms the same angle as the incident does with the normal to that surface, however it is of inverse direction. This



ray does not enter the surface of the medium, and is known in literature as the *specular lighting* term. The *diffuse lighting* term is used to refer to the refraction ray which enters the surface and is involved in scattering, absorption (in the common BRDF model, but not in the real-world), and transmission.

In the real-world, the refracted light ray does not immediately get absorbed at surface contact. As discussed in Section 2.2.1 previously, the light ray transport occurs at the same pace and intensity until a collision with another surface takes place; light energy is then converted to other forms of energy, and the incoming light ray loses its intensity and changes direction. Figure 2.9 illustrated such a collision between refracted light ray and the material's defining particles. These microscopic constructs absorb the light ray energy for each collision, energy that in this case is subsequently converted into heat.

Outside of computer graphics simulations which aim to approximate light interaction with surfaces, not all diffuse radiant energy is absorbed once an incident ray enters a surface. In the real-world case, the light ray will continue to traverse the medium, scatter within it depending on the index of refraction and density of the medium. In most cases the variation of trajectory is abrupt, and collision with other particles can take place multiple times until the ray's energy is depleted or the ray exits the medium again with less intensity and can interact with other surfaces or re-enter the initial medium.

During the transmission process, the scattered light rays can exit the medium's surface and in this particular case they do aggregate in the diffuse term of the surface's overall appearance (the observed colour). Due to the energy conservation model in place during the physically-based rendering phase of the forward pipeline, refracted light rays are immediately absorbed and scattered locally, whilst the transmission effect (would follow the phase of scattering inside the medium) is ignored. Generally, methods that tackle both initial refraction and transmission aim at decoupling the two, and compute the initial refraction and reflection step using an energy conservation BRDF model, while the transmission term is calculated as part of the *subsurface scattering* model. These methods can be seen when materials such as paper, skin, or wax are being rendered, however, they do incur a significant time penalty.

A single BRDF model is not a generic solution capable of representing multilayer materials, or complex appearance. In addition to subsurface scattering methods, metallic surfaces can be represented through specialised models. This is because electrics interact differently with the incident light compared to di-electrics - the refracted rays are absorbed upon contact with the surface meaning that scattering and transmission do not take place. Only specular areas are calculated, and the diffuse colour is not the result of an aggregation as is observed in di-electric materials.

Due to this difference, physically-based rendering pipelines in both production (offline) as well as in real-time applications choose to make a clear distinction and use specialised kernels to represent metals and di-electrics. This aspect will be discussed into greater detail in Section 2.4.4.

The energy conservation principle that lies at the foundation of the BRDF involved in physically-based rendering techniques makes the reflected light rays and refracted light rays to be mutually exclusive when it comes to contributing to the final fragment sample colour - the energy undergoing reflection escapes the transport through the medium and therefore cannot be absorbed by that specific medium. The remaining energy that is refracted and traverses the medium represents the result of the total amount of incoming energy without the reflected energy.

## 2.2.3 Mathematical model of light-material interaction

### 2.2.3.1 The micro-facet model for surface representation

In the real-world, optically geometrically flat surfaces are hard to come by, and are generally used in precise optical devices such as telescopes. Due to the rare occurrence of these types of surfaces, it's natural to assume that at a microscopic scale most surfaces present irregularities. In order to model non-uniform surface reflectivity, the assumption is that the surface currently evaluated can be represented through a number of microscopic statistically distributed facets (mirrors), each oriented according to a given probability distribution function [150]. This surface can be either *isotropic* or *anisotropic*. Ultimately, it is this orientation that accounts for the roughness of a surface in the visible spectrum.

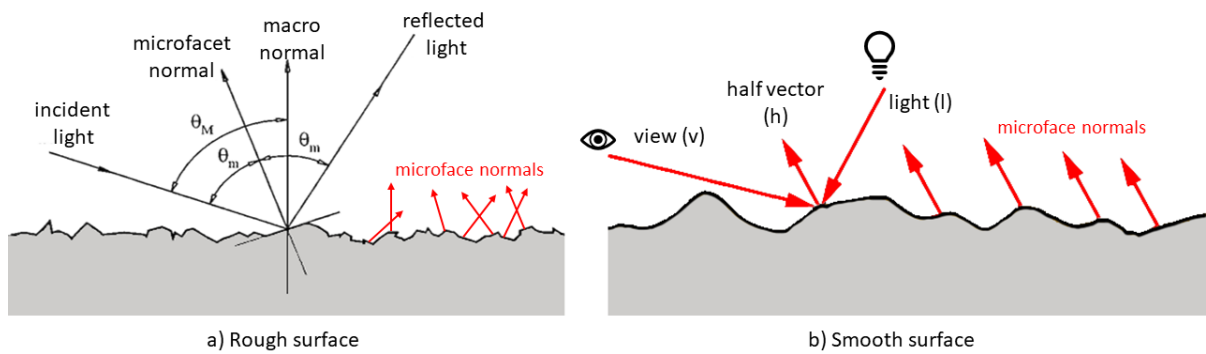


Fig. 2.10 Specular light-matter interaction. a) Incident ray coming in contact with a point on an anisotropic rough surface. The microfacets are distributed with different orientations causing light to bounce or scatter in various directions across the surface. b) Scattering in a smooth surface. The microfacets' orientation is even, therefore the light bounces in one direction.

If we take into consideration the energy conservation principle described in the previous Section 2.2.1, we can efficiently simulate light interaction by calculating the reflectance based on the incoming light, viewing direction, and amount scattered for each micro-facet and then average the result. In computer graphics, this behaviour is elegantly encompassed in a single equation known as *the rendering equation*[73]:

$$L_o(x, \omega_o, \lambda, t) = L_e(x, \omega_o, \lambda, t) + \int_{\Omega} f_r(x, \omega_i, \omega_o, \lambda, t) L_i(x, \omega_i, \omega_o, \lambda, t) (\omega_i \cdot n) d\omega_i \quad (2.11)$$

where the left term of the equation describes the total spectral radiance of wavelength  $\lambda$  directed outward along the direction vector  $\omega_o$ , at time  $t$ , from a particular position  $x$ . The right hand side term can be divided into the emitted spectral radiance term denoted by  $L_e$  as a function of wavelength, and the integral over a unit hemisphere denoted  $\Omega$  around the surface normal  $n$  containing all possible values for the incoming negative direction of light  $\omega_i$ . The term  $f_r$  is a the bidirectional reflectance distribution function which represents a proportion of light reflected from  $w_i$  to  $w_o$  at position  $x$ .

The rendering equation is generic, and for that reason we cannot resort to it for capturing all possible light-matter interactions. Some of the cases which require a specialized formulation include (Figure 2.11): transmission (light exits the surface without being scattered), polarization, and non-linear effects such as emission (which breaks the energy conservation principle). Predominantly, the materials exhibiting complex interaction are dielectrics, which are a common occurrence in the real-world.

### From the rendering equation to the reflectance equation

We recall the rendering equation 2.11, and each contributing member:

$L_o$	outgoing light ray exiting from a point $p \in$ surface $S$
$L_e$	the ray being emitted from the point on the surface
$\int_{\Omega} (...)$	the functions will be integrated over all directions in the hemisphere $\Omega$ above the surface, where $\Omega$ is orientated by $n$
$f_r$	first term of integral is the BRDF. Calculates a ratio of reflected light to amount received from direction $w_o$ .
$L_i$	second term of integral is an incoming ray at point $x$ having the same direction as the receiving light used in the BRDF. This light does not necessarily have to come from a light source; it can be either reflected or refracted (indirect).
$\omega_i \cdot n$	third term of integral is the normal attenuation based on the angle measured from the incoming ray and the medium's surface.

Due to performance concerns, the integral can be approximated by splitting it into two domains: direct and indirect light. Moreover, PBR models do not take into consideration

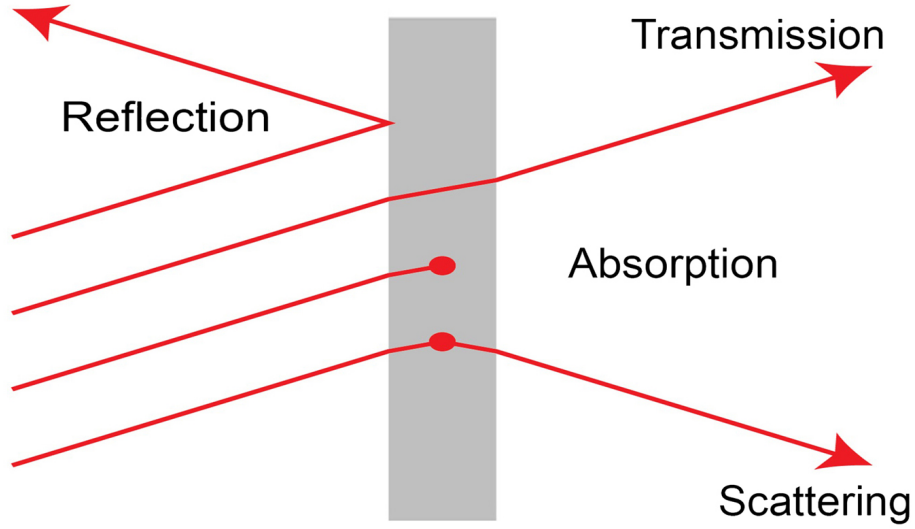


Fig. 2.11 Primary types of interaction upon the incident light making contact with the surface. Reflected light does not enter the surface, instead scatters at an angle equal to the incident angle between the ray and the normal to surface. Absorption happens once the incident ray enters the surface. Scattering occurs after the incident ray entered and bounced inside the surface, then exited. Transmission occurs when the incident light ray traverses the medium without losing intensity due to little to no collision or internal scattering.

the emitted light, and usually rely on textures to model the change in BRDF over a surface. By rewriting the rendering equation with these new observations we get the expanded **reflectance equation**:

$$L_o(x, \omega_o) = \int_{\Omega} f_r(x, \omega_i, \omega_o) L_i(x, \omega_i) n \cdot \omega_i d\omega_i \quad (2.12)$$

In real-time rendering literature, the above equation is further compressed by replacing  $\omega_i$  with  $l$  to denote the light direction, and  $\omega_o$  with  $v$  to denote the view direction. Both of these values are unit-length vectors, and will be used in replacing the BRDF with  $f(l, v)$ . This means that the directions can be defined by two parameters such as polar coordinates. By replacing the terms in the initial reflectance equation 2.12, we obtain the simplified form:

$$L_o(v) = \int_{\Omega} f(l, v) \otimes L_i(l) (n \cdot l) d\omega_i \quad (2.13)$$

Where the component wise operator  $\otimes$  specifies multiplication between the BRDF and light colour (RGB) vector components.

### 2.2.3.2 The bidirectional reflective distribution function

In radiometry, there are a number of radiometric quantities that measure light over a surface, however, for PBR techniques only one such quantity is of interest, namely *radiance*.

**Radiance** describes the magnitude of light along a single ray. In the context of the rendering equation, radiance is denoted as  $L$ , where variations such as  $L_i$  stands for incoming radiance, and  $L_o$  for outgoing radiance. Due to its spectral nature, radiance varies as a function of wavelength. To represent this value, normally, dense spectral samples would be used, however, for performance reasons in rendering, the RGB channels are used instead.

The bidirectional reflective distribution function (**BRDF**) represents the surface's response to incoming and outgoing light that interact with it. These two sides of the BRDF can solve different problems:

1. Given an incoming light ray defined by a direction and an angle away from the surface, the BRDF describes a distribution of the amount reflected and scattered through all the outgoing rays above this surface.
2. Given a viewing vector, the BRDF describes a contribution of light rays from each incoming direction following the outgoing light.

**Light energy:** As seen in Figure 2.12, the transported light beam (energy) is denoted  $\Phi$  and represents the radiant flux of energy characterised different wavelengths.

Physically-based rendering techniques are interested in modelling the visible colours on the electromagnetic spectrum. The flux originating from a light source is defined as a function of wavelengths, measuring the entire area under which light interacts with surfaces, from the incoming light rays to the emitted rays. When calculating this discretely, the simplified version of radiant flux is adopted - each colour channel is encoded as an RGB value ranging from 0 to 255. This incurs a penalty of accuracy, however, it does exploit shortcomings in the human visual system, yielding believable lighting conditions.

**The solid angle.** The solid angle seen in Figure 2.12 as  $\omega$  measures the size of the area of a shape projected (stretched) on a unit sphere (radius is one). The surface area of the projection forms the solid angle - a mathematical construct defined by a direction and its corresponding volume.

**Intensity of the radiant flux** is a term that refers to the amount of radiant flux inside the solid angle. This is used in calculating the energy (strength) over the projected area, and can be defined as seen in equation 2.14

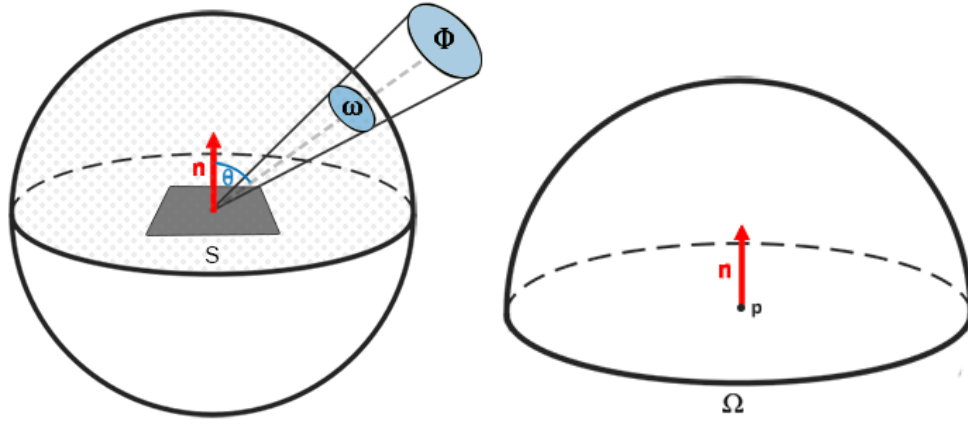


Fig. 2.12 Left image represents the sphere volume encapsulating a scene. Right image represents the hemispherical surface over which the Rendering Equation is defined.

$$I = \frac{d\Phi}{d\omega} \quad (2.14)$$

where  $\Phi$  denotes radiant flux corresponding to the volume described by the solid angle  $\omega$ . The radiant flux and intensity at a solid angle are knowns and can be plugged in the radiance equation  $L$  for the total observed energy over the area:

$$L = \frac{d^2\Phi}{dA d\omega \cos\theta} \quad (2.15)$$

One important observation is that surface appearance depends on the incident angle. Radiant flux dictates the quantity of light energy in a volume or across a surface scaled by the incident angle  $\theta$  (distance between surface and incident light ray), or sometime calculated as  $\cos\theta$  when relative to the surface's global normal vector and the incident light ray. The intensity is dimmer when the angle value is small, and stronger when the incident vector coincides with the global normal (orthogonal to the surface  $A$ ).

When the volume defined by the solid angle  $\omega$  in relation to the area  $A$  is small, the intensity of a single incident light ray coming in contact with  $A$  and point  $p$  can be calculated using the radiance equation in a fragment shader program. The solid angle's direction vector, the plane orientation and the point on the plane are the minimum parameters required to calculate the ray's contribution per-fragment sample (pixel).

**Irradiance** The rendering equation 2.12 is defined by the radiance  $L$  at point  $p$  and the solid angle  $\omega_i$  (an oriented volume obtained from light's projection). The irradiance

equation computes an aggregate of reflected radiance denoted  $L_o(p, \omega_o)$  at point  $p$ , facing in the viewing direction  $\omega_o$  (emitted rays).

### 2.2.3.3 Surface reflectance - the specular term of the BRDF

The incoming radiance vector  $\omega_i$ , together with the viewing direction (of the camera usually)  $\omega_o$ , and the micro-surface's defining parameters  $a$  (roughness) and  $n$  (normal vector) represent the minimum input values to most BRDF models. Using this input, the function estimates the quantity contributed to the final surface appearance per each incoming light ray that is reflected at contact with the surface. When the roughness value is small and the appearance of the surface is smooth (isotropic), the result obtained from the BRDF would approach a value of zero for the entire incident light, with one exception - for the ray that has the same incident angle value as the outgoing light ray (reflectance model will always be equal to one).

Micro-facet theory for physically-based rendering techniques makes an assumption regarding the uniformity of the micro-geometry - the variation of the geometry at this scale cannot be immediately observed, however, its value is greater than visible light frequency. In this way, physical rules of optics still govern whereas the possible effects of wave transport such as diffraction can be ignored. Furthermore, this theory is only used to model single ray bounces of reflected incoming light where the components of the incoming light vector as well as the view vector (outgoing light) are known *a priori*.

### The halfway vector and micro-facet shadowing

Out of all of the points on the surface, only those that are oriented in such a way as to reflect the incoming light into the viewing direction could potentially contribute to the BRDF total value. In such a case, the surface normal  $m$  is oriented halfway between  $l$  and  $v$ . This is called the *halfway vector* or *half-angle vector*  $h$ .

Even though  $m = h$  for a point  $p$  on surface  $A$ , that is no guarantee that the point will contribute to the final reflected vector; some points are occluded by neighbouring facets' orientation from the incoming direction  $l$  causing *shadowing*. In other cases, they will be obstructed from the outgoing direction  $v$  causing *masking*. In some cases, both interactions can take place concomitantly. Micro-facet theory cannot deal with interreflections due to the energy conservation principle, and the BRDF models assume that all shadowed light is does not contribute to the specular term. In real-world situations, due to interreflections, the surface will eventually be visibly affected. According to these assumptions, a BRDF

can be derived from first principles [6], and the specular BRDF term can the following form:

$$f_{\mu facet}(l, v) = \frac{F(l, h)G(l, v, h)D(h)}{4(n \cdot l)(n \cdot v)} \quad (2.16)$$

$h$	is the half-vector.
$l$	is the incident ray.
$v$	is the outgoing view vector.
$D(h)$	is the micro-geometry <b>normal distribution function</b> (NDF) at $h$
$G(l, v, h)$	is the <b>geometry term</b> . It represents a ratio of surface points with $m = h$ that are <i>not</i> occluded by micro-geometry.
$G(l, v, h)D(h)$	calculates the concentration of <i>active points</i> on surface contributing to reflectance (direct $l$ into $v$ ).
$F(l, h)$	is the <b>Fresnel reflectance</b> of the subset of points as a function of the light direction and micro-geometry normal $m = h$ . amount reflected (scalar).
$4(n \cdot l)(n \cdot v)$	is a correction factor for transforming between local and micro-geometry space.

### Fresnel reflectance term

focuses on calculating the ratio of the amount of light reflected from an at contact with micro-surfaces to the refracted amount and is conditioned by two factors: the angle  $\theta$  measuring the distance from the incident ray and the surface contact point, and the refractive index of the material. Due to its spectral nature, we treat the Fresnel reflectance as an RGB triple. Moreover, each value of the triple has to be in the  $[0.0, 1.0]$  range (surface reflectivity cannot be negative and more than 100%). Numerous materials, present a constant reflectance value for incoming angles  $\in 0^\circ$  and  $45^\circ$ . Thea appearance of surfaces changes significantly for reflectance values between  $45^\circ$  and  $75^\circ$ . In the interval 75 to 90 , the reflectance increases abruptly to 1 (white, if viewed as an RGB colour).

In order to calculate the reflectance value per surface point using the Fresnel model, the surface normals have to be known in order to determine the half-vector  $h$ . The incident light angle for the Fresnel equation measures the value between the half-vector and the incident light ray  $l$ . In BRDF models, the reflectance coefficient  $F_0$  tends to have a value close to  $0^\circ$  which characterises most surfaces that can demonstrate a colour value (fragment samples can contain any value in the range 0 to 255 for the RGB triplet or normalized between 0 and 1). In physically-based rendering, this term is known as the *specular colour*



(parameter) characterising a surface. Schlick [133] formulates an estimation for  $F_0$  that works for most materials:

$$F_{Schlick}(F_0, l, n) = F_0 + (1 - F_0)(1 - (l \cdot n))^5 \quad (2.17)$$

In a BRDF, the active surface points with a normal of  $h$  must substitute it for the surface normal  $n$  and therefore obtain:

$$F_{Schlick}(F_0, l, h) = F_0 + (1 - F_0)(1 - (l \cdot h))^5 \quad (2.18)$$

Metallic materials absorb all incident light. This means that they present no sub-surface reflectance, and no diffuse colour, in favour of a bright specular colour. A relatively small amount of materials present values between 20% and 40% reflectance; these values are typically specific to semiconductors or complex materials, which do not appear in production shading and are not accounted for in the general BRDF model. Materials with values lower than 2% (the  $F_0$  value of water) are typically found in conductors. Except for metallic materials, every other material instance fits a narrow reflectance value range of  $F_0$  between 2% and 5%.











Material	$F(0^\circ)$ (Linear)	$F(0^\circ)$ (sRGB)	Color
Water	0.02,0.02,0.02	0.15,0.15,0.15	
Plastic / Glass (Low)	0.03,0.03,0.03	0.21,0.21,0.21	
Plastic High	0.05,0.05,0.05	0.24,0.24,0.24	
Glass (High) / Ruby	0.08,0.08,0.08	0.31,0.31,0.31	
Diamond	0.17,0.17,0.17	0.45,0.45,0.45	
Iron	0.56,0.57,0.58	0.77,0.78,0.78	
Copper	0.95,0.64,0.54	0.98,0.82,0.76	
Gold	1.00,0.71,0.29	1.00,0.86,0.57	
Aluminum	0.91,0.92,0.92	0.96,0.96,0.97	
Silver	0.95,0.93,0.88	0.98,0.97,0.95	

Fig. 2.13 The Fresnel equations can be parameterised in order to determine the specular colour characterizing a specific material (metals can have different shades - gold, silver, etc.). At the other spectrum, dielectric materials have a low intensity shade. From “Real-Time Rendering, 3rd Edition”, A K Peters 2008

### Normal distribution function

In equation 2.16, the term  $D(m)$  represents the microgeometry's normal distribution function. It encompasses the statistical distribution of independent surface normals (orientation). In most surfaces, the microgeometry is not uniformly distributed in term of facet orientation. Generally, a large number of points belonging to the micro-surface have normals pointing upward and aligned to the global(macroscopic) surface orientation, than away from the normal  $n$ .

The normal distribution function yields a result, which is has to be positive but is not restricted to values in the range 0 to 1 (as was previously observed in the case of  $F(l, v)$ ). A large value for the term  $D(h)$  translates into a region on the micro-surface containing points with normals facing in similar directions. Another particularity of the normal distribution function is that the result is a scalar value as opposed to an RGB triplet. It indicates an amount of light at active surface points in the direction of the half-vector  $h$ . In the BRDF equation 2.16, the normal distribution takes  $h$  as parameter, because it only considers the case where  $m = h$ . The scalar value is then aggregated in the brightness term, as well as the size or scale of the specular highlighted area.

The distribution of normals for guiding surface orientation has been approximated in numerous models in Computer Graphics literature. In Gaussian models, only one term is used in the approximation (**roughness** or variance value). In isotropic and anisotropic models, two roughness terms are used to account for locality [8, 19, 103].

Upon observing the transition between a rough and smooth surface, the average micro-facet orientation (concentration of all normals in a particularly rough micro-geometry) decreases around the macro-geometry normal. The value of  $D(h)$  becomes noticeably high [152]. Walter et al. [158] demonstrate the correct method of normalization for  $D(h)$ , and benchmark several scenarios [89, 160].

### The geometry function

Equation 2.16 defines the geometry term of the BRDF model denoted  $G(l, v, h)$ . This function calculates the probability that points on the micro-surface with a their corresponding normal  $m$  will be visible from both the incident light vector  $l$  as well as from the the view direction  $v$  (Figure 2.10 b) where  $m = h$  in order to account for micro-facet orientation, not just the macro-geometry.

As discussed for the normal distribution, the geometry function yields a scalar term representing a probability, therefore limited in the range 0 - 1. There are various approximation models for  $G()$  in the related literature, however, in all cases the surface geometry is greatly simplified for performance reasons [150, 82].

It is possible for the geometry term  $G()$  to take no parameters or use the roughness parameter denoted *roughness parameter*( $s$ ) from the normal distribution function  $D()$ . It

plays an important role in maintaining the energy conservation principle for the BRDF model - if  $G()$  would not cancel the denominator in equation 2.16, the reflectance function could allow for more light to be emitted than the total number and intensity of the incident rays. In terms of surface area, the active area that is involved in reflecting the incident light energy towards the view direction may not undergo any shadowing from the macro-geometry and therefore exceed the total surface area.

#### 2.2.3.4 Subsurface reflectance - the diffuse term of the BRDF

In order to estimate reflection locally, common BRDF functions rely on the Lambertian surface model [93], which does not add complexity to the computation, as it is a constant term - the cosine or  $(n \cdot l)$  factor is part of the reflectance equation, not the BRDF (as seen in Equation 2.19). The value used broadly to define the Lambertian BRDF is:

$$f_{Lambert}(l, v) = \frac{c_{diffuse}}{\pi}, c_{diffuse} \in [0, 1] \quad (2.19)$$

The Lambertian model is not generic, and cannot define the behaviour of light interaction with surfaces at grazing angles and the energy exchange taking place between the diffuse and specular terms. In this case, the specular term has priority on the incoming light energy, and the diffuse term can only use its remainder.

According to the Fresnel equations, the specular term has a higher value at grazing angles, while the diffuse term decreases proportionally. One method for modelling this exchange can be achieved by performing a linear interpolation between the diffuse and specular terms, however, this does not render accurate results in all situations [6], [7], [82], [138].

## 2.3 Inverse rendering and image decomposition

Similar in nature to the inverse kinematic problem, inverse rendering problems do not have access to all the known variables *a priori* (geometry, number and position of light sources, surface material properties, etc.). In most cases, it is ill-posed to solve inverse rendering with no known parameters, and generally, at least one is known and the others are estimated based on assumptions and inference from observer relationships between them. Often times, the desired result in terms of light-surface interaction is known (the final appearance of a lit surface), and the approach is to backtrack in order to solve for the other unknown parameters. The inverse rendering problems can be classified depending on which parameter they are aimed at solving: shape from shading (SfS) if coarse geometry or object delimitations have to be identified along with the corresponding surface normals, illumination estimation, and material property modelling. There are techniques where a joint estimation of two of these is possible concomitantly.

Recall that the radiance value over the surface of a hemisphere is denoted  $L(r, \omega)$ , where  $r$  represents the point where power is radiated from, on a trajectory described by  $\omega$ . The direction of propagated radiance is per unit of projected area perpendicular to that direction per unit solid angle from a given frequency. If we consider the radiance integral, then the boundary conditions can be expressed in the following form:

$$L(r, \omega) = L_e(r, \omega) + \int_{S_i} f_r(r, \omega, \omega_i) \times L(r, \omega_i) \cos \theta d\omega_i \quad (2.20)$$

- $r$  represents all the points in the surface described by the hemisphere.
- $f_r$  BRDF
- $\theta$  is the incident angle measuring a distance  $\in$  surface normal and  $\omega$ .
- $S_i$  is the hemisphere encompassing all incident rays.

Based on the radiance equation at 2.20, a number of linear operators used in inverse rendering can be defined to account for reflectivity, field radiance, and emittances of the light sources present in a scene [5]. Equation 2.21 rewrites the Rendering Equation as a linear operator equation based on the reflection operator  $\hat{K}$ . In this case,  $\hat{K}$  denotes the amount of scattered light of the incident radiant energy as a function of  $h$ , the field radiance characterizing all incident rays. As the name suggests,  $\hat{K}$  represents reflection by mapping the incoming ray distribution to the relevant emitted light.

$$(\hat{K}h)(r, \omega) \equiv \int_{S_i} k(r; \omega' \rightarrow \omega) h(r, \omega') d\mu(\omega') \quad (2.21)$$

The field radiance operator  $\hat{G}$  is expressed as a function responsible for converting the distribution of emitted light to the distribution of incoming light rays, as seen in equation 2.22. The field radiance operator depends on the visible surface function for each surface in the environment. The implicit function  $r(r, \omega)$  can be factored out from the integral transport equation, and we can express it in a simplified form as seen in 2.23.

$$(\hat{G}h)(r, \omega) \equiv \begin{cases} h(r, \omega), & \text{when } \nu(r, \omega) < \infty \\ 0, & \text{otherwise} \end{cases} \quad (2.22)$$

$$L = L_e + \hat{K}\hat{G}L \quad (2.23)$$

When the problem solves for the total emitted light  $L_e$ , and the other operators are known, it becomes an *inverse lighting problem*. One real-world scenario to illustrate this situation would be if  $L$  is known from an environment map or HDR photograph, and the scene geometry  $\hat{G}$  is known along with the surface reflectivity or material property expressed through  $\hat{K}$ . In this case, it is also possible to make assumptions about estimating reflectivity or limiting it to the Lambertian reflectance model.

When surface reflectivity  $\hat{K}$  is unknown, the problem becomes an *inverse reflectometry problem*. If all other parameters are known, and specifically for  $L$  the input comes from a series of images, then the problem can be classified as an *image-based reflectometry problem* [105]. This is the most difficult sub-problem to tackle compared to the other inverse rendering ones, due to the dual nature of the variance of reflectance - spatial and directional, encoded as  $\hat{K}$ . Therefore, it is common to divide solutions based on this criteria into two categories: *inverse texture measurement* for constraining directional variation, and *inverse BRDF measurement* in which case the surface is assumed to be uniform.

When  $\hat{G}$  is unknown, the problem becomes an *inverse geometry problem*. This type of scenario is tackled through user annotation, or coarse geometry estimation based on box model fitting, or surface representation through perpendicular planes. In cases where a single image is provided, computing the geometry of a surface becomes a *shape from shading* (SfS) problem.

The straightforward case is when the three parameters are known and the problem solves for  $L$ . In contrast, the most challenging one is to solve for  $\hat{K}$ .

### 2.3.1 Estimating surface geometry through Shape-from-Shading

Shape-from-Shading stems from the larger problem of photometric stereo reconstruction of a scene, and it occurs when the input is a single image, usually representing a single object. In many approaches, the actual image used for the SfS technique is converted to grey-scale values and is used to represent the brightness factor. This was an incipient technique pioneered by Horn et al. [63] which aims to calculate a single solution for a first-order PDE known in literature as the *brightness equation*. This initial approach is ill-posed as in the real-world, the equation can have more than one solution and sometimes no solution at all.

$$I(x_1, x_2) = R(n(x_1, x_2)) \quad (2.24)$$

$(x_1, x_2)$	represent the position of a point $x$ in the 2D image.
$R$	is the reflectance map.
$I$	is the brightness image.
$n$	is the normal vector to the point $x$ .

The later works of Belhumeur [12] illustrate that neither the identified shadowed regions nor the overall shading of an object in the scene seen from a single viewport (since the input is a single image) reveal exact 3D structure.

It is a common assumption in SfS methods to model the scene as Lambertian. The reflectance map  $R$  then becomes the cosine of the angle between the incoming light vector  $L(x)$  and the normal vector  $n(x)$  to the surface at  $(x_1, x_2)$ . Considering this setup, most SfS techniques recover surface normals at each point  $p = (x, y, z)$  on the surface in order to estimate the geometry to scale. Since the surface normal is dictated by orientation, it is better to express its values in a spherical and not Cartesian coordinate system, hence,  $n$  can be defined in terms of its pitch (*azimuth angle*)  $\tau_n$  and yaw (*polar angle*)  $\sigma_n$  value respectively.

The azimuth angle describes the arc length measuring the distance between the horizontal axis and the projected area of  $n$  on to the image plane. The polar angle measures the distance between the z-axis and the normal. From the two observations it follows that  $n(\sigma_n, \tau_n)$ , where  $\sigma_n \in [0, \pi/2]$ ,  $\tau_n \in [0, 2\pi]$ , and can be formally defined as seen in equation 2.25.

$$n = (n_x, n_y, n_z) = [\sin\sigma_n \cos\tau_n, \sin\sigma_n \sin\tau_n, \cos\sigma_n]^T \quad (2.25)$$

$$p = \frac{\partial Z(x, y)}{\partial x} \quad q = \frac{\partial Z(x, y)}{\partial y} \quad n = \frac{1}{\sqrt{1 + p^2 + q^2}} [-p, -q, 1]^T \quad (2.26)$$

$Z(x, y)$  surface  $Z$  characterized by two axis  $(x, y)$  in a Cartesian system.  
 $(p, q)^T$  is the gradient (surface orientation)) at point  $(x, y, Z(x, y))$ .

The slopes describing the surface of interest:  $(1, 0, p)$  and  $(0, 1, q)$ , are obtained from the partial derivatives in two directions defining the gradient (Equation 2.26) or the surface orientation. The surface normal in relation to the surface gradient (normal vector perpendicular to point  $\in$  surface) can be calculated through the cross product of the two vectors  $\in$  the tangent plane at coordinates  $(x, y, z)$ . The result of the cross product then undergoes normalization in order to become a unit vector, represented as the surface gradient  $P(p, q)$ .

The SfS techniques rely on assumptions or constraints of the illumination and the material properties of the surface in order to recover data about normals. Illumination can be described in similar terms to the normal - with a pitch and a yaw  $I(\omega, \tau)$ , where  $\omega \in [0, \pi/2]$  and  $\tau \in [0, 2\pi]$ .

$$I = (I_x, I_y, I_z) = [\sin \omega \cos \tau, \sin \omega \sin \tau, \cos \omega]^T \quad (2.27)$$

The surrounding environment is only visible if light sources are present. In this situation, there is always a trade-off, an interaction based on objects that will absorb the incident light or reflect it, however, in most cases both processes take place in different ratios. This phenomena encompasses what is known as surface reflectivity - a fraction of incident light radiation bouncing off a surface. Mathematically, this can be modelled by in terms of the reflected direction vector and the incoming light vector. Based on this phenomena, and more specifically the ratio of absorbed to reflected light, materials can be classified as *diffuse* or *specular*.

Surface appearance encompassed by the Lambertian model presents uniform distribution of reflectivity - equal amount of light is transmitted in all directions. In real-world cases, the surface materials are complex and often exhibit both specular highlights as well as diffuse areas. Despite this fact, the diffuse surface model has been adopted in most SfS techniques due to the localized area of interest when it comes to intensity variation. The reflectance value in the Lambertian model can be defined as a function of  $\alpha$  and the specific surface reflectivity (*albedo*)  $\rho$  as follows:

$$R = (R_x, R_y, R_z) = \rho(R_x, R_y, R_z) \cos \theta \quad (2.28)$$

Equation 2.28 is built on the notion that the collection of active surface points of the form  $(R_x, R_y, R_z)$  will receive the same amount of light from this direction, however, the albedo will vary per point. A number of incipient SfS techniques further assume that the albedo is fixed in order to simplify the problem at the cost of result plausibility. This equation can be rewritten by employing unit vectors for the directions of the normal and illumination (equation 2.29). When the albedo and illumination directions are known, the reflectance map at each point  $(p, q)$  can be defined by a gradient space obtained from the dot product of the global normal vector and the incoming light direction.

$$R_{\rho, I}(p, q) = \rho I^T n = \frac{\rho}{\sqrt{1 + p^2 + q^2}} I^T [-p, -q, 1]^T \quad (2.29)$$

In the simplest form of SfS, the problem assumes that the scene comes from an orthographic projection, that the surface reflectivity is Lambertian, and that the albedo can be a known threshold. Under these constraints, the SfS problem can be defined formally as:

$$E(x, y) = R_{\rho, I}(p, q) = \rho I^T n = \frac{\rho}{\sqrt{1 + p^2 + q^2}} I^T [-p, -q, 1]^T \quad (2.30)$$

Equation 2.30 defines the irradiance in terms of the amount of light seen by the camera in an orthographic mode which is proportional to the gray-scale gradient. In this context, the non-linear PDE defines the gradients to be  $p$  and  $q$  respectively on an unknown surface  $Z$ . In numerous cases the albedo, and direction vectors for normals and light sources have to be estimated. These parameters should be available *a priori* for the domain boundary (image boundary). SfS aims to identify the surface  $Z$  by synthesizing a shaded image in which the assumption is that  $Z$  will be Lambertian. The albedo and directions are estimated *a priori* which are plugged in the partial derivatives over a pixel grid according to equation 2.30 in order to obtain the corresponding brightness maps. The surface partial derivatives are calculated in the x-direction ( $p$ ) and y-direction ( $q$ ) after which they are plugged back to solve for  $E$ .



### 2.3.2 Estimating surface materials through image decomposition

Shape-from-Shading technique assume reflectance and illumination data *a priori* and attempt to determine a surface shape according to these two parameters. When the geometry is known, and the problem solves for materials properties and illumination simultaneously from one image by parametrising the expected output, the technique is known as intrinsic image decomposition.

In a large number of cases the decomposition parameters are illumination, reflectance, and specularity. The radiance term generally accounts for all shading effects of geometry including interreflections. Reflectance, just like in SfS, is the *albedo* factor exhibited by a surface independent of viewpoint and illumination. Lastly, the specularity components dictates the highlights observed under specific illumination on the geometry. When these three elements are combined, the problem is formally defined as:

$$I_p(x) = S(x)R_S(x) + K_p(x) \quad (2.31)$$

In the initial setup described by equation 2.31,  $I(x)$  represents the intensity observed at each pixel,  $R(x)$  is the reflectance component over the surface, and  $K(x)$  is specularity. As is the case with most inverse rendering problems, intrinsic image decomposition in the absence of constraints is an ill-posed problem, as there are multiple unknowns for each observed component in expression 2.31. Despite the complexity of this problem, researchers have created different assumptions in order calculate each component. One early example is the assumption of a piecewise constant environment, known as a *Mondrian world* in the Retinex theory developed by Land and McCann [94]. In this work, the slow variation in gradient is mapped to illumination while spikes in this gradient (steep variations) correspond to reflectance. The Mondrian world is a considerable restriction with little application to ubiquitous day-to-day scenarios, therefore efforts were directed towards analysing object boundaries (edges) in order to infer illumination conditions based on a number of heuristics.

When dealing with a single image as input, the incipient approach of Sinha et al [139] consider as input a world of painted polyhedra, and aim to search for globally consistent features (edges or edge junctions) which are an effect of shape or reflectance variation over time. Later techniques developed by Tappen et al. [146, 145] relied on machine learning and non-linear regression to interpret gradients from local cues. Another set of approaches based on multiple images was pioneered by Weiss [162]. In this case, the images capture the initial scene with a variety of illumination setups. Weiss et al demonstrated that the

reflectance can be inferred from the input stream if there is an assumption of a random distribution for the BRDF (Figure 2.14).

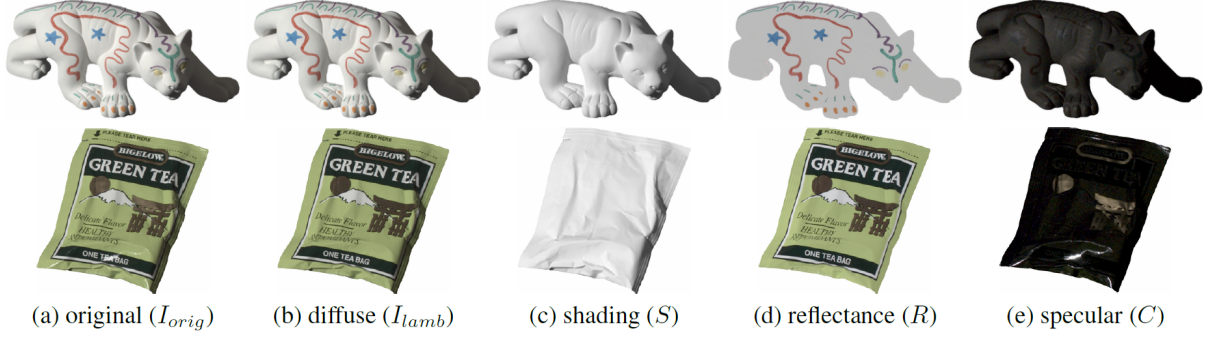


Fig. 2.14 Image decomposition in the work of Weiss et al [162]. (a) The original image captured with a dedicated filter that brings forth details of the specular highlights. (b) The image of the diffuse properties, captured with a dedicated filter that removes specular highlights. (c) Painted object completely diffuse for shading. (d) Reflectance mask. (e) Specularity mask.

The specular and diffuse terms play an important role in material appearance as a response to light interaction. What is referred to by the common notion of *highlights* is defined by the specular term, and represents the rays reflected from a surface. The diffuse term, governs a more complex interaction of the incident light rays that reflect and refract at angles dictated by the normals of each microfacet defining the surface. Due to its complexity, the diffuse term is decomposed into a shading function and a reflectance function. Intrinsic image decomposition techniques focus on retrieving two aspects: shading and reflectance. These two terms can take values which are scalar multiples of the captured values.

The intrinsic image decomposition workflow takes as input a single diffuse image, and assumes a Lambertian world with a single light source. The final fragment sample is a scalar value denoted  $S(x, y)$  (pixel coordinates). The following expression defines the decomposition of shading and reflectance in a grey-scale 2D domain:

$$I_p(x, y) = S(x, y)R_S(x, y) \quad (2.32)$$

The common algorithmic approach to intrinsic decomposition could be summarized as a number of iterative steps:

1. The horizontal and vertical gradients ( $i_x, i_y$ ) of the log grey-scale image are calculated.
2. Interpret the gradients from the previous step by estimating gradients  $\hat{r}_x$  and  $\hat{r}_y$  sampled from the reflectance map.

3. Compute  $\hat{r}$  that has the highest similarity to the gradients:

$$\hat{r} = \underset{r}{\operatorname{argmin}} \sum_{x,y} |r_x(x,y) - \hat{r}_x(x,y)|^p + |r_y(x,y) - \hat{r}_y(x,y)|^p \quad (2.33)$$

Where  $p$  is the least-squares weight involved in the geometry reconstruction from the image through Poisson approximation on a 2D grid. The gradients  $\hat{r}_x, \hat{r}_y$  are only estimated for the pixels on the surfaces since the background is ignored.

## 2.4 Related work

Convincing rendering of 3D artificial objects into legacy photographs takes into account a number of key components such as the scale and perspective at which the object is composited into the scene (the perspective has to match up to the camera), perceptually consistent illumination (captured or estimated), simulation of the cast shadow and new illumination setup after the object has been composited (common illumination interaction demonstrated by Slater et al. [140]), and the material properties of the surfaces involved in the light-shadow interaction post compositing.

Taking this into consideration, we can devise a pipeline for compositing 3D artefacts into photographs by focusing on the significant factors: geometry acquisition or estimation, determining the properties of the illumination model, and synthesizing light-material interaction present in the local (original) scene.

In accordance with the classification of object compositing methods covered in Section 2.1, the present body of work belongs to the 3D hybrid techniques category, where image scene properties are estimated rather than measured or acquired via specialized hardware. In addition to this, the described pipeline is semi-automatic (allows the user to specify artefact coordinate system, or adjust the inferred properties), and computes a rough geometric estimation of the scene.

### 2.4.1 Scene geometry estimation and structural reconstruction

Scene acquisition techniques used for compositing depend on the input specified at the beginning of the framework. In some cases, for ease of use, a single image is specified, stripped away of its original meta-data. In other cases, a video input is specified (stream of images), or the original image data was produced by specialized hardware able to capture depth information. All of these techniques are encountered in compositing, and whereas there is no single approach that works for every problem, depending on the desired end-result and the domain of application, some of these techniques may present significant advantages over their alternatives.

As a general overview, the methods which rely on a single image require either heavy user annotation, or a large dataset of ground truth scenes in order to identify a close match for the problem at hand. For this reason, the single image input approach is well suited for indoor scenes defined by planar horizontal and vertical surfaces, usually describing a room with little to moderate amount of furniture that does not overlap or interleave. Pipelines which take as input a video stream have a wide dataset for carrying out foreground-background segmentation which can be useful in the context of compositing, namely for inferring shadow regions, and leveraging user annotation of geometry. Lastly, data (either single or stream) emerging from RGB-D capable hardware offers even more

information regarding the structural setting of a scene. More data is not always desirable, since it is not always organized in accordance with the application domain, and often times requires denoising or analysis in its entirety, which translates into a significant time penalty without guaranteeing the exclusion of user customization or post edits.

The described approaches, regardless of level of automation, require user input, therefore it is generally safe to assume that the user should be involved in the pipeline to the extent where minimal input is encouraged but only for tweaking the result produced by an automated process. This ensures that the user has control over the quality of the result without requiring system expertise for scene properties' annotation.

One of the early techniques to tackle geometry estimation from 2D photographs was developed by Horry et al. [65] and it involved the user annotating vanishing points in a photograph so that the structure of the scene could be easily represented through grid meshes. This would facilitate the animation in 2D images - a technique entitled "Tour Into the Picture". Based on the algorithmic approach to measuring the metric properties of world planes from a perspective image [98], Criminisi and Zisserman [31] rely on user annotation of the vanishing line of a reference plane and a point for a direction intersecting this plane in order to compute distances between parallel planes describing scene structure.

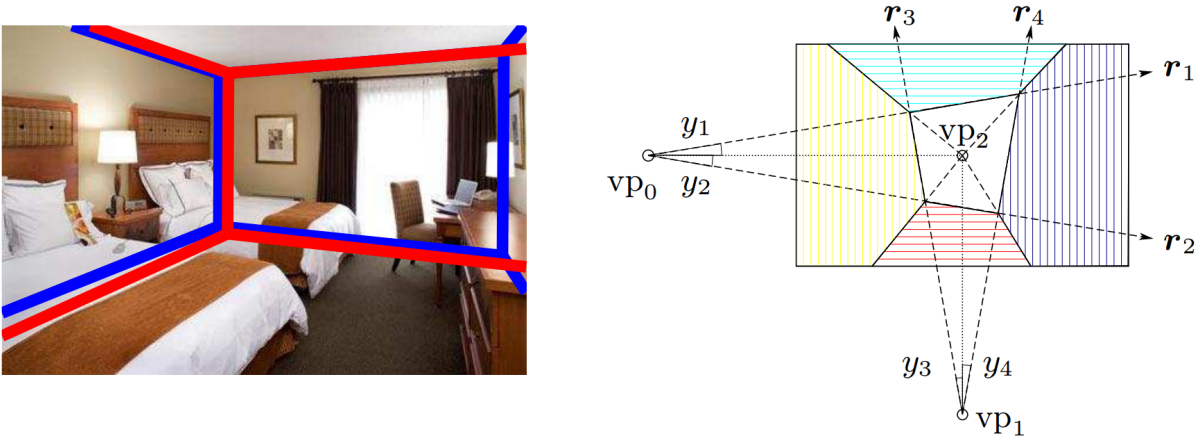


Fig. 2.15 Indoor scene layout estimation [175]. Left hand side image illustrates the process of layout detection from a single image - the blue overdrawn lines indicate the best fitted result, and the red lines highlight the predicted result. The right-hand side image reveals the parameters involved in defining the problem setup.

Oh et al. [116] opted for a different approach to depth estimation from a single 2D image through user-assisted operations such as per-pixel depth specification and layered grouping of different regions in the input domain. While this approach renders plausible results, the drawbacks relate to the large amount of time for depth acquisition and the required user expertise [115]. Depth information in its rawest form can be represented

through a texture (as seen in Oh et al.'s work), or can be further expanded, in a brute-force manner, to the 3D domain into a point cloud structure (where each pixel is given a third component -  $z$  in a Cartesian coordinate system). Such an approach was developed by Zwicker et al. [175] in order to allow the user to refine the point cloud and prepare it for texturing. These approaches of reconstruction are well suited in the context of architecture and outdoors building modelling.

A more recent approach introduced by Hedau et al. [58] and extended by Karsch et al. [77] modelled indoor scene structure through a set of vertical and horizontal planes creating bounding volumes. This method allows for a better scene structure recovery than its alternative for single image depth-estimation which relied on the ground-wall boundary detection, often occluded in images presenting cluttered rooms. This method was later improved by Schwing and Urtasun [136] - the first of its kind to solve the problem of exactly fitting an indoor scene layout from a single image.

Geometry estimation of an indoor scene can also be accomplished in a data-driven approach as seen in the work of Satkin et al. [129] who use a large collection of 3D models (i.e. furniture, common items found indoors) to reason about the scene in a 2D photograph. Del Pero et al. [124] rely on a Bayesian generative model to understand scene structure. In their work, the authors propose a stronger representation to that of 3D bounding box-like volumes, which offers finer level of detail.



Fig. 2.16 Video image depth estimation in the works of Karsch et al. [78]. The top row represents the video stream input. The bottom row illustrates the per-pixel depth estimation. This approach is entirely founded on a ground truth (GT) depth dataset, and does not rely on cues from motion parallax.

Single image-based depth estimation for scene reconstruction is an ideal candidate for applications where accuracy of scene detail recovery is not as important as understanding the overall scene structure organization (position and orientation of large flat surfaces). This category of techniques relies on observed relationships between image features and geometry in order to estimate depth [62]. The efforts were focused in the direction of establishing geometric constraints and assuming a Manhattan World (i.e. every plane is perpendicular to one of the axes of a single coordinate system) [30] in order to estimate

scene depth [37] for both single and multiple [46] images. The contemporaneous work of Karsch et al. [78] trains and evaluates a large dataset originating from Kinect - RGB-D capable hardware, where the ground truth depth is known (seen in Fig.2.16). The authors then effectively match new single images or video data to the database based on non-parametric sampling.

In addition to the previously described methods for depth estimation (user-assisted techniques, single image-based inference), another established approach is to operate on RGB-D input (complex geometry, densely reconstructed without missing patches). A large number of algorithms can be invoked on this type of input ranging from shape from shading, to surface albedo as illustrated in the work of Barron and Malik [11]. Other systems of this category rely entirely on Simultaneous Localization and Mapping (SLAM) - a method known for its ability to obtain camera poses and scene geometry concomitantly. Niessner et al. [113] prove that its well suited for real-time volumetric fusion of large scenes. In this context, the application takes a continuous stream of overlapping depth maps and incrementally fuses them into a single 3D model.

The field of robotics relies on SLAM systems for obstacle avoidance and scene mapping. Despite the large use cases, indoor scenes defined by walls presenting uniform textures that lack structural information can be challenging to capture using SLAM. Modern RGB-D sensor cameras such as the ones used for SLAM are not wide to the extent of capturing entire regions at once. Therefore, these techniques require bespoke hardware, or multiple sensor deployment to cover the entire area of interest.

Specialized hardware, such as Google's Project Tango has been used as a better candidate to SLAM, due to its visual-inertial odometry with a wide field-of-view camera for a more robust localization [87]. In more recent works, Cabral and Furukawa [21] edit and modify indoor scene geometry acquired from a stream of images, however, they assume a Manhattan World. This drawback was addressed by Zhang et al. [172] who rely on Project Tango (Figure 2.17) for a complete pipeline of 3D HDR radiometric calibration for indoor scene geometry and material appearance acquisition in real-time.

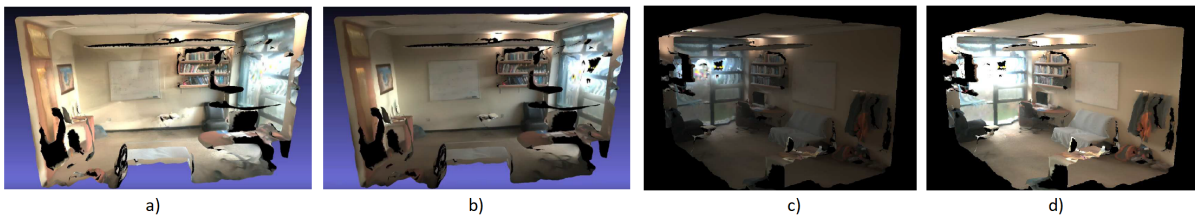


Fig. 2.17 Comparison of a recovered mesh acquired with Project Tango in the works of [172] before and after radiometric calibration. From left to right: a) is the un-calibrated mesh texture, b) is the same mesh after calibration, c) is a calibrated mesh with low exposure, and d) is the same mesh with higher exposure

### 2.4.2 Illumination estimation models

In the previous 3D compositing classification Section 2.1.2, we have devised two main categories for compositing based specifically on how illumination conditions were modelled. The present subsection focuses on detailing approaches relevant to our body of work, namely, estimated light models.

The straightforward approach to estimating illumination conditions from photographs is to identify point light sources based on established contour detection algorithms, or silhouette segmentation [70]. Lopez-Moreno et al. [110] propose a refinement of the technique by estimating illumination from a single image as a combination of achromatic light parametrization through 3D directions and relative intensities. This method assumes constant albedo, and relies on the user tracing the contours encompassing each light area. Illumination estimation for Augmented Reality (AR) applications has built on the previous approach of light parametrization. The more recent work of Nguyen and Le [112] assumes that the object used as a light probe has a globally convex shape, and therefore can employ it to determine the illumination information in terms of the direction and the intensity.

In the absence of a user-specified light probe such as a highly reflective sphere, Nishino et al. [114] show that a skybox of the scene could be captured from the reflections present on the surface of the eyes. Their work is based on anatomical studies that have demonstrated how the contour of the human cornea can be estimated with a 2D ellipsoid at an arbitrary scale. Based on this assumption, the technique infers a set of geometric priors that define the imaging system on the cornea. The capture environment map has a wide-field of view (stretched towards the sides). This map is sufficient to recover reflectance of the entire scene. Lighting setup of a scene can be inferred from a variety of factors, including detection of shadow regions. Panagopoulos et al. [118] considered an approach where illumination sources are coupled with the observed image and the latent variables corresponding to the shadow detection.

Illumination setup can be estimated using a number of assumptions about the environment and its structure, however, a common assumption present in literature is that of a distant-world. In a distant-world, the incoming light intensity affecting the surface it comes into contact with depends entirely on the direction vector of the ray, and not the point on the surface. This is where IBL approaches come into play (discussed in Section 2.1.2.1). Measured IBL techniques that aim to represent illumination through an environment mapping are used for compositing objects into real-world scenes. The distant-world approach is ill posed for this situation, as it does not provide sufficient information about scene-scaled effects (no variation of light across the scene and no variation is allowed after compositing). In other words, the global illumination captured is fixed.



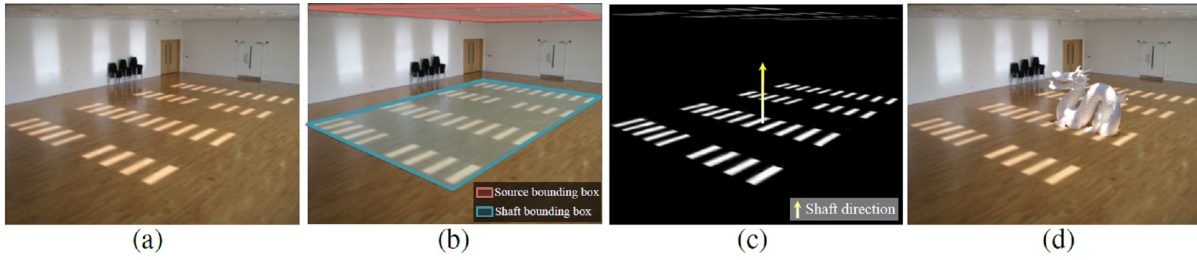


Fig. 2.18 Light estimation based on a user-assisted technique [77]. (a) Input image representing the captured scene as a 2D image. (b) User specifies bounding boxes around the source of light and its intercepting surfaces. (c) Light source direction is estimated using the centre of the bounding box, while the silhouette of the light shafts are detected. (d) Mask of spot light representation of light is applied to a composited artefact.

Surface light fields fit into the broader field of IBL-related techniques and have been invoked for estimating illumination in the context of artefact synthesis in photographs. Wood et al. [164] use a generic model for measuring vector properties coupled with principal component analysis in order to build the minimal representation of an object's structured light field from images or 3D scans (RGB-D). While rendering plausible results, the main disadvantage of the technique is the large size of input data which does not make this an ideal candidate for real-time applications. Spatially-varying lighting issues observed during artefact-scene interaction have been addressed in the more recent work of Cossairt et al. [29]. Their method for 4D light field capture and projection relies on the availability of an array of specialized cameras, a video camera, and a digital projection. This setup allows the capture of fixed real-world global illumination (GI). With this technique it is possible to represent specular materials and estimate a limited set of effects resulting from the interaction of synthetic artefacts composited into the field. Unger et al. [155] perfect the previously discussed methods and capture spatially-varying environment maps. The authors' approach expands on Cossairt et al. work, accommodating for the capture of illumination variance at scene-scale. Moreover, the extracted illumination data can be replaced with common BRDF models to allow for a better fit, and consistency once compositing takes places. The drawback of this approach consists of the scene representation through proxy models, which require prior knowledge or a rough scene estimation *a priori*.

Mixed approaches that rely on both assuming uniformly emitting point lights and estimate an environment map pre-date the light field techniques [142]. By assuming a Lambertian surface model, Weber and Cipolla [161] introduce a general model for point light sources that encapsulates direction and location of the source. In addition to this, the authors approach the problem from a rendering pipeline point of view, and proceed in using three entities (emitter, reflector, and receiver) in solving the lighting equations. This

approach is motivated by the geometry reconstruction problem (i.e. 2D image to rough estimation of the 3D structure of the scene); this can be viewed as the *inverse* process of the physical process that generates the image.

Previous illumination estimation techniques made the common assumption of modelling distant light sources, and in some cases considered only circular shaped emitters. In the contemporary work of Wu and Saito [168], nondistant lights are recovered from a single low-dynamic range (LDR) photograph, allowing for customization or complete relighting of the scene in a plausible manner. In their work, camera, geometry, and light sources are approximated sequentially based on user guided input and annotation (Fig.2.19). Unlike the previously discussed approaches, their framework relies on a masking layer that is used to minimize the effects of shadowed regions or overlapping geometry, and carry a weight assignment to pixels influenced by these effects.

The illumination is assumed to act in a conic-shaped volume in the 3D environment representation of the 2D scene, and as an elliptical area upon surface interaction within the resulting image. In the framework, this behaviour is estimated through the collection illumination sources (point or directional) distributed uniformly on a curve centred at  $L_p$ , orthogonal to axis  $L_f$  and characterized by a width and a height value denoted  $L_w$ , and  $L_h$  respectively. The diffuse illumination estimation function then follows:

$$diff(S(K(x)), L') = L'_K \sum_{l \in \Gamma} \frac{V_{l_p} \cdot N_p}{V_{l_p}} \times \frac{1}{||V_{l_p}||^2} \times Beam(V_{l_p}, L_a, L_k) \quad (2.34)$$

$$V_{l_p} = L_p + \Delta l_p - S(K(x)) \quad (2.35)$$

$$Beam(V_{l_p}, L_a, L_k) = \begin{cases} (1 - \frac{\theta}{L_a})^{L_k}, & \theta < L_a \\ 0, & \theta \geq L_a \end{cases} \quad (2.36)$$

$V_{l_p}$	vector from the estimated light area $S(C)$ to the light source $l$ .
$\Gamma$	set of light sources that cumulatively defining one area $L'$ .
first fraction	Lambert's cosine function.
second fraction	Inverse-Square Law for decrease of intensity over distance of light rays.
$Beam(...)$	decrease of luminance from optical axis (based on cone orientation).

Despite the plausible results and its wide applicability to image editing (i.e. specifically to bright light source removal or dimming), the framework has a number of limitations

due to assumptions related to the scene structure - the scene is represented through a Lambertian surface reflectance model, in which the foreground geometry has to fit into a number of perpendicular planes (i.e. horizontal floor, vertical walls, etc.). In addition to this specification, the camera is assumed to fit the pinhole camera model described in terms of the scene's origin and a look-at angle (pitch value or azimuth described around the system's horizontal X axis assuming Y is up and the system is left-handed).

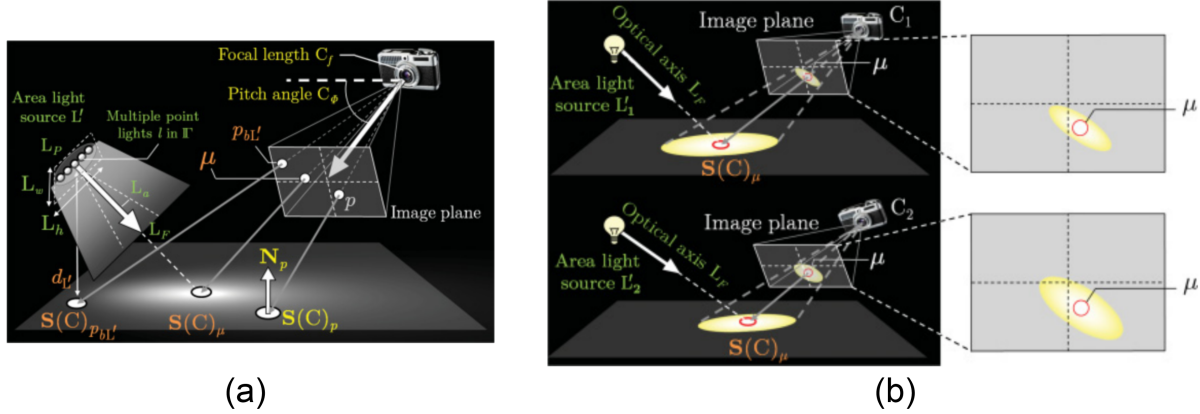


Fig. 2.19 Estimation and relighting [168]. Left hand side image (a) illustrated the conditions of existence for the physically-based model parameters characterizing each pixel  $p$ . The light source parameters can be observed in green, whilst camera and geometry are rendered in yellow. The remaining parameters are the initial estimates for the light source's positions and spread. Right hand side image (b) illustrates the function responsible for correcting the direction  $T(\cdot)$  which modifies the optical axis in world space of the light source  $L'$  to transform its orientation such that it always emits toward  $S(C)$ . In this manner, the origin of the light's area is rendered at the appropriate 2D space within the image.

A recent approach that is similar to our framework's domain of application is developed by Boom et al. [18], and tackles the point light source position estimation relying on an RGB-D sensor in an AR context. The algorithm takes as input a pair of images describing intensity and depth from which a point cloud with surface normals can be computed. Afterwards, a number of hypothesised light source positions are rendered and the one that best matches the conditions from the initial scene is selected. Although the estimation process takes less than a second, the surface normal calculation and the image segmentation based on albedo at every point are not multi-threaded and therefore cannot be well suited candidate for real-time applications.

Single-image geometric reconstruction and reflectance estimation techniques have evolved [153] over the course of the past decade, however, a large number of challenges have not been addressed. Due to the nature of the assumptions made, errors can propagate into the estimations, generating results that are later plugged in the rendering-based optimizations. When looking specifically at indoor lighting, the illumination conditions

vary widely depending on the geometric and photometric properties that define the environment (i.e. light sources can vary from rays crossing through a window, to diffuse illumination produced by a lamp). The variation of wavelength characteristic of these light sources cannot be rendered through a low dynamic range setup.

To address this issue, Gardner et al. [47] infer HDR lighting from a single, LDR photograph of specifically indoor scenes. The method proceeds to represent the data through an environment map which is then used in conjunction with IBL in order to composite synthetic artefacts. Their work achieves this by estimating illumination using a deep neural network. This network learns the representation of images in terms of illumination from a panoramic LDR set. Due to possible out of sight light sources, before training the illumination prediction, the initial input set is first used to train out-of-view light source estimation. The input for the synthetic compositing also comes from the panoramas, namely, cropped regions. The network undergoes corrections after the initial training on the low dynamic range data. In this way, the HDR input allows the network to regress and LDR cropped image in order to obtain the scene's global illumination.

The network's strength lies in its ability to predict light locations, not necessarily the sources' intensity. Moreover, if the sources are out-of-view, their orientation and extent pose a problem to the estimation approach. As a consequence, directional light sources presenting a high intensity are smoothed and cannot define accurate shadows that exhibit both a strong umbra and a soft penumbra region. Since the application's motivation is illumination prediction and not accurate compositing of 3D artefacts, the synthetic images produced work well in uncluttered scenes that can easily be fitted into the box model.



Fig. 2.20 Comparison between the ground truth data (a), Gardner et al. method for estimation (b), their method with user guided customization (c), Khan et al. illumination estimation [83] (d), and Karsch et al. automatic inference [79] (e). The results observed in (d) and (e) render the most plausible results when compared to ground truth data.

### 2.4.3 Material property representation and inverse rendering

Material appearance modelling and inference are processes tightly related to illumination estimation in image synthesis. The early research of Patow and Pueyo [119] bring into focus the challenges of inverse rendering. In particular, these techniques play a fundamental role in lighting setup, where costly mistakes often render it unfeasible to prototype design ideas on a model. The inverse problem infers parameters from observation rather than relying on given input parameters in order to simulate an effect.

Traditional forward rendering pipelines (generally found in real-time applications) represent lighting through a computation of a radiance distribution, where parameters defining the environment are known *a priori* (geometry of the scene and parameterised material information). This problem is well-posed, as there is enough information present to find solutions that do not introduce ill-elements. In contrast, the inverse techniques lack one or more of the Hadamard [56] criteria, making this problem ill-posed [72] - seemingly small errors of measurement and calibration will directly transfer in errors in the final solution.

Generally, inverse rendering problems could be classified into three categories: shape-from-shading (SfS) established by Horn and Brooks [64], direction of illumination from images, and surface property estimation from images. Shape from shading leans towards solutions often employed in computer vision techniques such as contour detection, classification, and filtering. The later two problem categories are interrelated and are generally solved through BRDF models as discussed in the previous Section 2.2.2.

Photometric stereo (Figure 2.21) aims to estimate the surface normals under various illumination conditions. By observing the amount of light reflected at a specific point on a surface, information could be inferred about the orientation of the surface in relation to the light source and the observer [24]. This method was first introduced by Woodham [165]. With the exception of SfS techniques, inverse rendering methods focus on input containing a detailed representation of a single object of interest, and do not factor in scene-wide effects that play a role in the object’s appearance (global illumination, shadowing, occlusion, etc.). In some cases these effects are only considered localised to the area where the object may be inserted. This renders them poor candidates for general-purpose compositing.

Wu et al. [167] demonstrate that it is possible to approximate time-varying incident lighting and use it for geometry refinement in real-time through the use of a specialized multi-threaded Gauss-Newton solver. This is one of the seminal approaches to real-time SfS techniques which treats the problem as a 3D depth map refinement through RGB-D capture. Consequently, the approach is subject to a number of limitations such as the presence of artefacts at high-frequency albedo changes, or constraints related to the custom albedo assumption and the single-bounce light transport model (i.e. caustics - set of light rays reflected or refracted onto a curved surface, are not physically accurate).

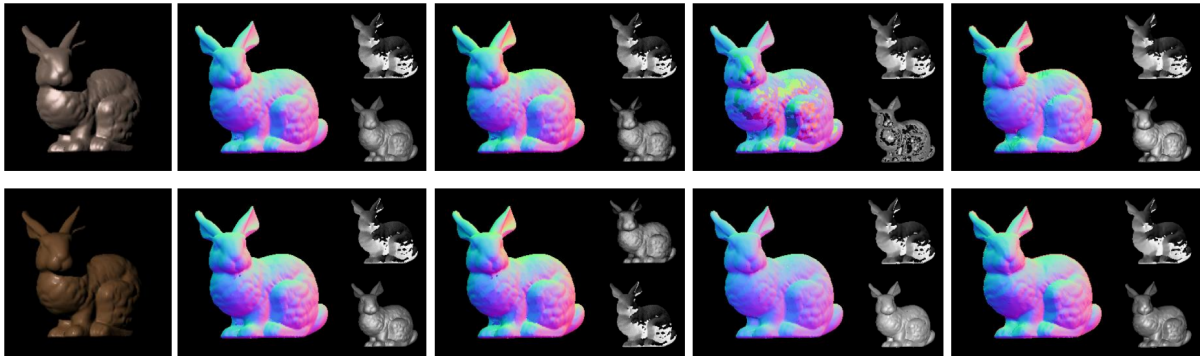


Fig. 2.21 Photometric stereo [96] in two different BRDF models. Each column from left to right illustrates various normal colour maps (N), azimuths (A), and elevation angle maps (E).

In more recent developments, Zollhöfer et al.[173] propose a different refinement rooted in an SfS technique applied directly to the implicit surface representation obtained from globally-aligned RGB-D images. Following this step, the inverse shading problem is formulated on the volumetric distance field and concomitantly solves both surface geometry and spatially-varying surface reflectance. The refinement method yields compelling results, however, the approach has a number of limitations regarding the lighting model. Spherical harmonics were used on an assumption of Lambertian surface representation with a distant, monochromatic light source. In this case, light-matter interactions such as specular, self-occlusion, and spatially-varying illumination are not considered. Unlike the work of their co-authors Wu et al., this method is offline, requiring precisely aligned RGB-D images



from scans, which can take several minutes for the setup. Both the works of Wu and Zollhöfer et al. were developed on the assumptions of distant illumination with a single bounce that did not permit occlusion or interreflections, however, they concomitantly recover scene geometry, temporally-varying lighting, and diffuse albedo from a scene.

In contrast, the earlier work of Ramamoorthi and Hanrahan [127] lay the theoretical foundation for analysing the reflected light field from surfaces under distant illumination through a signal processing formulation based on Fourier transform analysis. The novelty of their approach relies in identifying the specific set of conditions that have to be met in order for inverse rendering to be a well-posed problem. Their framework assumes the input is generated by a scanner and the resulting geometry is achieved through volumetric merging, after which, the two unknowns - lighting and material properties are estimated. Another assumption made with regards to light representation is that the scene contains only distant illumination, homogeneous in nature, easily modelled through an isotropic BRDF with 3 parameters. As with the previously discussed work, no interreflections and caustics are covered as part of the automated framework.

Although applicable to localized parts of the scene and to single objects, the work of Xu and Wallace [169] goes beyond the assumption of distant illumination and demonstrates that it is possible to recover point and directional light sources' positions and reflectance parameters. The input to their proposed framework is dependent on the availability of an intensity sensor used in conjunction with an RGB-D capable camera. This setup guarantees the elimination of user's participation when it comes to geometry or illumination annotation, or the necessity of calibration targets to be present in the scene post-capture.

A number of fairly recent techniques have shifted from the approach of modelling single objects relying on inverse rendering to a broader model that operates on the scene-scale. In the seminal work of Lombardi and Nishino [101], a Bayesian network is employed to jointly infer reflectance and illumination in the real world (Figure 2.22). Due to the difficulty of fitting a reflectance function from a single image even under known lighting conditions [25] (the known lighting is only observed within a portion of the scene), assumptions such as the Lambert or Torrance-Sparrow [149] model will not suffice. The statistical, data-driven approach allows for canonical inference by means of maximum a posteriori estimation from the reflectance model priors, which is assumed to be a directional statistics BRDF (DSBRDF), where reflectance is represented through the sum of *lobes* - each as a distribution in the half vector BRDF parameterization. The reflectance priors are gathered from the further separation into colour and intensity values per lobe. The limitation of the approach consists of an assumption of distant illumination for the generalized BRDF model.

In their later work [100], the authors extend the previous application which takes as input a stream of RGB-D images capturing the real-world scene with varying illumination.

To achieve this a computationally-expensive gradient solver is used during path tracing. Complex light-surface interactions (such as inter-reflections, and light transmission) are possible at the expense of offline rendering. Furthermore, the approach relies on the distant-world assumption for modelling the global illumination in the scene.

Since SfS techniques are generally employed for a single object of interest, they may not always be a feasible first choice for compositing frameworks due a number of reasons:

**Scene clutter.** In real-world cases, scenes are often cluttered, containing a varied amount of objects exhibiting heterogeneous surfaces. These objects can partially be occluding one another, potentially withholding important information regarding illumination.

**Multiple materials.** Objects already present in the scene can be made of one or more materials (submaterials), with largely varying indices of reflection and refraction. This common case can result in false negatives when using an SfS technique.

**Special materials.** Semi-opaque objects alone present difficulties for SfS in absence of a strict set of assumptions regarding illumination.

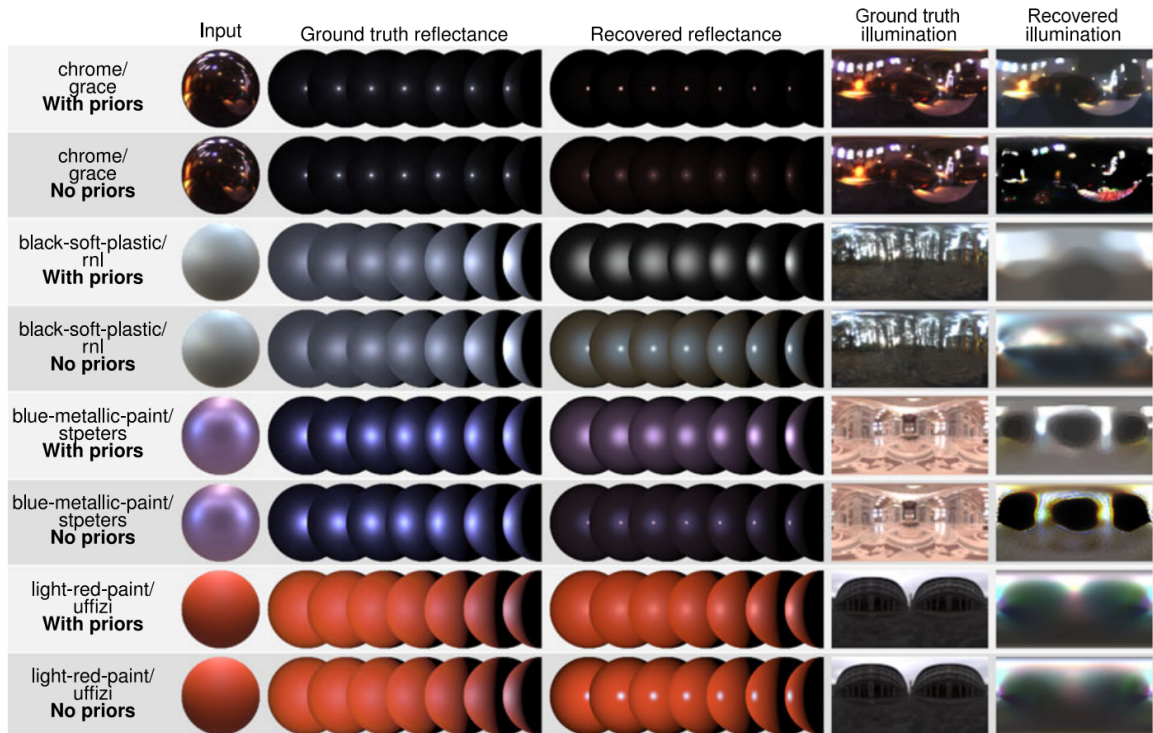


Fig. 2.22 Reflectance and illumination recovery [101]. Each material comes from the MERL BRDF database [106]. The environment maps used originate from the Light Probe Gallery [34]. The third column holds the illumination estimates.

Research has been carried to address these shortcomings by computing scene models from a single image. These techniques are usually estimating illumination and not physically



measuring it. Boivin and Gagalowicz [16] propose a hierarchical inverse rendering approach on a single image of a scene where the light source positions and geometry are known. They proceed to approximate the BRDF through a set of parameters that allow the recovery of photometric properties including the diffuse and specular factors along with isotropic or anisotropic surface information. A notable aspect of the framework consists of the camera parameters that were recovered through a technique proposed by Dementhon and Davis [38] in conjunction with the a downhill simplex minimization method developed by Press et al. [125].

The computed BRDF is built on the initial assumption that the synthetic artefacts introduced in the scene are diffuse. Using a global illumination approach that minimises the error between the real and the synthetic image, the algorithm makes iterative changes to the hypothesis regarding the initial BRDF. The number of intermediate synthetic images is dependent on the number of objects composited. In this approach, the user defines a global minimum error threshold. The work has a number of limitations including rendering time (approximately 4 hours for 2 composited objects), and the lack of possibility to discern between shadows and highlights on a given surface. Despite the limitations, the work produces plausible results.

The early work of Barron and Malik [11] tackles BRDF modelling through the technique of intrinsic image decomposition. This category entails an extended model of SfS that recovers not only shape but also illumination and reflectance from shading, gaining the common term of SIRFS in the related literature. SIRFS gained ground through an earlier work conducted by Barron [10] on estimating shape, albedo, and illumination from a single grayscale image that contained one (unknown) object of interest. The recovery of these parameters is known as the *intrinsic images* problem [45], and has been addressed beyond computer science as a *lightness constancy* approach. Albedo and illumination can be estimated after establishing priors regarding the local smoothness and global sparsity of the albedo regions in conjunction with priors on the shape of the object that encourage flatness and local smoothness. In this case, the illumination model is represented through spherical harmonics.

In the context of SIRFS, Barron and Malik [9] apply this technique on an RGB-D image whilst using soft segmentations of the input image, ensuring that an environment map is generated for each illumination segment. In this manner, the framework recovers spatially-varying lighting that approximates occlusion and interreflections. The main limitation of this approach to image decomposition is the lack of means for recovering reflectance since only albedo is targeted through local smoothness priors.

Karsch and Forsyth [76] address the recovery of reflectance from a single image, by relaxing the constraints imposed in the previous SIRFS methods. Even though the main

goal of their work is to recover material appearance, their approach has to solve for two other unknowns *a priori* - illumination and shape. Goldman et al. [50] propose a similar technique that aims to estimate per-pixel mixture weights for the parametric materials, however, multiple HDR images under unknown illumination are used as input. In contrast, Karsch and Forsyth’s approach only relies on a single LDR image and infers illumination before progressing to material estimation. In their approach, only isotropic materials with monochromatic specular components are considered which limits the BRDF estimation to very specific shapes. In addition to this, the recovered lighting conditions along with shape representation are not physically correct, but rather consistent with one another. The illumination estimation model relies on the assumption of distant and spherical light sources and does not account for interreflections of colour consistency issues.

The contemporary work of Bonneel et al. [17] looks at the best intrinsic decomposition approach for creating seamless image edits. In this context, the authors separate the effect of the scene illumination from that of the material reflectance. In the assumed model, each pixel is the result of the product of an RGB triplet for colour and one for reflectance. Moreover, it analyses only the last bounce of the light transport and makes a number of simplifying assumptions such as “all materials are Lambertian” and “no participating media”, but it nonetheless provides a powerful means to reason about light and object colours that has proven to be useful to many image editing applications.

In the context of hybrid composition methods, Liao et al.[97] propose a novel application that allows a user to insert an object from a 2D image into a 3D scene recovered from a different 2D environment. In their work, the authors propose an alternative to SIRFS which explores an approximate shading model aimed at circumventing the complexity faced by 3D reconstruction from image. This approach was inspired by a number of studies indicating a high tolerance for non-physical rendering of the human visual system [117, 27], provided a number of structural rules are met. Instead of SfS, the framework relies on illumination cone representation in order to decompose the input image into albedo and shading relying on a Retinex algorithm [53].

The author’s postulation that inserting a real object from a flat 2D image into a 3D scene will create a result that is closer to the real-world expectation is correct as long as the object does not create inter-reflections, cast shadows, or is itself an illumination source. This aspect greatly decreases the application range.



Fig. 2.23 Results from the works of Liao et al.[97]. Left-hand image - input data is given as one larger image in the background representing an initial scene along with an array of images of single objects that are to be composited. The method then builds an estimation in 3D for each of the 2D models (teapot models and an ostrich). Right-hand side image - the result of insertion of the array of objects in a scene, which is consistent with the original illumination settings. This approach allows for reflection to occur (front teapot), as well as refraction (back teapot) and the user can specify depth-of-field effects (ostrich).

#### 2.4.4 PBR techniques in the wild

The theoretical foundation and motivation behind physically-based rendering (PBR) techniques was covered at the start of the Background Section 2.2. The present section of the related work aims at describing and comparing ubiquitous PBR systems developed and used in the games(real-time) and movie (offline rendering) industries, in order to offer insight into existing state-of-the-art techniques.

##### 2.4.4.1 Production PBR

Production PBR emerged roughly in 1990 along with the increasing complexity of the modelled effects displayed in early motion picture titles such as *Antz*, or *Toy Story*. The intricate behaviour between light and matter had to be captured and rendered with physical accuracy, and as such, this demand propelled research in a number of areas including high-performance computing (HPC), dedicated multi-threaded graphic core architecture for mathematical operation acceleration, numeric models for light-surface interaction, and novel graphics APIs.

One of the pioneers of production PBR is DreamWorks Animation (DWA) studios. Their incipient framework, *D\_RENDER* relied on a Lambertian diffuse surface model coupled with Phong specular lighting and an added Fresnel component. This approach was unable to model light-skin interaction, and as such Jensen and Buhler [68] developed an octree-based implementation for diffusion-based subsurface scattering which allowed for accelerated rendering of not only light-surface interaction on skin, but would also account for the diffuse light exiting the surface. Early work produced by the studio would rely on two types of illumination: point and directional lights. As a consequence, indirect

illumination was manually authored by an artist relying on *fill lights* placement which was a slow process and often resulted in physically inconsistencies. This issue was later addressed by Tabellion and Lamorlette [144] who constructed an irradiance cache-based global illumination estimation pipeline. Subsurface scattering for skin modelling and the illumination constrained directed the research and development efforts towards more physically-grounded approaches. This trajectory was also motivated by the desire to create richer content aimed at character design, and to solve inconsistencies between surfacing and production lighting.

In a number of DWA’s productions, glass and metal represented ubiquitous materials in their environments, and consequently were modelled through the first classic metallic-roughness PBR with a BRDF Fresnel term. This however, was not a general solution to capture material appearance, so Donner and Jensen [39] proposed new work on multi-layered translucent materials, where skin would be treated as a three-layered material giving artists full control on the micro-granularity of shading. In all the previous PBR models, DWA accounted only for direct light interaction, and there was no model in place for integrating indirect lighting. Křivánek et al. [88] addressed the hard surface reflections under IBL-based area and environmental light maps through an importance-sampled Ward BRDF approach leveraged by octree-traversal within the point-based (PBGI) global illumination framework.

Similar to Disney’s layered material system [20], DWA’s PBR framework relies on a multi-layered material system in conjunction with an uber material shader. A number of commonly encountered material types are derived from this class including: metal, refractive, and solid dielectric, and can be combined in a network, or used independently (as observed in 2.24).

<b>Metal</b>	this specialized material relies on the anisotropic Cook-Torrance microfacet model with Beckmann BRDF [158]. The Fresnel term is parameterized using metallic reflectivity and edge tint colours [54].
<b>Refractive</b>	relies on the same microfacet model as the metallic material with the added roughness map for specular highlights.
<b>Solid dielectric</b>	similar to the refractive material in terms of specular reflections. The diffuse reflection relies on a scattering parameter. When it is set to 0, the default is a Lambertian BRDF model, otherwise a BSSRDF is used instead for dipole model [69].

When combined, the shading is resolved in two stages. The initial parameter resolution stage performs an exhaustive iterative depth-first traversal of all the materials, analysing their parameters and attached texture data. The resulting parameters are interpolated resulting in a single set of parameters at the end of the resolution stage. The next step takes the resulted parameters from the previous part, and creates a set of closures specifically

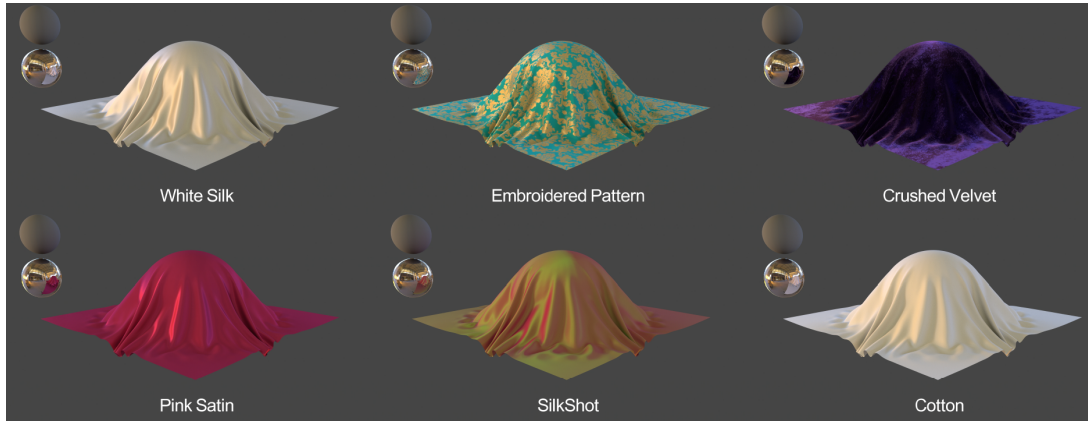


Fig. 2.24 DreamWorks Animation Studio importance sampling fabric BRDF [60]. Various fabric appearances under artist-authored parameters such as fabric density, fabric thread direction, orientation, and roughness. For each case illustrated, the fabric appearance is modelled as a combination of the nodes defined in the works of Krivánek et al.[88]. The intense specular areas come from the metal node value, characteristic of the metallic reflectivity parameter in the Fresnel equations used to solve the node. The self-shadow highlighted areas are characteristic of a roughness parameter derived from the refractive node in the layered material model. Combinations of different values for these nodes will yield a wide range of appearances for materials under the same illumination conditions.

for the parameters. This ensures that only a single instance of the shader program exists to define a specific material at run-time.

In contrast to the specialized BRDF models adopted by DreamWorks, a number of studies [6] demonstrate that production PBR frameworks could benefit technical artists' in terms of productivity if their design brought together intuitiveness and physical accuracy from real-world interaction. Walt Disney Animation Studios based their approach on measured BRDF studies and developed a framework capable of comparing simulated numeric BRDF solutions to real-world datasets in order to allow the user to validate and author materials.

As previously described in the background section 2.2.1, specifically in the microfacet model, given a surface reflection covering the area described by the light vector  $l$  and view vector  $v$  then there exists a vector halfway between the two, which is perpendicular to a microfacet. This vector is known as the *half-vector*  $h = \frac{l+v}{|l+v|}$ . In the case of an isotropic material, the microfacet models can be summarised as follows:

$$f(l, v) = \text{diffuse} + \frac{D(\theta_h)F(\theta_d)G(\theta_l, \theta_v)}{4 \cos \theta_l \cos \theta_v} \quad (2.37)$$

Where the diffuse term is assumed in many cases to be Lambertian,  $D$  represents the specular term (through a distribution function),  $F$  represents the amount of light reflected

according to the Fresnel reflection laws, and  $G$  denotes the geometric attenuation or self-shadow factor. The parameter  $\theta_h$  is the angle describing the normal to the microsurface and the half-vector,  $\theta_d$  is the angle resulted from the difference of the incident light vector and the half-vector. Microfacet models differ from other types of representations only when it comes to explicitly relying on the term  $\frac{1}{4 \cos \theta_l \cos \theta_v}$  that comes from the microfacet derivation.

This approach differs from the one created by DreamWorks in that it follows a set of principles, such as:

1. This model favours intuition from real-world compared to physical accuracy.
2. Relies on as few parameters as possible with a range clamped in the range  $[0, 1]$ .
3. The combination of parameters should still render a plausible result.

The layering parameters available for a user include: base coat, subsurface layer, metallic, specular, roughness, anisotropic, sheen, clear coat. These layering parameters are also spatially varying and allow blending through masks.

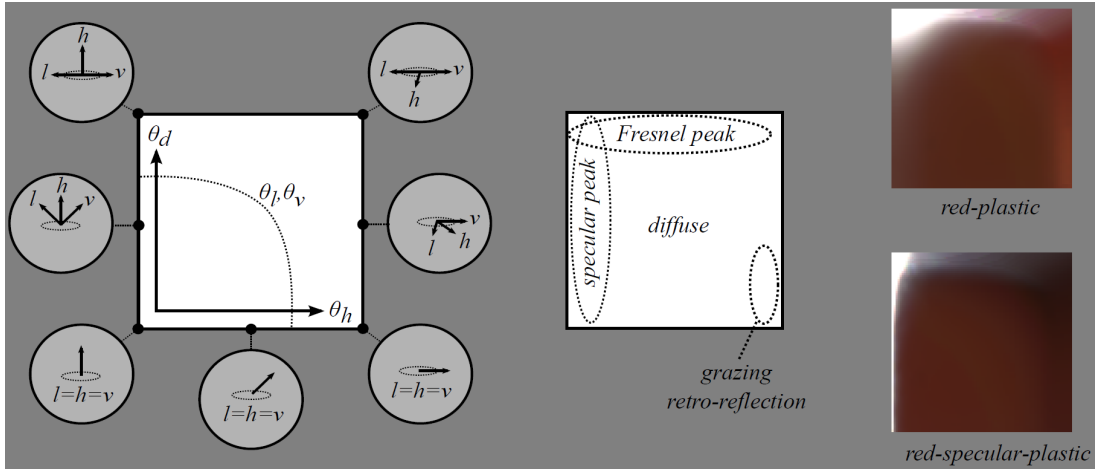


Fig. 2.25 Image slice from the BRDF viewer developed by Disney [20]. The slice space illustrates how to understand the different parts of the microfacet model for a variety of materials conveyed in the MERL dataset. In the present case, the image slice is characteristic of red-plastic, expressed schematically under different incident light ( $l$ ) and viewing angles ( $v$ ).

#### 2.4.4.2 Real-time PBR

One of the more popular real-time implementations of PBR was developed by Epic for the Unreal Engine (UE) [75]. Their method is similar in spirit to Disney's offline PBR system, and follows a list of guidelines adapted for real-time. The main difference between Disney's and UE's approach is that UE requires to use IBL and analytic light sources interchangeably while maintaining parameter's behaviour consistent. Additionally, UE relies on algorithms that optimize for a vast amount of light sources in real-time (deferred rendering, forward plus rendering, tiled rendering, etc.) which means material transparency becomes a challenge to model with traditional PBR, as geometry of a specific material type opacity has to be presorted and rendered twice.

In their material representation framework, UE rely on a classic Lambertian model for the diffuse term. The specular microfacet BRDF model relies heavily on Disney's functions for modelling the specular D, G, and F terms, with modifications only where optimization for real-time was crucial at the small penalty of physical accuracy. The normal distribution function (NDF) relies on the GGX/Trowbridge-Reitz instead of Blinn-Phong, and the value of  $\alpha$  is now replaced by *roughness*<sup>2</sup>. By relying on the simplified version of the expression 2.37 (often used in literature for real-time material appearance modelling), we can express the specular D term as a function of the normal to the microfacet halfway between view and light vectors:

$$D(h) = \frac{\alpha^2}{\pi((n \cdot h)^2(\alpha^2 - 1) + 1)^2} \quad (2.38)$$

Unlike Disney's approach, UE represents the specular geometric attenuation term G through Schlick's model [133] where the specular highligh value is remapped to  $\alpha/2$ . This essentially matches the Schlick model to the Smith model [158] for  $\alpha = 1$  and anywhere in the range  $[0, 1]$ . Analytic light sources also remap the roughness to  $\frac{roughness+1}{2}$  before squaring.

$$G(l, v, h) = \frac{n \cdot l}{(n \cdot l)(1 - k) + k} \cdot \frac{n \cdot v}{(n \cdot v)(1 - k) + k} \quad (2.39)$$

$$k = \frac{(roughness + 1)^2}{8} \quad (2.40)$$

The Fresnel term (F) is model through a conjunction of both Schlick's approximation and Spherical Gaussian approximation [90] for replacing the exponent. This approach was chosen as a speed optimization as the quality difference is almost imperceptible. The term is expressed as a function of the specular reflectance at normal incidence  $F_0$ :

$$F(v, h) = F_0 + (1 - F_0)2^{-5.55473(v \cdot h) - 6.98316(v \cdot h)} \quad (2.41)$$

In UE's implementation of the material model, efficiency and a reduced size of the required screen buffers are important aspects for real-time rendering. As a consequence, the number of material types allowed is a simplification of one used by Disney, encompassing: a base colour layer, a metallic layer, a roughness, and cavity (used for self-shadow). The novel parameter is the last one - cavity (seen in Figure 2.26). Unlike screen-space ambient occlusion (a post process technique that approximates a point's exposure to ambient lighting) or real-time shadow mapping (a technique described in detail in Section 3.3), the cavity material type deals with the shading of micro-geometry such as the narrow space between cloth fibres defining the same material type. This parameter is used in lieu to the well-known specular layer. Their findings suggest that the specular term was confusing the overall control over material appearance, having a default value of Burley's 0.5 corresponding to 4% reflectance.

One cause for concern with the material layering approach is that some parts of the same mesh may require different material properties, which leads to splitting the mesh into submeshes. In turn, this approach generates more draw calls per frame, causing a penalty to the real-time performance. Despite this concern, the more generalized and layered model has proven to increase productivity and add realism to the assets.

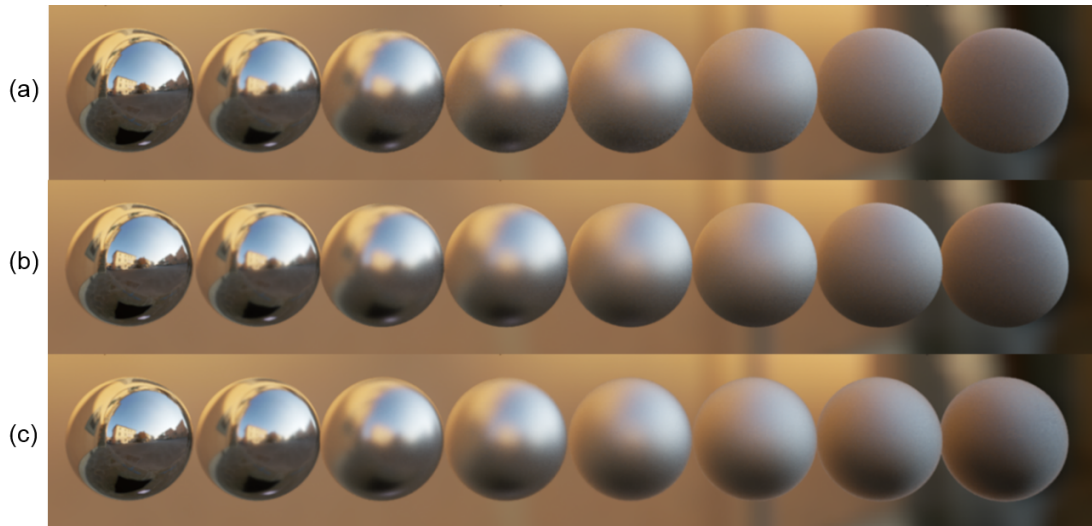


Fig. 2.26 Materials look development in Unreal Engine [75] (a) Row of real-world captured and measured material properties. (b) Row of computed material appearance using the split-sum approximation. (c) Row of computed material appearance using the full approximation. Both approximations rely on the assumption of radial symmetry.

Another notable advancement for real-time PBR is developed by Activision Infinity Ward (AIW) [41] based on Disney's offline production framework. Unlike Unreal Engine's



approach, AIW can model a variety of materials while still optimizing for speed, however, they are not guided by the same set of principles as either Disney or UE. They achieve a generalized material system through their specialized Material Compiler.

The multilayer material approach in AIW's case was driven by the need to handle complex materials which occur in the real-world and are multilayered in their physical nature. Such complex materials include but are not limited to: frosted glass, thin films (i.e. oil patches, soap bubbles, coatings/varnishes), impure metals (where the surface is galvanized, oxidized, covered in oil or dirt). Instead of relying on specialized material models to handle each complex case that arises (snow, ice, sand, car paint, hair/fur), AIW researched a way to tie all of the models into a single concept, so that the shader pipeline could run auto-optimizations and also allow for the engine to not interfere with the PBR framework when blending and other core GPU functionalities were specified.

The initial and stand-out step towards the unified layered material approach is to address macro geometric structures that conceptually would be mapped by the same material. In a specific and ubiquitous case this would include textile modelling - adapting the material properties/PBR model based on the cloth's stretch (change of density), and view angle which affects the perceived opaqueness. By deciding on this granularity, the layering system was designed with three different domains in mind: a *surface layer* which describes the interface of the material, the *macro layer* which defines the structure inside the material, and a *micro layer* which addresses the medium's lighting model (Figure 2.27).

A challenging problem posed early-on was how to unify each of these domains, specifically the macrostructure. This has been addressed by considering the blend factor - the further away a macrostructure is viewed from, the more it looks like a consistent, uniform aggregate, as if it's constituent structures were blended together. Materials that conceptually adhere to this rule include textiles, grainy structures such as sand, salt, snow, ice. The blend factor is a function of both the thickness and density [40] parameters of the material to-be-modelled in a specific layer. The microstructure layer handles scattering and transmittance of refracted rays from one medium to the next until it is fully absorbed. This approach is similar in spirit to Dupuy's [42] mix between volume and surface rendering.

Once the three domains are validated, a material definition for the compiler can be formulated in terms of classes of parameters belonging to either [19]: surface properties (normal and roughness maps), micro properties (albedo, specular, and refraction maps), or macro properties (density and thickness values). These categories were chosen for ease of surface decomposition and generalisation of a material model that is physically-based, intuitive to author, and suitable for real-time interactive applications.

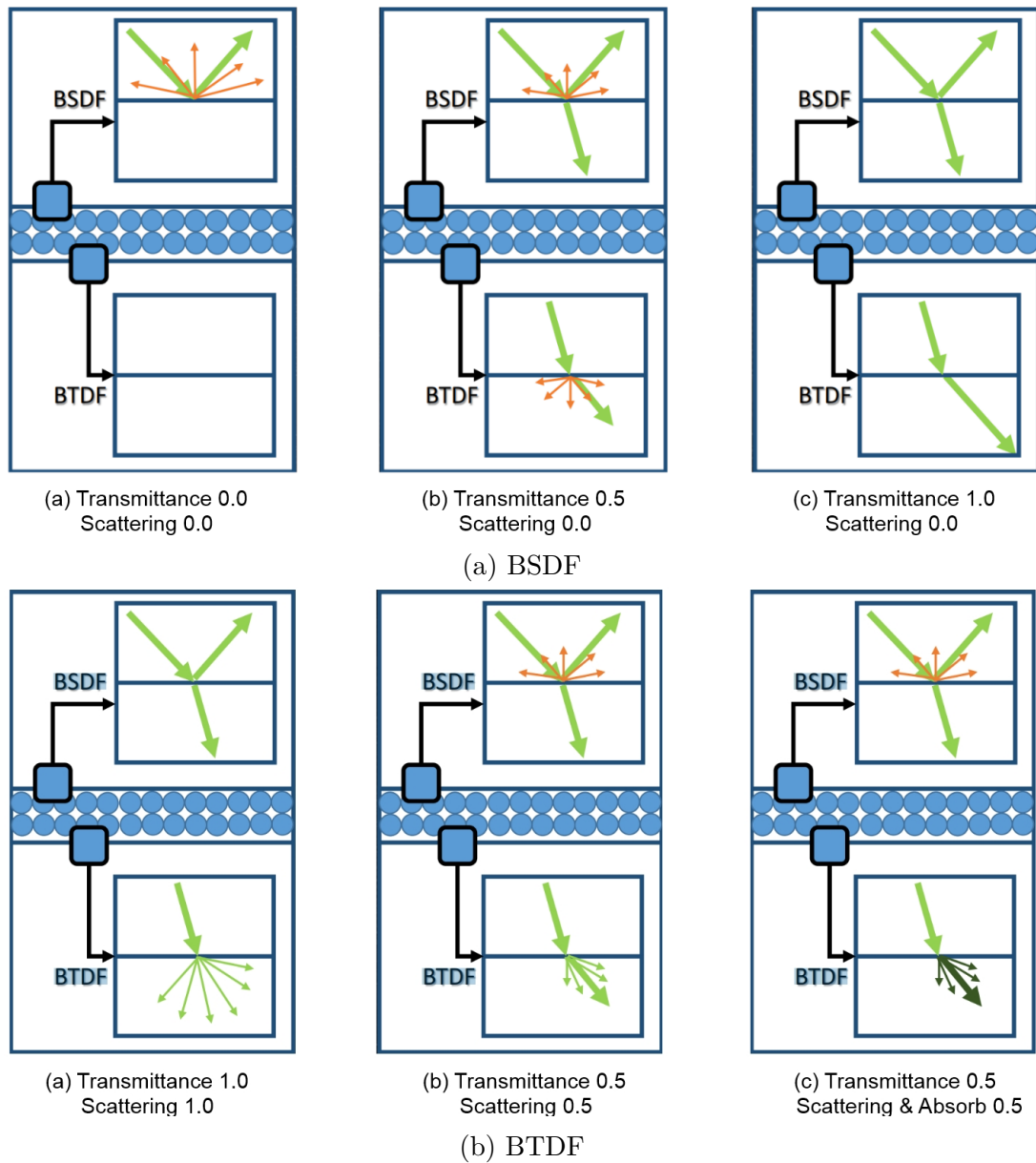


Fig. 2.27 Diagrams of surface decomposition models used in Activision's Infinity Ward [41]. The first row of diagrams represents light interacting with the surface at various transmittance values where no scattering takes place. In these cases a simplified BSDF model will be used to simulate the interaction. The second row of diagrams presents the interaction when both transmittance and scattering vary. The latter case is simulated using a BTDF model which accounts for both surface properties as well as macro properties. In both cases, the diagrams can be interpreted as an amount of light absorption upon entering and exiting the surface that is modelled in two ways depending on the type of material used.

## 2.5 Background summary

The present chapter is divided into two sections, the first of which lays the theoretical foundation required to understand the related work covered in the second part of this chapter. The related work section aims to describe, in chronological order, the most stand-out advancements made in the different areas which directly relate and influence the process of compositing.

Commonly, the results of the compositing process are tied to the 2D domain (video or single-frame), however, for any compositing to offer consistent results, a conversion from the 2D to the 3D domain is assumed. Due to this requirement, the process of compositing can be best defined as a pipeline, or a series of steps where at each stage a single problem is solved. Research in compositing pipelines is sparse, but despite this, novel methods emerge in the individual areas that comprise a part of the stages involved in the pipeline, most notably stereo reconstruction, and scene property inference. By structuring the related work to fit into these categories, it is straightforward to map the related work to each individual stage in the pipeline described in Chapter 3.

Section 2.4 covers a variety of methods that have application to the compositing process, however, only a limited number of these techniques are directly relevant to the work described in the present document. In the work of Karsch et al. [79], the first automated compositing pipeline was introduced. The pipeline presents elements that have undergone automation (lighting), as well as elements which are user-guided (annotation, artefact placement).

Another closely-related work is that of Zhang et al. [172], in which compositing takes place in the context of room visualisation. In this scenario, a room captured with the Google Project Tango hardware is represented in 3D for visualisation and allows a user to de-clutter the captured room, and proceed in compositing (introducing) 3D artefacts (i.e. authored 3D elements of furniture) in the virtual setup. Although the work of Zhang et al. takes place only in the 3D domain, the authors have created an example of a compositing pipeline that is hardware oriented, as opposed to Karsch et al. who have strived to generalise the compositing process. Nonetheless, the work of both groups illustrates different ways to define compositing pipelines, along with a set of assumptions and limitations specific to their chosen direction, which is what inspired the approach described in the following chapters.

# Chapter 3

## Methodology

### 3.1 Chapter Overview

The present chapter spans across each step of the proposed compositing pipeline and describes in more detail the approach and technical contributions per stage. Similarly to the inverse rendering techniques discussed in Section 2.3, the software stages are separated to fit an inverse rendering problem (Figure 3.2). The scene geometry term is characterised by the initial input. After this stage, robust reconstruction takes place, converting from 2D to 3D space, in order to enable more accurate lighting calculations. The following step solves jointly for illumination and material estimation, and relights the scene after composition takes place. The last step is reprojection - converting the 3D calculations to a specific 2D system that matches the initial input parameters (viewing direction and orientation, viewing volume).

The framework takes as initial input a pair of stereo images that are rectified [102]. This ensures that the search-space for the reconstruction stage is trimmed considerably. There are a number of state-of-the-art techniques for computing the image rectification which are out of the scope of the present research [74], however, it is worth noting that these solutions are not hardware-dependent, and therefore simplify the acquisition process of rectified images in the wild. Stereo images were chosen as the preferred input data, as they can easily be accessed in an AR or VR setup that comes from either phone/tablet front sensors, or from a head-mounted display or pair of AR goggles where the image is formed per each lens (left, and right eye accordingly). Inferring depth from a *single* stereo pair is difficult in comparison to a stream of images (video), as there is more data to sample and a far greater range of techniques in the latter case. One such technique is background and foreground segmentation [22], which is invoked in video matting and compositing software for the movie industry [28]. Once the input undergoes a custom matching process that transforms the 2D image data into 3D geometry that is compatible

with the OpenGL depth buffer construct (i.e. an image that holds values compatible with a depth image format), the data is ready for dense reconstruction. In addition to geometry, the framework relies on the stereo pair to infer the light distribution in the scene, along with the detection of strong luminance areas, indicative of potential primary or secondary light sources. The framework relies on HDR images, however, if the images are in an LDR format, the framework will match the recovered map to a similar one available from the HDR dataset [33]. From depth, an initial occlusion map can be inferred, and this is further used after the 3D artefacts are added in order to inform which areas of the scene are occluded after compositing (an artefact can block a light source). The last step that follows after relighting is the reprojection - converting the result obtained in the 3D system back to a 2D setup that matches the viewing volume and orientation of the initial input.

In contrast to the relevant related work, the proposed method does not rely on bespoke hardware for depth acquisition ([172]), and aims at providing the results of compositing in real-time while solving the issue of multiple object addition that is raised in the works of Karsch et al. [79]. The relighting stage of the pipeline, based on an inferred occlusion map, informs which areas will be occluded post compositing and allows the computation of physically-accurate shadows to accompany the 3D artefacts yielding a consistent result.

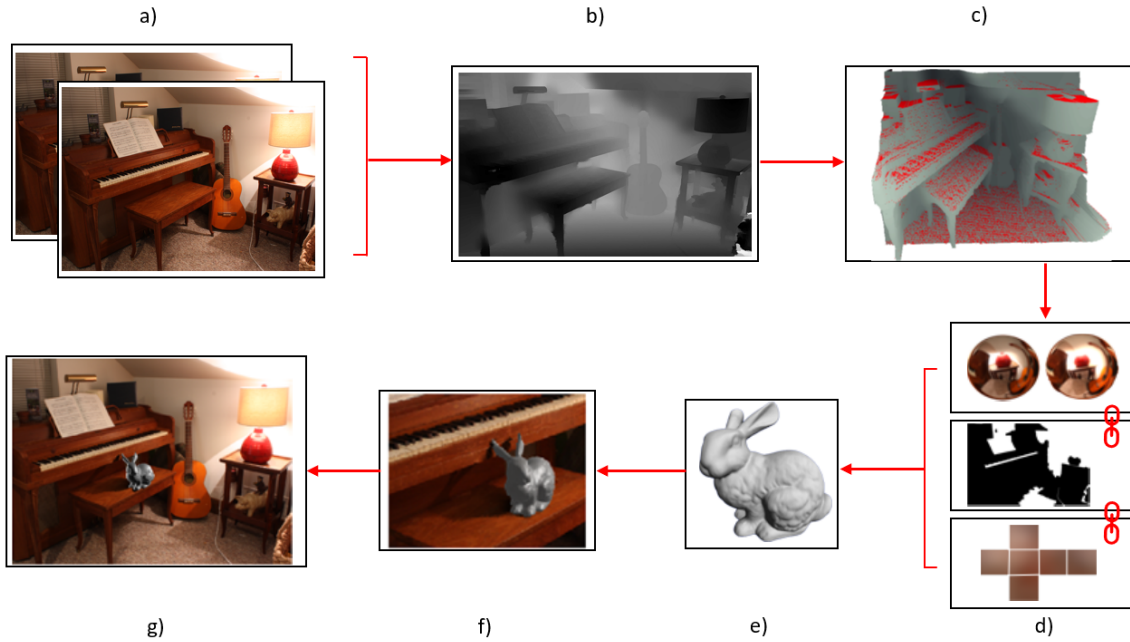


Fig. 3.1 Overview of the pipeline stages. a) Input consisting of a stereo pair. b) Recovered depth buffer. c) Dense reconstruction with normal data (in red). d) Images in the stack represent the different types of properties inferred from so far in the framework: radiance map, estimation of the occlusion map, environment map. e) Example of a 3D artefact used for compositing. f) Preview of the composition results. g) Result of reprojection.

## 3.2 Scene depth reconstruction

Acquiring depth information from a pair of images in the absence of hardware meta-data or proprietary RGB-D capable technology is an established domain of problems in computer vision. The stereo vision process closely mimics its biological counterpart - stereopsis through a number of steps that aim at recovering a disparity map, equivalent of the binocular disparity based on parallax in the human visual system. Generally, the two images captured with two separate sensors undergo a number of domain transformation before the core comparison algorithm can produce an accurate disparity:

**Projection matching.** An optional step to remove tangential distortion allowing the observed image to match the projection in the pinhole camera model.

**Rectification.** Using epipolar geometry constructs Figure 3.2, the images are projected back to a common plane in order to facilitate feature analysis. Moreover, this stage determines the fundamental ( $F$ ), the essential ( $E$ ), and the intrinsic calibration ( $K$ ) matrices that are then used further in the pipeline for reprojecting the final output.

**Correspondence.** The core algorithm for comparing and extracting similar features. Can be classified as either a correlation-based or feature-based approach.

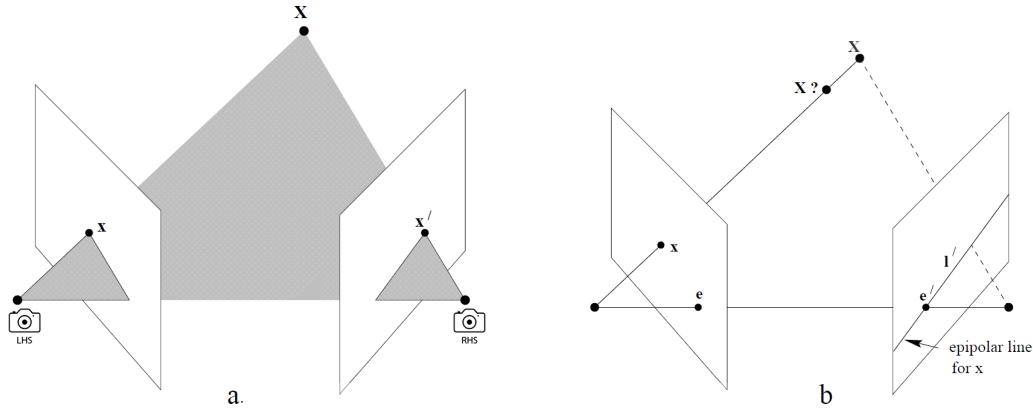


Fig. 3.2 Point correspondence overview with a) Centred left hand side and right hand side cameras along with their corresponding image lines  $x, x'$  in a common plane outlined in grey. b) The centres of the cameras are marked as the epipolar centres in  $e, e'$  in order to describe the back-projection of the point  $x$  to the ray  $l'$  defined by the LHS camera and that point. When projecting  $X$  it overlaps  $x$  and belongs to ray  $l'$ .

The setup described in Figure 3.2 can solve jointly for both initial camera pose estimation, as well as depth reconstruction. In order to compute the difference in pixels from the left image (originating from the left camera) and the right image (acquire a disparity value), the two images must undergo projection to a common plane. The process of calculating the binocular disparity connects the depth at a point to corresponding location

identified by the counterpart camera or eye. This approach assumes that parameters defining locations of cameras are known *a priori*.

In the approach discussed in the upcoming Section 3.2.1, the assumption is that the images used as input are rectified (matched on a height basis). If this is not the case, the images will undergo a straight-forward process of rectification, where a rotation matrix  $R$  is computed based on the coordinate system of the first and second camera matrices. The rectification process imposes a constraint that the epipolar lines in the rectified images are vertical and have the same x-coordinate. By doing this, the matching algorithm illustrated in Figure 3.3, will search on a row basis instead of iterating over adjacent neighbours in 8 directions.

For every pixel in the resulting image, the disparity at that pixel is given by the delta value expressed as an energy minimization function of the sum-of-squared-differences for the current pixel. The process of identifying corresponding pixels in a given image pair is known formally as *stereo matching* and it can be classified as local (uses a small window of comparison in which specific features are considered) or global (keep track of region continuity) depending on the desired accuracy and constraints of the problem. This type of algorithm is employed in the present framework for acquiring a depth map compatible with the graphics API's depth buffer.

### 3.2.1 Initial depth acquisition and refinement

The first contribution in the broader context of automating compositing of 3D artefacts in images (stream of images) for real-time applications is to ensure that the obtained depth map from the correspondence matching algorithm is smooth without the loss of significant structural information. This endeavour is motivated by the real-time nature of the application domain - most techniques run an intermediate refinement or correction routine after the correspondence matching in order to prepare the scene for dense reconstruction from the disparity map (smoothing a mesh based on the constituent primitives' orientation). This incurs a critical time penalty in commercial compositing applications limiting them to offline post-processing. In contrast, our approach brings together correspondence matching and cascaded convolutions for ensuring a smooth depth map without topologic loss that is faster than the additional refinement step and completely removes the need for one in the context of dense reconstruction.

As seen in Figure 3.3, the *matching cost function* calculates the similarity percentage between two areas, corresponding to each image. In order to carry a preliminary matching test, normalized cross correlation can be applied to the values of the two windows. This is achieved by computing a row-wise mean and standard deviation for each pixel intensity. The mean intensity is subtracted from each pixel's intensity value. Mean intensities from corresponding windows are multiplied then summed over the area of the entire image.

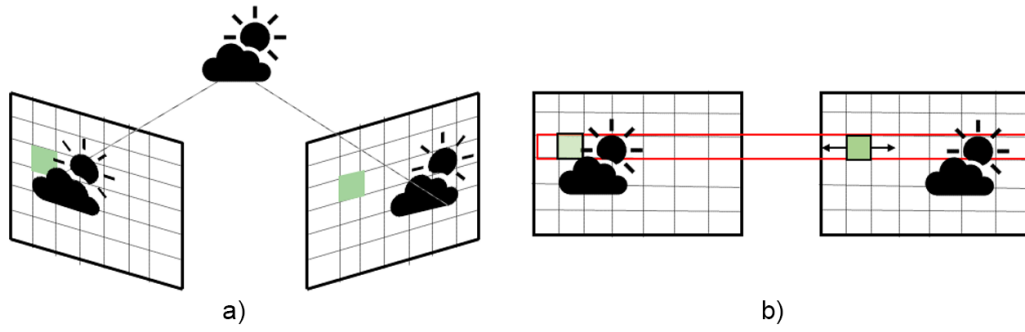


Fig. 3.3 a) The matching process commonly involves enclosing a region of pixels belonging to the left image in a window. b) The defined window is then checked, horizontally on a *scanline* with several windows of the same size in the right image beginning from the same position which is considered to have a disparity value of 0. The window never slides more than one pixel at a time (increments disparity by 1 unit a step). The right window value which minimizes the cost, matches the left image with the highest similarity.

Lastly, in order to get the cross correlation value, the sum is divided by total number of pixels in either image (since they have the same area) and by each standard deviation. The similarity of a match is directly proportional to the normalized cross correlation result, and should be carried on images that are not height-aligned before matching takes place. This is encouraged in order to reduce the problem space.

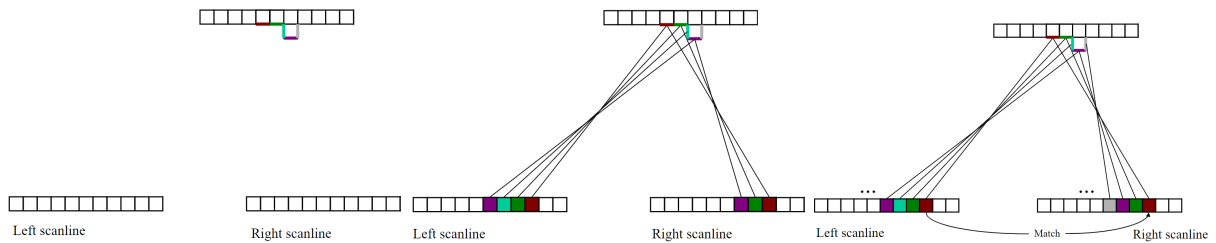


Fig. 3.4 Building disparity from two stereo images in classic SGBM. The three images describe two scanlines going through  $N$  pixels in each line (spanning across the entire image width). The pixels within the arrays are up for comparison and can be considered as matched or ignored (they may be missing in one of the two images). The routine reiterates per row in the image yielding the final disparity (horizontal shift) of each scanline. This displacement should minimize the mean-squared error between the right block area and its corresponding left block.

The incipient step relies on two stereoscopic images and constructs a disparity map through the modified semi-global block matching (SGBM) approach [61]. Due to its resilient result in the presence of large homogeneous textures, the SGBM becomes an ideal candidate for disparity map acquisition (Fig. 3.3). In order to obtain a high-fidelity map for the rendering stage, it is important to handle the remaining disparity occlusions, often present at this stage due to lack of pixel correspondence within the image pair. Correlation matching is a technique that aims to identify the degree of similarity on a collection of



pixel values by considering adjacent pixels within a window, and not a horizontal scanline. This approach is desired in dense reconstruction due to its ability to decrease the search space. When the input data contains depth discontinuities or large uniform diffuse texture areas that contain little structural information, correlation matching can consider the wrong content introducing significant ill-elements.

As opposed to regular block matching (BM) techniques, semi-global matching (SGM) relies on a number of sequential steps: identifying highest matching pixel values, and invoking a smoothness constraint through the aggregation of sub-constraints. Mathematically, the SGM can be expressed as a 1D variation of a 2D Markov Random Field (MRF) since the 2D version is NP-hard:

$$E(D) = \sum_p C_p(d_p) + \sum_{p,q} V(d_p, d_q) \quad (3.1)$$

$$V(d, d') = \begin{cases} 0, & d = d' \\ p_1, & |d - d'| = 1 \\ p_2, & |d - d'| \geq 2 \end{cases} \quad (3.2)$$

$E(D)$	energy minimisation function
$d$	disparity value of each pixel $p$ , $d \in D[d_{min}, d_{max}]$
$C_p(d_p)$	unary term for the matching cost of $p$ to $d$
$V(d, d')$	pairwise smoothness term penalizing disparity mismatch of adjacent pixels

When considering the 1D MRF case, the SGBM solves the minimisation in 8 directions on a block window  $r$  via a lookup table. This step is known as pathwise aggregation based on a matching cost heuristic  $L_r(p, d)$ . Equation 3.1 then becomes:

$$L_r(p, d) = C_p(d) + \min(L_r(p - r, d') + V(d, d')) \quad (3.3)$$

We rely on the simplified formula and define the cost volume term (sum of 8 cardinal directions representing the smoothness) as  $S(p, d)$ , where the disparity value at each pixel  $d_p = \operatorname{argmin} S(p, d)$ :

$$S(p, d) = \sum_r L_r(p, d) \quad (3.4)$$

Our method builds intuition from the observations regarding the relation between the eight individual local minima cost  $L_r(p, d)$  and the sum of all minimum costs  $S(p, d)$ , namely that the local minima is a lower bound of the cost volume at each pixel  $p$ . Therefore,

we can express the difference between the two as the uncertainty term:

$$U_p = \min_r \sum L_r(p, d) - \sum_r \min L_r(p, d) \quad (3.5)$$

Areas where the minimum-cost paths contain a match will yield a small value for  $U_p$ . The sum of all costs  $S(p, d)$ , as mentioned in the case of correlation matching, is a function of the entire set of 1D cost paths.

The classic SGBM returns integer-values after block matching which causes pronounced contouring where there is lack of pixel correspondence (areas of different disparities do not align causing noise). In order to address this introduced artefact, the present work samples not only the minimum cost but also the next two adjacent matching costs. Once these are obtained, a correction computation is carried by fitting a parabola through the samples and minimizing the result.

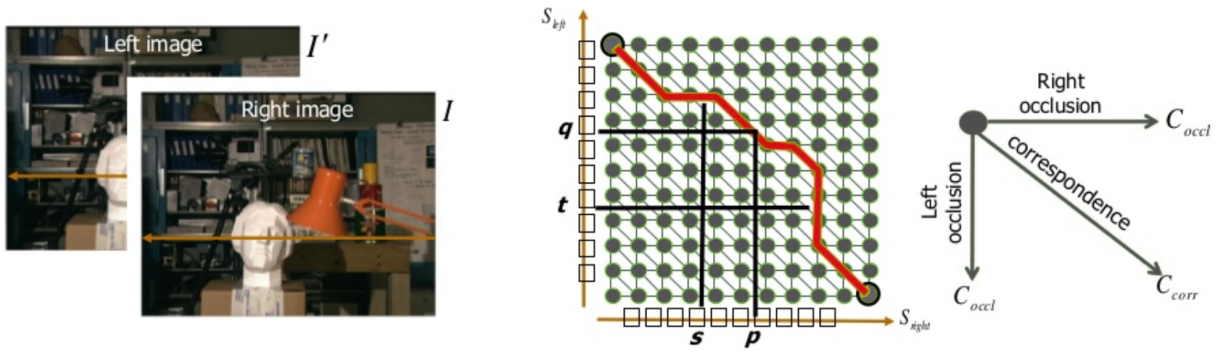


Fig. 3.5 Dynamic Programming (DP) is used to minimize  $E(D)$  per scanline (described independently in Fig. 3.3). The result is the optimal path in the grid of matches for the ordering constraint when using a scanline for each input image.

The scanline problem that leads to identifying optimal approximations for a line of pixel values translates into identifying a path of intermediate solutions across the 2D domain. The block matching metric is used as the cost function in order to map the disparity values to a small variance. This is computed using a dynamic programming approach, similar in spirit to the proposed method of O. Veksler [157].

Dynamic programming, as illustrated in Fig. 3.5, can give rise to ill-elements that manifest as undesired blurred contours (presumably where objects of different textures are delimited) due to the constraint introduced in the solution calculation. Moreover, it does not diminish the necessity of smoothing between rows (upper and lower window edge), which is why a striation pattern can be present systematically on the foreground blocks. Despite these limitations, the DP modification from the original matching process produces a result with the noise along the vertical block side eliminated, and with foreground features being well delineated.

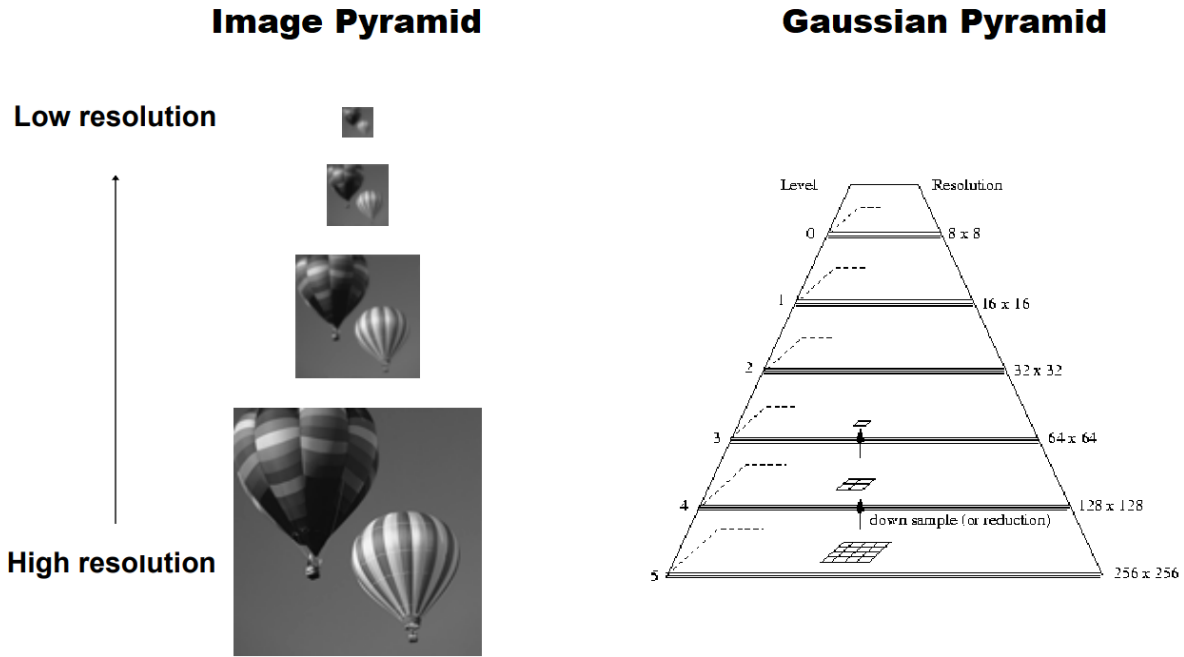


Fig. 3.6 Cascaded Convolutions. Left hand side overview of samples in the spatial domain. Right hand side is the equivalent image pyramid in the spatial-frequency domain (first is lowest frequency with a smooth detail, and last is finest detail.). Our approach defines a reduce operation using Gaussian pyramids, and an expand operation using Laplacian pyramids.

The work implements cascaded convolutions (known commonly as the "image pyramiding" approach first formulated by Thevenaz et al. [147]) in conjunction with block matching and dynamic programming. As described in Fig. 3.6, we modify the initial approach of Thevenaz et al. and rely on a Gauss pyramid constructed with the reduce operation, and use it as the input for constructing the Laplacian pyramid (filtered pass) using the expand operation. During the reduction process, the pyramid layer is obtained by running a low-pass Gauss kernel to the upper part and reducing the size in half. Each layer will be in a different low-pass filter range. Each layer of the Laplacian Pyramid is constructed from the difference of two consecutive Gauss pyramid layers. The subtraction is valid if the lower layer is upsampled first to be the same size as the upper one (the expand operation). The band-pass layers defining the Laplacian pyramid allow for different frequencies to be modified independently, without affecting the local features (from spatial domain). When considering the entire domain for disparity matching, the search-space incurred a significant time penalty. Through constraining the domain to half its initial size, only an area of a quarter of the adjacent pixels would be contributing to the active list used in the search. The estimated disparity values found within the scaled domain can be reintroduced in the search for the original domain. Finally, the approaches described can be merged as described in Algorithm 1.

**Algorithm 1** Depth map reconstruction

---

```

1: left ← imread("left.png")
2: right ← imread("right.png")
    ▷ Convert the image matrices from RGB channels to greyscale by averaging
3: leftI ← mean(left, 3)
4: rightI ← mean(right,3)
5: range ← 50
6: halfWindowSize ← 4
    ▷ Initialize dynamic programming process
7: Ddynamic ← zeros(size(leftI), 'single')
8: [imgHeight, imgWidth] ← size(leftI)
9: finf ← 1e3
    ▷ All values in the disparity cost matrix are initialized to MAX INFINITY
    ▷ The cost matrix contains one minRow/image column, one column/disparity
    value
    ▷ Used for a single minRow of the image, then reinitialized for the next
10: cost ← ones(imgWidth, (range * 2) + 1)
11: penalty ← 0.5
12: for m ← 1 : imgHeight do
13:     ▷ Re-initialize the disparity cost matrix
14:     cost(:) ← finf
15:     minRow ← max(0, m - halfBlockSize)
16:     maxRow ← min(imgHeight, m + halfBlockSize)
17:     for n ← 1 : imgWidth do
18:         minCol ← max(1, n - halfBlockSize)
19:         maxCol ← min(imgWidth, n + halfBlockSize)
20:         ▷ Limit the search so that we do not exceed image bounds
21:         mind ← max(-range, 1 - minCol)
22:         maxd ← min( range, imgWidth - maxCol)
23:         ▷ Cache matching costs
24:         ▷ Compute the SAD between the template at pixel (n, m) and all blocks in range
25:         for d ← mind : maxd do
26:             cost(n, d + range + 1) ← ...
27:             sum(sum(abs(rightI(minRow:maxRow,(minCol:maxCol)+d) ...
28:                 - leftI(minRow:maxRow,minCol:maxCol))))
29:         end for
30:     end for
    ▷ Compute disparity costs with DP for the scanline
31:     ▷ optimalIndeces will be a lookup table which will hold partial
32:     ▷ solution for disparity for the pixel in column k+1 given pixel k's disparity.
33:     optimalIndices ← zeros(size(cost), 'single')
34:     ▷ Start with the SAD values for the rightmost pixel on the current line of the image.
35: end for

```

---

**Algorithm 2** Continuation...

---

```

1:  $cp \leftarrow cost$ 
2: for  $j \leftarrow imgWidth-1:1$  do  $\triangleright$  Select the SAD values for the next pixel to the left, and
   modify:
3:    $\triangleright$  Replace the leftmost and rightmost block SAD value with 'cfnf' (which grows
   linearly in magnitude as we move left).
4:    $\triangleright$  Add the minimum values from the above matrix to all of the SAD values
5: end for
                                      $\triangleright$  backtrack path of intermediate solutions

6:  $[ ,ix] \leftarrow \min(cp)$ 
7:  $DP(m,1) \leftarrow ix$ 
8: for  $k \leftarrow 1:(imgWidth-1)$  do                                      $\triangleright$  Set the next pixel's disparity
9:    $\triangleright$  Lookup the disparity for the next pixel by indexing into the optimalIndices table
   using the current pixel's disparity
10:   $DP(m,k+1) \leftarrow optimalIndices(k, ...$ 
11:   $\max(1, \min(\text{length}(optimalIndices,2), \text{round}(DP(m,k)) ) ) )$ 
12: end for
13:  $DP \leftarrow DP - range$ 

```

---

The steps described in the depth map reconstruction algorithm follow a set of operations that strive to accurately find matching pixels while also removing any artefacts introduced. Depending on the area in which matching takes place, intermediate results can cause visible patches around the area of interest, blocks, or streaks. Each matching heuristic introduced is aimed at removing or minimising these unwanted effects.

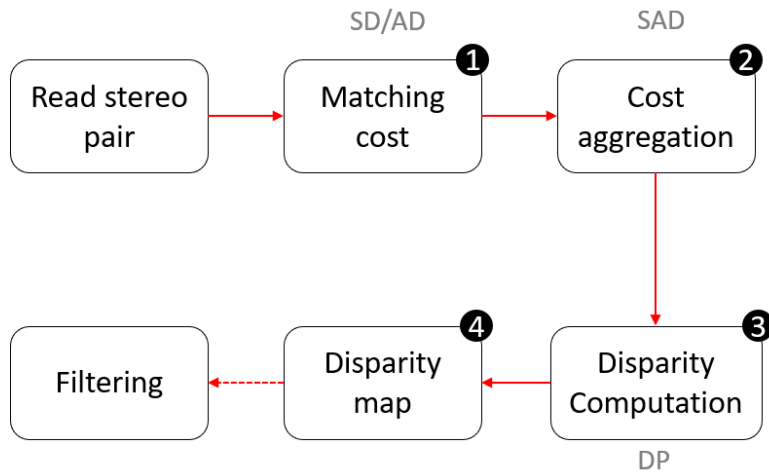


Fig. 3.7 Overview of the reconstruction steps. 1) The matching cost for assigning a disparity hypothesis to each pixel value. 2) Aggregation of the initial matching costs spatially across matched regions. 3) Computation of the best (unique) disparity hypothesis per pixel - global or local minimum cost function. 4) Obtained result can then be further refined if it still presents noisy regions.

As seen in Figure 3.7, in order to define similarity based on intensity between the pixels belonging to the left-hand image and the ones present in the right-hand image, a matching cost is introduced. This pixel-based matching cost is expressed as an absolute difference. This step introduces unwanted noise, and in order to obtain better results, the sum of absolute difference (SAD) is invoked. From each pixel in the block, we subtract the corresponding pixel in the template and take the absolute difference. In the following step, all differences are summed up (aggregated) resulting in a single value which is a measure of similarity characteristic of the two blocks (matching block and template block). In this situation, a lower value is preferred as it is an indication of higher similarity.

Global methods look for disparity  $d$  that minimizes the global energy function. The dynamic programming approach is executed for each scan line (row in a rectified image) independently, where the assumption is that of the ordering constraint between neighbouring pixels of the same row. Sub-pixel interpolation (refinement) represents the fourth step in the reconstruction pipeline. The cost of matching is interpolated with the parabola function, essentially fitting a curve to the cost of matching in an the discrete disparity stage in order to smooth the result). This approach differs considerably from general block matching, and was tailored specifically to ensure a denoised result that is compatible with a depth buffer format and can be ready to use by the rendering API straight away. Block matching relies on a windowed algorithm, which can introduce noisy patches around the window border. Moreover, in the absence of sub-pixel interpolation, contours would be visible at every texture discontinuity (considerable discrepancy between neighbouring intensities). The noise introduced by the classic block matching approach comes from relying on the local optimal cost (optimal disparity per pixel based on own cost function). This is addressed in Algorithm 1 by allowing pixels to have a disparity with sub-optimal cost locally but increasing pixel's agreement with neighbours - constrain the disparity to change a small amount by considering a specific number of neighbouring pixels to be factored in.

Even after refinement, large disparity changes may translate to noise around the scanline boundaries. In order to ensure the result does not contain these, the disparity map undergoes a process of cascaded convolutions, through which a bandpass cascaded map (pyramid) is achieved. This smoothing approach was motivated based on feature enhancement: the noise introduced in the disparity map has to be smoothed out but not at the cost of loosing accurate feature values.)

### 3.2.2 Creation of the counterpart 3D scene projection

In order for new 3D geometry to be composited within a depth/image map pair, two challenges must be overcome:

1. The added geometry must be projected correctly in relation to existing objects contained within the image.
2. The depth map must occlude geometry correctly.

#### 3.2.2.1 Preliminaries

In the present framework, there are a number of assumptions made when it comes to scene property inference as well as the characteristics of the composited geometry. The artefacts to-be composited must come in the form of a common 3D geometry format such as FBX or OBJ, and can be static or animated. Moreover, the objects can present their own material properties through the shape of a set of textures, or can be untextured, in which case, the colour data for each vertex will be read from the mesh file. Regardless of texture presence, the artefacts can have a translucent value defined at the top level or per sub-mesh, and the geometry can be of any type (i.e. not restricted to being watertight). The limitations on artefact type applies to objects which have the potential to become primary or secondary light sources. Secondary light sources are objects with a highly reflective and opaque surface that can bounce the light coming in from the primary light source. Compositing a new light source as an artefact will change/override the illumination inferred from the input and represents a scenario for future work. The compositing process allows for iterative artefact insertion, however, the artefact geometry file can contain many separate meshes.

In terms of content of the initial input, the preferred images are structured (usually indoor), and captured without lenses that have the potential to distort or warp the image. A straightforward example of inputs that cannot be used are macro photography, images of landscapes, images taken with a fish-eye lens. In the absence of structural variation, the depth cannot be inferred accurately, whilst when the image is distorted, the estimation of initial viewing volume and orientation becomes challenging.

In rasterization APIs, the process of correctly orienting and projecting 3D geometry into the scene is the result of performing a number of homogeneous transforms upon the vertices of the geometry to be added. This transformation process generally takes the form of:

$$P_v = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \cdot MVPW \quad (3.6)$$

Where  $P_v$  represents the vector form of a vertex, and  $MVPW$  are homogeneous transformation matrices, with  $M$  representing a matrix that transforms geometry vertices into a common 'global' space,  $V$  representing the inverse of the camera's transformation in global space, and  $P$  the projection transform from 3D to 2D space, setting up the resulting transformed vertex such that the following perspective divide equation will produce vertex positions in the range  $[-1,1]$  for all visible locations:

$$P'_v = \begin{bmatrix} x/w \\ y/w \\ z/w \\ 1 \end{bmatrix} \quad (3.7)$$

For the geometry projected into the depth / colour map pair to produce a seamless result, the matrices  $V$  and  $P$  used to transform and project vertices must be calculated such that they could accurately represent the simulated geometry within the depth / colour map pair; without this, the composition will not result in a plausible image.

The depth calculation outlined in Section 3.2.1 generates a depth map containing linear values; that is, a map entry representing geometry twice as far away as a neighbouring entry will have a value twice that of its neighbour. In order for the depth map and photographic source to be correctly composited within a rasterized scene, it must be converted into values compatible with a rasterization depth buffer, allowing additional rasterized geometry to occlude, and be occluded, as necessary.

### 3.2.3 Creation of the counterpart topology

To solve the issues raised in section 3.2.2, we devise a custom triangulation method from the unordered point cloud set representing the previously computed depth map. Our method does not rely on the Poisson surface reconstruction [80] because the nature of the scene is unknown and non-watertight geometry might be present. Instead, we propose a straight-forward grid-based triangulation method which has two advantages: speed and fidelity for initial topology based on a vertical axis displacement heuristic. This results in an image mesh of  $x * y$  vertices interconnected as triangle primitives, where  $x$  and  $y$



match the input image's width and height, respectively. Each vertex  $v(x, y, z) \in \{R^3\}$ , where  $\{x, y\} \in [-1.0, 1.0]$ , and  $z$  the pixel value taken from the depth map, assumed to be a linear real value  $\in [0.0, 1.0]$ .

### Image to mesh reprojection

To seamlessly insert an object into the original scene, complete with lighting and shadowing as appropriate, the image geometry generated in section 3.2.2 must undergo a process of reprojection, whereby the 3-dimensional details present within the original stereo image pair are transformed such that they are positioned correctly within a 3D Cartesian coordinate system, and negating any foreshortening and perspective effects present within the original images. This allows lighting and shadow calculations to be performed upon the image as if it were any other geometry. The reprojection step is achieved by transforming each vertex of the scene geometry by the inverse of the projection assumed to have been applied to it during image capture. This is a multiple stage process, as while the vertices of the image mesh are assumed to represent geometry that has been captured from a perspective viewpoint, the depth map computation described in section 3.2.1 outputs a normalized linear value; it therefore cannot be used as a depth as-is by rasterization, as traditionally the pipeline writes the nonlinear value  $z/w$  to the depth buffer.

Accurate depth values are obtained using a Z-Buffering algorithm through a process that can be described as follows:

$$z' = \frac{P_F + P_N}{2(P_F - P_N)} + \frac{1}{z} \left( \frac{-P_F P_N}{P_F - P_N} \right) + \frac{1}{2} \quad (3.8)$$

Where  $z$  is the linear distance from the viewpoint origin, and  $P_N$  and  $P_F$  define the near and far clipping planes that bound the viewable region of space, with  $z'$  being the resulting value stored within the depth map. Rasterized geometry is tested against pre-existing values in the depth buffer to determine whether occluding geometry has already been written to the destination pixels that the geometry would cover, with the rasterizer writing or discarding pixels as necessary to produce a consistent image.

To transform the linear values to correct non-linear values, we define matrix  $M_{image}$ , used to transform each of the vertices of the image mesh.  $M_{image}$  is composed of sub matrices  $C_{image}$ ,  $P_{image}$ , and  $V_{image}$ , where  $V_{image}$  and  $P_{image}$  are the assumed view and projection matrices that represent the original camera's position and projection properties in relation to the image, and the correction matrix  $C_{image}$ , used to accommodate shifts in the vanishing point of the image away from its centre. Matrix  $P_{image}$  is constructed as described in the Annex section, with the ratio being that of the image, and the field of vision being that afforded by the original camera setup.

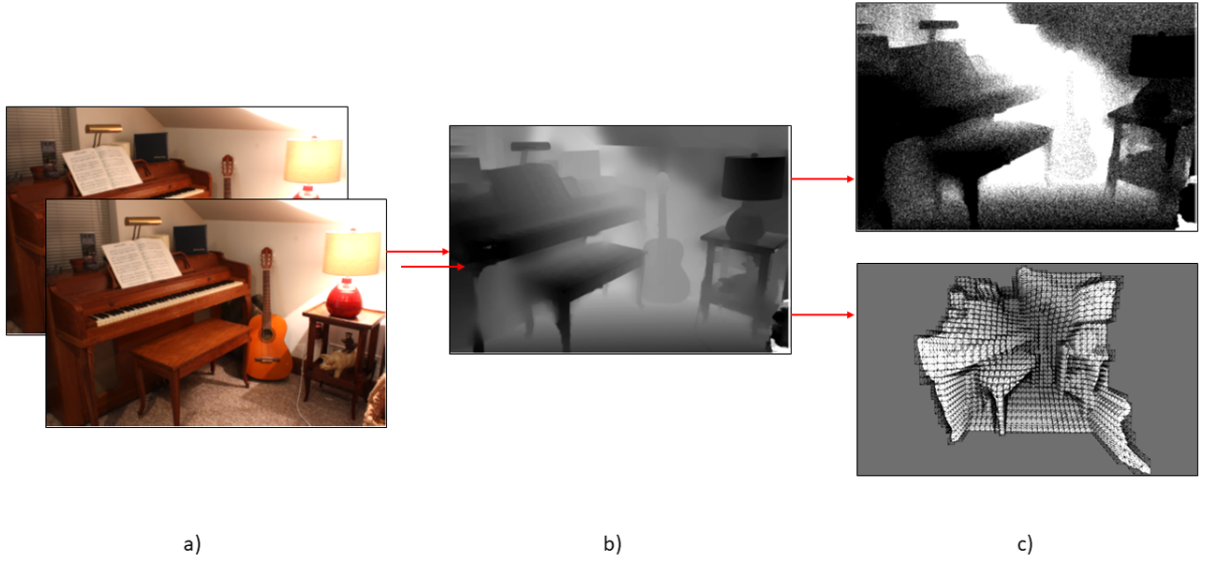


Fig. 3.8 Overview of the reconstructive stage in sequential order. a) Stereo image pair as input. b) Depth map acquired following the DP method with the auxiliary operators for ensuring a denoised outcome. c) Upper image represents the point cloud from depth. The lower image represents the dense reconstruction of the mesh in a wire-frame setup.

The near plane is set to a value of 1.0, with the far plane then being determined as an image size ratio. The far plane can be set empirically by manual adjustment at runtime, in a viewpoint that allows viewing of the projected scene in 3D prior to object addition and final image composition, however, our proposed approach calculates the intrinsic matrix  $K$  to relate between 3D world coordinates to homogenized camera coordinates.

Matrix  $M_{image}$  is first utilized to transform only the  $z$  coordinate, leaving  $x$  and  $y$  intact, which at this point will have values that range between -1.0 and 1.0. Matrix  $M_{scene}$  is now introduced, defined as  $P_{image}^{-1}$ . This, when used to transform the  $x$ ,  $y$ , and  $z$  coordinates of the image mesh vertices, will produce inverse clip space coordinates, which upon a division of the  $w$  coordinate gained during transformation, effectively inverses the traditional projection method utilized in rasterization, distorting the image mesh from a flat plane, to a facsimile of the original geometry within the stereo image pair; and undoing the convergence of parallel lines to a vanishing point.

To allow for the resulting image mesh vertices to be utilized in a rasterization pipeline, they are transformed from their original values via the transform feedback API<sup>1</sup>, allowing efficient computation of the necessary reprojected vertex coordinates on the GPU device. After this projection stage, a triangular mesh representing the geometry of the original image pair can be placed within a 3D scene, where it can interact with any other added 3D meshes that the user wishes to insert into the original images.

<sup>1</sup> In the OpenGL API, transform feedback represents the process through which the primitive or geometric elements are stored into a buffer after the vertex shader program has run, essentially allowing the data to be preserved post-rendering.

### 3.3 Shadow mapping and relighting

Compositing software rely on a sequential state transition - from input data, to scene acquisition and scene properties' parameterization (illumination, surface reflectivity, field of view), followed by the composition step (user guided placement and scaling), and lastly reprojection of the result (from the 3D scene, back to a plausible 2D image). The composition step is split into two phases - the initial user customization of the artefact's position in the 2D image, and the aggregation of the artefact according to the parameterization carried from the scene description.

The lighting step relies on a hybrid system that offers a local solution to illumination modelling if direct light sources are present in the image. The HDR nature of the input set allows the retrieval of a spherical map that can be used to model radiance and irradiance as an ambient term. If HDR images are not available, the system matches the filtered spherical map to a similar map from an open-source IBL dataset. Even in the absence of directly visible primary light sources, the spherical map will inform the diffuse ambient term. Studies discussed in the Background Chapter state that compositing pipelines are most valuable when the user is allowed to customise the results of an automated step, and the present pipeline follows this direction by allowing the user to modify the light source position, orientation, and intensity with the limitation to one light type, namely a directional light.

The notable contributions of this step consist of a fast solution to the relighting step post compositing (i.e. the implementation follows a real-time rasterization approach), as well as the ability to compute physically accurate shadows during the relighting step. The latter approach addresses the issue of multiple object compositing which the relevant related work does not tackle.

#### 3.3.1 Illumination model

In order to confer consistent ambient lighting to the newly composited artefacts, our framework relies on an environment map sampling method, where each pixel represents an incoming light source. In the previous section we have defined the reflectance equation (Equation 3.9), which we will be using in order to retrieve a scene's radiance and store it in the environment map. Since computing this in real-time may be slow or require newer hardware on mobile devices, the reflectance integral in this case will be pre-computed.

$$L_o(p, \omega_o) = \int_{\Omega} \left( k_d \frac{c}{\pi} + k_s \frac{DFG}{4(\omega_o \cdot n)(\omega_i \cdot n)} \right) L_i(p, \omega_i) n \cdot \omega_i d\omega_i \quad (3.9)$$

One observation is that the diffuse and specular term can be calculated independently in

two separate integrals. Since further in the framework the shadows require light source emitters, the specular integral calculation will be deferred until after the artefacts are composited, while the environment map will consist only of the diffuse term.

$$C_d = k_d \frac{c}{\pi} \quad (3.10)$$

$$F_{diffuseIrradiance} = C_d \int_{\Omega} L_i(p, \omega_i) n \cdot \omega_i d\omega_i \quad (3.11)$$

The term defined by Equation (3.10) is the *lambertian* constant defined by a colour  $c$ , and the refraction factor (derived from the Fresnel term)  $k_d$ . Since all the participating terms are constants, the diffuse irradiance part of the reflectance function can be simplified further as illustrated in equation 3.11. The assumption made in this work is that  $p$  represents the centre of the environment map.

### Pre-processing the radiance map

Physically-based rendering operates with real-world values, therefore, the images used to create the environment map have to be in a HDR format. When an HDR format cannot be supplied, the framework template matches for the pre-computed irradiance map with highest similarity to the LDR image. The HDR format for environment maps allows graphics APIs to specify colour values beyond the standard  $[0.0, 1.0]$  range. This is important for storing real-world light intensity and colour information.

Once the framework computes or receives the HDR image, it converts it to a suitable environment map inside a specialized vertex and shader program. The shader program achieves this by creating a unit cube and projecting the HDR image (wrapping it) onto each face of the inside of the cube. Each face of this unit cube is rendered not to screen, but to graphic pipeline's frame buffer object (FBO) attachments - a technique used for both direct and indirect drawing.

For estimating the 3D illumination model from the 2D images, the diffuse indirect light term has to be calculated relying on the environment map (representing diffuse irradiance - see Figure 3.9). This means that ideally, the scene's radiance has to be sampled across all directions  $w_i$  within the surface  $\Omega$ . That would be computationally intensive, however, the diffuse irradiance integral can be solved discretely by retrieving irradiance from  $\Omega$  oriented around the normal. Each pixel value is calculated using the surface's normal orientation ( $N$ ), and all these values are convoluted to form a cubemap. This is analogous

to computing the average radiance distribution for each incoming direction  $\omega_i$  under the hemisphere. This method requires two shader passes and can be computationally expensive.

In order to approximate the integral, the shader takes a fixed amount of sample vectors per texel inside  $\Omega$  and average the result. Furthermore, instead of directly relying on the solid angle  $d\omega$ , the values integrated over are its corresponding spherical coordinates (a polar azimuth angle  $\in [0, 2\pi]$ ) and an inclination zenith angle  $\in [0, \frac{1}{2}\pi]$ . The diffuse irradiance term can discretely be expressed as a sum, given samples  $v_1$  and  $v_2$  per spherical coordinate:

$$v_o = \{\phi_o, \theta_o\}, v_i = \{\phi_i, \theta_i\} \quad (3.12)$$

$$L_o(p, v_o) = C_d \frac{1}{v_1, v_2} \sum_{\phi=0}^{v_1} \sum_{\theta=0}^{v_2} L_i(p, v_i, \theta_i) \cos(\theta) \sin(\theta) d\phi d\theta \quad (3.13)$$

The sampling region projected onto the hemisphere's surface decreases proportionally to the elevation value of the zenith angle (during their convergence towards the pole). This issue can be addressed by scaling the smaller area by  $s \sin \theta$  (zenith angle).

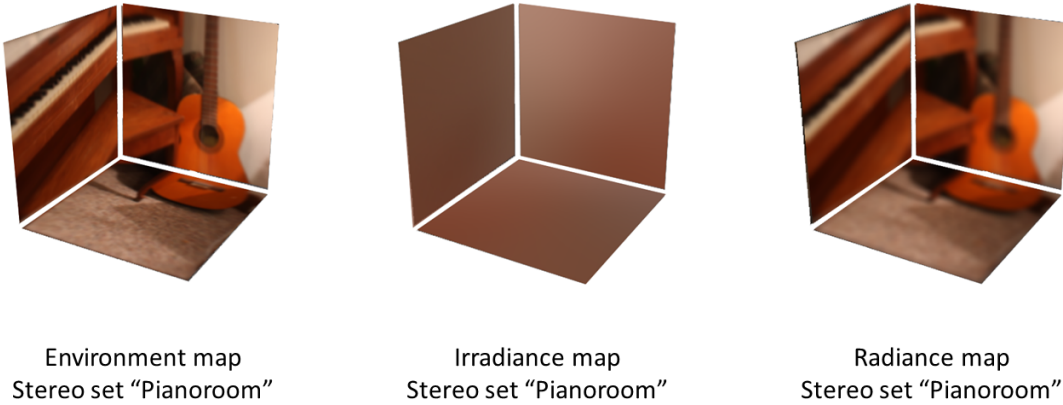


Fig. 3.9 Left-most image is an illustrative example of the environment map from the stereo input set. The centre image is a capture of our irradiance map formed by each texel originating in the convoluted version of the environment map sampled from outgoing light directions  $\omega_o$ . The right-most image represents the radiance map with one level of roughness.

As seen in Figure 3.10 and in the additional results section C.1, the appearance of gloss or dullness is dictated by the interaction of the indirect light and the surface properties - the higher the roughness factor, the more light will be absorbed, toning down the crisp

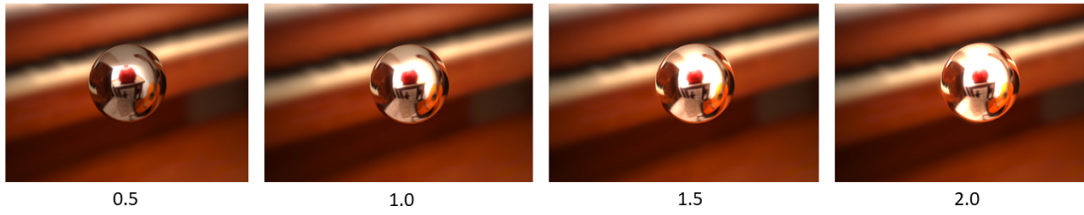


Fig. 3.10 Indirect lighting levels

reflections. In order to handle the lack of a roughness map factor, the framework simulates it based on the angle formed by the view vector and the normal to the surface.

---

**Algorithm 3** Roughness approximation
 

---

```

1:  $\cos\Theta \leftarrow (\max(\cos(\text{normal}, \text{view}), 0.0))$ 
2:  $\text{specular} \leftarrow \cos\Theta * (1.0 - \cos\Theta)^5$ 
3:  $\text{diffuse} \leftarrow \text{irradianceMapSampler.inRGB}$ 
4:  $\text{ambient} \leftarrow (1.0 - \text{specular}) * \text{diffuse}$ 

```

---

### 3.3.2 Relighting step

All compositing pipelines, during the aggregation stage, superimpose an area soft shadow at the contact boundary between the artefact and the scene mesh. This approach is the common accepted practice due to creating perceptually plausible results in a timely manner. However, a number of challenging issues arise with this aggregation approach due to the lack of physical accuracy when calculating the shadow coverage and intensity:

1. When compositing a single object in a scene, it is not immediately obvious that the superimposed shadow is not physically accurate. When a large number of objects have to be composited, the shadows may overlap or stretch according to both previous and current topological state (Fig. 3.11).
2. In numerous real-world cases, there is more than one point light source in a scene. This means that potentially for every object present there will be up to  $n$  shadows cast, where  $n$  represents the number of lights present in the input data. With the superimposed shadow approach, in a scene where the light sources are clearly identifiable, a single shadow will no longer create a plausible result (Fig. 3.12).
3. From 2. stems another scenario where the composited objects are added in independent passes and therefore the shadows do not influence one-another. This is due to both the aggregation method and the sequential state of the pipeline (Fig. 3.13).

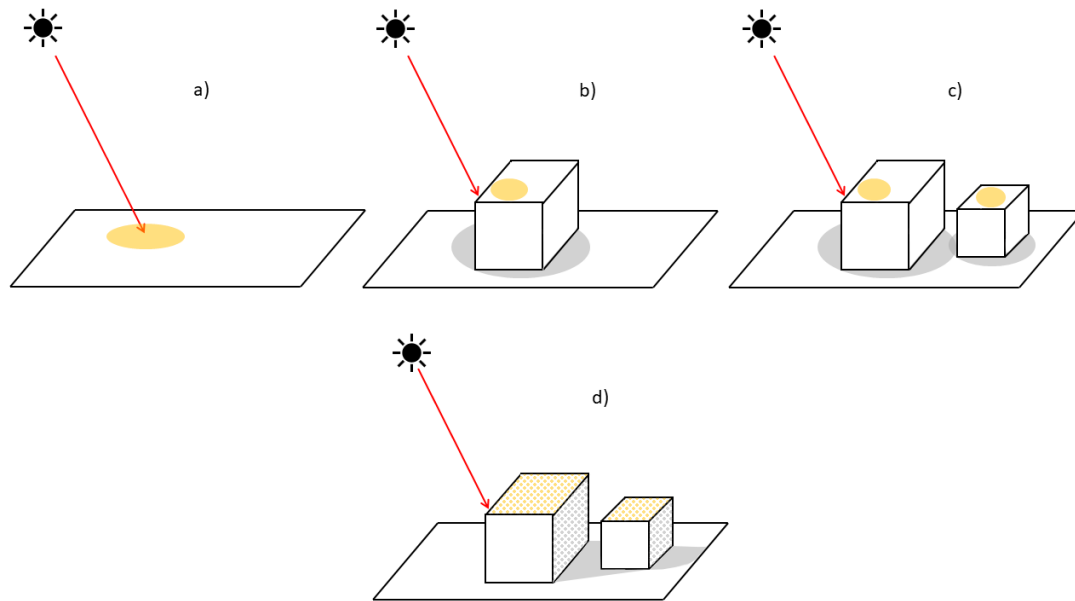


Fig. 3.11 Scenario illustrating inconsistencies caused by non-physically accurate shadows. a) Initial scene with a directional light. b) Composited artefact (cube) with non-physically accurate shadow, which can still look plausible. c) Composited two artefacts with non-physically accurate shadow, which can now render an inconsistent result. d) Our solution

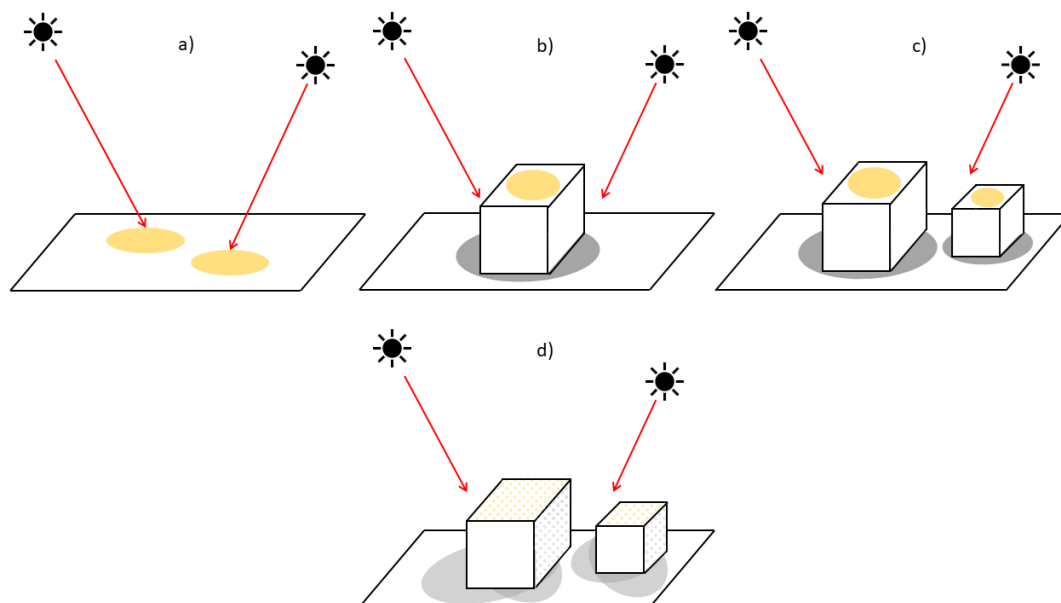


Fig. 3.12 Scenario illustrating inconsistencies in the presence of multiple light sources. a) Initial scene with two directional lights. b) Composited one artefact; result could still be plausible. c) Inconsistency upon compositing multiple artefacts. d) Our solution

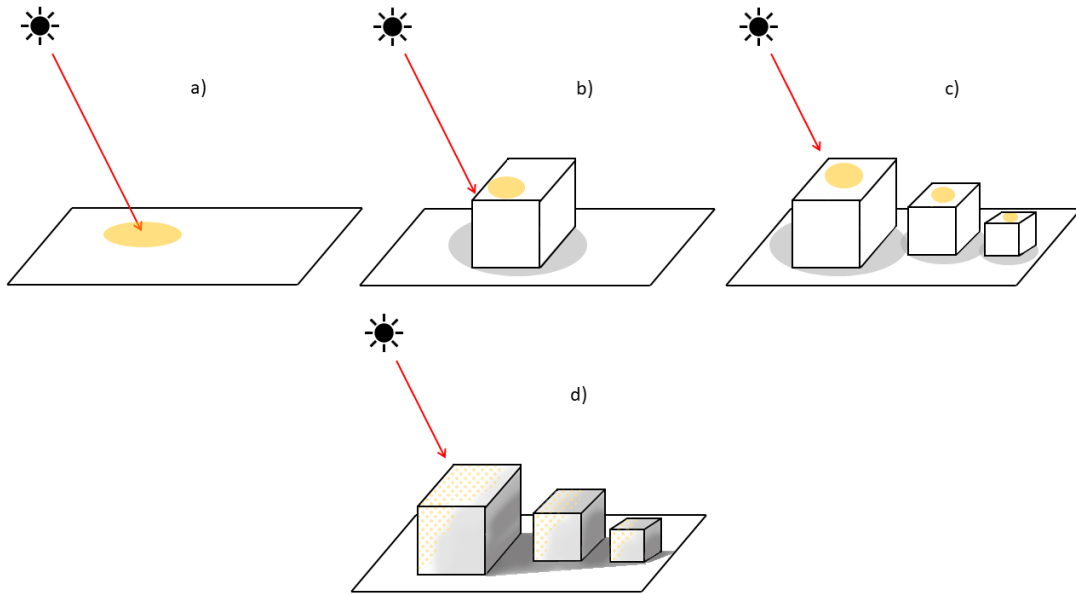


Fig. 3.13 Scenario illustrating inconsistencies caused by sequential state transition. a) Initial scene with one directional light. b) Scene with composited artefact; the result could be plausible. c) Scene with numerous composited artefacts where cast shadows are inconsistent due to the linear state. d) Our solution.

The contribution described in this section addresses the challenges enumerated above by modifying the sequential state transition paradigm in order to achieve physically-accurate shadows. Upon reaching the first phase of the compositing stage (user-defined artefacts are introduced at a specific location), the second phase (aggregation) will solve an inverse rendering problem where the geometry and illumination terms are known. During aggregation, intermediate states (partial solutions) of the scene will be stored and compared (state transition can go backwards), ensuring that there is a correlation between the number and/or type of illumination sources, the total number of artefacts composited, and their respective cast shadow.

### 3.3.3 Overview of the shadow casting process

In the context of the present work, a shadow is defined as an area on a surface for which at least a part of a light source is occluded [57]. This definition implies that only primary light sources (casters) are considered in the occlusion test. Secondary sources of illumination (light bouncing due to a high factor of reflectivity on a smooth surface) are treated as a special case. Another implication that stems from the definition is the nature of the occluder - in this scenario the occluder is assumed to be opaque, and translucent occluders are treated as a separate case. Furthermore, subsurface scattering is not accounted for in the automated pipeline flow, however, user guidance is required to enable that capability.



In order to formalize the problem, the work assumes that the surface geometry (denoted  $S$ ) is a set of connected points ( $p$ ) that form triangle primitives, each with its own facet normal ( $n_p$ ). The primary light sources ( $L$ ) can be generalized as encompassing an area defined by a set of points ( $l$ ) on the light's surface. Light exiting points in sample set  $l$  has directionality (treated as emitted radiance). The problem of shadow casting in this setup relies on a binary check to test whether light can travel from a point on the surface ( $p_S$ ) to another ( $q_S$ ), in a straight line, without obstructions, as illustrated in Equation 3.14. The expression defines  $p_S$  and  $q_S$  as two points on a line segment (defined over surface geometry  $S$ ) consisting of all points  $r_S$  located between  $p_S$  and  $q_S$ .

$$(p_S, q_S) := \{r_S | r_S := p_S + \alpha(q_S - p_S), 0 < \alpha < 1\} \quad (3.14)$$

The binary shadow check can then go through each light sample from the set  $l$  and each point  $p_S$  and verify whether  $l$  is visible to  $p_S$ . If it is not visible, then we can conclude that  $p_S$  belongs to a shadow region cast as a result of an occluder in  $l$ 's path. In a variety of cases, however, a light source is not fully occluded. This can be accounted for by splitting the shadow test into an *umbra* region and a *penumbra* region (Eq. 3.15 and Fig. 3.14).

$$O_L(p) = \{l \in L | p \cap l = \{\emptyset\}\} \quad (3.15)$$

$O_L(p)$	set of $\forall$ light samples $l \in L$ not seen by $p_S$
$O_L(p) = L$	the whole light ( $L$ ) is obstructed fully $\Rightarrow p$ is in <i>umbra</i>
$L \neq O_L(p) \neq \emptyset$	some amount of light is obstructed $\Rightarrow p$ is in <i>penumbra</i>
$O_L(p) = \emptyset$	$p$ is fully lit.

### Encapsulating hard shadows and soft shadows in the Rendering Equation

The lighting formulation described in the previous sections did not involve a shadow term to define an area inside the hemisphere or viewing volume (frustum). As a reminder, the work relies on the following expression for representing illumination in a real-time setup:

$$L_o(p_S, \omega) = L_e(p_S, \omega) + \int_{\Omega_+} f_r(p_S, \omega, \dot{\omega}) L_i(p_S, \dot{\omega}) \cos(n_p, \dot{\omega}) d\dot{\omega} \quad (3.16)$$

The expression 3.16 has been redefined to account for the shadow area - the outgoing

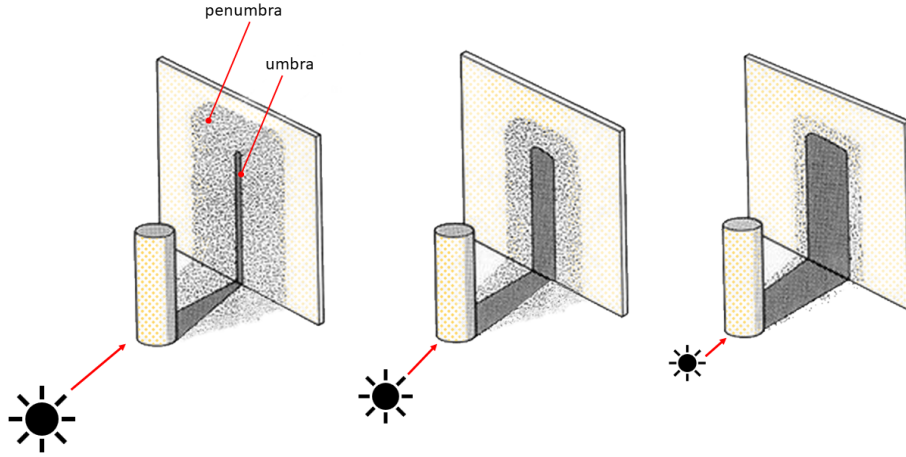


Fig. 3.14 Three cases illustrating the relation between the distance from light source to occluder and how the spread of umbra and penumbra vary accordingly.

radiance  $L_o$  is a function of position  $p_S$  and direction  $\omega$ , where  $p_S \in$  scene geometry  $S$ . The terms for emitted radiance  $L_e$ , and incoming radiance  $L_i$  remain unchanged. The surface integral is specifically defined in terms of the surface normal  $n_p$  at point  $p_S$ , where the hemisphere ( $\Omega_+$ ) is above point  $p_S$ .

We consider two points  $p_S, q_S \in$  surface geometry  $S$ . The following expression holds when  $p_S$  has visibility of  $q_S$  without obstructions:

$$p_S \rightarrow q_S := \frac{q_S - p_S}{\|q_S - p_S\|} \quad (3.17)$$

In terms of radiance, the Equation 3.17 implies that  $L_o$  from one side of the segment defined by  $p_S$  and  $q_S$  is equal to  $L_i$  from the other side (Equation 3.18).

$$L_o(q_S, q_S \rightarrow p_S) = L_i(p_S, p_S \rightarrow q_S) \quad (3.18)$$

Using the observation described in Equation 3.18, the equilibrium of energy in the scene (Equation 3.16) can be reinterpreted using  $L_o$  as substitute for  $L_i$  as follows:

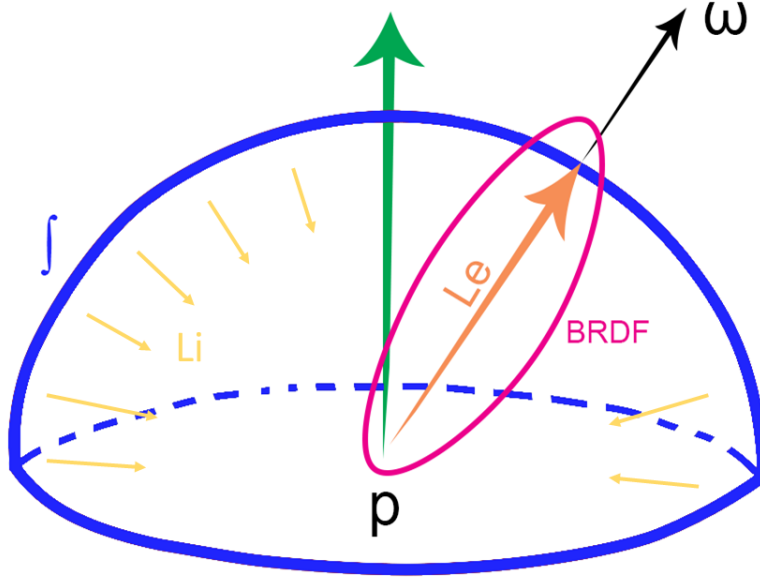


Fig. 3.15 The light towards the viewer  $L_o$  from a point  $p_s$  is equal to the sum of the emitted light  $L_e$  from that point and the integral over surface geometry  $S$  within the unit hemisphere of the incident light  $L_i$  coming from any direction multiplied by the probability of such light rays bouncing towards the viewer ( $f_r$ ) in relation the the irradiance factor over the normal  $p_n$ .

$$L_o(p_s, \omega) = L_e(p, \omega) + \int_S f_r(p_s, \omega, p_s \rightarrow q_s) L_o(q_s, q_s \rightarrow p_s) G(p_s, q_s) V(p_s, q_s) dq_s \quad (3.19)$$

$$G(p_s, q_s) = \frac{\cos(n_p, p_s \rightarrow q_s) \cos(n_q, q_s \rightarrow p_s)}{\|p_s - q_s\|^2} \quad (3.20)$$

$$V(p_s, q_s) = \begin{cases} 1, & p_s \cap q_s = \emptyset \\ 0, & p_s \cap q_s \neq \emptyset \end{cases} \quad (3.21)$$

As the work only takes into account primary light sources, and treats subsurface scattering or bouncing light as a separate specialized case, the term  $L_e$  can be added at the end as a constant, simplifying the expression further for the direct lighting:

$$L_o(p_S, \omega) = \int_L f_r(p_S, \omega, p_S \rightarrow l) L_e(l, l \rightarrow p_S) G(p_S, l) V(p_S, l) dl, l \in L \quad (3.22)$$

When the distance from the light source to the receiver is significant (with respect to the light's solid angle), then the geometric term  $G$  from the reflectance function  $f_r$  does not vary. When the reflectance function is assumed to be diffuse, the Equation 3.22 can be split into a product of  $G$  and  $L_e$ :

$$L_o(p_S, \omega) = \int_L f_r(p_S, \omega, p_S \rightarrow l) G(p_S, l) dl \cdot \frac{1}{|L|} \int_L L_e(l, l \rightarrow p_S) V(p_S, l) dl \quad (3.23)$$

Equation 3.23 decouples shading (the first term of the product) from the shadow region (the second term of the product). For real-time applications, the emitted radiance can be simplified to a function of position if the light source is assumed to have homogeneous directional radiation over the surface. In addition to this, most light sources are assumed to have a uniform colour, which means that the emitted radiance can be treated as a constant, denoted  $L_c$ .

$$L_o(p_S, \omega) = \text{shadingIntegral}(p_S, \omega, L, L_c) \cdot \text{visibilityIntegral}(p_S) \quad (3.24)$$

$$\text{visibilityIntegral} = V_L(p_S) = \frac{1}{|L|} \int_L V(p_S, l) dl \quad (3.25)$$

$$\text{shadingIntegral}(p_S, \omega, L, L_c) = L_c \int_L f_r(p_S, \omega, p_S \rightarrow l) G(p_S, l) dl \quad (3.26)$$

Equations 3.25 and 3.26 first calculate the direct illumination (region outside of shadow), then the soft shadow. Most compositing pipelines rely on the decoupled version of the shadow region computation (Equation 3.25). The issue introduced by this approach concerns the geometry term  $G$  - only the amount of visibility is computed and not the physical locality of which part is blocked.

The present work relies on the coupled equation for calculating both soft shadow and the location of the blocker through  $G(p_S, l)$  which influences the light source on the point  $p_S$  and accounts for the fall and orientation of the shadow region.

### 3.3.4 Initial lighting setup

The unordered point cloud does not provide enough information regarding scene-illumination interaction. Thus, our method reinterprets the cloud as a multi layered mesh which will offer sufficient surface connectivity in the context of object compositing.

In order to provide the appearance of light and shadow interactions on the surface of the new mesh geometry, it must be rendered using an appropriate lighting model. While the image mesh counterpart is already assumed to have the correct lighting contribution from its source image, it must also be rendered taking into consideration the shadows cast across it by the inserted object. We utilize a modified Blinn-Phong lighting model on each rendered fragment of the inserted object, defined as follows:

$$I = k_a + \sum_n k_d (L_n \cdot N + c_s k_s (R_n \cdot V)^\alpha c_f) \quad (3.27)$$

Where  $n$  is the total number of light sources considered,  $I$  is the incident vector,  $N$  the fragment normal, and  $V$  the view direction. The constants  $k$  denote the specular reflection ( $k_s$ ), diffuse reflection ( $k_d$ ), and diffuse ambient ( $k_a$ ) colours, while  $\alpha$  denotes the specular power. This model relies on a degree of energy conservation for a more accurate specular lighting contribution, via the approximation factor  $c_s$ , defined as:

$$c_s = (\alpha + 8)/8\pi \quad (3.28)$$

To further increase specular accuracy, the Fresnel effect included in the equation using approximation factor  $f$  is defined as:

$$c_{base} = (1 - (R_n \cdot V)^\beta), \beta = 5 \quad (3.29)$$

$$c_f = (c_{base} + c_0)(1 - c_{base}) \cong 0.018 \quad (3.30)$$

When combined with appropriate texture, bump, and specular maps applied to the mesh, a plausible simulation of lighting upon the inserted object can be calculated.

In order to determine positions within the image from which to calculate lighting contri-

butions, the original source images are processed to identify pixels of sufficient brightness as to be potential sources of illumination for the inserted object. These pixels are then coalesced into a number of light volumes via a blob detection technique, each defined by the mean position and color of the pixels, and volume radius. In cases where no light areas are detected within the image, a single light volume at the top of the image is instead assumed to be present. The resulting illumination from each of the light volumes onto the inserted object can then be calculated using the lighting model described above.

### 3.3.5 Relighting during the composition step

At this stage, the framework has a dense representation of the scene in 3D space along with light source volumes placed consistently with the initial scene setup. In addition to this, the view frustum derived from  $vImage$  and  $pImage$  has a good estimate for the depth of field, aspect ratio, pitch, and yaw. This is sufficient data for compositing an object while also allowing the object to participate in the relighting process.

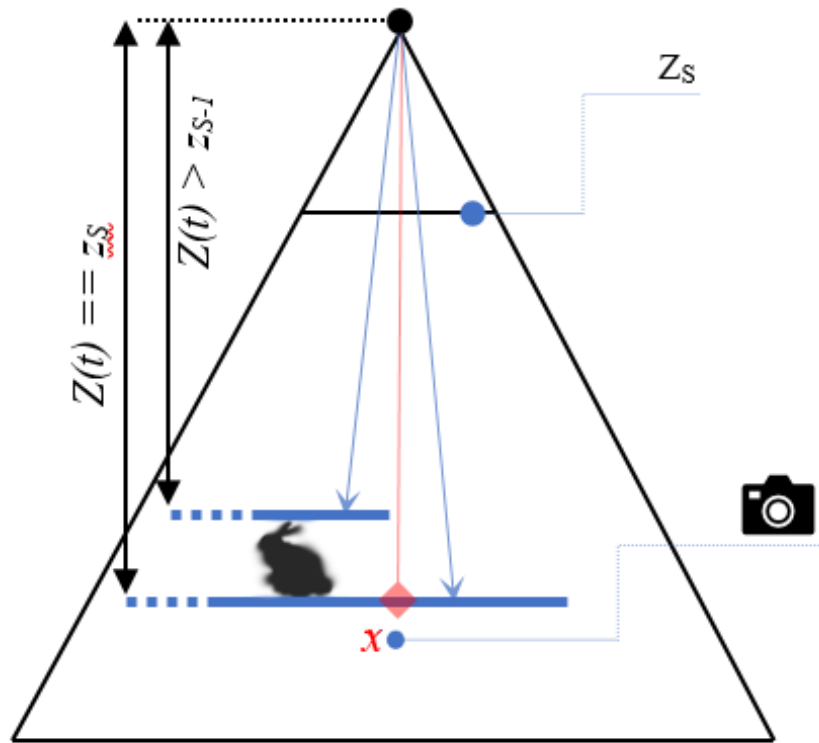


Fig. 3.16 Exponential shadow mapping. The two blue horizontal lines illustrate the delimitation of a correct shadow filter, where the initial assumption regarding the distance between occluder depth and light source hold true.

Often in rasterized lighting a straight-forward shadow mapping technique will be implied, however, such an approach will not be suitable for object compositing due to the hard nature of the resulting shadow, and the disadvantage of resampling errors that come with

this technique. In contrast, our work focuses on implementing a layered exponential soft shadow algorithm which is cost-efficient, not prone to aliasing errors, and resembles the result of physically based soft shadows at lower computational cost.

Most statistical approaches to shadow-mapping techniques involve filtering, namely a blurring kernel. Our implementation follows this standard. Assume  $t = (u, v)$  to be the coordinate within the shadow-map after projecting a world point  $p$  into it, and its characteristic depth space  $z_s$ . We can define a shadow-mapping function  $f$  as follows:

$$f_s(t, z_s) = H(z(t) - z_s) \quad (3.31)$$

Where  $H(x)$  represents a shadow comparison function with the depth value at  $z(t)$  sampled from the shadow map as seen in (5):

$$H(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \quad (3.32)$$

Exponential shadow mapping builds on this initial formulation, and solves the probability of an incoming depth  $z_s$  being in front or behind values already stored in the shadow-map:

$$P(z_s \leq z(t_{filter})) \quad (3.33)$$

The probability result is then used as a light calculation scaling factor which characterizes the final shadow fragment. In the present framework, the depth-map automatically discards fragments with a  $z$  (depth) value smaller to the values already stored.

$$z - z_s \geq 0 \rightarrow H_s(z - z_s) = \exp(-c(z - z_s)) \quad (3.34)$$

When the filtering step is added to the shadow projected value as described in (8), the result will take the form:

$$f_{S_{filter}}(t, z_s) = \exp(cz_s) \sum_{t_i \in K_{filter}} k(t_i, t) \exp(-cs(t_i)) \quad (3.35)$$

Classic exponential shadow mapping suffers from a drawback whereby due to the shadow value increasing towards 1 as equation (8) reaches 0 in order to produce a fully lit

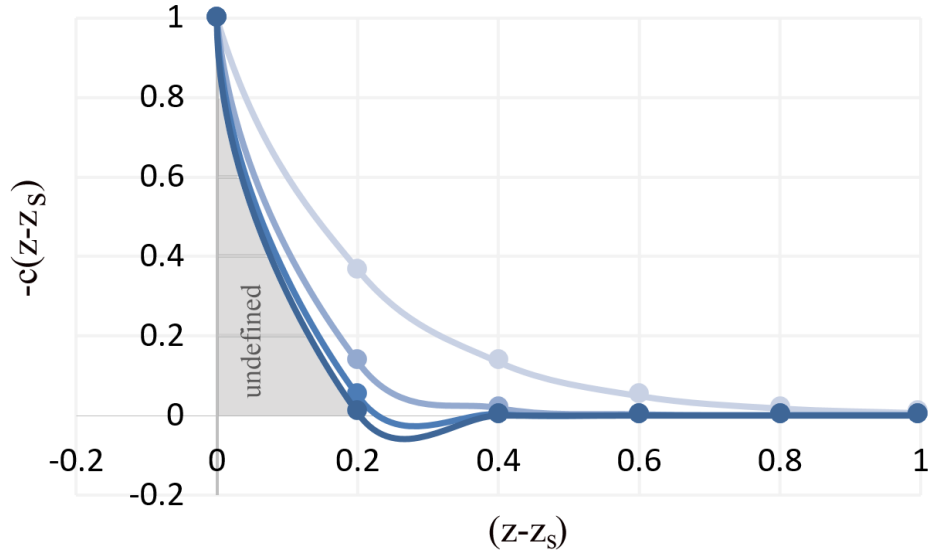


Fig. 3.17 Exponential shadow map function. The horizontal axis represents the test function between the distance from the light source to a point on the surface, and the distance between the light source and the depth of the occluder. The vertical axis represents the exponential function, where a larger  $c$  value increases the accuracy of the approximation.

fragment in cases where a fragment is unoccluded within the shadow map, the shadows umbra appears to fade towards the occluding object. This results in a loss of a contact shadow on the scene geometry, producing an inaccurate image that is missing a key feature of plausibility to the inserted object. Contact shadows are usually obtained by increasing the value of  $c$  until the change over distance is no longer perceptually visible. We instead modify equation (11) as follows:

$$s_{orig} = \exp(-c * diff) \quad (3.36)$$

$$s_{clamp} = \begin{cases} 1, & s_{orig} > 1 \\ s_{orig}, & 0 < s_{orig} < 1 \\ 0, & s_{orig} < 0 \end{cases} \quad (3.37)$$

$$z - z_S \geq 0 \rightarrow H_s(z - z_S) = \begin{cases} 1 - s_{clamp}, & s_{clamp} < 1 \\ s_{clamp}, & s_{clamp} > 1 \end{cases} \quad (3.38)$$



This modification inverts the amount of shadow calculated within the region  $[0, 1)$ , instead allowing the value to increase as distance increases, providing a shadow that fades away at the extents of the projected shadow, which when combined with equation (8) affords a softer shadow umbra at a smaller filtering kernel size. This allows the lights generated from the source image to utilize a real-time soft shadow model. As observed in Fig. 3, the exponential approximation function is left-bounded, which means that when the test function fails, the exponential approximation will yield values that are incompatible with the filtering kernel.

### 3.3.6 Geometry reprojection

At this stage, it is possible to render a single 3D scene with consistent lighting and shadowing that includes both the original photograph geometry and newly inserted object geometry, using the reprojected image mesh to provide correct depth and light occlusion on the inserted geometry. Once the object(s) to be inserted into the scene have been placed and oriented correctly, a consistent final image can be constructed. However, due to minor aberrations in the calculation of the linear image depth, and in the approximation of a suitable matrix  $P_{image}$  as in section 3.3, an image rendered at a point approximating that of the original camera may produce error, as the projected vertices may not be projected exactly to their original image coordinates. To accommodate this, the final step of producing a manipulated image, projects the vertices of added object(s) using matrix  $M_{image}$ , which when combined with the perspective divide, produces a projection equivalent to that contained within the original image.

In this step, the original, color image is rendered to screen via a full-screen quad, and using a shader which is aware of the projected vertices produced by section 3.3., which are used to calculate a vector from which to derive a depth value suitable for writing to the depth buffer, and for passing to the fragment shader for processing of lighting and shadowing, with the actual quad vertices projected via an orthographic matrix directly to the screen. The additional inserted object(s) are then rendered in similar manner, calculating depth and world position values for fragment shading using matrix  $M_{scene}$ , allowing the object(s) to be rendered using depth testing techniques to occlude correctly. In order for the inserted object(s) to maintain consistency with the geometry within the original image, it is then projected using matrix  $M_{image}$ , while maintaining the z value derived from  $M_{scene}$  to ensure correct depth testing. This method allows for lighting and shadowing calculations to be performed in a linear space common to all rendered objects, while still affording the ability to seamlessly project new geometry into the original image, with the image appearing to occlude newly inserted geometry that has been added logically 'behind' objects in the image from the viewpoint contained within the original image.

# Chapter 4

## Results and evaluation

### 4.1 Evaluation overview

In order to evaluate our approach, the quality and efficiency of the pipeline’s capabilities, three types of tests will be taken into consideration:

**Based on standard stereo images.** In this test, we will be relying on stereo image pairs originating from established datasets, as well as on stereo images taken with a regular mobile device. Due to the wide range of visual contexts available in these datasets, this type of test is ideal for showcasing how the techniques deal with cluttered scenes and scenes presenting different types of lights, materials and geometry.

**Based on synthetic scenes.** Using a commercial 3D engine we will be designing a number of box-like indoor scenes and generate the ground truth depth in addition to a capture of two cameras that represent the left and right eye in traditional binocular vision. We use these captures to then run our approach for depth reconstruction. This test will illustrate the accuracy of our technique. Moreover, this type of test enables us to have control over the light sources and verify the correctness of the shadows cast.

**Stage-by-stage comparison.** Lastly, we can obtain intermediate results of each pipeline stage and compare it to similar techniques in terms of quality and, where possible, even performance. Through this approach, we hope to make clear the differences between our work and the previously established frameworks, as well as underline the overall capabilities and limitations.

We create a dataset of scenes which vary across structure, number and types of light sources present, and visual clutter (high frequency of colour changes, complexity of surface shapes present in the initial image, etc.). The first scenes are part of the Middlebury 2014 [132] stereo dataset, while the last scenes are artificial box-like representations developed in the Unity 3D engine due to its capability to enable the visualisation and exporting of a depth buffer from its own rendering pipeline, an output suitable for comparison against our own depth reconstruction. We have deliberately chosen synthetic scenes in order to

compare the quality of our depth-map acquisition stage along with the relighting step to a correct and consistent depth map. The last scene was acquired in an ad-hoc setting – the input image pair is near-stereo as it was acquired via a commercial mid-range mobile phone without prior measurements of the scene or alignment (purposely chosen to demonstrate sensitivity to less-than-ideal stereo pairs and lighting conditions).

The scenes obtained from the dataset are visually complex and are predominantly indoor, cluttered scenes in comparison to the fabricated scenes which contain geometric shapes. The latter were specifically made for observing compositing between objects, and this method allowed us full control over the scene parameters. The synthetic scenes constructed in Unity make use of 3 cameras (left, right, and main) in order to represent the left and right input in a stereo pair. The main camera is half-way between the other two and serves the purpose of ground truth check. Because this approach is modelled from a binocular vision system, we have used a distance of 5.80 units (equivalent to 58mm) between left and right cameras to mimic the correct minimum average interpupillary distance.

In the present system, the user is expected to input the initial stereo image pair, a 3D mesh, and to participate in positioning (scaling, rotating, translating) of the desired synthetic object at the penultimate stage of the pipeline. For further accuracy, the user may also choose light source positions and directions, however, this step is optional, as these are approximated by our pipeline as covered in section 3.3.

#### 4.1.1 Dataset criteria

The Middlebury dataset was chosen to provide stereo input as it is the most well-known and documented open source dataset in the field of Machine Vision. It contains ideal images of indoor scenes, however, only a single image which includes a primary light source that illuminates the scene. From this dataset we have chosen five distinct images that have the following features: a light source that is visible and actively illuminates the scene, medium density when it comes to clutter, varying and uniform background, high density clutter, and low density clutter. Each of these scenarios mimics real-world situations and, as a whole, these scenarios should cover most cases: cluttered rooms where the shadows have to project on more than one surface, illumination that influences the shadow orientation, clutter which allows for compositing in-between existing scene geometry.

The synthetic tests comprise of three scenarios aimed towards looking at compositing with accurate shadow in the presence of one light source, many light sources, and light sources with medium cluster. In the synthetic scenes, there is complete control over each parameter ranging from initial camera locations to capture the stereo images, to adjusting the size and direction of the lights.

The near-stereo dataset consists of three scenarios aimed at reconstruction in the presence of more challenging features such as large reflective areas, areas with high density of clutter and detail.

## 4.2 Evaluation of the scene acquisition consistency

Initially, the user provides a valid stereo image pair which will be processed for the acquisition of a coarse disparity map. This intermediate result will be filtered locally and normalized at the very end in order to obtain a smooth depth-map suitable for becoming a depth buffer attachment in our OpenGL-based pipeline.

In both sets of test data (based on real-world stereo captions and based on the synthetic scenes created in an open-source rendering engine) the scene acquisition stage of the pipeline recovers an estimate of the baseline which is then used to construct the initial camera position in the 3D world. When timing the application during performance analysis, the baseline calculation is coupled with the reconstruction of the scene. The reconstruction step encompasses the input acquisition, generating the depth map compatible with hardware depth (catered for the OpenGL API), and the mesh connectivity.

### 4.2.1 Evaluation based on a real-world stereo dataset

The set of input images, illustrated in Figure 4.1, were deliberately selected as they have different sizes, and present both visual complexity (a large number of objects of various shapes, materials, and sizes, partially obstructing scene geometry). The run-time complexity is dictated primarily by the input size (width x height), and secondarily by the size and contents of the confidence map.

Table 4.1 Performance overview of the reconstruction stage for the stereo dataset. The mesh size is represented by the total number of polygons after dense reconstruction.

Scene	Width (px)	Height (px)	Matching Time (s)	Filtering Time (s)	Mesh Size
Middlebury Piano	2820	1920	5.2	2.0	1,804,800
Middlebury Playroom	2800	1908	6.0	2.1	1,780,800
Middlebury Cable	2796	1984	10.7	0.9	1,825,279
Middlebury Adirondak	2880	1988	9.0	1.0	1,908,479
Middlebury Plant	710	496	0.25	0.05	117,385

An ample discussion about these results is covered in Section 4.4, and from a brief overview of Figure 4.2, the minimum entry point is represented by the Middlesbury Plants dataset - the depth map recovery and filtering took the shortest amount of time

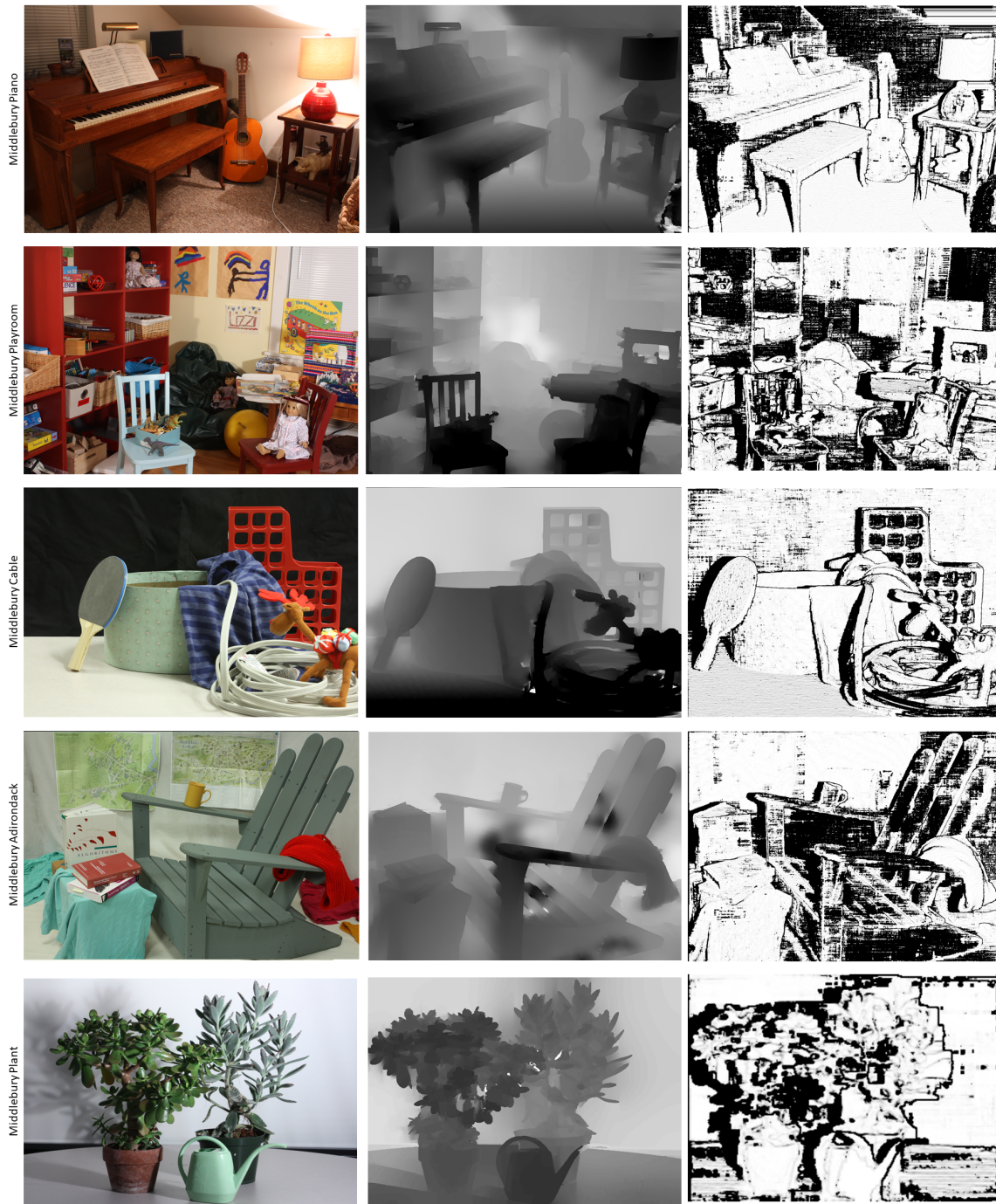


Fig. 4.1 Evaluating scene consistency after reconstruction based on the stereo Middlebury dataset. The left column contains the left image from the stereo set. The middle column contains the reconstructed scene in our framework. The right column contains the confidence map for filtering. This is obtained from the histogram of the initial input image ran under a Canny edge detector [23, 99] in order to enhance the contour of the objects present in the scene. The contour represents the texture discontinuity boundary that will be preserved in the filtered disparity.

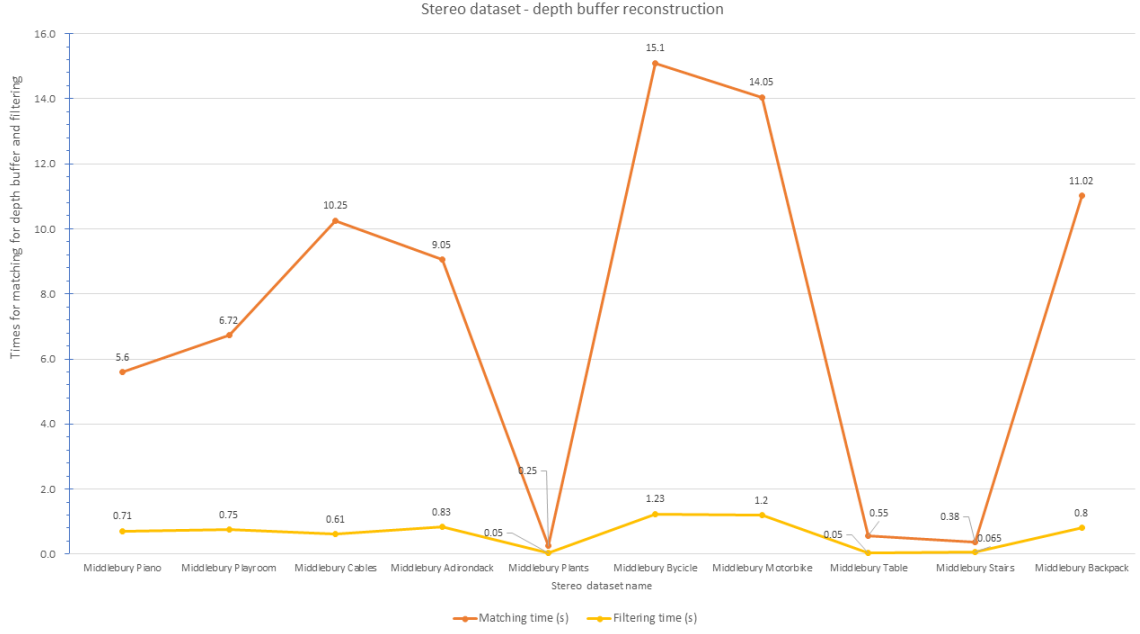


Fig. 4.2 Stereo Depth Reconstruction. The lower line plot represents the total filtering time metric. The upper line plot represents the matching time (loading the image and performing DP stereo block matching). The global minimum  $\min_{filtering}, \min_{matching}$  corresponds to the Plants stereo set, with a width and height value of  $710px \cdot 496px$ , 0.05 seconds for filtering and 0.25 seconds for matching and reconstruction. The global maximum is represented by the Bicycle dataset with a resolution of  $2988px \cdot 2008px$ , a reconstruction time of 15.1s, and a filtering time of 1.23s.

to complete. This is directly influenced by the modest size of the image entry pair. In contrast, Middlebury Cable, and Middlebury Backpack, are similar in size but with a different disparity, which means that in the case of Middlebury Cable dataset where the disparity is larger, the matching process starts a bit further from the edge of the left hand side image, essentially truncating the matching and filtering process which results in a smaller time compared to the Backpack dataset.

#### 4.2.2 Evaluation based on synthetic scenes and near-stereo

In real-world stereo datasets, it is often quite difficult to control, fine-tune, and test edge cases. For this reason, the second set of scenes was created synthetically (manually) inside Unity3D. Even though the scenes contain fundamental 3D shapes instead of complex meshes, this test (see Figure 4.4) is tailored to verify consistency of reconstruction against a known collection of parameters including: ground-truth depth, illumination sources, depth of every object's bounding volume, camera setup, and scene size. Compared to the evaluation criteria formulated at subsection 4.2.1, this test is not aimed at run-time analysis, therefore the input data is small in size (  $1024px \times 768px$  maximum).

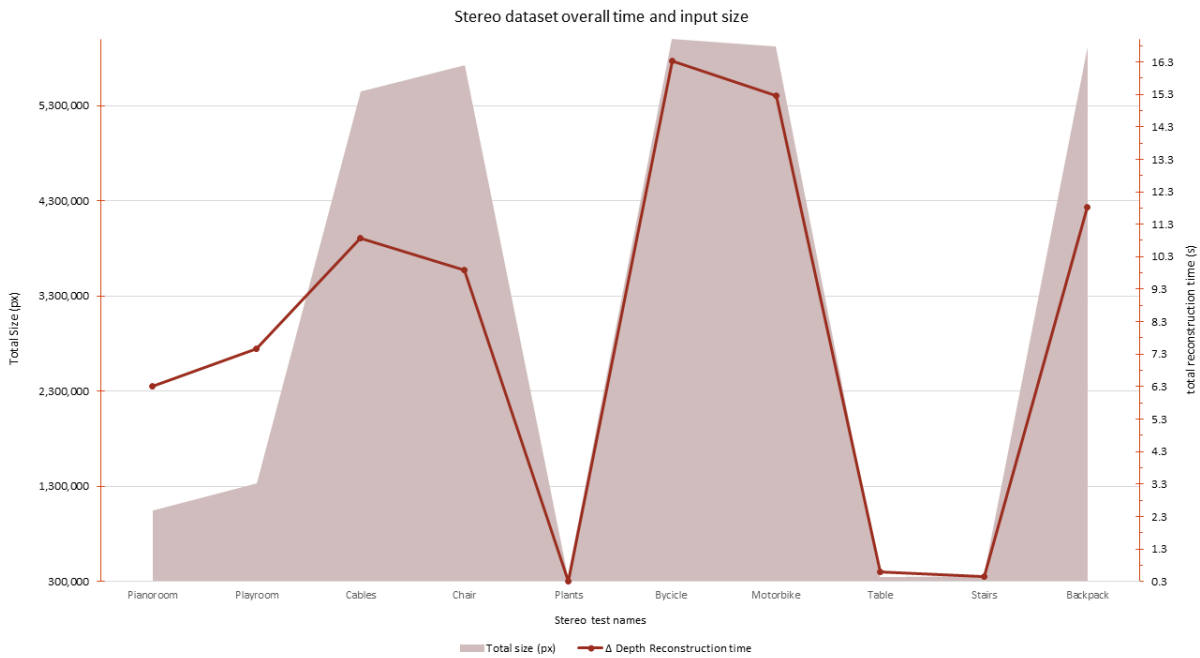


Fig. 4.3 Overview of total image input size and the overall time of reconstruction (matching, filtering, mesh creation).

As can be observed in Figure 4.4, the reconstructed scenes (centre column) present more noise, and are overall sharper in aspect compared to the ground-truth column (right-most column) originating from the Unity3D engine, however, no discontinuities are present at texture boundaries (around the contour of participating objects present within the original scene). This means that shadow regions can be easier to identify and mask in the eventuality of artefacts occluding pre-existing shadow-casting geometry.

Table 4.2 Performance overview of the reconstruction stage for the synthetic dataset. The mesh size is represented by the total number of polygons after dense reconstruction.

Scene	Width (px)	Height (px)	Matching Time (s)	Filtering Time (s)	Mesh Size
Synthetic 1	512	512	0.1	0.03	87,360
Synthetic 2	1024	768	0.1	0.04	262,216
Synthetic 3	512	512	0.2	0.02	87,360

The reconstructed scenes compatible with the depth buffer layout are filtered based on the confidence map which may contain inconsistencies that translate as dark or bright high-contrast regions which may bring objects slightly closer or further in the depth buffer compared to their real position. However, since this offset is added to all pixels, it acts as



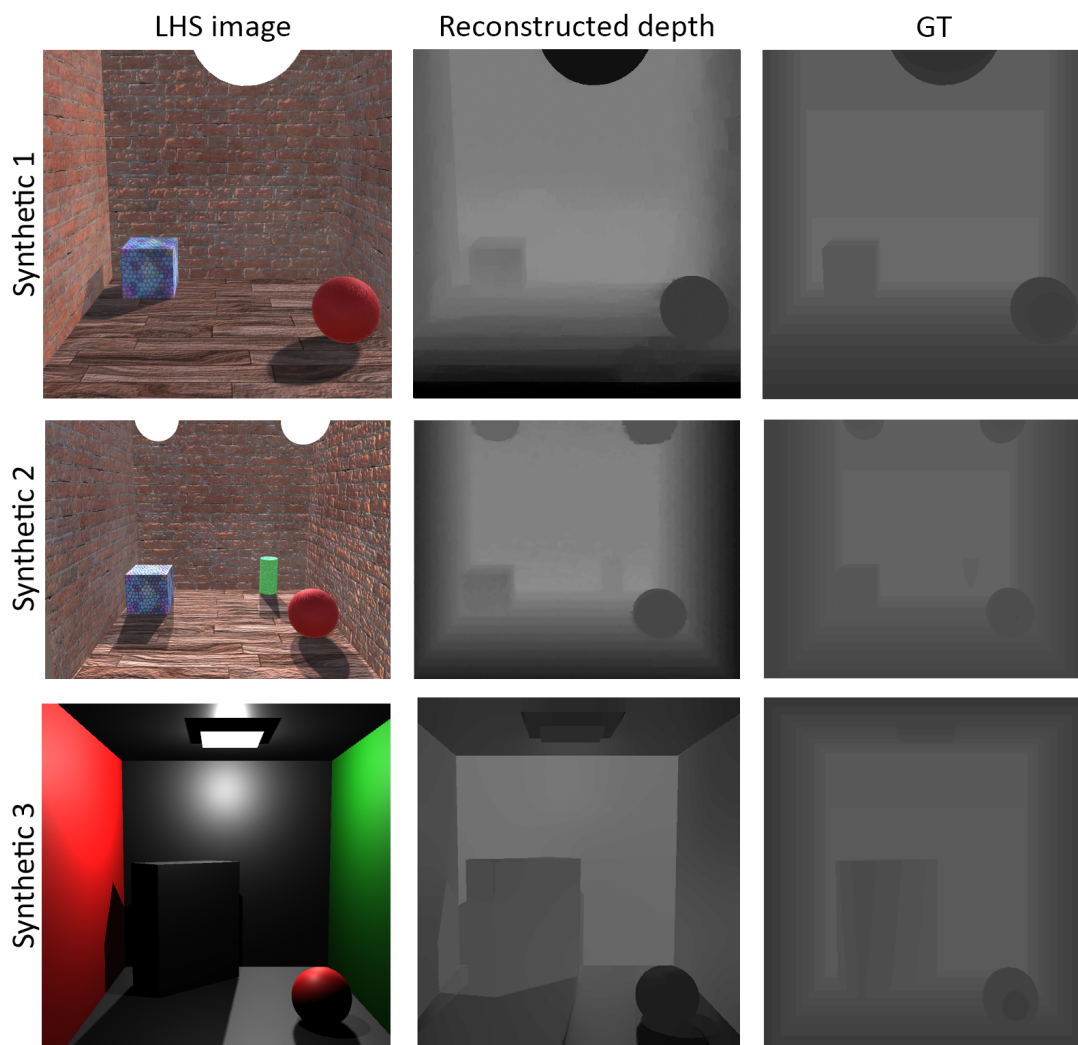


Fig. 4.4 Evaluating scene consistency after reconstruction based on synthetic scenes created manually in Unity3D (commercial game engine). The left column contains the left image from the stereo pair. The middle column illustrates the reconstructed depth in our framework. The right column contains the ground-truth(GT).

a scaling factor which, upon reprojection to the 2D scene as the final result, will not be noticeable.

A potential risk posed by slightly noisier regions present at the foreground of widespread flat areas could directly affect the shadow projection falloff making soft shadows appear inconsistent across their surface. From the results demonstrated by Figure 4.4, the noise is present predominantly at the back, not at the front of the scene.

Since the fragment program responsible for calculating shadow projection and sampling mip-maps will perform interpolations according to depth, this means that we obtain a smooth background which will not show inconsistencies when the final result is reprojected in 2D space under the initial camera parameters. The initial input images contain noticeable, well defined shadows in the foreground region. During the reconstruction



process, the entire input matrix is considered for matching, however, the pipeline does store an initial shadow mask (a binary map of the detect shadow features) as seen in Appendix B. Figure B.3, which is used in the relighting step to test whether or not a shadow already exists from the input at the region where an artefact occludes a light source. Shadow regions in images obstruct structural information which makes it challenging to reconstruct the geometry. One approach to addressing this is to ensure the input is in an HDR format, and that a pre-filtering step is ran before reconstruction (i.e. enhance colour balance). Alternatively, when using an LDR format, the image space can be changed to greyscale and the luminance channel can be increased. This operator affects the intensity of all pixel values, revealing variations in the shaded regions.

For the last set of tests aimed at evaluating reconstruction consistency, we look at how robust the reconstruction stage is in the presence of error-prone, uncalibrated input data. For this test, the device used to capture the input image pairs is a commercial mid-range smart-phone (see Appendix A for hardware specification). The assumptions made were that the images taken would not undergo any modification of lens distortion (macro or fish-eye), as well as the baseline would be horizontal and parallel with the ground. The latter is straightforward to achieve by enabling a nowadays common helper mobile phone camera function known as "grid-mode" which draws a set of horizontally and vertically parallel lines, equally separated, which allows the user to keep the device aligned to real-world surfaces used as reference.

The near-stereo dataset is challenging compared to the Middlebury dataset (calibrated stereo), or the synthetic scenes (known GT and low-complexity scene) due to ambiguous input (uncalibrated), and high-complexity when it comes to the present geometry. In numerous real-world cases, objects occlude one-another, can contain or intersect, and can participate in the illumination model as secondary light sources - bouncing or emitting incoming light or blocking and casting self-shadow.

This added complexity can introduce false-positives in matching due to similarity of localized content. An example of such a case can be observed in the first row of Figure 4.5 (Office 1). The lower right corner of the LHS image presents an ambiguous region where the black office chair occludes a similarly shaded screen, a number of dark objects on the desk, and itself.

When matching takes place, pixels in this region would ideally need to be matched using a smaller window for accuracy, however, this cannot be the rule for the entire image since it will create a noisy disparity. Since the matching process will yield a global optimal solution, there are situations where regional texture similarities will translate into the same depth locally, but a wrong value compared to neighbouring regions. In the case of the Office 1 image, the foreground chair will have regions with a greater depth compared to the background chair which will contain depth values closer to 0 in the depth buffer.



Fig. 4.5 Evaluating fault-tolerance near-stereo in the wild. The left column contains images from a near-stereo (uncalibrated - unedited or measured) pair. The middle column presents the reconstructed depth relying on our method. The right column illustrates the confidence map during the filtering stage.

In the specific context of compositing, this does not present an issue in the average cases - as seen in the other results from Figure 4.5, the optically-flat surfaces (table, floor, etc.) where compositing can take place in order to yield a believable result, were reconstructed adequately without noise or loss of texture data.

Table 4.3 Performance overview of the reconstruction stage for the near-stereo dataset. The mesh size is represented by the total number of polygons after dense reconstruction.

Scene	Width (px)	Height (px)	Matching Time (s)	Filtering Time (s)	Mesh Size
Office 1	2480	1256	5	1.7	1,038,731
Office 2	2480	1379	2.8	2.2	1,039,273
Office 3	1256	2148	2.3	0.44	899,295

The scenario on the second row in Figure 4.5 presents a more complex setting. During matching, the borders on the scanline are restricted in terms of number of adjacent neighbours to sample, and noise is expected around the edges. This can be addressed during the filtering step. In the second scenario, the noise persists due to the lack of correspondence around the border and the matching that takes places in the similar texture space. This type of noise is characteristic of images with large uniform textured areas, and is more frequent in real-world scenes compared to the Middlebury dataset in which scenes present texture variation. A way to avoid introducing this specific type of noise is to further filter the input set in terms of histogram luminance channel and contrast, and recover as much texture variation as possible before commencing the matching process.

In the third scenario described in Figure 4.5, the glass panel and the wall behind it present similar textures whilst the glass introduces reflections. This represents an ambiguous region to match and would benefit from pre-enhancement of the input data's histogram to alleviate noise.

### 4.3 Evaluation of the relighting consistency

The present lighting model is a synergy between a statistical approach to shadow-mapping, namely exponential shadow maps, and an augmented directional-light model. Taking this into consideration, our system yields realistic shadows when compositing is done on flat areas. On slanted surfaces over-stretching of the projected shadow may occur and this is due to the shadow function equating to a value greater than 1. Generally, compositing techniques make the assumption that the surface where artefacts are placed is flat, and horizontal.

Similar to the data used in carrying the evaluation for scene reconstruction (section 4.2), the relighting consistency evaluation will be relying on data from the Middlebury dataset (Figure 4.6), synthetically-made data in the Unity3D engine (Figure 4.7), and near-stereo data taken with a consumer smart-phone (Figure 4.8).

The two sets of stereo image pairs chosen from the Middlebury dataset for carrying out the relighting consistency (Figure 4.6) contain both illumination sources that are present in the initial input (Piano scene) and also outside of the viewing volume (Playroom scene). Furthermore, these two inputs vary in terms of amount of visually-noticeable clutter which in turn influences the possible suitability of compositing while yielding a plausible result. While it is reasonable to assume that an artefact must also belong (from an aesthetic perspective) with the collection of objects already present in the original scene, the present work is concerned with determining physically-accurate soft shadows with application to real-time techniques and not how well an artefact is assimilated in the scene.

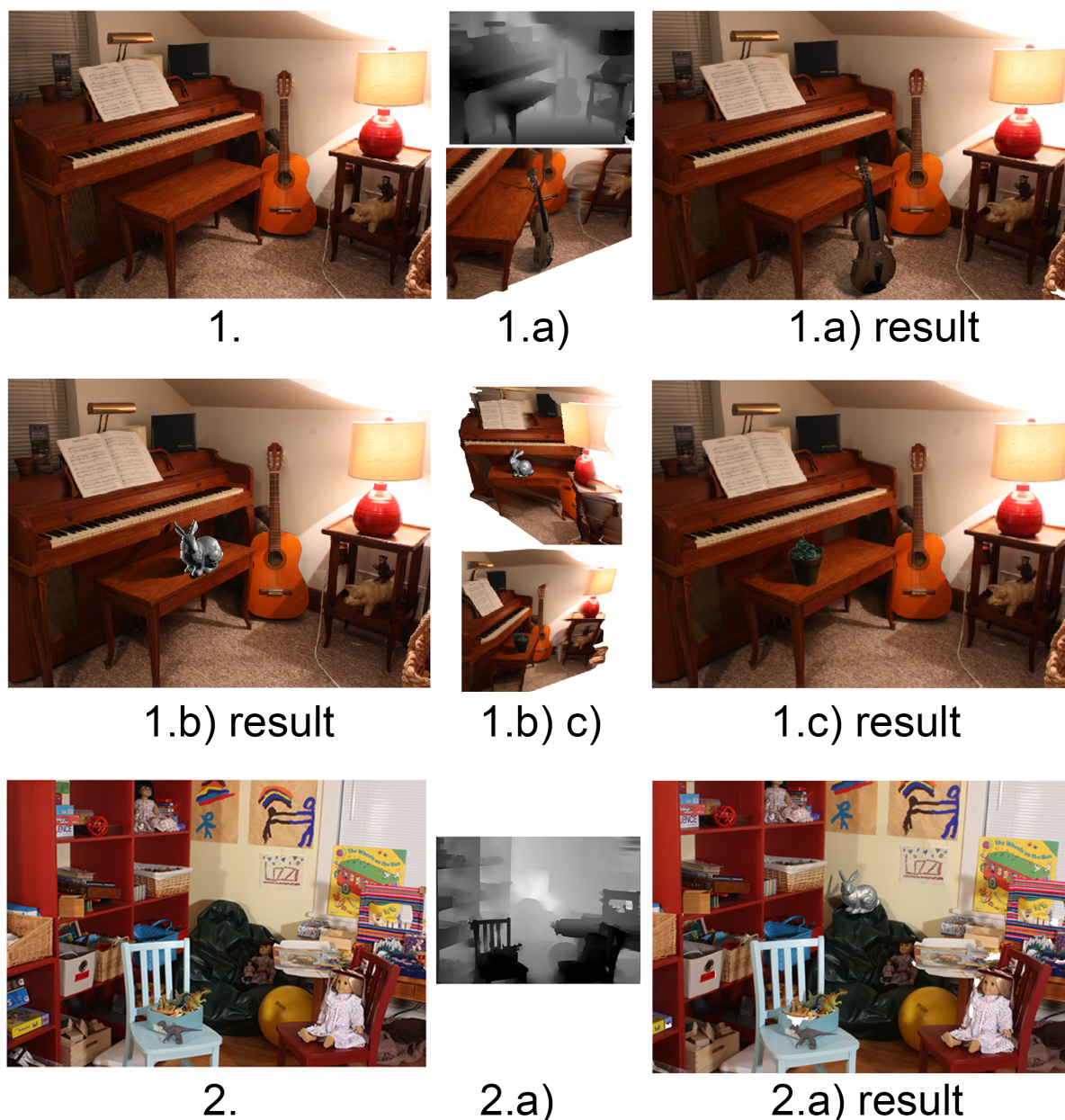


Fig. 4.6 Evaluating relighting consistency on the Middlebury dataset. 1. LHS: initial left image from the stereo pair. RHS: its counterpart reconstructed scene. 1a) LHS: Result of compositing the Stanford Bunny model. RHS: captured intermediate view from our framework as viewed by one of the cameras in the reconstructed scene. 2. Left-most image represents the left image from the stereo pair. Middle image is the reconstructed scene. Right-most image is the result of compositing the Stanford Bunny model.

Studies have shown that artefact compositing (in the context of matting and compositing - 2D domain) does not need to render physically-accurate results as long as the original scene presents a large number of visual queues (presence of a combination of factors including large number of objects - clutter, moderate amount of objects varying across surface material, complex lighting) [110]. The work in this thesis demonstrates that previous



compositing frameworks which aim at approximating light interaction by decoupling the illumination model from the physically-based counterpart, will not allow for compositing that scales with the number of artefacts added, unless a minimal physical model governs the rules of the relighting step.

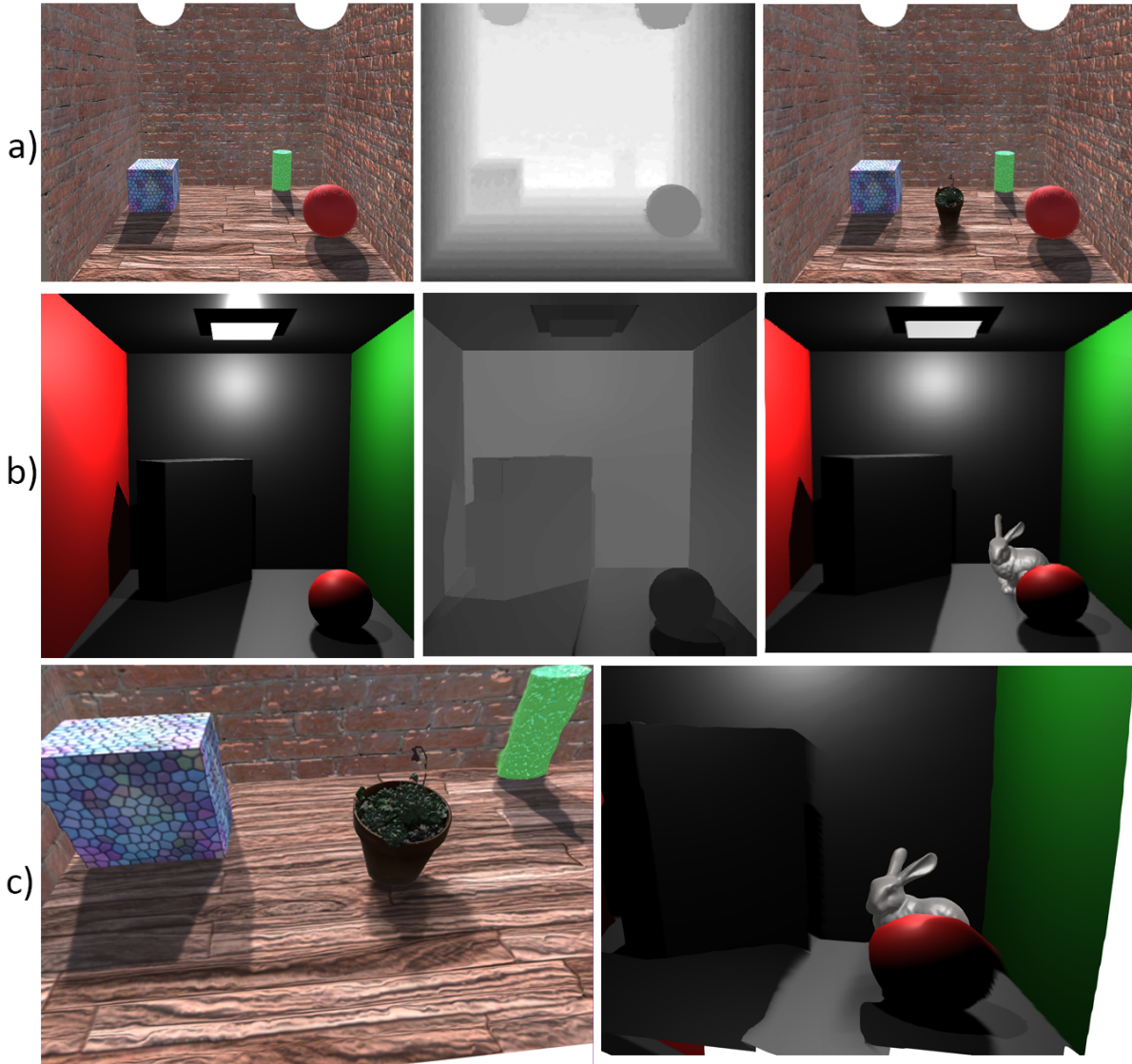


Fig. 4.7 Evaluating relighting consistency in the synthetic scenes. a) and b) from left to right: initial left image from the input stereo pair, our reconstruction of the scene, and the result of compositing an artefact. c) Last row contains closer views captured within the 3D scene before the final result was projected back into 2D to match initial camera parameters.

The first row in Figure 4.7 describes a scene with two light sources, that are visibly dictating the shadow falloff on the three geometric shapes. This was a deliberate construct to ensure that the composited artefact would contain consistent shadows, in accordance with the number of lights present. In the last row c), left-most image we captured a view

from within the 3D reconstructed scene that already contained the cube with its respective shadow map. The composited flowerpot artefact present a similar shadow projection which will hold for a large number of directional or point light sources in our framework. The second row b) in Figure 4.7 looks at the compositing artefacts behind pre-existing geometry - an aspect that has not been tried in similar compositing frameworks.



Fig. 4.8 Evaluating relighting consistency for near-stereo input. a) The initial left image of the input stereo pair. b) Result of compositing an artefact assuming one diffuse light. c) A close-up capture from within the reconstructed scene before the final compositing result was reprojected in 2D to fit the initial camera parameters.

The near-stereo dataset presents a greater challenge due to the lack of stereo calibration and possible erroneous disparity approximation. In the worst-case scenario, the algorithm for reconstruction will present inconsistencies between the foreground and the background in the initial input image pair. When this occurs, the point cloud will contain areas that appear denoised (smooth) from the initial camera parameters in the 2D setting, but visibly distorted within the 3D counterpart, making those areas incompatible with soft shadow projection.

Figure 4.8 c) illustrates such cases, where the visibly flat areas such as the wooden coffee table, and the yellow sofa seat were reconstructed consistently, however, in the vertical plane, the backrest of the seat presents a high amount of noise, making it difficult to composit artefacts that would have their respective shadow projected onto that vertical plane.

## 4.4 Discussion

Automated compositing of 3D artefacts into 2D images is carried out through several stages designed to infer as much data as possible in order for the final result to be consistent and believable. As covered in previous sections (3), when dealing with real-world data, the challenge of compositing consists of reaching a plausible result relying on uncalibrated or often times missing information which leads to solving an ill-posed inverse rendering problem.

As illustrated by Karsch et al in a study for compositing ([79]), user input is desirable under the condition that expertise or prior knowledge are not required in order to perform the fine-tuning in between framework stages. Our work follows this finding, and allows the user to adjust the camera parameters for a correct projection, as well as adjust the lighting after the initial illumination model was estimated. This is a suitable approach for both offline and real-time rendering, however, it is not ideal for AR applications which are primarily gesture-controlled.

The techniques underpinning each stage of the framework have a potential to be used in areas related to solving inverse rendering problems relying on various types of hardware. The present section reasons about this potential and aims to identify the advantages and shortcomings of the implementation. This is achieved by analysing the performance data in tandem with similar existing work.

## Observations on the reconstruction stage

The three datasets used in the reconstruction consistency evaluation (Section 4.2) are constructed in order to offer different insights:

**Middlebury dataset.** The real-world calibrated stereo input is characterized by high resolution predominantly landscape oriented pictures that present varying degrees of occlusion between objects as well as self-occlusion. Furthermore, the materials dictating surface appearance of the constituent objects pre-defined in the initial input do not present translucency nor high reflectivity. The latter observation implies that the likelihood of identifying secondary light sources that contribute to shadow-casting bears a significantly low value. If the number of shadows present would've taken up a large percent of the scene, these would require masking before matching can proceed.

Despite decoupling the processes that are related to reconstruction from the ones related to illumination modelling, the reconstruction stage will indirectly bring forward persistent information regarding the illumination settings of the initial scene. One example of this is the shadow projections already present - these textureless areas can introduce errors in the later stages, and they may not need to be taken into consideration during reconstruction at

all as compositing an artefact would cancel them (artefact can be inserted between them and a light source therefore requiring a cancellation or removal of the original shadows).

This dataset is ideal due to its calibrated nature - it implies that the average correct disparity is in the range [80 - 112], and there is consistency between the method used to acquire the dataset (assume same distance and orientation between cameras capturing the left and right side of the stereo pair). Taking this into consideration, we can identify and reason about cases where the reconstructed scene contained erroneous information. For example, in the penultimate row of Figure 4.1 the part of the chair that is in the foreground is at very similar depth with the seat. This is incorrect, as the seat should be behind (greater depth value) the foreground arm-rest. This error is introduced due to a number of factors including self-occlusion and lack of local variation in texture space (neighbouring pixel values to the current window position present a high similarity). This poses a challenge for compositing only if the area that the artefact is to be placed on presents a high degree of noise, or if it contains significantly different values compared to the foreground and background parts of the object that are involved in self-occlusion.

**The synthetic dataset.** The scenes created in the Unity3D engine allow the present framework to have a comparable ground-truth depth buffer, not just a disparity map. A noticeable difference between the ground-truth(GT) set and the one produced with our framework (see Figure 4.4) is that Unity uses the 3D scene geometry only, excluding shadows, in order to populate the depth buffer. As the cameras defined in Unity (LHS and RHS corresponding to the left and right eye in the human vision system) produce a 2D image stereo set used by our framework to reconstruct the 3D scene in a similar manner to real-world input, the shadow projections were present.

The two sets have a high structural similarity, which means that when the input image set does not present textureless regions (lack of local texture variation) and no self-occluding topology, the compositing can yield plausible results following physically-accurate properties. Studies [172] demonstrate that compositing cases involve indoor scenes (where the room outline is visible and fits an geometric volume approximation) more frequently than close-up of objects or outdoor scenes. The synthetic scene evaluation is an appropriate candidate for illustrating the suitability of indoor scene compositing in our framework due to the nature of the box-like scene, and diffuse or directional illumination present at the top or lateral sides of the scene. Another aspect that reinforces the suitability of the evaluation approach is the number of planar surfaces available in such scenarios.

**The near-stereo dataset.** This dataset was captured using a Samsung S8 mobile-phone camera, with no prior calibration. Despite using the RAW format to store and send the near-stereo image pairs for matching, the metadata is not used in our pipeline; it operates on the assumption that no metadata exists, and that the required parameters for reconstruction can be estimated, increasing its portability. Because the orientation (tilt)



and position of the device vary more compared to the Middlebury and synthetic datasets, this makes the near-stereo evaluation quite challenging.

As expected from the observations made on the Middlebury dataset - self-occluding objects (the table and sofa in the 3rd and 2nd row of Figure 4.5 respectively) present high amounts of local noise. Since these regions are difficult to access for compositing, our framework does not rectify them separately. Despite this shortcoming, the foreground of each scene is reconstructed consistently, and in all cases the background is over-smoothed causing information loss (i.e. base of the tree in the first row of Figure 4.5). This problem could be addressed as a separation of filtering based on background segmentation relying on the near-stereo input.

Across all evaluated scenes in this section, the uncluttered planar surfaces were sufficiently denoised in order to allow for a correct soft shadow projection. Assuming the compositing takes place in the foreground and not in the background of the input near-stereo pair, the relighting stage will not introduce noticeable errors. This opens up the possibility to capture a depth estimation (targeted for depth buffer storage) using commercial, accessible, mid-range devices.

**Comparison to similar work.** Other compositing pipelines rely on coarse estimation of constituent scene properties and attempt to reconstruct the full scene, or the areas where the artefacts will be added. This section draws insight from the different approaches and compares the results for this stage with our technique.

Figure 4.9 illustrates two featured works of Karsch et al. (2015, 2012). The method developed in 2015 relies on a set of RGBD images (depth value already captured) and an algorithm developed by Lee et al. [95] in order to detect surface orientation. In contrast, our method produces reliable depth information without relying on bespoke depth sensors. Moreover, as seen in Figure 4.10 our surface orientation is captured by just relying on surface normal estimation after dense reconstruction.

The depth estimation visible in Figure 4.9 c) comes from a machine learning algorithm capable of reconstructing depth from single images. This algorithm was not used in compositing pipelines due to its noisy surface (predominantly present in the foreground) and loss of information in the scene’s background. In d), e) the estimated depth of the planar surface using the RGBD data is still ambiguous in small localised areas of the foreground.

## Observations on the relighting stage

Our framework presents a novel approach for computing shadows in the context of 3D artefact compositing in 2D images. This approach is motivated by two factors - ensuring

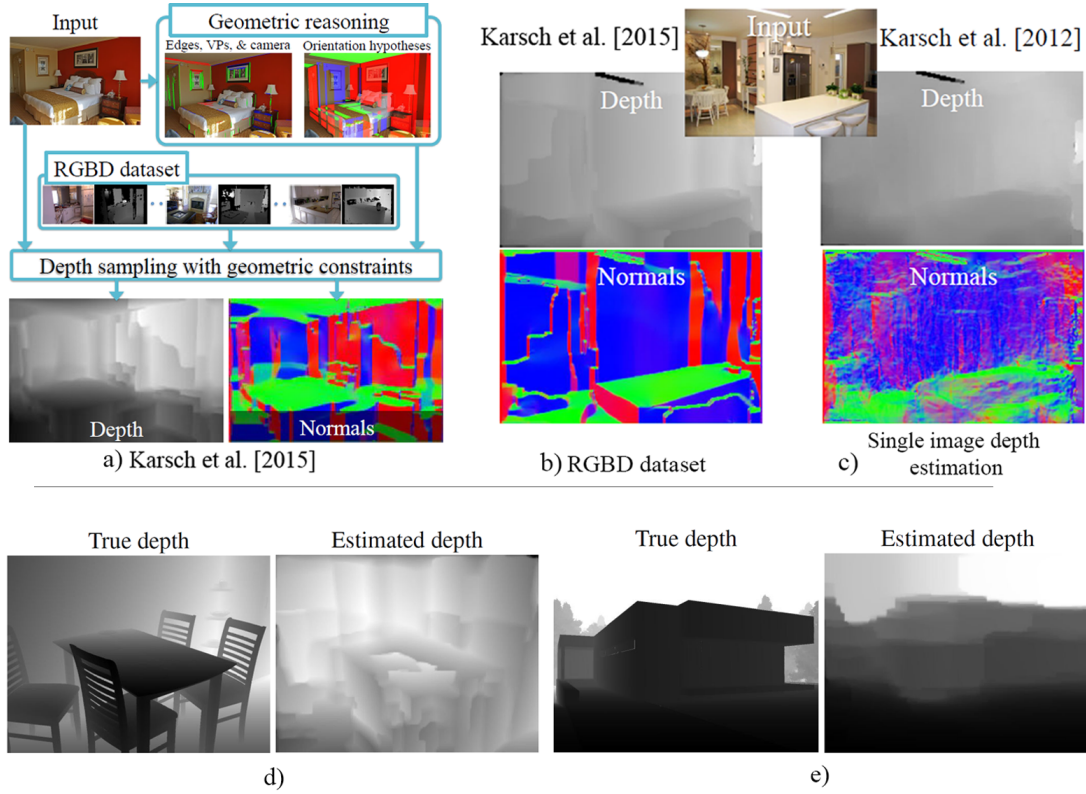


Fig. 4.9 Overview of results for reconstruction from Karsch et al. 2015 [79] and Karsch et al. 2012 [78]. a),b) Preview of depth reconstruction in the work of Karsch et al. 2015. c) Comparison of Karsch et al. 2012 reconstruction from single image. d), e) More scenarios of reconstructed scenes in the works of Karsch et al. 2015.

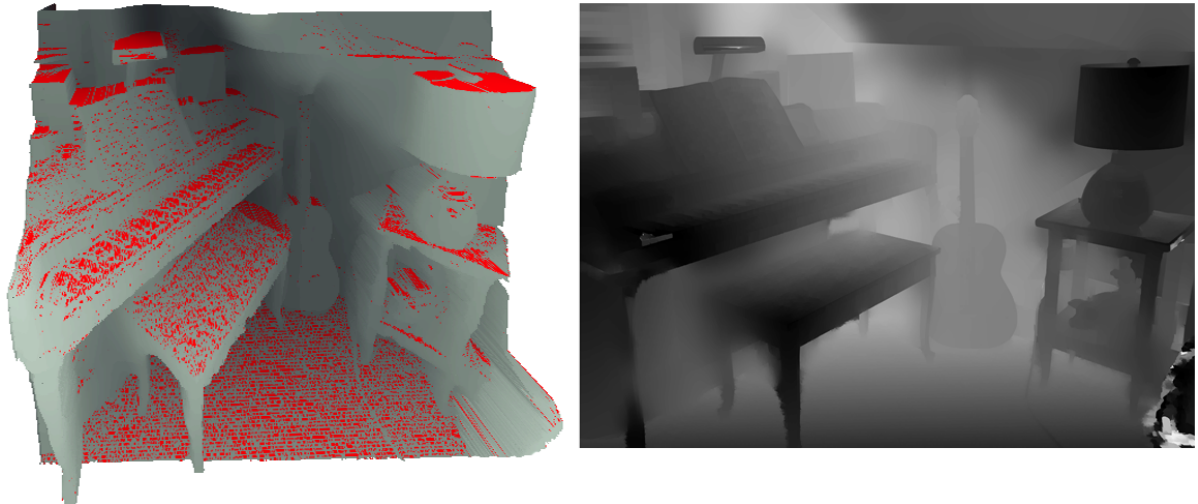


Fig. 4.10 Surface orientation and depth reconstruction. Left-most image represents the reconstructed scene with planar surface detection based on normal orientation after the dense reconstruction from point cloud (red areas). Right-most image represents the counterpart depth-buffer ready map reconstructed by our framework.

consistency after multiple artefact insertion, and making compositing a viable solution for real-time applications.

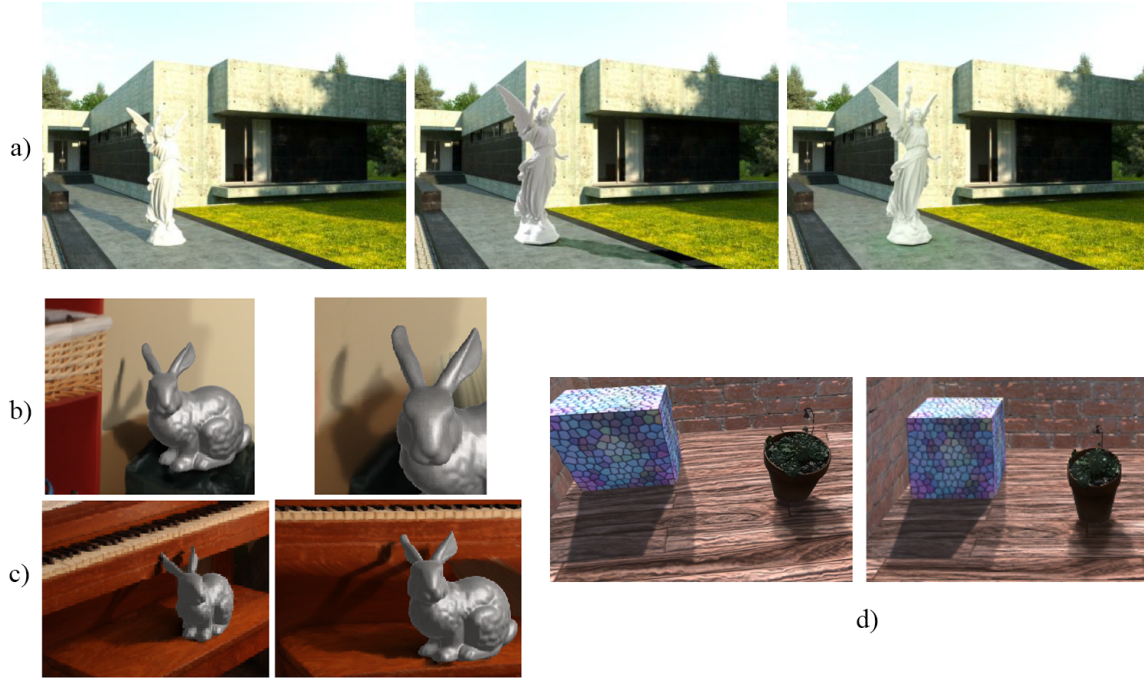


Fig. 4.11 Overview of compositing consistency in different frameworks. a) Left-most image is the ground truth scene created synthetically in the benchmarking section of Karsch et al. 2015 [79], middle image is the composition result for Karsch et. al 2015, right-most image is the result in the works of Lalonde et al. [92]. b),c),d) Left-most image the result of compositing produced by our framework. Right-most image is a close-up capture of the cast shadow inside the reconstructed scene before reprojection.

Figure 4.11 row a) shows plausible but physically inaccurate projected shadows. The user study presented by Karsch et al. [79] indicates that relative cues are more important than physical accuracy. Based on that particular work, if numerous artefacts were composited instead of a single one, the relative cues would not be the only criteria for creating a plausible result. The last image on row a) showcases a superimposed shadow at the boundary with the contact surface.

Furthermore, that set of images was obtained while relying on offline rendering techniques, taking up significant amounts of time.

In contrast, our results indicate that physical accuracy achieves consistent results in significantly less time compared to offline rendering, since statistical shadow mapping techniques were designed for real-time applications.

Another note-worthy aspect is the shadow consistency - our framework produces soft shadows with noticeable falloff, whereas the results of other frameworks presented in Figure 4.11 row a) illustrate a uniform soft shadow. Depending on the material properties of the artefact added to the scene, a uniform soft shadow may not be the desirable solution.



Fig. 4.12 Left-most image illustrates the classic exponential shadow mapping, showing lack of contact shadow at the base of the object [43]. Centre image: composited object within our framework demonstrating the contact shadow issue when relying on traditional exponential shadow mapping. Right-most image: our modified statistical shadow mapping algorithm with blurring kernel to enhance the penumbra effect.

Moreover, our approach to statistical soft shadows presents a solution to the sparsity of the contact shadow introduced by the classic exponential shadow mapping technique as seen in Figure 4.12.

# Chapter 5

## Suitable domains of application

Compositing involves a number of processes ubiquitous to the post-production stage in the movie industry. They address challenging problems that deal with maintaining scene consistency after artefacts were introduced in the original scene. As such, our approach to compositing offers portable solutions for both reconstruction and relighting. Other areas where our approaches can be applied include:

**Augmented Reality and real-time applications.** Augmented Reality (AR) technology enables the user to have an immersive experience through the use of bespoke hardware such as Microsoft's HoloLens, tablets or mobile phones. These technologies rely on one or more cameras, placed side by side (if they are mounted on a headset and mimic the human visual system) or one at the front of the device (to see the surrounding environment) and one on the backface of the device (for gesture interaction). If the device has a single camera, chances are, it is an RGBD sensor in order to record depth information. In the case of Microsoft's HoloLens, the two cameras acquire a stream (video) of stereo images (much like our framework).

Figure 5.1 shows the spatial mapping carried by HoloLens using Unity 3D - the device continuously scans the surrounding environment building a triangulated mesh stored locally on the device as an .OBJ mesh file. This is an advantage as it allows the scene to be saved and shared at any time. The 3D scene reconstruction from stereo in the HoloLens device does not cope well with textureless areas that offer no predominant features (dark surfaces - shadow, bright surfaces - reflective materials or light sources, and translucent objects). When such a surface is encountered, an empty patch is added to the mesh in the Spatial Mapping view.

In the context of the office desk illustrated in Figure 5.1, the keyboard, mouse and desk lamp will be represented as patches instead of densely reconstructed topology - their overall appearance will be flat. This situation poses a challenge for compositing any artefact that casts a shadow which can potentially intersect or project over the patched area.



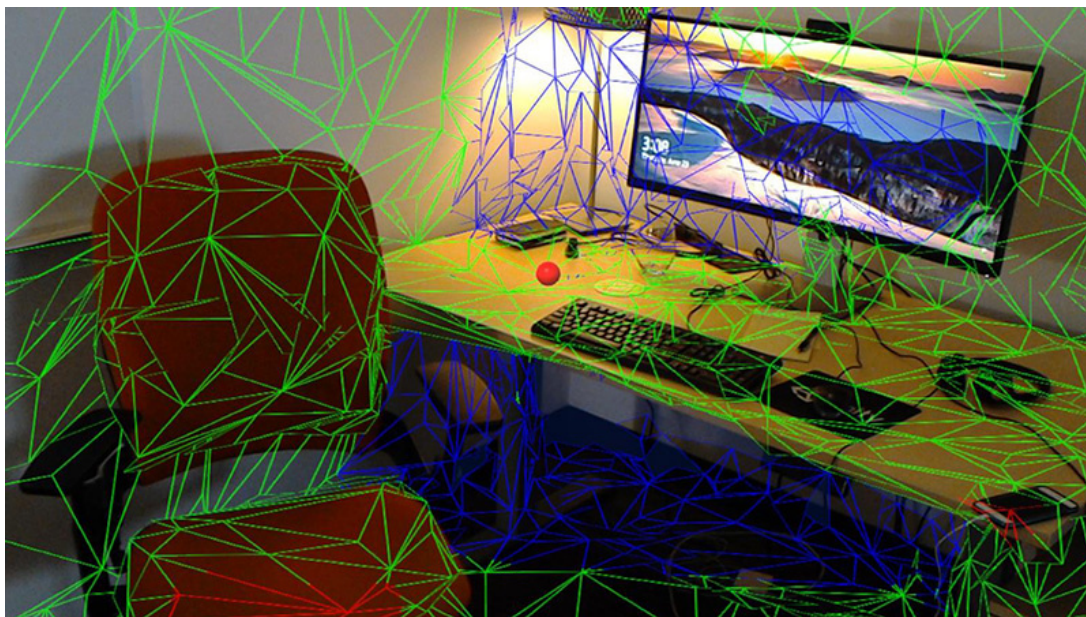


Fig. 5.1 Spatial mapping recovered by HoloLens under Unity 3D engine (Appendix A.1).

The dense reconstruction captured with HoloLens has no reinforcement when it comes to surface orientation. Therefore, surface patches may improperly intersect and overlap especially around areas where the real-world data changes orientation (i.e. from a horizontal desk to the vertical wall behind it).

The issues encountered by surface mapping could be addressed locally (only when artefact compositing is requested) using our reconstruction method, or even adopted for the continuous scanning. When relying on a single image or a stereo pair of images, the input becomes restrictive compared to a video stream. A stream of images offers the opportunity for foreground and background segmentation, for feature matching and detection with a greater confidence factor (larger sample set). This means that shadows could be detected, masked, and removed with high accuracy. The same goes for detecting object boundaries. Since our framework produces a reliable reconstruction from just two images, it could be an ideal candidate for HoloLens reconstruction. Furthermore, HoloLens images produced by each camera have a medium resolution (634px x 360px). When comparing our reconstruction time on similar image sizes, the average reconstruction takes 0.01 seconds which demonstrates that at least part of our implementation can have a positive impact for this technology.

**Optimizations for post-production.** Rendering props or incorporating edits into post-production represent time-consuming operations - offline rendering can take tens of hours. Our compositing method could be used as an alternative to the classical post-production rendering for inserting new props. This cuts production time, as there is no need to ray trace every frame (for animation) or re-render the whole footage.

**Image forgery detection.** Addition of artefacts to an image represents a permanent alteration, however, the addition methods can be traced and isolated. Compositing pipelines follow a similar, incremental stage approach to reaching a plausible result. They offer insight into the alteration process, and by following the reverse of the stages, alterations can be traced with a high degree of accuracy. An example involves identifying areas inconsistent with the overall lighting, or inconsistent with the average scene perspective projection. Johnson and Farid [70] demonstrate that it is possible to detect point light sources from the analysis of silhouettes and shading along object contours. This method was further developed to estimate complex lighting using a spherical harmonics model [71]. The research efforts in the area of image forgery detection rely on lighting inconsistency as the primary criteria for validating an image [81], while geometric inconsistencies tend to not be factored in any joint detection models.

**Product visualisation and rapid prototyping.** As discussed in the related work section, mobile technology used in conjunction with compositing software enables the user to declutter rooms and fit virtual furniture, and in general apply compositing to a targeted and well-defined problem aimed at product placement and visualization. In addition to this, compositing can be used in both movie and game industries for rapid scene prototyping in real-world settings.

# Chapter 6

## Conclusion

### 6.1 Contributions overview

The process of artefact compositing has applicability across a broad range of creative industries. creative industries. It arose from the necessity to alter images and video footage in post-production. Along with the growing needs of these creative industries, image capture and rendering technologies have expanded, offering solutions to these needs. Modern rendering technologies generate large quantities of data representing high-resolution textures and meshes of complex models. There is a tradeoff to this increase in quality and complexity, however, in terms of computation time to achieve photorealistic results, and this is why post-production software relies on offline rendering, and more generally, on solutions that are not running fully in real-time. In order to obtain plausible results in post-production compositing, affected scenes have to be re-rendered in order for the illumination model to account for the newly added geometry and surface material properties. This is a costly operation and not entirely a fail-proof method (i.e. animated models, submashes and specialized materials such as oil films, irridenscent surfaces, and generally surfaces that are secondary light sources affect illumination in ways that are hard to predict or avoid).

The present body of work contributes solutions to a number of the challenges in modern compositing techniques. Based on the compositing pipeline stages, the contributions can be classified as follows:

**Scene acquisition from stereo.** As the first stage of our framework, the acquisition and reconstruction process operates on a minimal input in the form of a stereo pair of images (or near-stereo). This pair goes through our matching system which outputs a specialized smooth disparity map (*depth map*) compatible with the depth buffer format in modern APIs. There are two stand-out aspects to this approach, namely that the method generates a denoised depth map straight from the matching process. This improves



performance, as the densely reconstructed mesh from the point cloud does not need to be denoised based on surface normal direction. The second notable aspect is that on the tests carried for large textures, the average run time from the data entry point up to mesh generation is under 5 seconds, and our tests indicate that even on lower resolutions (medium size tests), the compositing still produces consistent and plausible results in near real-time, making it suitable for use on a range of devices, such as workstations for post-processing work, to mobile devices for live editing of photos as they are taken.

**Physically-based relighting.** The second contribution regards the relighting step, specifically the soft shadow projection. The key insight of our approach is the ability to represent accurate soft shadows through a statistical shadow mapping algorithm that is consistent with the illumination setup present in the scene (i.e. number of light sources). This step provides an important visual queue to the existing scene after an object has been composited. As with the first contribution, this approach is novel for compositing pipelines (as traditional compositing relies on either path-traced effects, or superimposing a shadowed area on the point of contact between the object and the surface it was inserted on to). This approach was motivated by the desire to composite multiple artefacts that still create a consistent and convincing result. In traditional compositing, as the number of artefacts added into a single scene increases, the greater the chances that these objects will start occluding one another or primary light sources, at which point it is crucial to render physically-accurate shadows. The present framework allows for multiple artefact compositing without creating inconsistent shadows in real-time.

The secondary contributions of the work stem as a by-product of the design and implementation of the framework presented in this thesis, and are listed in decreasing order of significance:

**Applicable in real-time technologies.** Devices such as Microsoft’s HoloLens AR kit (two front cameras), smart-phones with HDR and RGBD capability make the perfect consumer devices for AR and compositing. Traditional compositing pipelines do not perform in real-time, require user expertise and annotation - an interaction which cannot occur on a mobile device or head mounted display, and are generally dependend on specialized hardware or third-party software to perform the rendering. In contrast, the present work carries both suitably detailed scene reconstruction on a much more restrictive input set (2 images only) as opposed to HoloLens (continuous image stream), and fast enough to be a viable candidate for this technology. Furthermore, the reconstruction stage does not introduce holes, or overlapping patches and can also account for non-watertight geometry such as ubiquitous items found in indoor scenes (cups, glasses, etc.), aspects which HoloLens cannot avoid or adjust as the reconstruction takes place.

**Portable and platform-agnostic.** The present work operates on the assumption that the data entered comes in the shape of two stereo images. In addition to this, all

the rendering API operations are performed in OpenGL ES which is an open-source API, found on every modern chipset for mobile platforms as well as workstation hardware. These aspects make the framework platform-agnostic. Due to the modular design, each stage is decoupled and can easily be attached or integrated to other code-bases or as a stand-alone plugin to different applications such as HoloLens or Unity3D engine. This makes the solution portable and easy to maintain.

**No user-annotated data.** Existing techniques for object composition rely on inference, annotation or extraction of scene geometry to provide additional data for successful resolution. Collectively, these operations are known as "*image-grounded editing*". Our technique shows that 3D object compositing, at the correct perspective, shading and illumination model, is possible in the absence of any such additional information, user annotation, or user experience.

## 6.2 Limitations

Our system is not generic to the extent of rendering reliable results when the input stereo image pair comprises of distorted images. We use the term distorted to refer to radial, tangential, and caustic distortions produced by fish-eye, wide-angle, and catadioptric cameras [137]. Throughout the framework, we assume perspective-projection manipulation, therefore lacking the means of estimating and reducing distortions through warping [? ]. In addition to this, our system cannot handle structurally insubstantial scenes such as close-up (macro) photography.

The scene reconstruction step present in our framework is a straight-forward greedy approach for dense reconstruction, followed by a planar surface detection and extraction step which is a minimum requirement for determining possible surfaces where objects could rest upon while producing a plausible result. However, this step could be greatly enhanced (at a time penalty) by an object detection approach coupled with a machine learning technique [66] focused on indoor and outdoor scene classification.

The illumination model has a set of drawbacks primarily caused by the approach of identifying appropriate bounding volumes for the possible light sources in the scene, and assumes a Phong-based single-view point light model when sources are identified in conjunction with IBL for sampling an environmental map. In the absence of primary visible light sources, the system assumes a global illumination model. Due to this assumption the relighting step can result in an incorrect shadow map projection. This issue can either be prevented by enforcing a light classification approach as described by Zhang et al. [172], or ameliorated during the shadow projection computation step by updating a texture of occluders (occluder bit-masking) [135]. Despite the capability of compositing synthetic meshes between existing topology, our system does not account for the transparent, or

possible reflective nature of the neighbouring areas. Therefore, as the material property variation across the scene increases, the chances are that the relighting step will not offer reliable results.

As previously stated in Section 4.2, exponential shadow mapping has a set of constraints. When these constraints are not met the shadow function fails, rendering undesirable results. As suggested by [3] the failure cases could be classified and depending on their respective type a number of solutions (usually filtering) can be exercised.

## 6.3 Further work

**Surface property identification for compositing special materials.** As described in the Limitations section (6.2), the illumination model estimates parameterized light properties through an environmental map and directional light structure. A fully-featured physically-based material system would provide a unified solution for modelling material appearance. A journal paper is currently being prepared to account for this change which would be a novel contribution for real-time compositing. Furthermore, this change enables complex light-material interaction with translucent objects that are difficult to capture in computer graphics however their presence is ubiquitous in real-world situations (i.e. glasses, plastic bottles, windows, etc.). To this moment, there are no compositing frameworks that can handle generic light-material interaction, the common assumption being of a Manhattan world with Lambertian surfaces.

**Machine learning for automating material property inference.** When attempting to automate the process of identifying material properties from a single image, stereo pair, or video stream, the process usually involves extracting objects using an edge detector or feature matcher, followed by texture analysis (some of these steps cannot be carried accurately on single image, and can be time consuming on a video file). This can work if the scene contains only one object presenting only one type of material, but it becomes an ill-posed problem if more objects are present, and the illumination is unknown or ambiguous. In their work, Karsch and Forsyth [76] demonstrate that it is possible to extract diffuse reflectance and specular parameters (spatially varying material properties) in a single image where shape and illumination are jointly inferred. The assumptions made by the authors is that the entry data will show only one object, and that the material model recovered is low-order and cannot be as robust as the BRDF of arbitrary shapes. Their recovered material estimate is not correct with respect to the physically-accurate real-world light sources and objects shape, and cannot handle interreflections.

Deep learning techniques have started to become a popular tool in the field of computer graphics, for such processes as the modelling of material appearance [134], and this has a significant impact on how compositing software will solve light-material interaction in the

future. It is therefore worth developing a ground truth dataset for learning how various materials appear under known illumination and apply the extraction algorithm similar to that of Karsch and Forsyth on a larger scale for expressing each material present in a scene as a parameterized BRDF.

**Extension to video compositing.** Data obtained via video capture offers more information compared to a pair of stereo images in cases where either the camera or objects within the video are moving over time. Background and foreground segmentation can be carried out with more accurate results, shadows can be defined and masked without user annotation, and in general, feature matching can allow for inferring data and relations between participating entities in each frame, that otherwise would not be possible in single or dual image input. The drawback is that most of the algorithms for single images do not scale well with video input, however, since video capture technology evolves (light-field capture for 3D video), the demand for fast and reliable compositing techniques increased significantly.

**Augmented reality compositing with special materials.** The framework presented in this thesis will be upgraded to allow for special artefacts to be composited. We define special artefacts as any 3D model that is a light source or drastically modifies the initial light setup (light sources such as lamps, candles, or reflective surfaces such as mirrors).

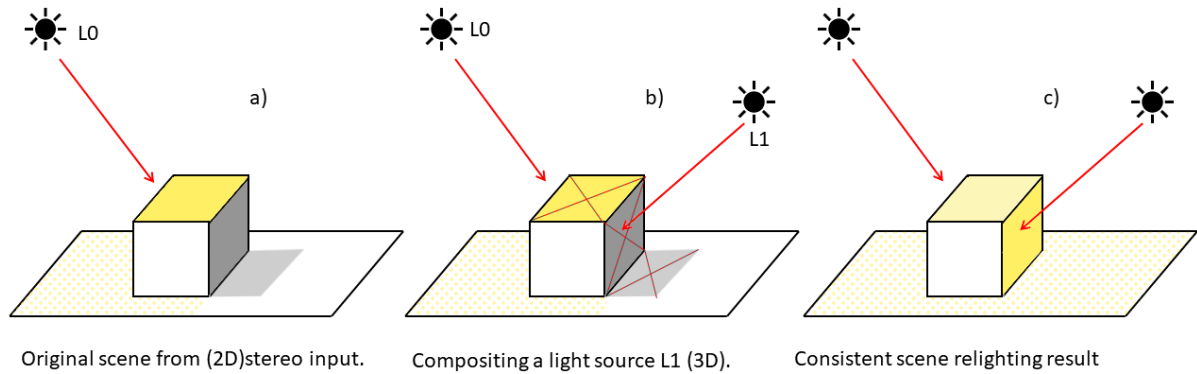


Fig. 6.1 Scenario describing the composition of a new illumination source as the 3D artefact. a) Original 2D image with one light source visible in the scene. b) Specifying a new illumination source as the 3D artefact. The red dotted lines illustrate the changes that need to take place in the scene for relighting. One of these changes involve the identification of the shadows in the initial input and their removal. c) The plausible and physically-accurate result of the composition and relighting.

The aim is to develop a generic PBR model that allows for light sources to be composited consistently (as demonstrated in Figure 6.1) without requiring a separate pipeline to the rest of the materials. Work is currently being undertaken in this novel approach for real-time AR applications, with the aim of submitting the work to the Transaction on Graphics Journal.

The present framework can identify the initial shadow areas (see Appendix C for approach) in the input image and store the shadow mask for the spatial partitioning data structure to check for light-object obstructions, however, the shadows are not removed from the input. This operation is necessary during the relighting step in order to composite primary and secondary(reflective) light sources, as well as accounting for the case where an artefact blocks the shadow caster and therefore prevents it from propagating shadow regions in the scene.

Another example involving complex light-surface material interaction is posed by the presence of areas that look translucent in the initial input image (Figure 6.2). In this case, it is challenging to determine the initial translucent area without relying on user annotation. If an assumption is made regarding minimum annotation, and the translucent area is identified, then the next challenge is understanding what is the full extent of the 3D translucent object from a single image (i.e. the objects in real-world cases do not present symmetry along an axis). When compositing a 3D object behind a translucent area, the 3D object's appearance and reflection have to be deformed according to the surface shape and refraction index of the translucent material.

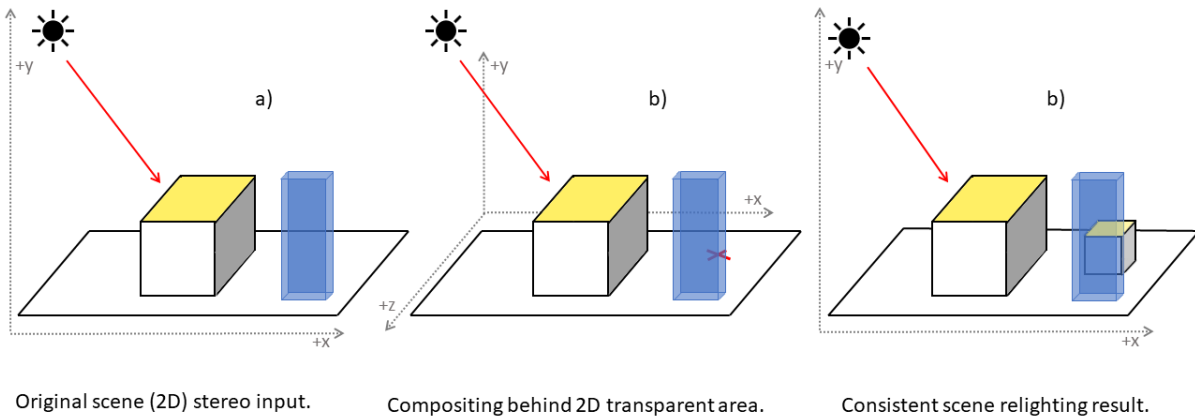


Fig. 6.2 Overview of compositing scenario in the presence of translucent media. a) The original 2D image depicting a light source, and two objects: one opaque and one translucent. b) The 3D counterpart scene from the initial image input. The red cross behind (along the  $z$  axis) the translucent surface represents the desired position of the artefact soon to-be composited. c) A plausible result of compositing a 3D artefact behind the initial translucent area from the 2D image with correct illumination.

# References

- [1] Abad, F., Camahort, E., and Vivó, R. (2003). Integrating synthetic objects into real scenes. *Computers and Graphics (Pergamon)*, 27(1):5–17.
- [2] Allisy-Roberts, P. and Williams, J. (2008). Chapter 1 - radiation physics. In Allisy-Roberts, P. and Williams, J., editors, *Farr's Physics for Medical Imaging (Second Edition)*, pages 1 – 21. W.B. Saunders, second edition edition.
- [3] Annen, T., Mertens, T., Seidel, H.-P., Flerackers, E., and Kautz, J. (2008). Exponential shadow maps. In *Proceedings of Graphics Interface 2008*, GI '08, pages 155–161, Toronto, Ont., Canada, Canada. Canadian Information Processing Society.
- [4] Apple Insider (2018). ARKit app on iPad. <https://www.wearable-technologies.com/2018/05/apple-could-release-an-ar-headset-in-2021-predicts-gene-munster/>.
- [5] Arvo, J. (1995). The role of functional analysis in global illumination. In *Rendering Techniques '95*, pages 115–126. Springer-Verlag.
- [6] Ashikhmin, M. and Shirley, P. (2000a). An anisotropic phong bidirectional reflectance distribution function model. *Journal of Graphics Tools*, 5(2):25–32.
- [7] Ashikhmin, M. and Shirley, P. (2000b). An anisotropic phong light reflection model. *Journal of Graphics Tools*.
- [8] Bagher, M. M., Soler, C., and Holzschuch, N. (2012). Accurate fitting of measured reflectances using a shifted gamma micro-facet distribution. *Comput. Graph. Forum*, 31(4):1509–1518.
- [9] Barron, J. and Malik, J. (2015). Shape, illumination, and reflectance from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37:1670–1687.
- [10] Barron, J. T. (2012). Shape, albedo, and illumination from a single image of an unknown object. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, pages 334–341, Washington, DC, USA. IEEE Computer Society.
- [11] Barron, J. T. and Malik, J. (2016). Intrinsic scene properties from a single rgb-d image. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(4):690–703.
- [12] Belhumeur, P. N., Kriegman, D. J., and Yuille, A. L. (1999). The bas-relief ambiguity. *Int. J. Comput. Vision*, 35(1):33–44.
- [13] Blinn, J. F. and Newell, M. E. (1976). Texture and reflection in computer generated images. *Commun. ACM*, 19(10):542–547.

- [14] Bohren, C. F. and Huffman, D. R. (1998a). *Absorption and scattering of light by small particles*. Wiley.
- [15] Bohren, C. F. and Huffman, D. R. (1998b). Absorption and scattering of light by small particles. Wiley Online Library.
- [16] Boivin, S. and Gagalowicz, A. (2002). Inverse rendering from a single image. In *Color in Graphics, Imaging, and Vision CGIV*.
- [17] Bonneel, N., Kovacs, B., Paris, S., and Bala, K. (2017). Intrinsic decompositions for image editing. *Comput. Graph. Forum*, 36:593–609.
- [18] Boom, B., Orts, S., Ning, X., McDonagh, S., Sandilands, P., and Fisher, R. (2017). Interactive light source position estimation for augmented reality with an RGB-D camera. *Journal of Visualization and Computer Animation*, 28.
- [19] Burley, B. (2015). Extending disney’s physically based brdf with integrated subsurface scattering.
- [20] Burley, B. and Walt Disney Animation Studios (2012). Physically-based shading at Disney.
- [21] Cabral, R. and Y., F. (2014). Piecewise planar and compact floorplan reconstruction from images.
- [22] Camplani, M. and Salgado, L. (2014). Background foreground segmentation with RGB-D Kinect data: An efficient combination of classifiers. *Journal of Visual Communication and Image Representation*, 2(1):122 – 136. Visual Understanding and Applications with RGB-D Cameras.
- [23] Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698.
- [24] Chakrabarti, A. and Sunkavalli, K. (2016). Single-image RGB Photometric Stereo With Spatially-varying Albedo. *ArXiv e-prints*.
- [25] Chandraker, M. and Ramamoorthi, R. (2011). What an image reveals about material reflectance. In *2011 International Conference on Computer Vision*, pages 1076–1083.
- [26] Chao, Y. W., Choi, W., Pantofaru, C., and Savarese, S. (2013). Layout estimation of highly cluttered indoor scenes using geometric and semantic cues. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8157 LNCS(PART 2):489–499.
- [27] Conway, B. and Livingstone, M. (2007). Perspectives on science and art. *Current Opinion in Neurobiology*, 17:476–482.
- [28] Corrigan, D. (2008). Video matting using motion extended grabcut. *IET Conference Proceedings*, pages 3–3(1).
- [29] Cossairt, O., Nayar, S., and Ramamoorthi, R. (2008). Light field transfer: Global illumination between real and synthetic objects. *ACM Trans. Graph.*, 27(3):57:1–57:6.

- [30] Coughlan, J. M. and Yuille, A. L. (1999). Manhattan world: Compass direction from a single image by bayesian inference. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, pages 941–, Washington, DC, USA. IEEE Computer Society.
- [31] Criminisi, A., Reid, I., and Zisserman, A. (2000). Single view metrology. *International Journal of Computer Vision*, 40(2):123–148.
- [32] Curless, B. and Levoy, M. (1996). A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 303–312, New York, NY, USA. ACM.
- [33] Dataset, I. (2019). <http://www.hdrlabs.com/sibl/archive.html>.
- [34] Debevec, P. (1998). Rendering Synthetic Objects into Real Scenes : Bridging Traditional and Image-based Graphics with Global Illumination and High Dynamic Range Photography. In *Computer Graphics Proceedings, Annual Conference Series (Proc. ACM SIGGRAPH '98 Proceeding)*, pages 189–198.
- [35] Debevec, P., Graham, P., Busch, J., and Bolas, M. (2012). A single-shot light probe. In *ACM SIGGRAPH 2012 Talks*, SIGGRAPH '12, pages 10:1–10:1, New York, NY, USA. ACM.
- [36] Debevec, P., Tchou, C., Gardner, A., Hawkins, T., Poullis, C., Stumpfel, J., Jones, A., Yun, N., Einarsson, P., Lundgren, T., Fajardo, M., and Martinez, P. (2004). Estimating Surface Reflectance Properties of a Complex Scene under Captured Natural Illumination. ICT Technical Report ICT TR 06 2004, University of Southern California Institute for Creative Technologies.
- [37] Delage, E., Lee, H., and Ng, A. Y. (2007). Automatic single-image 3d reconstructions of indoor manhattan world scenes. In Thrun, S., Brooks, R., and Durrant-Whyte, H., editors, *Robotics Research*, pages 305–321, Berlin, Heidelberg. Springer Berlin Heidelberg.
- [38] Dementhon, D. F. and Davis, L. S. (1995). Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15(1):123–141.
- [39] Donner, C. and Jensen, H. W. (2005). Light diffusion in multi-layered translucent materials. *ACM Trans. Graph.*, 24(3):1032–1039.
- [40] Drobot, M. (2009). Quadtree displacement mapping with height blending. *Game Developer Conference GDC*.
- [41] Drobot, M. and Micciulla, A. (2017). Practical multiplayered materials.
- [42] Dupuy, J., Heitz, E., and d'Eon, E. (2016). Additional progress towards the unification of microfacet and microflake theories. In *Proceedings of the Eurographics Symposium on Rendering: Experimental Ideas & Implementations*, EGSR '16, pages 55–63, Goslar Germany, Germany. Eurographics Association.
- [43] Engel, W. (2013). *GPU Pro 4: Advanced Rendering Techniques*.
- [44] Fitzpatrick, P. (2002). Indoor and outdoor scene classification project. Technical report.



- [45] G. Barrow, H. and M. Tenenbaum, J. (1978). Recovering intrinsic scene characteristics from images.
- [46] Gallup, D., Frahm, J. M., and Pollefeys, M. (2010). Piecewise planar and non-planar stereo for urban scene reconstruction. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1418–1425.
- [47] Gardner, M.-A., Sunkavalli, K., Yumer, E., Shen, X., Gambaretto, E., Gagné, C., and Lalonde, J.-F. (2017). Learning to predict indoor illumination from a single image. *ACM Trans. Graph.*, 36:176:1–176:14.
- [48] Glassner, A. S. (1994). *Principles of Digital Image Synthesis*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- [49] Glazer, A. M. (1988). Linear and circular birefringence and crystal structures. In Salje, E. K. H., editor, *Physical Properties and Thermodynamic Behaviour of Minerals*, pages 185–212. Springer.
- [50] Goldman, D. B., Curless, B., Hertzmann, A., and Seitz, S. M. (2010). Shape and spatially-varying brdfs from photometric stereo. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(6):1060–1071.
- [51] Gortler, S. J., Grzeszczuk, R., Szeliski, R., and Cohen, M. F. (1996). The lumigraph. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 43–54, New York, NY, USA. ACM.
- [52] Griffiths, D. J. (1998). Introduction to electrodynamics. In *Introduction to Electrodynamics*, pages 1 – 40. Prentice Hall, third edition edition.
- [53] Grosse, R., Johnson, M., Adelson, E., and Freeman, W. (2009). Ground truth dataset and baseline evaluations for intrinsic image algorithms. *ICCV Proceedings of the International Conference on Computer Vision*.
- [54] Gulbrandsen, O. (2014). Artist friendly metallic fresnel. *Journal of Computer Graphics Techniques (JCGT)*, 3(4):64–72.
- [55] Gutierrez, D., Seron, F. J., Lopez-Moreno, J., Sanchez, M. P., Fandos, J., and Reinhard, E. (2008). Depicting procedural caustics in single images. *ACM Trans. Graph.*, 27(5):120:1–120:9.
- [56] Hadamard, J. (1902). Sur les problemes aux derivees partielles et leur signification physique. *Princeton university bulletin*, pages 49–52.
- [57] Hasenfratz, J.-M., Lapierre, M., Holzschuch, N., Sillion, F. X., Sillion, F. X. A., Hasenfratz, J.-M., Lapierre, M., Holzschuch, N., and Sillion, F. (2003). A survey of Real-Time Soft Shadows Algorithms. 22(4):4–753.
- [58] Hedau, V., Hoiem, D., and Forsyth, D. (2009). Recovering the spatial layout of cluttered rooms. In *2009 IEEE 12th International Conference on Computer Vision, ICCV 2009*, pages 1849–1856.
- [59] Henry, P., Krainin, M., Herbst, E., Ren, X., and Fox, D. (2014). RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments. *Springer Tracts in Advanced Robotics*, 79:477–491.

- [60] Hill, S., McAuley, S., Conty, A., Drobot, M., Heitz, E., Hery, C., Kulla, C., Lanz, J., Ling, J., Walster, N., Xie, F., Micciulla, A., and Villemain, R. (2017). Physically based shading in theory and practice. In *ACM SIGGRAPH 2017 Courses*, SIGGRAPH '17, pages 7:1–7:8, New York, NY, USA. ACM.
- [61] Hirschmuller, H. (2008). Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341.
- [62] Hoiem, D., Efros, A. A., and Hebert, M. (2005). Automatic photo pop-up. *ACM Trans. Graph.*, 24(3):577–584.
- [63] Horn, B. K. P. (1989). Shape from shading. chapter Obtaining Shape from Shading Information, pages 123–171. MIT Press, Cambridge, MA, USA.
- [64] Horn, B. K. P. and Brooks, M. J., editors (1989). *Shape from Shading*. MIT Press, Cambridge, MA, USA.
- [65] Horry, Y., Anjyo, K.-I., and Arai, K. (1997). Tour into the picture: Using a spidery mesh interface to make animation from a single image. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, pages 225–232, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.
- [66] Ikehata, S., Yang, H., and Furukawa, Y. (2015). Structured indoor modeling. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 1323–1331, Washington, DC, USA. IEEE Computer Society.
- [67] Izadi, S., Davison, A., Fitzgibbon, A., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., and Freeman, D. (2011). Kinect Fusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera. *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11*, page 559.
- [68] Jensen, H. W. and Buhler, J. (2002). A rapid hierarchical rendering technique for translucent materials. *ACM Trans. Graph.*, 21(3):576–581.
- [69] Jensen, H. W., Marschner, S. R., Levoy, M., and Hanrahan, P. (2001). A practical model for subsurface light transport. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 511–518, New York, NY, USA. ACM.
- [70] Johnson, M. K. and Farid, H. (2005). Exposing digital forgeries by detecting inconsistencies in lighting. In *Proceedings of the 7th Workshop on Multimedia and Security*, pages 1–10, New York, NY, USA. ACM.
- [71] Johnson, M. K. and Farid, H. (2007). Exposing digital forgeries in complex lighting environments. *Trans. Info. For. Sec.*, 2(3):450–461.
- [72] Kabanikhin, S., Tikhonov, N., K Ivanov, V., and M Lavrentiev, M. (2008). Definitions and examples of inverse and ill-posed problems. 16:317–357.
- [73] Kajiya, J. T. (1986). The rendering equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '86, pages 143–150, New York, NY, USA. ACM.

- [74] Kang, Y. and Ho, Y. (2011). An efficient image rectification method for parallel multi-camera arrangement. *IEEE Transactions on Consumer Electronics*, 57(3):1041–1048.
- [75] Karis, B. (2013). Real shading in unreal engine 4. *SIGGRAPH 2013 Key Note*.
- [76] Karsch, K. and Forsyth, D. (2014). Blind recovery of spatially varying reflectance from a single image. In *SIGGRAPH Asia 2014 Indoor Scene Understanding Where Graphics Meets Vision*, SA '14, pages 2:1–2:10, New York, NY, USA. ACM.
- [77] Karsch, K., Hedau, V., Forsyth, D., and Hoiem, D. (2011). Rendering synthetic objects into legacy photographs. *ACM Transactions on Graphics*, 30(6):1.
- [78] Karsch, K., Liu, C., and Kang, S. B. (2012). Depth extraction from video using non-parametric sampling. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part V*, ECCV'12, pages 775–788, Berlin, Heidelberg. Springer-Verlag.
- [79] Karsch, K., Sunkavalli, K., Hadap, S., Carr, N., Jin, H., Fonte, R., Sittig, M., and Forsyth, D. (2014). Automatic Scene Inference for 3D Object Compositing. *ACM Transactions on Graphics*, 33(3):1–15.
- [80] Kazhdan, M. and Hoppe, H. (2013). Screened poisson surface reconstruction. *ACM Trans. Graph.*, 32(3):29:1–29:13.
- [81] Kee, E. and Farid, H. (2010). Exposing digital forgeries from 3-d lighting environments. *2010 IEEE International Workshop on Information Forensics and Security*, pages 1–6.
- [82] Kelemen, C. and Szirmay-Kalos, L. (2001). A Microfacet Based Coupled Specular-Matte BRDF Model with Importance Sampling. pages 576–581.
- [83] Khan, E. A., Reinhard, E., Fleming, R. W., and Bühlhoff, H. H. (2006). Image-based material editing. *ACM Trans. Graph.*, 25(3):654–663.
- [84] Kholgade, N., Simon, T., Efros, A., and Sheikh, Y. (2014a). 3d object manipulation in a single photograph using stock 3d models. *ACM Trans. Graph.*, 33(4):127:1–127:12.
- [85] Kholgade, N., Simon, T., Efros, A., and Sheikh, Y. (2014b). 3d object manipulation in a single photograph using stock 3d models. *ACM Transactions on Computer Graphics*, 33(4).
- [86] Koenderink, J. J., Pont, S. C., van Doorn, A. J., Kappers, A. M. L., and Todd, J. T. (2007). The visual light field. *Perception*, 36(11):1595–1610. PMID: 18265841.
- [87] Kottas, D. G., Hesch, J. A., Bowman, S. L., and Roumeliotis, S. I. (2012). On the consistency of vision-aided inertial navigation. In *ISER*.
- [88] Křivánek, J., Fajardo, M., Christensen, P. H., Tabellion, E., Bunnell, M., Larsson, D., and Kaplanyan, A. (2010). Global illumination across industries. In *ACM SIGGRAPH 2010 Courses*, SIGGRAPH '10, New York, NY, USA. ACM.
- [89] Kurt, M., Szirmay-Kalos, L., and Křivánek, J. (2010). An anisotropic brdf model for fitting and monte carlo rendering. *SIGGRAPH Comput. Graph.*, 44(1):3:1–3:15.
- [90] Lagarde, S. (2012). Spherical Gaussian approximation for Blinn-Phong, Phong, and Fresnel. <https://seblagarde.wordpress.com/2012/06/03/spherical-gaussian-approximation-for-blinn-phong-phong-and-fresnel/>.

- [91] Lalonde, J.-F. and Efros, A. (2010). Synthesizing environment maps from a single image.
- [92] Lalonde, J.-F., Efros, A. A., and Narasimhan, S. G. (2012). Estimating the natural illumination conditions from a single outdoor image. *Int. J. Comput. Vision*, 98(2):123–145.
- [93] Lambert, J. H. (1760). *Photometria sive de mensura de gratibus luminis, colorum umbrae*. Eberhard Klett.
- [94] Land, E. H. and McCann, J. J. (1971). Lightness and retinex theory. *J. Opt. Soc. Am.*, 61(1):1–11.
- [95] Lee, D. C., Hebert, M., and Kanade, T. (2009). Geometric reasoning for single image structure recovery. *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, 2009 IEEE:2136–2143.
- [96] Li, S. W. and Shi, B. (2015). Photometric stereo for general isotropic reflectances by spherical linear interpolation.
- [97] Liao, Z., Karsch, K., and Forsyth, D. (2015). An approximate shading model for object relighting. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5307–5314.
- [98] Liebowitz, D. and Zisserman, A. (1998). Metric rectification for perspective images of planes. In *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No.98CB36231)*, pages 482–488.
- [99] Lindeberg, T. (1994). Edge detection. *Encyclopedia of Mathematics*.
- [100] Lombardi, S. and Nishino, K. (2016a). Radiometric Scene Decomposition: Scene Reflectance, Illumination, and Geometry from RGB-D Images. *2016 Fourth International Conference on 3D Vision (3DV)*, pages 305–313.
- [101] Lombardi, S. and Nishino, K. (2016b). Reflectance and illumination recovery in the wild. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(1):129–141.
- [102] Loop, C. and Zhang, Z. (1999). Computing rectifying homographies for stereo vision. Technical Report MSR-TR-99-21.
- [103] Löw, J., Kronander, J., Ynnerman, A., and Unger, J. (2012). Brdf models for accurate and efficient rendering of glossy surfaces. *ACM Trans. Graph.*, 31(1):9:1–9:14.
- [104] Magic Leap (2019). Magicleap see signal. <https://www.magicleap.com/news/partner-stories/bring-mobile-data-to-life-with-seesignal-from-badvr>.
- [105] Marschner, S. (1998). *Inverse Rendering for Computer Graphics*. PhD thesis, Ithaca, NY, USA. AAI9839924.
- [106] Matusik, W., Pfister, H., Brand, M., and McMillan, L. (2003). A data-driven reflectance model. *ACM Trans. Graph.*, 22(3):759–769.
- [107] Michl, J. (1999). Optical spectroscopy, linear polarization theory. In *Encyclopedia of Spectroscopy and Spectrometry*, pages 1701–1716. Elsevier.

- [108] Microsoft (2019). Microsoft HoloLens 2. <https://www.microsoft.com/en-us/hololens/developers>.
- [109] Miller, G. S. and Hoffman, C. R. (1984). Illumination and reflection maps: Simulated objects in simulated and real environments. *SIGGRAPH*, Course Notes for Advanced Computer Graphics Animation(10).
- [110] Moreno, J. L., Hadap, S., Reinhard, E., and Gutierrez, D. (2010). Compositing images through light source detection. *Computers and Graphics*, 34(6):698–707. Graphics for Serious Games Computer Graphics in Spain: a Selection of Papers from CEIG 2009 Selected Papers from the SIGGRAPH Asia Education Program.
- [111] Nakamae, E., Harada, K., Ishizaki, T., and Nishita, T. (1986). A montage method: The overlaying of the computer generated images onto a background photograph. *SIGGRAPH Comput. Graph.*, 20(4):207–214.
- [112] Nguyen, R. M. H. and Le, M. N. (2012). Light source estimation from a single image. In *2012 12th International Conference on Control Automation Robotics Vision (ICARCV)*, pages 1358–1363.
- [113] Niessner, M., Zollhöfer, M., Izadi, S., and Stamminger, M. (2013). Real-time 3d reconstruction at scale using voxel hashing. *ACM Trans. Graph.*, 32(6):169:1–169:11.
- [114] Nishino, K. and Nayar, S. K. (2004). Eyes for relighting. *ACM Trans. Graph.*, 23(3):704–711.
- [115] Oh, B. M. (2002). *A System for Image-based Modeling and Photo Editing*. PhD thesis, Cambridge, MA, USA. AAI0804146.
- [116] Oh, B. M., Chen, M., Dorsey, J., and Durand, F. (2001). Image-based modeling and photo editing. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 433–442, New York, NY, USA. ACM.
- [117] Ostrovsky, Y., Cavanagh, P., and Sinha, P. (2005). Perceiving illumination inconsistencies in scenes. *Perception*, 34(11):1301–1314. PMID: 16358419.
- [118] Panagopoulos, A., Wang, C., Samaras, D., and Paragios, N. (2011). Illumination estimation and cast shadow detection through a higher-order graphical model. In *CVPR 2011*, pages 673–680.
- [119] Patow, G. and Pueyo, X. (2003). A survey of inverse rendering problems. *Computer Graphics Forum*, 22(4):663–687.
- [120] Payne, A. and Singh, S. (2005). Indoor vs. outdoor scene classification in digital photographs. *Pattern Recogn.*, 38(10):1533–1545.
- [121] Peers, P. and Dutré, P. (2003). Wavelet environment matting. In *Proceedings of the 14th Eurographics Workshop on Rendering*, EGRW '03, pages 157–166, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.
- [122] Pellacini, F. (2010). envylight: An interface for editing natural illumination. *ACM Trans. Graph.*, 29(4):34:1–34:8.
- [123] Perez, R., Seals, R., and Michalsky, J. (1993). All-weather model for sky luminance distribution—preliminary configuration and validation". *Solar Energy*, 50(3):235 – 245.

- [124] Pero, L. D., Bowdish, J., Kermgard, B., Hartley, E., and Barnard, K. (2013). Understanding bayesian rooms using composite 3d object models. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 153–160.
- [125] Press, W., Teukolsky, S., Vetterling, W., and B.P., F. (1992). *Numerical Recipes in C, The Art of Scientific Computing*. Cambridge University Press.
- [126] Ramamoorthi, R. and Hanrahan, P. (2001a). On the relationship between radiance and irradiance: determining the illumination from images of a convex lambertian object. *J. Opt. Soc. Am. A*, 18(10):2448–2459.
- [127] Ramamoorthi, R. and Hanrahan, P. (2001b). A signal-processing framework for inverse rendering. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 117–128, New York, NY, USA. ACM.
- [128] Ramanarayanan, G., Ferwerda, J., Walter, B., and Bala, K. (2007). Visual equivalence: Towards a new standard for image fidelity. *ACM Trans. Graph.*, 26(3).
- [129] Rashid, M. and Hebert, M. (2014). Detailed 3d model driven single view scene understanding. In *2014 2nd International Conference on 3D Vision*, volume 1, pages 139–146.
- [130] Reinhard, E., Akyuz, A. O., and Colbert, M. (2004). Real-time color blending of rendered and captured video.
- [131] Sato, I., Sato, Y., and Ikeuchi, K. (2001). Modelling from reality. chapter Acquiring a Radiance Distribution to Superimpose Virtual Objects Onto a Real Scene, pages 137–160. Kluwer Academic Publishers, Norwell, MA, USA.
- [132] Scharstein, D. and Szeliski, R. (2002). A Taxonomy and Evaluation of Dense Two-Frame Stereo. *International Journal of Computer Vision*, 47(1):7–42.
- [133] Schlick, C. (1994). An inexpensive brdf model for physically-based rendering. *Computer Graphics Forum*, 13(3):233–246.
- [134] Schwartz, G. and Nishino, K. (2018). Recognizing Material Properties from Images.
- [135] Schwarz, M. and Stamminger, M. (2007). Bitmask Soft Shadows. Technical Report 3.
- [136] Schwing, A. G. and Urtasun, R. (2012). Efficient exact inference for 3d indoor scene understanding. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part VI*, ECCV'12, pages 299–313, Berlin, Heidelberg. Springer-Verlag.
- [137] Seitz, S. M. and Kim, J. (2002). The space of all stereo images. *Int. J. Comput. Vision*, 48(1):21–38.
- [138] Shirley, P., Smits, B., Hu, H., and Lafortune, E. (1997). A practitioners' assessment of light reflection models. In *Proceedings The Fifth Pacific Conference on Computer Graphics and Applications*, pages 40–49.
- [139] Sinha, P. and Adelson, E. (1993). Recovering reflectance and illumination in a world of painted polyhedra. In *1993 (4th) International Conference on Computer Vision*, pages 156–163.

- [140] Slater, M., Usoh, M., and Chrysanthou, Y. (1995). The influence of dynamic shadows on presence in immersive virtual environments. *Selected papers of the Eurographics workshops on Virtual environments*, 95:8–21.
- [141] Smith, A. R. and Blinn, J. F. (1996). Blue screen matting. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 259–268, New York, NY, USA. ACM.
- [142] Stauder, J. (2000). Point light source estimation from two images and its limits. *International Journal of Computer Vision*, 36(3):195–220.
- [143] Szeliski, R. and Shum, H.-Y. (1997). Creating full view panoramic image mosaics and environment maps. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, pages 251–258, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.
- [144] Tabellion, E. and Lamorlette, A. (2004). An approximate global illumination system for computer generated films. *ACM Trans. Graph.*, 23(3):469–476.
- [145] Tappen, M. F., Adelson, E. H., and Freeman, W. T. (2006). Estimating intrinsic component images using non-linear regression. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1992–1999.
- [146] Tappen, M. F., Freeman, W. T., and Adelson, E. H. (2005). Recovering intrinsic images from a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(9):1459–1472.
- [147] Thevenaz, P., Ruttimann, U. E., and Unser, M. (1998). A pyramid approach to subpixel registration based on intensity. *IEEE Transactions on Image Processing*, 7(1):27–41.
- [148] Tipler, P. and Mosca, G. (2004). Volume 1: Mechanics, oscillations and waves, thermodynamics. In *Physics for Scientists and Engineers*, pages 470 – 480. Macmillan, first edition edition.
- [149] Torrance, K. and Sparrow, E. M. (1967). Theory for Off-Specular Reflection From Roughened Surfaces. volume 57. *Journal of the Optical Society of America*.
- [150] Torrance, K. E. and Sparrow, E. M. (1967). Theory for Off-Specular Reflection From Roughened Surfaces. *Journal of the Optical Society of America*, 57(1105-1114):10.
- [151] Torrence, A. (2006). Martin newell's original teapot. In *ACM SIGGRAPH 2006 Teapot Copyright Restrictions Prevent ACM from Providing the Full Text for the Teapot Exhibits*, SIGGRAPH '06, New York, NY, USA. ACM.
- [152] Töral, . A., Ergun, S., Kurt, M., and Öztürk, A. (2014). Mobile GPU-based importance sampling. *2014 22nd Signal Processing and Communications Applications Conference (SIU)*, pages 510–513.
- [153] Tsesmelis, T., Hasan, I., Cristani, M., Bue, A. D., and Galasso, F. (2017). Lit: A system and benchmark for light understanding. *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 2953–2960.

- [154] Turk, G. and Levoy, M. (1994). Zippered polygon meshes from range images. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '94, pages 311–318, New York, NY, USA. ACM.
- [155] Unger, J., Kronander, J., Larsson, P., Gustavson, S., Löw, J., and Ynnerman, A. (2013). Spatially varying image based lighting using hdr-video. *Computers and Graphics*, 37(7):923 – 934.
- [156] Unger, J., Wenger, A., Hawkins, T., Gardner, A., and Debevec, P. (2003). Capturing and rendering with incident light fields. In *Proceedings of the 14th Eurographics Workshop on Rendering*, EGRW '03, pages 141–149, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.
- [157] Veksler, O. (2005). Stereo correspondence by dynamic programming on a tree. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 384–390 vol. 2.
- [158] Walter, B., Marschner, S. R., Li, H., and Torrance, K. E. (2007). Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, EGSR'07, pages 195–206, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.
- [159] Wang, J. and Cohen, M. F. (2007). Simultaneous matting and compositing. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.
- [160] Ward, G. J. (1992). Measuring and modeling anisotropic reflection. *SIGGRAPH Comput. Graph.*, 26(2):265–272.
- [161] Weber, M. and Cipolla, R. (2001). A practical method for estimation of point light-sources. *British Machine Vision Conference*, 2:471–480.
- [162] Weiss, Y. (2001). Deriving intrinsic images from image sequences. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 68–75 vol.2.
- [163] Wexler, Y., Fitzgibbon, A. W., and Zisserman, A. (2002). Image-based environment matting. In *Proceedings of the 13th Eurographics Workshop on Rendering*, EGRW '02, pages 279–290, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.
- [164] Wood, D. N., Azuma, D. I., Aldinger, K., Curless, B., Duchamp, T., Salesin, D. H., and Stuetzle, W. (2000). Surface light fields for 3d photography. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 287–296, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.
- [165] Woodham, R. J. (1989). Shape from shading. chapter Photometric Method for Determining Surface Orientation from Multiple Images, pages 513–531. MIT Press, Cambridge, MA, USA.
- [166] Wright, S. (2006). *Digital Compositing for Film and Video*. Number v. 10 in Digital compositing for film and video. Focal Press.
- [167] Wu, C., Zollhofer, M., Niessner, M., Stamminger, M., Izadi, S., and Theobalt, C. (2014). Real-time shading-based refinement for consumer depth cameras. *ACM Trans. Graph.*, 33(6):200:1–200:10.



- [168] Wu, J.-H. and Saito, S. (2017). Interactive relighting in single low-dynamic range images. *ACM Trans. Graph.*, 36.
- [169] Xu, S. and Wallace, A. (2008). Recovering surface reflectance and multiple light locations and intensities from image data. *Pattern Recognition Letters*, 29(11):1639 – 1647.
- [170] Yeung, S.-K., Tang, C.-K., Brown, M. S., and Kang, S. B. (2011). Matting and compositing of transparent and refractive objects. *ACM Trans. Graph.*, 30(1):2:1–2:13.
- [171] Yu, Y., Debevec, P., Malik, J., and Hawkins, T. (1999). Inverse global illumination: Recovering reflectance models of real scenes from photographs. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '99*, pages 215–224, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.
- [172] Zhang, E., Cohen, M. F., and Curless, B. (2016). Emptying, refurnishing, and relighting indoor spaces. *ACM Transactions on Graphics*, 35(6):1–14.
- [173] Zollhofer, M., Dai, A., Innmann, M., Wu, C., Stamminger, M., Theobalt, C., and Niessner, M. (2015). Shading-based refinement on volumetric signed distance functions. *ACM Trans. Graph.*, 34(4):96:1–96:14.
- [174] Zongker, D. E., Werner, D. M., Curless, B., and Salesin, D. H. (1999). Environment matting and compositing. In *Proceedings of ACM SIGGRAPH 99*, Computer Graphics Proceedings, Annual Conference Series, pages 205–214. ACM Press / ACM SIGGRAPH / Addison Wesley Logman.
- [175] Zwicker, M., Pauly, M., Knoll, O., and Gross, M. (2002). Pointshop 3d: An interactive system for point-based surface editing. *ACM Trans. Graph.*, 21(3):322–329.

# Appendix A

## Full listing of supplemental materials

### A.1 Website resources

1. Microsoft HoloLens  
<https://www.microsoft.com/en-us/hololens>
2. Unity3D Engine  
<https://unity3d.com/>
3. Epic's Unreal Engine  
<https://www.unrealengine.com/en-US/what-is-unreal-engine-4>
4. LuxCoreRender  
<https://github.com/LuxCoreRender>
5. Middlebury dataset  
<http://vision.middlebury.edu/stereo/data/>
6. Apple ARKit 2  
<https://developer.apple.com/arkit/>
7. Google ARCore  
<https://developers.google.com/ar/>
8. Samsung S8  
<https://www.samsung.com/uk/smartphones/galaxy-s8/spec-plus/>
9. Sketchfab - free 3D models  
<https://sketchfab.com/>
10. sIBL Archive - free HDR image database  
<http://www.hdrlabs.com/sibl/archive.html>

## A.2 Hardware specification

Table A.1 Specification for the hardware used in this framework.

Device name	Device Type	Year released	Specification details
GeForce GTX 770	Nvidia GPU	2013	7 Gbps memory speed GDDR5, 225 GB/s band
Core i7-4790K	Intel CPU	2013	4 Cores, L2: $4 \times 256$ KiB, L3: 8 MiB, 4GHz
Samsung S8 Rear Camera	Mobile Camera	2016	Dual Pixel 12MP AF, Pixel size: $1.4\mu\text{m}$ RAW, HDR, Sensor size: $1/2.55''$ FOV: 77, F1.7 aperture

## A.3 Libraries, APIs, 3rd-party software

Table A.2 APIs and external library dependencies in our framework

Name	Type	Location	Usage details
OpenGL 4.5	API	ready with GPU and CPU	cross-platform language to interface with the graphics hardware. Transfers resources from CPU to GPU.
GLSL	API	ready with GPU and CPU	OpenGL shading language for writing shader programs.
Intel MKL	Library	<a href="http://www.software.intel.com/mkl">www.software.intel.com/mkl</a>	Intel's Math Kernel Library, used for matrix operations (pixel ops).
QT	Library	<a href="http://www.qt.io">www.qt.io</a>	Cross-platform library for UI creation.

### A.3.1 Compatibility

The libraries and 3<sup>rd</sup> party software listed in Table A.2 were built for x64 devices in both Debug and Release for a Windows 10 platform. All libraries listed are cross-platform, compatible with newer versions of most popular OS': Linux Ubuntu, Linux Fedora, Linux Arch, iOS, Windows 8.1, Windows 10.

### A.3.2 Development tools

Table A.3 holds a detailed description of all the tools used in the creation of the framework presented in this thesis. These tools include programming languages, development environments, and generally required information for ensuring that our framework can be ran and used, and that test results can be reproduced easily.

Table A.3 Details regarding all the development tools required in the creation of the framework described in the present document.

Name	Type	Location	Usage details
Visual Studio 2017	IDE	<a href="https://visualstudio.microsoft.com/">visualstudio.microsoft.com/</a>	Microsoft IDE
Visual Studio Code	Text Editor	<a href="https://code.visualstudio.com/">code.visualstudio.com/</a>	Microsoft Editor
Nvidia Nsight	Plugin	<a href="https://developer.nvidia.com/">developer.nvidia.com/</a>	Debug Analysis Profiler VS Integration
Render Doc	Plugin	<a href="https://renderdoc.org/">renderdoc.org/</a>	External Frame Debugger
vcpkg	Build Mgr	<a href="https://github.com/Microsoft/vcpkg">github.com/Microsoft/vcpkg</a>	Scripts for managing global dependencies
glsl validator	Plugin	<a href="https://github.com/KhronosGroup/glslang">github.com/KhronosGroup/glslang</a>	Scripts for managing GLSL shader validation
gltf	Standard	<a href="https://github.com/KhronosGroup/glTF">github.com/KhronosGroup/glTF</a>	Implemented in our framework for asset loading
cmake	Build Tool	<a href="https://cmake.org/">cmake.org/</a>	Command line tool and VS plugin
Jenkins	C.I.	<a href="https://jenkins.io/">jenkins.io/</a>	Continuous Integration and build
GitHub	Source Mgr	<a href="https://github.com/">github.com/</a>	Source hosting

The Results Section of this work relies on a number of open-source models that are ubiquitous in Computer Graphics, and the author wishes to acknowledge their original provenance: the Stanford Bunny [154], the Newell Teapot [151], and the Stanford Dragon [32].

# Appendix B

## Supplemental code

The present appendix section contains a number of algorithms developed specifically for the framework presented in the Thesis. The details of implementation are outside the scope of the Methodology section, however, they can offer insight into the approach, specifically where the hooks between the CPU and GPU work reside.

During each stage of the framework, data is being read or written to by both CPU and GPU through shader programs and indirect draw calls (Figure B.1). With the exception of the transform feedback stage, the general execution flow commences at the CPU level, which issues commands to the GPU. These commands are executed every frame. In this framework, the transform feedback (round trip CPU - indirect draw to GPU then write back to CPU) is ran a single time at the beginning of the pipeline, and never again until the state of the scene is modified (a new scene is loaded).

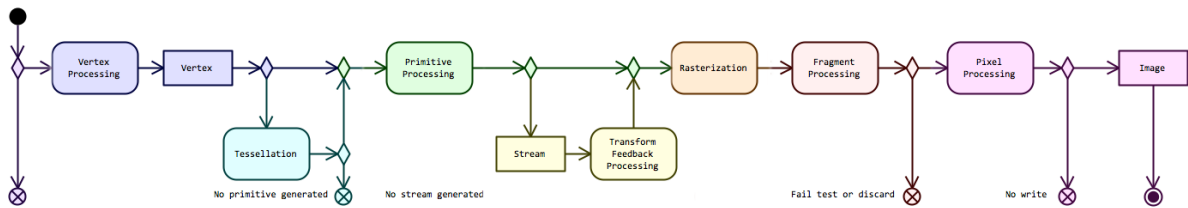


Fig. B.1 Overview of the modern OpenGL pipeline states. From left to right: geometric data along with properties are specified and a draw call is issued from the CPU. The Vertex data is processed through a vertex shader program invocation. An optional tessellation shader program stage can be invoked to amplify existing vertex data. Vertices are assembled as primitives, the type of which was specified in the initial draw call. Transform feedback can be ran on existing data, the result of which is written and offloaded back on the client or host device (CPU). The TF stage is optional, and if not enabled, the primitives are rasterized into fragments, which are then shaded by a pixel (fragment) shader program.

## B.1 Intrinsic scene representation

After the depth map is obtained (as discussed in the Methodology chapter), the mesh and its underlying topology have to be created. In order to achieve this, the mesh undergoes a number of transformations that create the point cloud, add connectivity in order to account for non-watertight geometry, and lastly, calculate the depth of each constituent vertex in the mesh.

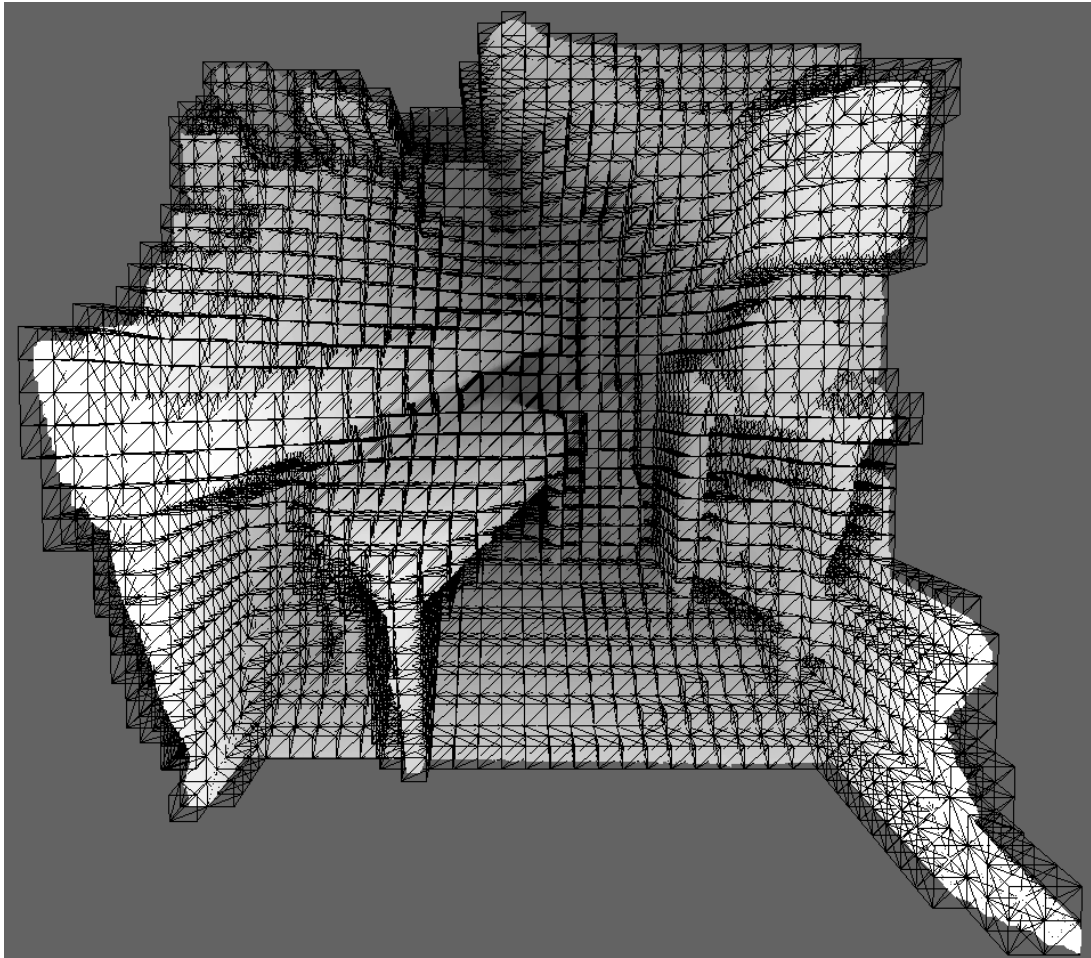


Fig. B.2 This intermediate caption comes from the processing of the stereo data set. It represents the dense reconstruction stored in a tree-like bounding volume. This allows easy access to any of the scene's regions for quick occlusion checks that take place at later stages in the pipeline.

As a secondary aspect of the scene representation, the mesh is stored in a spatial partitioning data structure, namely a derived octree of 6 levels, as seen in Figure B.2. We serialize and build the results of the octree progressively from level 1 all the way to level 8, but rely on using the middle LOD representation for most operations.

One of the operations which makes octree a bespoke spatial partitioning data structure is the ability to check whether a node (*octant*) is in shadow or not. This requires  $O(1)$  for

**Algorithm 4** Dense point cloud reconstruction

---

```

1:  $width, height \geq 0$ ;
2: function GENERATE POINT LIST( $width, height$ )
3:   Mesh  $m.primitiveType \leftarrow POINTS$ 
4:    $m.numVertices = width \cdot height$ 
5:   allocate memory for vertices, texture coordinates, colour
6:   for  $y \in [0 \rightarrow height]$ ,  $y \leftarrow y + 1$  do
7:     for  $x \in [0 \rightarrow width]$ ,  $x \leftarrow x + 1$  do
8:        $index \in (y \cdot width) + x$ 
9:        $m.vertices \leftarrow Vec3(-1.0 + ((x/width) \cdot 2.0), -1.0 + ((y/height) \cdot 2.0), 0.0)$ 
10:       $m.texCoords \leftarrow Vec2(x/width, y/height)$ 
11:     end for
12:   end for
13:   return  $m$ 
14: end function
15: function GENERATE MESH TOPOLOGY( $pointCloud, width, height$ )
16:    $numVertices \leftarrow width \cdot height$ 
17:    $numIndices \leftarrow (width - 1) \cdot (height - 1) * 6$ 
18:   allocate memory for vertices, texture coordinates, indices
19:   for  $z \in [0 \rightarrow height]$   $z \leftarrow z + 1$  do
20:     for  $x \in [0 \rightarrow width]$   $x \leftarrow x + 1$  do
21:        $offset \leftarrow (z \cdot width) + x$ 
22:        $vertices[offset] \leftarrow pointCloud.vertices[offset]$ 
23:     end for
24:   end for
25:    $numIndices \leftarrow 0$ 
26:   for  $z \in [0 \rightarrow height - 1]$   $z \leftarrow z + 1$  do
27:     for  $x \in [0 \rightarrow width - 1]$   $x \leftarrow x + 1$  do
28:        $a \leftarrow (z \cdot width) + x$ 
29:        $b \leftarrow ((z + 1) \cdot width) + x$ 
30:        $c \leftarrow ((z + 1) \cdot width + x + 1)$ 
31:        $d \leftarrow (z \cdot width) + x + 1$ 
32:        $indices[numIndices++] \leftarrow c, b, a, a, d, c$ 
33:     end for
34:   end for
35:   calculate normals and tangents for each vertex in the topological mesh
36:   allocate data on GPU DRAM cache
37: end function

```

---

checking if it is in shadow or not and  $O(N)$  for lookup where  $N$  is the total number of octants in the tree. At most, the total number of octants cannot exceed the total number of pixels (width \* height of image).

## B.2 Shadow masking

Recovering a shadow mask from a single RGB image is a difficult problem, due to complex interactions of geometry, albedo, and illumination. Many techniques have been proposed over the years, but shadow detection still remains an extremely challenging problem. In the present framework, the image is converted to the luminance space, then based on the histogram of this colour space, the contrast is increased. After computing the full average at the Y channel, a sliding window operation is performed, to decide which pixels belongs to the shadow. The noise is then reduced by applying a median filter.



Fig. B.3 Overview of shadow masking. Left column represents the input image from the stereo image pair. The second column represents the binary mask resulted from our framework. The first row originates from the stereo dataset, while the second row comes from the synthetic dataset. Although not exact, in both cases the light sources have been masked out, as well as the surfaces with highest reflectivity, which is the bare minimum information required to guide the algorithm for a rough estimation shadow presence which leads to a more robust illumination model inference overall. It is noteworthy that previous compositing techniques do not take this into account.



The algorithm identifies the minimum intensities areas in the picture, and this can describe darker objects too (false detection). For such situations, it can further reason if about the presence of a shadow by taking into consideration that its border is smooth (the border is usually rough for objects). The result of the shadow detection is a binary shadow mask suitable for the later stages of the pipeline (Algorithm 3).

---

**Algorithm 5** Binary shadow masking
 

---

```

    height, width, step, channels, stepMask  $\leftarrow$  0
    data, dataFiltered, dataMask, dataGray, dataHist  $\leftarrow$  nullptr
    data  $\leftarrow$  imageMatrix.ReadFile(filePath)
4: orig  $\leftarrow$  dataGray[step · height]
    dataFiltered  $\leftarrow$  medianBlur(orig)
    edge  $\leftarrow$  cannyEdgeDetection(dataFiltered)
    for j  $\in$  [0, height] j  $\leftarrow$  j + 1 do
8:     for i  $\in$  [0, width] i  $\leftarrow$  i + 1 do
        dataMask[j · step + i]  $\leftarrow$  0
    end for
    end for
12: fullAverage, fullCount, average, count, maxA, maxB  $\leftarrow$  0
    maxWindow  $\leftarrow$  31
    for j  $\in$  [0, height] j  $\leftarrow$  j + 1 do
        for i  $\in$  [0, width] i  $\leftarrow$  i + 1 do
16:     orig[j · step].y  $\leftarrow$  data[j · step + i · channels + 1] + data[j · step + i · channels + 2]
        orig[j · step].u  $\leftarrow$  data[j · step + i · channels + 1] + data[j · step + i · channels + 2]
        orig[j · step].v  $\leftarrow$  data[j · step + i · channels + 1] + data[j · step + i · channels + 2]
        dataHist[j · stepMask + i]  $\leftarrow$  orig[j · step + i].y
20:     end for
    end for
    dataHist  $\leftarrow$  calculateHistogram(height, width, fullAverage, fullCount, orig)
    fullAverage  $\leftarrow$  fullAverage / fullCount
24: average  $\leftarrow$  average / count
    dataMask  $\leftarrow$  detectShadow(dataHist, stepMask, channels, width, height, orig, average,
    fullAverage)
    return dataMask
  
```

---

## B.3 Additional depth reconstruction



Fig. B.4 Additional results for reconstruction. a) Middlebury Cable. b) Middlebury Adirandok. c) Middlebury Plants. d) Middlebury Backpack. e) Middlebury Motorcycle. f) Middlebury Bicycle.

# Appendix C

## Supplemental results

### C.1 Illumination setup across datasets

During the relighting stage of the framework, a number of steps are taken in order to capture the radiance and irradiance terms, and ensure that the initial input image can be converted to a suitable environment map. Using a parameterized material appearance model, the composited objects can take properties specific to metallic, rough, soft, intensely reflective or dull surfaces.

In figure C.1 the sphere grid in each figure illustrates material appearance under different light intensities, roughness, and glossiness. These are the possible appearances that a composited object can inherit.

The framework relies on the GLTF 2.0 standard A.2 for loading meshes along with their corresponding material definition (textures and shader programs if available), however, it discards the specified shader program in favour of the framework’s general material representation. The only data used is the bare GLTF file for creating the mesh, and the path to the textures.

Moreover, the implementation based on the GLTF format brings the additional benefit of supporting a wide range of texture types, ranging from common formats for diffuse maps (.PNG or .JPG), to HDR environmental maps (.DDS, .KTX, .HDR).

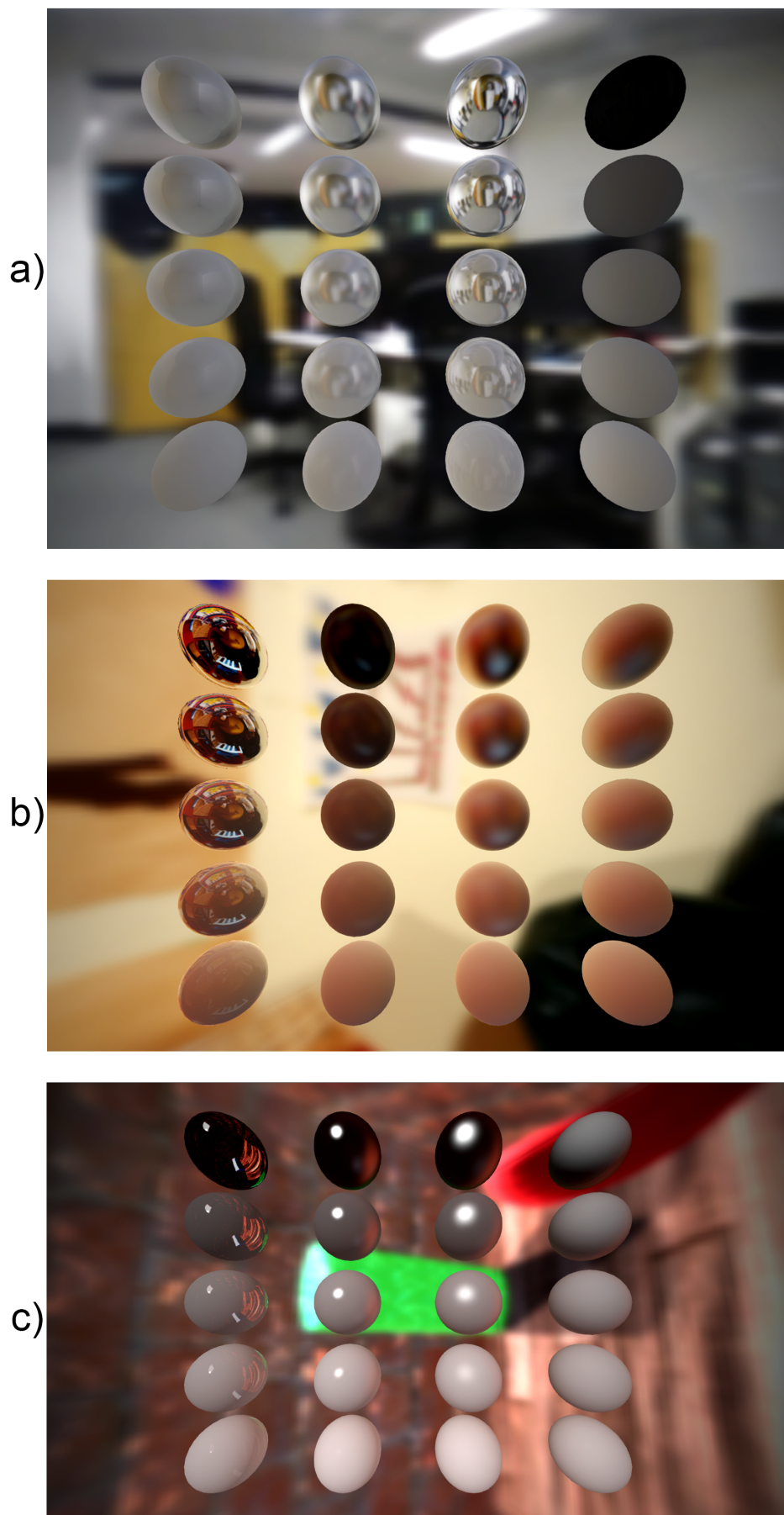


Fig. C.1 Recovered illumination and material appearance in the presented framework. a) The environment map originates from the near-stereo office dataset. b) The environment map is the cluttered playroom from the stereo dataset. c) Final row presents an environment map captured from the synthetic dataset.