



Masterarbeit

Mobile Reflectance Estimation

Universität Ulm

Fakultät für Ingenieurwissenschaften und Informatik

Institut für Medieninformatik

Tobias Häußler, tobias.haeussler@uni-ulm.de, 24. April 2012

Erstgutachter: Prof. Dr. Hendrik Lensch, Universität Ulm

Zweitgutachter: Prof. Dr. Heiko Neumann, Universität Ulm

Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Arbeit selbständig, ohne fremde Hilfe und ohne Benutzung anderer als der von mir angegebenen Quellen angefertigt zu haben. Die Arbeit wurde noch keiner Prüfungsbehörde in gleicher oder ähnlicher Form vorgelegt.

Tobias Häußler (Matrikelnummer 715518), 24. April 2012

Abstract

Measuring the bidirectional reflectance distribution function (BRDF) of surfaces usually is a complex procedure that demands for calibrated setups and high-quality cameras. This work investigates how to simplify the process by using cameras of mobile devices. For this purpose, an approach for reconstructing normals and spatially varying reflectance of a textured, almost planar surface from two images taken under distant, known illumination is presented. The illumination is measured by using a camera equipped with a fish-eye lens to capture the environment in a single shot. Diffuse shading is extracted from the input images by using a bilateral filter to extract details in image intensity that are not contained in chromaticity. The normal map is reconstructed with a shape from shading algorithm based on a low-order spherical harmonics approximation of the environment. BRDFs are represented by a linear combination of basis materials. This allows to estimate its coefficients with linear least-squares optimization. Two approaches are presented to handle spatially varying reflectance. In the first, surface points are clustered by analyzing residual histograms of random models generated from a RANSAC-like sampling process. In the second, diffuse and specular components of the BRDF are independently estimated, which allows to recover a texture with albedo color per pixel and a specular component shared by multiple surface points. Furthermore, differences between the two input images are analyzed in order to separate the surface into glossy and matte regions.

Acknowledgments

First of all, I would like to thank Manuel Finkh for his constant support, motivation and guidance as well as many discussions on small and large problems. I would also like to thank Prof. Hendrik Lensch for introducing me to the fascinating field of computer graphics and for making this project possible. I am also grateful for many helpful hints and discussions. In addition, I would like to thank Katharina Schwarz for many useful suggestions regarding writing and for proofreading. Besides, I am grateful to Christian Fuchs for his technical support in various areas. Finally, as this thesis would not have been possible without their permanent help and encouragement, I am very thankful to all my friends and family.

Contents

1. Introduction	1
1.1. Problem Statement	1
1.2. Outline	3
2. Background	5
2.1. Radiometry	5
2.2. Bidirectional Reflectance Distribution Function	7
2.2.1. Properties of BRDFs	7
2.2.2. BRDF Representation	8
2.2.3. Acquiring BRDFs	10
2.2.4. Other Scattering Functions	10
2.3. Linear Least-Squares Optimization	10
2.4. Multiple Model Estimation	11
2.5. Rendering Equation	13
2.6. Environment Mapping	13
2.7. Spherical Harmonics	15
2.8. Homography Decomposition	17
3. Related Work	19
3.1. Measuring BRDFs	19
3.2. Normal Reconstruction and Shape From Shading	20
3.3. Intrinsic Image Decomposition	21
3.4. Combinations	21
4. System Overview	23
4.1. Image Capturing and Scene Reconstruction	23
4.2. Normal and BRDF Estimation	24
4.3. Problems	26
4.4. Notes on Evaluation	26
5. Scene Reconstruction	27
5.1. Generate High Dynamic Range Images	28
5.2. Estimate Homography	28
5.3. Decompose Homography	29
5.4. Extract Texture	30
5.5. Build Environment Map	31

Contents

5.5.1. Handling Imperfect Calibration	32
5.6. Convert Environment Map	33
6. Normal Estimation	35
6.1. Intrinsic Image Decomposition	35
6.2. Estimating Normal Maps of Diffuse Surfaces	42
7. BRDF Estimation	49
7.1. A Linear BRDF Model	49
7.2. Estimating BRDF Weighting Coefficients from Images	51
7.3. Evaluation	54
7.3.1. Synthetic Scenes	54
7.3.2. Real-World Scenes	60
8. Estimating Spatially Varying BRDFs	67
8.1. Clustering Similar Points	68
8.2. Diffuse-Specular Separation	74
8.2.1. Handling Textured Surfaces	74
8.2.2. Combining Separation and Clustering	80
9. Combining Normal and BRDF Estimation	89
9.1. Homogeneous Materials	90
9.2. Spatially Varying Materials	95
10. Conclusion	99
A. Iterative Solutions for Normal Estimation	101
A.1. Initial Normal Map	101
A.2. Height Map	102

1. Introduction

High-quality material models are a basic requirement in photorealistic image synthesis. If correctly modeled, optical properties help us to deduce physical characteristics of objects like hardness, weight, elasticity, or brittleness. Otherwise, if modeled imprecisely, images look artificial. Typically, optical properties of materials are represented by the bidirectional reflectance distribution function (BRDF). It describes to what extent a light ray is reflected in a direction when hitting the material from a particular direction. In this way, color and glossiness of the material are defined.

Synthesizing images with realistic looking materials can be achieved by the use of measured BRDFs. However, this results in large datasets with high memory requirements and involves complex measurement procedures. Producing high-quality datasets can take hours and needs accurately calibrated measurement devices and light sources. In addition, the optical properties may vary with position on the surface, increasing the complexity of the measurement process even more.

The surface of objects is not limited to changes in optical properties of the material. Variations in height of nearby surface points lead to a bumpy appearance. This can be captured and simulated by normal maps that contain the direction of the normal at each surface point. Accurate measurement of normals requires multiple images of the surface illuminated from different directions. It works best if the material does not have a glossy appearance.

In contrast, people only need to see a small number of images of the object's surface to infer its optical properties and surface structure. This implies that a few images already contain enough information for inference. Consequently, precise measurement is not necessary to reproduce an image of the material that looks natural to us. Instead, only the effects needed for correct inference have to be captured. Creating a computer vision system capable of estimating optical properties and normals from only a few images would allow to produce realistic renderings of materials without complex BRDF and normal measurement.

1.1. Problem Statement

This work investigates how to estimate normals and spatially varying BRDFs of nearly planar surfaces from photographs captured under distant natural illumination. At least two images of the surface and one image of the environment have to be

supplied. In order to capture the environment in a single shot, the camera is equipped with a fish-eye lens. Given the input images, the system does not need any further user interactions. To reduce complexity, inter-reflections and shadows are ignored.

Although images could be captured using arbitrary cameras, this work focuses on the use of mobile devices like smartphones or tablets and its integrated cameras. Besides capturing, the increasing processing power of these devices allows to run the computation on the device. Therefore, the algorithms are designed with keeping low-end hardware in mind. However, the prototypical implementation currently runs on standard PC hardware.

The system can be divided into two parts: estimation of normals and estimation of spatially varying BRDFs. In general, normals and BRDF are different at each surface point. Given the great number of unknowns, it is impossible to reconstruct the normal and the BRDF at each point. Fortunately, in the majority of cases there is some coherency in points of real surfaces, at least approximately. By ignoring some subtle effects, surfaces can be grouped as shown in Fig. 1.1. Limiting the input data to a certain surface type allows to recover meaningful normals and reflectance properties from only two images. However, different types demand for different constraints. Therefore, multiple approaches are developed in this work, allowing to generate the best possible result for the appropriate surface type. Nevertheless, even by restricting the surface type, the problem remains ill-posed. Consequently, it is not always possible to generate a good solution.

In order to reconstruct normals, the reflectance properties have to be known and vice versa. As both are unknown, diffuse shading information is extracted from the input images by applying some heuristics. This information is used to estimate the normal map. For this purpose, an intrinsic image decomposition algorithm for images of surfaces with details in shading and albedo is introduced, followed by a shape from shading approach that is capable of estimating normals under arbitrary illumination.

Similar to normal reconstruction, the estimation of the BRDF is done via optimization. In this work, a BRDF model based on a linear combination of basis materials is used, so fitting can be performed by linear least-squares optimization. A precomputation step further decreases the running time. For surfaces with homogeneous BRDF (diffuse and specular components do not vary), all surface points can be used as constraints in the optimization process, yielding a feasible solution to the problem.

Extending the approach to reconstruct spatially varying BRDFs is difficult, because large regions on the surface do not carry any information about the specular part of the BRDF. As mentioned before, coherency between surface points can be exploited to estimate the specular component. In this thesis, two approaches are presented. In the first, surface points are clustered into regions that share the same BRDF and a diffuse as well as a specular component are computed for each region. In the second, diffuse and specular components of the BRDF are independently estimated. While the diffuse color is estimated per point, the specular component is shared

by multiple pixels. This approach produces a texture with diffuse per-pixel color (albedo) and a specular map that separates the surface into regions with matte and glossy materials. Afterwards, for each of the regions, a set of parameters describing its shape is estimated.

1.2. Outline

This thesis begins by introducing the fundamental concepts and techniques used in the subsequent chapters (Sec. 2). This is followed by an overview on related work (Sec. 3). After that, the prototypical system developed in this thesis is presented, starting with a summary of the basic components (Sec. 4) and being followed by a detailed explanation. First, it is shown how a 3-dimensional scene is reconstructed from two images (Sec. 5). Second, the algorithms for normal (Sec. 6) and BRDF estimation (Sec. 7) are explained. Afterwards, BRDF estimation is extended to estimate spatially varying BRDFs (Sec. 8). This is followed by more details on how to combine the described approaches (Sec. 9). Finally, the work is reviewed and an outlook on future work is given (Sec. 10).

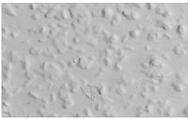
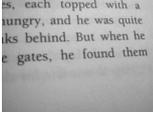
	planar surfaces	bumpy surfaces
diffuse: homogeneous specular: -	 <i>paper</i>	 <i>ingrain wallpaper</i>
diffuse: homogeneous specular: homogeneous	 <i>plastic</i>	 <i>Lego</i>
diffuse: homogeneous specular: spatially varying	 <i>paper with tape</i>	 <i>wrinkled paper with tape</i>
diffuse: spatially varying specular: -	 <i>printed paper</i>	 <i>stone wall</i>
diffuse: spatially varying specular: homogeneous	 <i>printed glossy paper</i>	 <i>coated wood with irregularities</i>

Figure 1.1.: Examples for surfaces grouped by its reflection properties and normals. Grouping is only possible if some effects are ignored, e.g., “diffuse” means that the surface is approximately diffuse. Furthermore, blurred highlights usually occur because of variations in normal direction on a micro-scale (microfacet distribution), i.e., surfaces showing blurred highlights are always rough. However, in this work “planar” and “bumpy”/“rough” refer to surfaces with clearly visible normal variations (macro-scale). See Sec. 2.2 for more details.

2. Background

2.1. Radiometry

Radiometry defines a set of physical quantities to measure light. Light is electromagnetic radiation that can be sensed by the human eye, with wavelengths ranging from 380 nm to 740 nm. The basic building block of light is a photon that carries an amount of energy depending on its wavelength. The radiometric quantities describe the number of photons (i.e., the energy) traveling through space per second. Some quantities needed to understand the concept of BRDFs and image synthesis are described in the following, based on [SM09] and [WLL⁺08].

Power/Radiant Flux $[\Phi] = J/s$ describes how many photons leave a light source or hit a surface per second.

Irradiance $[E] = W/m^2$ describes how many photons arrive at a surface point with area dA per second and is defined as power per unit area:

$$E = \frac{d\Phi}{dA}. \quad (2.1)$$

Radiance $[L] = W/(m^2 sr)$ describes how many photons arrive at a point of the surface from a particular direction per second. Light travels through space along a straight line without changing until a surface is hit. Radiance is defined as power per projected unit area per unit solid angle:

$$L = \frac{d^2\Phi}{d\Omega dA \cos\theta}, \quad (2.2)$$

where θ is the angle between surface normal and incoming light direction and Ω is the solid angle on the hemisphere subtended by the incoming light. The solid angle on the hemisphere in three-dimensional space is comparable to the angle in two dimensions. A 2D angle is the length of a circular arc of a circle with radius 1, ranging from 0 to 2π (in radians). A 3D solid angle is the area of a region on the surface of a sphere with radius 1, ranging from 0 to 4π (in steradians sr , see Fig. 2.1).

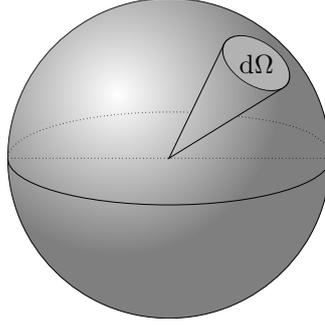


Figure 2.1.: The area $d\Omega$ of a patch on the surface of a unit sphere is named solid angle, measured in steradians (sr).

The area dA hit by the light ray needs a correction term $\cos \theta$ to account for the change in area that results from the slanted incoming angle. This eliminates the influence of θ on the radiance, so the radiance is independent of the incoming angle.

Radiance is the basic quantity in radiometry. Both irradiance and power can be given in terms of radiance. The irradiance of a point \mathbf{x} on a surface is calculated by integrating the incoming radiance over the upper hemisphere Ω_+ above the surface:

$$E(\mathbf{x}) = \int_{\Omega_+} L(\mathbf{x}, \boldsymbol{\omega}) \cos \theta d\boldsymbol{\omega} \quad (2.3)$$

$$= \int_{\Phi=0}^{2\pi} \int_{\theta=0}^{\frac{\pi}{2}} L(\mathbf{x}, \boldsymbol{\omega}) \cos \theta \sin \theta d\theta d\Phi. \quad (2.4)$$

For constant radiance $L(\mathbf{x}, \boldsymbol{\omega}) = L_c(\mathbf{x})$, the irradiance becomes $E(\mathbf{x}) = \pi L_c(\mathbf{x})$. The power hitting a surface is found by integrating the irradiance across the surface area S :

$$\Phi = \int_S E(\mathbf{x}) dA, \quad (2.5)$$

where \mathbf{x} is a point on the surface and dA its differential area.

All quantities can be based on wavelength, giving

- Spectral Power (W/nm),
- Spectral Irradiance (W/(m² nm)), and
- Spectral Radiance (W/(m² sr nm)).

Similar to radiometry, photometry measures perceived light, i.e., it incorporates the eye's perception properties. Each radiometric quantity has its photometric counterpart, e.g., radiant flux corresponds to luminous flux measured in lumen (lm),

irradiance corresponds to illuminance measured in lux (lux) and radiance corresponds to luminance measured in candela per square meter (cd/m^2). In this work, only radiometric quantities are used, as the physical quantities of light should be estimated and simulated.

2.2. Bidirectional Reflectance Distribution Function

The distinct appearance of materials is defined by the bidirectional reflectance distribution function (BRDF) [NRH⁺77] [WLL⁺08]. This function describes the reflecting properties of a surface point. It is defined as the ratio between the reflected radiance and the irradiance:

$$f_r(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = \frac{dL(\boldsymbol{\omega}_o)}{dE(\boldsymbol{\omega}_i)} = \frac{dL(\theta_o, \phi_o)}{L_i(\theta_i, \phi_i) \cos \theta_i d\omega_i}, \quad (2.6)$$

where $\boldsymbol{\omega}_i$ and $\boldsymbol{\omega}_o$ denote incident and outgoing direction with respect to the surface normal, $dL(\boldsymbol{\omega}_o)$ denotes the outgoing radiance in direction $\boldsymbol{\omega}_o$, $dE(\boldsymbol{\omega}_i)$ is the incident irradiance from direction $\boldsymbol{\omega}_i$, $L_i(\theta_i, \phi_i)$ is the incident radiance from direction $\boldsymbol{\omega}_i$, and $d\omega_i$ is the solid angle within which the incident radiance is confined.

2.2.1. Properties of BRDFs

All physically possible BRDFs share the following properties:

- The values are always positive, as outgoing radiance and incident irradiance are positive.
- The reflected light has to be equal or smaller than the incoming light:

$$\int_{\Omega_+} f_r(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) \cos \theta_o d\omega_o \leq 1.$$
- Interchanging the incident and outgoing direction does not change the BRDF. This property is called Helmholtz reciprocity and is necessary to guarantee that the light transport is symmetric.

A BRDF can either be isotropic or anisotropic. Isotropic BRDFs are invariant with respect to rotation about the surface normal. Therefore, an isotropic BRDF is a function of θ_i , θ_o and the difference between ϕ_i and ϕ_o : $f_r(\Delta\phi, \theta_i, \theta_o)$.

Fig. 2.2 shows example plots of various BRDFs. These plots are generated by fixing the incident angle $\boldsymbol{\omega}_i$ (e.g., $f_r(45^\circ, 15^\circ, \theta_o, \phi_o)$) and visualizing the function of the remaining two dimensions in polar coordinates.

The BRDF of an ideal Lambertian reflector is shown in Fig. 2.2(a). The surface reflects light equally in all directions ($f_r(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = c = \frac{\rho}{\pi}$), so to an observer the surface looks the same independent of the viewing direction. In contrast, an ideal

mirror (Fig. 2.2(b)) reflects light in an infinitely narrow peak. If the peak becomes broader, the reflection is called specular or glossy (Fig. 2.2(c)). The BRDFs of real materials show more complicated reflection properties (Figs. 2.2(d)-2.2(f)) due to the complex scattering interactions of light rays with the material.

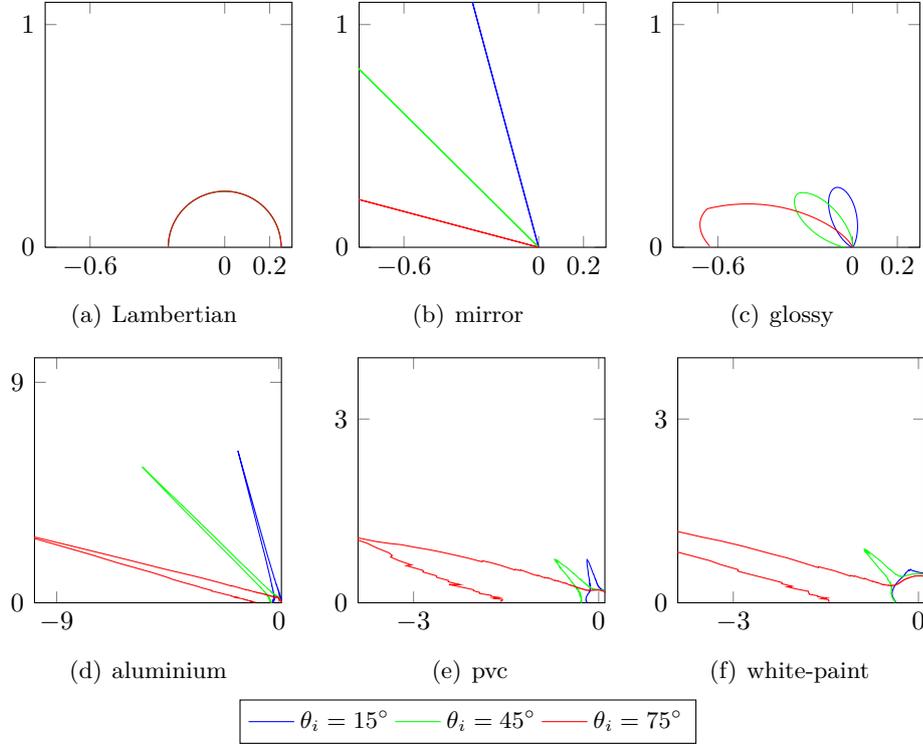


Figure 2.2.: Examples for different bidirectional reflectance distribution functions, scaled by cubic root, $\theta_i = \{15^\circ, 45^\circ, 75^\circ\}$, $\phi_i = 0^\circ$, $\phi_o = 180^\circ$. BRDFs (d)-(e) were measured by Matusik et al. [MPBM03a].

2.2.2. BRDF Representation

BRDFs can be represented in two ways: either as tabulated values or as an analytic function. Tabulating the values would result in high storage requirements if sufficiently many samples were stored to reproduce the reflection properties adequately. Therefore, analytic functions were developed that preserve the properties of real-world materials more or less accurately without the need to store all values. These parametric models are usually parameterized by only a few parameters. Typically, they consist of a diffuse component and one or more specular lobes:

$$f_r(\omega_i, \omega_o) = \underbrace{\frac{\rho_d}{\pi}}_{\text{diffuse}} + \underbrace{\rho_s f_s(\omega_i, \omega_o)}_{\text{specular}}. \quad (2.7)$$

2.2. Bidirectional Reflectance Distribution Function

In this example, ρ_d and ρ_s are the intensities of diffuse and specular component, respectively. ρ_d is also called albedo. The shape of the specular lobe is given by $f_s(\omega_i, \omega_o)$.

Popular analytic models are the Ward model [War92], Blinn-Phong model [Bli77], Lafortune model [LFTG97], Cook-Torrance model [CT82] and Ashikhmin-Shirley model [AS00]. They differ in computational complexity, accuracy in representing real-world materials [NDM05], support for anisotropic materials, and number of parameters.

Ashikhmin-Shirley BRDF As a simplified version of the Ashikhmin-Shirley BRDF model is used in this work, a small summary of its properties is given in the following. The anisotropic BRDF model by Ashikhmin and Shirley has proven to perform well in representing a large variety of materials [NDM05]. It consists of a specular term accounting for Fresnel-effects (Schlick’s approximation [Sch94]) and a diffuse term with a few improvements over Lambertian reflection. These improvements ensure energy conservation in the presence of a Fresnel-weighted term. The shape of the specular lobe is controlled by three parameters, the Fresnel reflectance and two Phong-like exponents to control the roughness of the material. The model can be easily integrated into Monte-Carlo based rendering methods.

Linear models The BRDF models mentioned above represent the specular lobe by one or more non-linear, exponential terms. In addition, some models like the Lafortune-BRDF [LFTG97] employ multiple lobes (see Eq. (2.7)). This is even necessary to represent some materials accurately [NDM05]. Fitting the parameters of non-linear models to measured data is realized with constrained non-linear optimization techniques. Hence, it is sensitive to the initial guess and the optimization process can get stuck to a local optimum. In contrast, linear models do not suffer from such problems. A BRDF can be represented as a linear combination of K basis functions $f_k(\omega_i, \omega_o)$ with weighting coefficients a_k :

$$f_r(\omega_i, \omega_o) = \sum_{k=1}^K a_k f_k(\omega_i, \omega_o). \quad (2.8)$$

Any function can be used as basis function, e.g., single lobes of a parametric model [HFB⁺09], measured BRDFs or radial basis functions [WWHL07]. Fitting the parameters a_k is a linear problem, solvable with standard linear least-squares optimization. Furthermore, materials with multiple specular lobes can be inherently represented and the lobe’s shape is not restricted by an analytic function. However, the number of parameters to control the shape of the function is higher than in a parametric model and they typically do not have an intuitive meaning.

2.2.3. Acquiring BRDFs

BRDFs of real-world materials are classically acquired with a gonioreflectometer. This measurement device consists of a light source and a detector that can be freely positioned on the hemisphere surrounding the material sample. Thus, by placing light source and detector at each possible combination of positions, the whole BRDF can be acquired. Two configurations of light source and detector are possible: either the sample is illuminated by a wide-angle light source and the detector has a narrow field of view or the sample is illuminated by a narrow light beam and the detector has a wide field of view [WSB⁺98]. As only a single combination of angles is acquired per time step, measurement is a lengthy process [WLL⁺08].

Instead of measuring only a single (ω_i, ω_o) -combination at once, multiple combinations can be measured simultaneously by specialized setups. In this case, the surface's material has to be homogeneous, i.e., each surface point has exactly the same BRDF: $f_r(\omega_i, \mathbf{x}, \omega_o) = f_r(\omega_i, \omega_o)$. The specific measurement device depends on the geometry of the surface. A planar surface, for example, can be acquired by a CCD sensor with a fish-eye lens [War92]. Each light beam is reflected by the surface into a hemispherical mirror, which reflects the light back into the fish-eye lens.

2.2.4. Other Scattering Functions

The 4D anisotropic BRDF or the 3D isotropic BRDF capture only a small amount of phenomena associated with surface reflectance. In order to store reflection properties of an inhomogeneous surface, two additional dimensions have to be added to the function, resulting in a spatial varying BRDF (SVBRDF). If subsurface scattering should be represented, another two dimensions are necessary, leading to the bidirectional scattering surface reflection distribution function (BSSRDF). This allows to characterize translucent materials, where light leaves the surface at another point as it entered. Fluorescence and Phosphorescence add even more dimensions (wavelength λ and time t). Summing up, the complete 12 dimensional scattering function looks as follows: $f_r(\underbrace{\theta_i, \phi_i, \theta_o, \phi_o}_{\text{BRDF}}, x_i, y_i, x_o, y_o, \lambda_i, \lambda_o, t_i, t_o)$

$$\begin{array}{c}
 \underbrace{\hspace{10em}}_{\text{BRDF}} \\
 \underbrace{\hspace{10em}}_{\text{BSSRDF}} \\
 \underbrace{\hspace{10em}}_{\text{Fluorescence}} \\
 \underbrace{\hspace{10em}}_{\text{Phosphorescence}}
 \end{array}$$

2.3. Linear Least-Squares Optimization

When a linear combination of basis materials is used for the BRDF model, fitting the model to measured data results in a linear least-squares optimization. In this case, the model consists of K weighting coefficients and should be fitted to M measured

function values. However, an exact solution does not exist, because $K < M$. Instead, the solution that best approximates the data is searched.

A linear system of equations has the following form:

$$\sum_{j=1}^K a_j x_{ij} = b_i, i = (1, 2, \dots, M). \quad (2.9)$$

The system is called overdetermined if there are more (linear independent) equations than unknowns, i.e., if $M > K$. As there is no exact solution, the solution coefficients a_j that minimize the squared difference between the observations b_i and the function values $\sum_{j=1}^K a_j x_{ij}$ can be calculated, leading to the best fit in least-squares sense. This means to minimize the following function:

$$E(\mathbf{a}) = \sum_{i=1}^M |b_i - \sum_{j=1}^K a_j x_{ij}|^2. \quad (2.10)$$

In matrix notation, this corresponds to the squared L2 norm of the residual vector \mathbf{r} :

$$\|\mathbf{r}\|^2 = \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2, \quad (2.11)$$

where $\mathbf{A} \in \mathbb{R}^{M \times N}$ and $\mathbf{b} \in \mathbb{R}^M$. The function E is minimal when its gradient vector is zero, leading to the following direct solution

$$\hat{\mathbf{x}} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}, \quad (2.12)$$

where $(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$ is called the pseudoinverse.

Non-negative linear least-squares Often there are additional constraints which the solution vector \mathbf{x} should satisfy. One such constraint is that its elements must not be negative. The problem has the form

$$\arg \min_{\mathbf{x}} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 \text{ subject to } \mathbf{x} \geq \mathbf{0}. \quad (2.13)$$

An iterative algorithm to compute \mathbf{x} is given in [LH74]. On initialization, \mathbf{x} is set to $\mathbf{0}$. Then, in each iteration, the elements of \mathbf{x} are successively updated by solving a linear least-squares problem with all elements computed so far and one new element. If an element becomes smaller than or equal to $\mathbf{0}$, the problem is slightly adjusted by setting conflicting elements of \mathbf{x} to zero. As the authors show, the algorithm is guaranteed to converge to a solution.

2.4. Multiple Model Estimation

A surface consisting of multiple materials has a spatially varying BRDF. It is not sufficient to estimate parameters for a single BRDF model, as these parameters

vary from point to point. Therefore, techniques to estimate multiple models from a collection of noisy data points can be used. Fig. 2.3 illustrates this problem by means of line fitting. The arrangement of the data points indicates three models for which the parameters should be estimated. The algorithm proposed in [ZK07] estimates these models by analyzing histograms of residuals generated from a RANSAC-like sampling process.

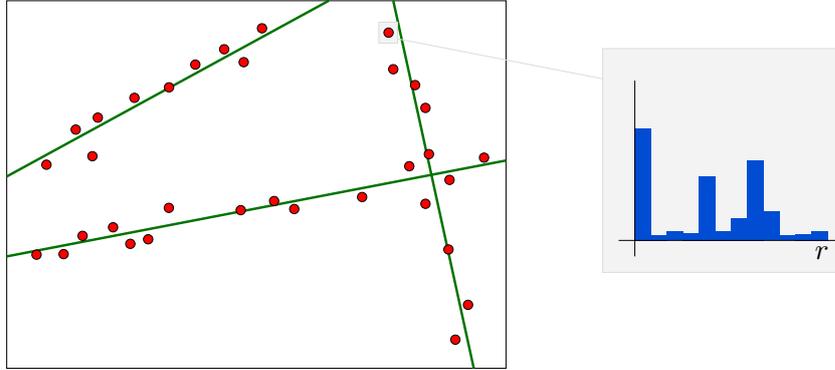


Figure 2.3.: Fitting multiple lines to noisy data points. Left: the data points (red) indicate three models (green). Right: the residual histogram for an arbitrary point contains modes for the models.

Let \mathbf{x}_i be the measured data points with $i = (1, 2, \dots, L)$, corrupted by noise and a small amount of outliers. These data points are generated from M models, each model is parameterized by its parameter vector \mathbf{p}_j , $j = (1, 2, \dots, M)$. Given the data points, the objective is to find the number of models M and their parameters \mathbf{p}_j . The algorithm operates in two steps: hypotheses generation and evaluation.

In the first stage, N hypotheses are generated. From all data points, a minimum number of points needed to instantiate a model are randomly selected. With these points, the hypothesis is formed and verified by calculating the residuals r_{ij} for each data point. Then, the residuals are stored and the next set of random points is selected to generate a new hypothesis. After N iterations, the first stage has finished.

In the second stage, the hypotheses are analyzed to create the true models. First, at each point a histogram is created from the residuals at that point. By analyzing these histograms, the number of models can be estimated. Hypothetical models that explain a significant amount of data points form modes in the histograms. In the line fitting example in Fig. 2.3, the residual histogram at point $i = 1$ shows three large modes that correspond to the three true models found in the data points. However, there is a large number of smaller modes that have to be ignored. Therefore, a few processing steps described in [ZK07] are necessary for a robust estimation of M . The final number of models is then the median number of modes of all histograms. Afterwards, the “best” histogram according to some criterion is chosen and the hypotheses corresponding to the modes are determined. However,

multiple hypotheses correspond to a mode. Besides a cluster of “good” hypotheses that describe the data point, there are other random models that also have the same residuals. By the help of the histogram of another data point, the “good” hypotheses can be separated from the “bad”. Finally, the true models are extracted from the “good” hypotheses.

2.5. Rendering Equation

The rendering equation describes the propagation of light in an environment. It defines the outgoing radiance $L_o(\mathbf{x}, \boldsymbol{\omega}_o)$ at a surface point as the sum of emitted radiance and reflected radiance [ICG86]:

$$L_o(\mathbf{x}, \boldsymbol{\omega}_o) = L_e(\mathbf{x}, \boldsymbol{\omega}_o) + \int_{\Omega_+} f_r(\boldsymbol{\omega}_i, \mathbf{x}, \boldsymbol{\omega}_o) L_i(\mathbf{x}, \boldsymbol{\omega}_i) (\boldsymbol{\omega}_i \cdot \mathbf{n}) d\boldsymbol{\omega}_i. \quad (2.14)$$

The reflected radiance is given as the integral over all incident directions on the upper hemisphere Ω_+ . The incident radiance $L_i(\mathbf{x}, \boldsymbol{\omega}_i)$ has to be multiplied with the BRDF $f_r(\boldsymbol{\omega}_i, \mathbf{x}, \boldsymbol{\omega}_o)$ and the cosine of the angle between surface normal \mathbf{n} and $\boldsymbol{\omega}_i$ ($\cos \theta_i = \boldsymbol{\omega}_i \cdot \mathbf{n}$). When dealing with global illumination, the incoming radiance is equal to the reflected radiance at another surface point in the scene. This makes the evaluation of the rendering equation complicated, as in order to compute the outgoing radiance at one point of a surface, the outgoing radiances of all visible surfaces in the scene have to be known recursively. The rendering equation can be solved with algorithms based on Monte-Carlo integration, like (Bidirectional) Path Tracing or Metropolis light transport [Vea98].

2.6. Environment Mapping

Environment mapping is a technique to approximate the lighting of objects in a scene. Under the assumption that the illumination is distant, the incident light at a point can be precomputed for each surface normal direction and stored in a texture [BN76]. With distant lighting, the incident illumination at a point does not vary with the position of the point in 3D space, but only with the normal of the surface at this point (for simplicity, self-shadowing and self-reflection is ignored). The object is then illuminated by looking up one or more pixels in the environment map. The environment map can be seen as the texture of a large sphere surrounding the object. In terms of the rendering equation, the process is formulated as

$$L_o(\mathbf{x}, \boldsymbol{\omega}_o) = \int_{\Omega_+} f_r(\boldsymbol{\omega}_i, \mathbf{x}, \boldsymbol{\omega}_o) L_i(\boldsymbol{\omega}_i) \cos \theta_i d\boldsymbol{\omega}_i. \quad (2.15)$$

In this case, $L_i(\omega_i)$ is the environment map. If the object is perfectly mirror-like, the integral will be zero except for a single angle $d\omega_i$, so only a single pixel has to be looked up. In other words, the object reflects the ray from the eye into the surrounding sphere with the environment map (see Fig. 2.4). The pixel at the intersection point determines the color of the reflection.

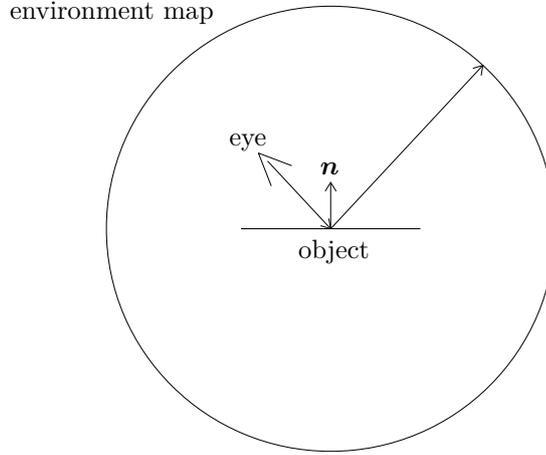


Figure 2.4.: The eye ray reflected at the object hits the surrounding sphere at a specific point. The corresponding pixel in the environment map is looked up and used to shade the object.

For diffuse objects or objects with complex specular BRDFs, the integral has to be evaluated. In the diffuse case, the BRDF is constant. Therefore, the integral does not depend on the outgoing direction ω_o , but only on the surface normal. Thus, the environment map can be pre-filtered, i.e., for each normal direction the irradiance is stored in a texture $E(\mathbf{n})$:

$$E(\mathbf{n}) = \int_{\Omega_+} L_i(\omega_i) \cos \theta_i d\omega_i. \quad (2.16)$$

The texture is called irradiance environment map [RH01b]. The normal

$$\mathbf{n} = (\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta)^T \quad (2.17)$$

and ω_i are given in global coordinates. Usually, the environment map $L_i(\omega_i)$ is given in global coordinates, but the integration is performed in the local coordinate system of the surface. From now on, local coordinates are indicated by primes. Consequently, with a rotation operator \mathbf{R}_n that translates from local coordinates ω'_i to global coordinates ω_i , the irradiance becomes

$$E(\mathbf{n}) = \int_{\Omega'_+} L_i(\mathbf{R}_n(\omega'_i)) \cos \theta'_i d\omega'_i. \quad (2.18)$$

The integral describes a convolution of the environment map L_i with $\cos \theta'_i$, resulting in the irradiance environment map $E(\mathbf{n})$. In the rendering process, only the pixel in the texture that corresponds to the normal at surface point \mathbf{x} has to be looked up:

$$L_o(\mathbf{x}, \boldsymbol{\omega}_o) = \frac{\rho}{\pi} E(\mathbf{n}). \quad (2.19)$$

Extending this technique to complex BRDFs is complicated, as the integral depends on \mathbf{n} and $\boldsymbol{\omega}_o$ and so a 4D texture would have to be stored, resulting in large memory requirements.

The reflected light from a diffuse surface varies slowly with respect to the normal. Therefore, an irradiance environment map $E(\mathbf{n})$ has only very low-frequency contents. This property can be exploited to compress E efficiently.

2.7. Spherical Harmonics

Spherical harmonics are a set of orthonormal basis functions Y_{lm} defined on the sphere, with $l \geq 0$ and $-l \leq m \leq l$ [RH01b]:

$$Y_{lm}(\theta, \phi) = \begin{cases} \sqrt{2}K_{lm} \cos(m\phi)P_{lm}(\cos \theta) & \text{if } m > 0 \\ K_{l0}P_{l0}(\cos \theta) & \text{if } m = 0 \\ \sqrt{2}K_{lm} \sin(|m|\phi)P_{l|m|}(\cos \theta) & \text{if } m < 0 \end{cases}, \quad (2.20)$$

where P_{lm} denote the Associated Legendre polynomials and K_{lm} are normalization constants:

$$K_{lm} = \sqrt{\frac{(2l+1)(l-|m|)!}{4\pi(l+|m|)!}}. \quad (2.21)$$

When written in Cartesian coordinates, they form polynomials of (x, y, z) with degree l :

$$Y_{00}(\theta, \phi) = \frac{1}{2\sqrt{\pi}} \quad (2.22)$$

$$Y_{1-1}(\theta, \phi) = \frac{\sqrt{3}}{2\sqrt{\pi}}y \quad (2.23)$$

$$Y_{10}(\theta, \phi) = \frac{\sqrt{3}}{2\sqrt{\pi}}z \quad (2.24)$$

$$Y_{11}(\theta, \phi) = \frac{\sqrt{3}}{2\sqrt{\pi}}x \quad (2.25)$$

...

A function $f(\theta, \phi)$ can be represented as a linear combination of the basis functions $Y_{lm}(\theta, \phi)$ with weightings f_{lm} :

$$f(\theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^l f_{lm} Y_{lm}(\theta, \phi). \quad (2.26)$$

The coefficients f_{lm} can be obtained by projecting the function $f(\theta, \phi)$ onto the basis functions:

$$f_{lm} = \int_0^{2\pi} \int_0^{\pi} f(\theta, \phi) Y_{lm}(\theta, \phi) \sin \theta \, d\theta d\phi. \quad (2.27)$$

Spherical harmonics can be used to compress irradiance environment maps. It has been shown that reasonable approximation can be achieved with only 9 coefficients ($0 \leq l \leq 2$) [RH01b]. Furthermore, the convolution in Eq. (2.18) becomes a multiplication in spherical harmonics domain (analog to the Fourier domain), which makes the rendering very efficient. In the following, a brief overview of the rendering with spherical harmonics is given, for a more complete explanation see [RH01a].

The environment map $L(\mathbf{n})$ and the cosine term $A(\theta')$ can be represented in the spherical harmonics basis:

$$L(\theta_i, \phi_i) = \sum_{l=0}^{\infty} \sum_{m=-l}^l L_{lm} Y_{lm}(\theta_i, \phi_i) \quad (2.28)$$

$$A(\theta'_i) = \max(\cos \theta'_i, 0) = \sum_{l=0}^{\infty} A_l Y_{l0}(\theta'_i). \quad (2.29)$$

A has no azimuthal dependence, so ϕ can be omitted and only $m = 0$ is required. Furthermore, $A(\theta'_i)$ has to be 0 over the lower hemisphere where $\cos(\theta'_i) < 0$, so the values are clamped to 0. The values for A_l are given in [RH01a]. The rotation operator $R_{\mathbf{n}}$ also has to be expressed in the new basis:

$$R_{\mathbf{n}}\{Y_{lm}(\theta'_i, \phi'_i)\} = \sum_{m'=-l}^l \tilde{D}_{mm'}^l(\mathbf{n}) Y_{lm'}(\theta'_i, \phi'_i), \quad (2.30)$$

with $\tilde{D}_{mm'}^l = \sqrt{\frac{4\pi}{2l+1}} Y_{lm}(\theta, \phi)$. This describes how a rotated spherical harmonic in level l is rewritten as a new linear combination of spherical harmonics of the same level, only the indices m have changed. $D_{mm'}^l$ are the coefficients for the rotated spherical harmonics. Only the coefficients for $m' = 0$ are of interest here, as the other

terms will vanish later. $E(\theta, \phi)$ can then be written as

$$E(\mathbf{n}) = \int_0^{2\pi} \int_0^\pi \sum_{l=0}^{\infty} \sum_{m=-l}^l L_{lm} \sum_{m'=-l}^l \tilde{D}_{mm'}^l(\mathbf{n}) Y_{lm'}(\theta'_i, \phi'_i) \sum_{n=0}^{\infty} A_n Y_{n0}(\theta'_i) \sin \theta'_i d\theta'_i d\phi'_i \quad (2.31)$$

$$= \sum_{l=0}^{\infty} \sum_{m=-l}^l \sum_{m'=-l}^l \sum_{n=0}^{\infty} L_{lm} \tilde{D}_{mm'}^l(\mathbf{n}) A_n \int_0^{2\pi} \int_0^\pi Y_{lm'}(\theta'_i, \phi'_i) Y_{n0}(\theta'_i) \sin \theta'_i d\theta'_i d\phi'_i. \quad (2.32)$$

Due to the orthonormality of the basis functions, $Y_{lm'}(\theta'_i, \phi'_i) Y_{n0}(\theta'_i)$ will be 1 if $l = n \wedge m' = 0$ and 0 otherwise. So what remains is

$$E(\mathbf{n}) = \sum_{l=0}^{\infty} \sum_{m=-l}^l L_{lm} \tilde{D}_{m0}^l(\mathbf{n}) A_l. \quad (2.33)$$

To sum up, the irradiance of a surface point with normal \mathbf{n} is calculated in the following way:

$$E(\mathbf{n}) = \sum_{l=0}^{\infty} \sum_{m=-l}^l \sqrt{\frac{4\pi}{2l+1}} L_{lm} A_l Y_{lm}(\theta, \phi). \quad (2.34)$$

As $Y_{lm}(\theta, \phi)$ is a polynomial function in Cartesian coordinates, rendering means to solve a quadratic (9 coefficients) or linear (4 coefficients) polynomial of the surface normal. Furthermore, the coefficients L_{lm} are calculated from the environment map $L(\theta, \phi)$ by solving the following integral:

$$L_{lm} = \int_0^{2\pi} \int_0^\pi L(\theta, \phi) Y_{lm}(\theta, \phi) \sin \theta d\theta d\phi. \quad (2.35)$$

Solving the integral means to sum up all pixels in the environment map and weight them according to the spherical harmonics basis functions and the solid angle subtended by the pixel. For RGB environment maps, this is done for each channel separately, so for each color channel there are 9 coefficients for $l = 2$.

2.8. Homography Decomposition

A homography describes the transformation between two views of a planar surface in space, assuming a pin-hole camera model. It captures the motion that a camera performed between two frames. Once a homography has been estimated, it can be decomposed into translation and rotation to build a 3D model of a scene.

Let $\mathbf{p}_p = (x_p, y_p, 1)^T$ and $\mathbf{p}'_p = (x'_p, y'_p, 1)^T$ be vectors containing the normalized projective coordinates of two relating points in the reference view F and current view F' , respectively. These coordinates are the projections of the 3D point \mathbf{p} onto the image planes in both camera spaces. The 3×3 -matrix \mathbf{H} that transforms \mathbf{p}_p into \mathbf{p}'_p up to a scale factor w is called homography matrix:

$$w \mathbf{p}'_p = \mathbf{H} \mathbf{p}_p. \quad (2.36)$$

\mathbf{H} can be estimated from the image by matching a minimum of 4 coplanar points, as there are 8 unknowns. It can then be decomposed into components \mathbf{R} , \mathbf{t} , \mathbf{n} :

$$\mathbf{H} = \mathbf{R} + \frac{\mathbf{t} \mathbf{n}^T}{d}, \quad (2.37)$$

where \mathbf{R} is a rotation matrix, \mathbf{t} is a translation vector, \mathbf{n} is the normal of the plane and d is the distance to the plane (see Fig. 2.5). There are a number of numerical and analytical methods to decompose \mathbf{H} . For more details see [MV07].

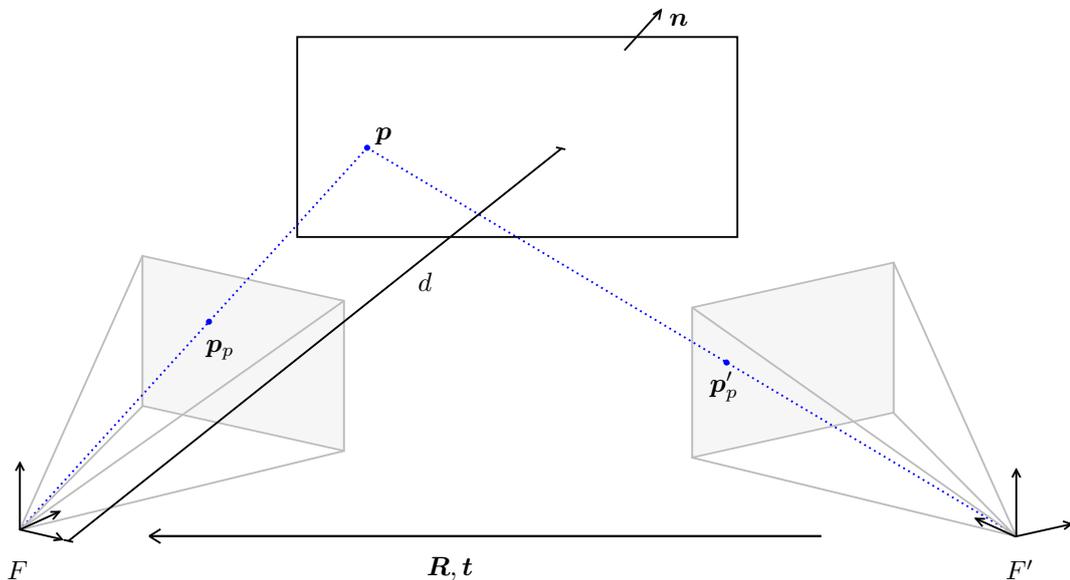


Figure 2.5.: Elements involved in homography decomposition. Point \mathbf{p} on the plane with normal \mathbf{n} is projected onto \mathbf{p}_p and \mathbf{p}'_p in views F and F' . The cameras are related by rotation matrix \mathbf{R} and translation vector \mathbf{t} .

3. Related Work

3.1. Measuring BRDFs

Accurate measurement of BRDFs is traditionally performed with active approaches, where lighting and sensor/camera position can be actively controlled. Contrarily, in passive approaches, lighting and camera position cannot be altered, instead the reflectance is recovered from given images of a scene.

A simple active approach to measure a BRDF is a gonireflectometer [WSB⁺98]. As the measurement with such devices is very time consuming, a number of image-based techniques were proposed. In order to sample a large number of directions of a planar surface with a single image, Ward uses a curved mirror [War92]. Otherwise, if the surface is curved, a mirror is not necessary, but the geometry of the object has to be known. In this case, the BRDF of cylindric surfaces or spheres [MWLT00] and even arbitrary convex surfaces [MWL⁺99] can be measured. However, these methods work on homogeneous surfaces only.

In order to measure spatially varying BRDFs in a reasonable time, Lensch et al. utilize the fact that in many cases surfaces consist of only a small set of basis materials [LKG⁺03]. By exploiting the coherence of surface points belonging to the same material, they recover the basis materials of an object with known geometry and assign a weighted combination of these basis materials to each surface point. Afterwards, the normals are refined with the help of the recovered BRDFs to add geometric detail that was lost during the initial range scan.

In passive approaches, lighting and camera position cannot be altered, instead the reflectance is recovered from given images of a scene. There are various approaches for the different combinations of input data: single or multiple views with varying viewpoint or lighting, known or unknown, static or varying illumination, single point-light illumination or complex natural illumination, known and unknown geometry.

Haber et al. estimate the reflectance of a scene with known geometry from multiple images with varying, unknown illumination [HFB⁺09]. The inverse rendering is formulated as an optimization process using a system of bilinear equations. It is solved iteratively by alternating between the estimation of reflectance and illumination. The BRDFs are represented as a linear combination of basis BRDFs. For fast computation the optimization operates in the Haar-wavelet basis [NRH04].

Similarly, Romeiro and Zickler use bilinear equations as likelihood in a probabilistic

approach to recover the reflectance of a homogeneous surface from a single image with unknown illumination [RZ10]. In this case, it is difficult to avoid a solution where the material acts like an ideal mirror and the observed reflection is projected onto the environment. By exploiting statistics of natural lighting, this solution is avoided. The basis functions used for the BRDFs are learned via non-negative matrix factorization of the entire MERL BRDF database [MPBM03a].

In contrast to the previous approaches, Goldman et al. estimate surface reflectance properties and normals from multiple images with fixed viewpoint but different illumination conditions [GCHS10]. Again, it is assumed that the BRDF at a point is composed of several fundamental materials, in this case modeled by a Ward BRDF [War92]. In an iterative process, the parameters of the fundamental BRDFs, the normals and material weight maps are updated.

The fact that surfaces of objects are often composed of a few basis materials is further investigated by Weistroffer et al. [WWHL07]. In their work, the BRDF at each surface point is composed of a weighted sum of materials, which are formed of several basis BRDFs. They describe a technique which decomposes scattered measurements into blending weights assigned to each point and weights for the basis BRDFs.

More complex scenes demand for the inclusion of effects from global illumination and shadows. Yu et al. recover the reflectance properties of surfaces in a small indoor scene from multiple images captured from varying viewing points, known geometry and lighting [YDMH99]. The algorithm detects specular highlights and uses these locations to update the parameters of a Ward BRDF model iteratively.

An example for recovering geometry and reflectance of large outdoor scenes is given by Debevec et al. [DTG⁺04]. They created a relightable model of the Parthenon from range scans and photographs captured under natural, known illumination. A Monte-Carlo global illumination algorithm is used for the inverse rendering task. A small number of BRDFs with distinct diffuse colors is measured under controlled point-light illumination and fitted to a Lafortune [LFTG97] model. Based on the diffuse color observed in the images, the per-point specular component is interpolated from the measured BRDFs.

3.2. Normal Reconstruction and Shape From Shading

All approaches listed above share the assumption that the geometry of the scene is known or can be computed via Photometric Stereo as in [GCHS10]. Photometric stereo was introduced by Woodham [Woo80] as a technique to reconstruct the shape of a Lambertian surface from images taken under varying point-light illumination. If the lighting is static or only a single image is given, shading information can be exploited to recover normals. This problem is named shape-from-shading.

Shape-from-shading is an ill-posed problem that can be solved for example via energy

minimization with smoothness constraints, such as proposed by Ikeuchi and Horn [IH89]. In a survey Zhang et al. compare several shape-from-shading algorithms [ZTCS99]. These algorithms assume surfaces with Lambertian reflectance and distant illumination from a known direction as input data. In their conclusion they note that all algorithms produce poor results for synthetic data as well as real-world scenes.

More recent work takes more information into account. Prados and Faugeras demonstrate that if a pin-hole camera model and a light source at the optical center is assumed, shape from shading can be modeled as a well-posed problem [PF05]. Vogel et al. extend this model by incorporating the Phong reflection model to account for non-Lambertian surfaces [VBW08].

Other work tries to resolve the ill-posedness by user interaction. In the method proposed by Wu et al., the normals corresponding to high-frequency details are estimated by an automatic minimization approach. Afterwards, the low-frequency components are specified by the user [WSTS08].

3.3. Intrinsic Image Decomposition

In order to estimate normals, the shading of a scene has to be known. Intrinsic image decomposition tries to separate images into albedo and shading information, resulting in an albedo image that shows only the diffuse color of the surfaces and a shading image that shows only the influence of the light sources on the surfaces. If only a single image is given, this task is inherently ill-posed, as one tries to recover two images from one. Therefore, the decomposition has to be based on a few assumptions. For Mondrian images (composed of patches with constant reflectivity), the reflectance is constant within a patch and the shading varies smoothly. Thus, isolating sharp edges keeps reflectance information and removes shading [Hor86]. Similarly, large derivatives in the chromaticity channel of color images can be used as an indicator for reflectance, as proposed by Funt et al. [FDB92]. In real-world scenes, these assumptions often do not hold and more complex classification schemes for edges were developed. Tappen et al. train a classifier that distinguishes shading from reflectance edges by their surrounding gray-scale pattern [TFA05]. Shen et al. use local texture information to group pixels that most likely have the same shading [STL08].

3.4. Combinations

This work deals with the problem of automatically recovering both normals and spatially varying reflectance from real-world photographs of planar objects and a captured environment map. Therefore, intrinsic image decomposition, normal reconstruction and BRDF estimation have to be combined. This is similar to the

AppGen system of Dong et al. [DTPG11], however, they use a single input image with unknown illumination. Therefore, they have to rely on user interaction in certain steps to refine the automatically generated results. Furthermore, they do not recover the specular component of the BRDF, instead the user has to choose the optical properties of the materials from a database of BRDFs. From the input image, they remove highlights by thresholding and retrieve diffuse shading as well as reflectance by an optimization process. The process is based on two assumptions: pixels in a local region with the same chromaticity share the same reflectance (local albedo assumption) and all pixels in a local region have the same shading (local shading assumption). Regions where this assumption is violated are refined by user interaction. Afterwards, the user specifies the lighting direction and the normals are reconstructed with an algorithm based on [WSTS08], but enhanced to recover more high-frequency normals. In the last step, the user selects small regions of constant specular reflectance and chooses the corresponding BRDFs. A combination of these basis BRDFs is then automatically assigned to the remaining pixels.

4. System Overview

In this work, a system is presented for reconstructing normals and spatially varying BRDFs of planar surfaces from two images and incident illumination. The lighting is assumed to be distant, and apart from that, self-shadowing and inter-reflections are ignored. Basically, a solution to the inverse rendering problem is searched:

$$L_o(\mathbf{x}, \boldsymbol{\omega}_o) = \int_{\Omega_+} f_r(\boldsymbol{\omega}_i, \mathbf{x}, \boldsymbol{\omega}_o) L_i(\boldsymbol{\omega}_i) (\boldsymbol{\omega}_i \cdot \mathbf{n}) d\boldsymbol{\omega}_i. \quad (4.1)$$

For each surface point \mathbf{x}_i , two measurements $L_o(\mathbf{x}_i, \boldsymbol{\omega}_o^0)$ and $L_o(\mathbf{x}_i, \boldsymbol{\omega}_o^1)$ are given, because the point is observed from different angles in the two images. Furthermore, the incident illumination $L_i(\boldsymbol{\omega}_i)$ is known. The task is to determine normal \mathbf{n} and BRDF $f_r(\boldsymbol{\omega}_i, \mathbf{x}, \boldsymbol{\omega}_o)$ for each point. Depending on the BRDF model, f_r has for example 7 unknowns per pixel: diffuse color, specular color and a coefficient to control the shape of the specular lobe (roughness parameter). Besides, there are three unknowns for normal \mathbf{n} , or two if the normal is expressed in spherical coordinates. All in all, there are more unknowns than knowns, so the problem is ill-posed and cannot be solved independently for each point. Therefore, it is necessary to find and utilize some coherence between points. This chapter provides the basic ideas of this work and gives an overview of the steps necessary to reconstruct normals and BRDFs (see Fig. 4.1).

4.1. Image Capturing and Scene Reconstruction

The process of capturing images should be as simple as possible. A plane surrounded by markers is captured from two or more different locations with a dynamic range that is sufficient to include all lighting effects. Afterwards, the incident illumination is captured at the center of the plane using a fish-eye lens. Furthermore, the user has to provide the size of the plane as well as the size of the environment, approximated by a sphere surrounding the plane. This means that the radius of the sphere has to be specified.

In the first step, a 3-dimensional representation of the scene is built from the captured images. The normal of the plane and the camera position in each view is estimated by exploiting the homography between the images. Afterwards, the incident illumination captured with a fish-eye lens is aligned to the views and stored in an environment

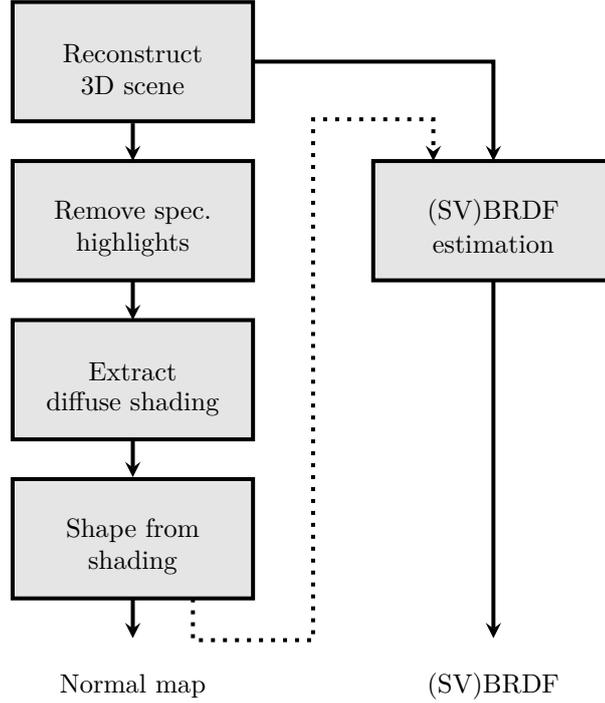


Figure 4.1.: Steps to estimate normal map and (spatially varying) BRDF. Diffuse shading information is used to estimate the normal map. BRDF estimation can either incorporate the normal map or assume a fully planar surface.

map. The reconstructed scene now contains all geometrical information necessary to estimate normals and BRDF.

4.2. Normal and BRDF Estimation

According to Eq. (4.1), the observed pixels in the images are the result of an integration in which normal direction and BRDF are combined. Now, this combination should be factorized into its individual parts. For this purpose, the integral is split into diffuse and specular contributions. Now, the two images can be described by its intrinsic images with the following equation [GJAF09]:

$$I_i(x, y) = \frac{\rho(x, y)}{\pi} E(x, y) + C_i(x, y), \quad (4.2)$$

where $I_i(x, y)$ is the color at pixel (x, y) in the i -th image, $\rho(x, y)$ is its albedo, $E(x, y)$ is the irradiance and $C_i(x, y)$ is the specular term. E and C_i are influenced by the normal direction, so information about the normal is contained in E and C_i . However, C_i is only present if the BRDF has significant specular contribution and if

light sources are reflected from the surface into the camera. Typically, this is only the case on small regions in the images. For this reason, E is used to reconstruct normals of the surface. Images I_i are decomposed into their intrinsic images to obtain an approximation of E . The decomposition algorithm is specifically designed to handle textured surfaces with high-frequency normals. Using the approximated E , the normals at each surface point are estimated and stored in a normal map. An algorithm to reconstruct normals from diffuse materials under arbitrary illumination is introduced. It makes use of the fact that the environment can be represented by low-order spherical harmonics. This allows to formulate the reconstruction problem as a system of linear equations.

BRDF estimation is performed by fitting a BRDF model to the observations. In this work, the BRDF is represented by a linear combination of basis materials, so linear least-squares fitting can be used. This representation shares advantages of analytic and data-driven models. Like most analytic models, a BRDF can be represented by a small number of coefficients, which results in low storage requirements, especially if the BRDF is spatially varying. Furthermore, it is convenient in optimization, because only a few coefficients have to be estimated. On the other hand, flexibility is similar to data-driven models, as the range of representable materials is only limited by the basis functions.

The estimation is performed in a two-step process. At first, for each basis material the plane is rendered under the captured illumination. Then, the difference between a linear combination of the rendered planes and the captured images is minimized, i.e., the BRDF is estimated by linearly interpolating between the rendered planes. Rendering is performed by Monte-Carlo integration. To speedup the process, the basis materials are expressed as Ashikhmin-Shirley BRDFs. Therefore, it is possible to generate samples according to the probability distribution of the BRDF.

If each surface point belongs to the same BRDF because the material is homogeneous, all points can be used in the optimization. This means that a large number of measurements is available to estimate a single BRDF. However, if the BRDF is spatially varying, the problem will become more complex. As mentioned earlier, a single point is not sufficient for estimation. In this work, two approaches to deal with spatially varying BRDFs are presented.

In the first approach, the surface is clustered into regions of pixels sharing the same BRDF. It is assumed that each point corresponds to exactly one BRDF. Clustering is performed by analyzing residuals of hypotheses generated by a RANSAC-like random sampling process.

The second approach makes use of Eq. 4.2 to independently estimate diffuse and specular components of the BRDF. While a single point of a diffuse material contains enough information to reliably measure its albedo, this is not the case for specular materials. The shape of the specular lobe, i.e., its roughness, can only be estimated if multiple points are taken into account. Based on this fact, it is shown how to extract

albedo (per-pixel information) and specular parameters shared by all pixels. Diffuse and specular regions on the plane are detected by examining differences between the captured images, as the observed color values will only vary for specular materials (assuming appropriate lighting conditions).

The following chapters discuss the algorithms to form a 3-dimensional representation of the scene, decompose intrinsic images, reconstruct normals, and estimate spatially varying BRDFs in more detail. Each chapter contains discussion and evaluation, showing possibilities and limits of the approaches. Examples are given for synthetic and real-world data sets.

4.3. Problems

There are some problems that can occur when dealing with photographs, especially if they are captured with mobile devices. Specular materials exhibiting highlights have a large dynamic range that has to be captured entirely to reconstruct the specular lobe as well as subtle changes in albedo at non-highlighted regions. The same applies to environments, which typically have an even larger dynamic range. Ideally, multiple images at different exposures are captured and merged. However, under low-lighting conditions, images suffer from noise. Therefore, the estimation algorithms have to be insensitive to noise. Another challenge is the imperfect calibration, particularly the relationship between views and environment. If the environment is not captured perfectly in the middle of the plane or the device is slightly rotated or tilted, the computed reflection does not match the reflected light in reality. The same happens if the environment can not be approximated by a sphere or if the estimated homography is not exact. Consequently, the system has to correct such errors.

4.4. Notes on Evaluation

The techniques presented in the following sections are evaluated with both synthetic and real scenes. Synthetic scenes are rendered using Monte-Carlo sampling and distant illumination from an environment map. Real scenes are captured with an NVIDIA Tegra prototype tablet running Android. It has a 1080p front-facing camera and two 5 MP back-facing cameras for stereo imaging. An inexpensive fish-eye adapter is attached to the front-facing camera. Only a single back-facing camera is used at the moment because of limited image resolution in stereo mode, so the two images needed for estimation are taken one after another. As the FCam library is available on the tablet [TPP12], most camera parameters can be controlled manually (exposure, ISO value, focus, and white balance). Furthermore, all high dynamic range images shown in this work are tone-mapped with $f(c) = c/(c + 1)$, $c = (r, g, b)$ (pixel color) and the brightness is adjusted to ensure that all details are visible.

5. Scene Reconstruction

After acquiring the images with a camera, the first step is to reconstruct a 3D scene. The images have to show at least two views of the scene, for example taken with a stereo camera in a single shot. For flexibility, it is not necessary to use a stereo camera. Two shots from slightly different viewpoints are sufficient.

Furthermore, the incident illumination at each surface point has to be known. Using a fish-eye lens with an 180° angle of view, the whole upper hemisphere can be captured in one shot. However, due to parallax effects, the incident illumination is different for each point, so ideally one image of the upper hemisphere per surface point is needed. Fortunately, if a small surface is located in a huge environment, it is feasible to assume that the incident illumination is constant for each point, therefore capturing it from a single point should be sufficient. An example for the three images necessary to reconstruct the scene and estimate a BRDF are shown in Fig. 5.1. The overall process to reconstruct the scene consists of the following steps:

1. Generate high dynamic range images
2. Find correspondences using feature points or markers and estimate homography
3. Decompose homography
4. Extract textures
5. Build environment map
6. Convert environment map to spherical harmonics representation

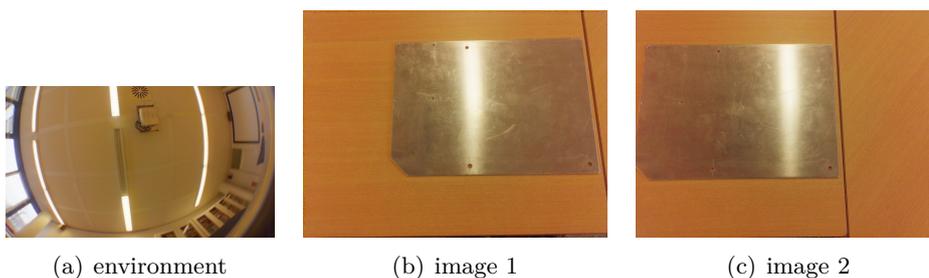


Figure 5.1.: At least three images are necessary to reconstruct the scene geometry and estimate the BRDF: the incident illumination captured with fish-eye lens and two images of the plane.

ach

5.1. Generate High Dynamic Range Images

For an ideal estimation of the BRDF, the images have to capture the whole dynamic range of the scene. In the prototypical implementation, the high dynamic range (HDR) images of real scenes are reconstructed from several photographs with different exposures [RHP⁺10]. For calibration and HDR image generation, `pfstools` [MKMS07] is used. Furthermore, the images are captured without a tripod and need to be aligned before HDR generation. This task is accomplished by `align_image_stack` from the Hugin project [D'A]. Note that taking multiple images with different exposure times is only necessary if it is not possible to capture the whole dynamic range within a single image.

5.2. Estimate Homography

In order to estimate the eight unknowns of the homography matrix, the coordinates of four corresponding points in both images have to be known [MV07]. This problem is known as “correspondence problem”, where a set of points in one image has to be found in the other image. For a homography, the points have to be coplanar, i.e., they have to lie in the same plane. Typically, corresponding points are detected via feature-based algorithms. A feature-detector is used to find salient image points in both images and extract a descriptor vector at their locations (e.g., SIFT [Low99]). Then, the features of one image are matched with the features in the other image, leading to several correspondences.

However, this only works for textured surfaces where feature points can be found. In mostly homogeneous regions, the algorithm will fail. To overcome this problem, markers can be placed onto the surface. If four unique markers are used and if all of them are visible in both images, it will be easy to extract their image coordinates and use them to estimate the homography. In the prototypical implementation, the 2D coordinates of four markers or point correspondences have to be supplied manually by the user. The points are not only used to estimate the homography, but also to define the bounding box on the plane in which the spatially varying BRDF is estimated.

The markers are located in the images at coordinates \mathbf{x}_i^I and $\mathbf{x}_i^{I'}$. With the intrinsic camera parameters in matrix \mathbf{K} , the coordinates on the image plane in the camera coordinate system can be calculated:

$$\mathbf{x}_i^C = \mathbf{K}^{-1}\mathbf{x}_i^I \quad , \quad \mathbf{x}_i^{C'} = \mathbf{K}^{-1}\mathbf{x}_i^{I'} . \quad (5.1)$$

The intrinsic parameters of the camera have to be known from calibration. This implementation relies on the algorithm from the open-source computer vision library OpenCV v2.31 [Bra00] to find the camera matrix and distortion coefficients.

The homography matrix \mathbf{H} is estimated by solving the linear system of equations up to the unknown scale factor w :

$$w \mathbf{x}_i^{C'} = \mathbf{H} \mathbf{x}_i^C. \quad (5.2)$$

5.3. Decompose Homography

Afterwards, \mathbf{H} is decomposed into \mathbf{R} , \mathbf{t} and \mathbf{n} with the decomposition algorithm of [MV07]. This yields four possible decompositions, two of them can be eliminated by taking into account that all the points have to be in front of the camera. In order to choose from the remaining two solutions, either a fifth marker or some heuristic has to be applied, as both solutions are perfectly valid for the four markers. In this implementation, the solution with the smaller rotation angle is chosen, as the viewpoints are either near to each other or from a stereo camera with no rotation at all. In case the heuristic fails, it is possible to choose the correct solution manually.

The position and rotation of the cameras are then transformed to world space. For simplicity, the local coordinate system of the first camera is defined as world space, so the first camera is always located at $(0, 0, 0)^T$. Fig. 5.2 shows the geometry of the scene.

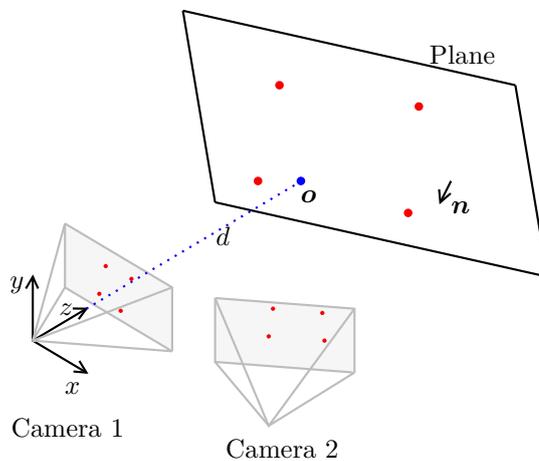


Figure 5.2.: Reconstructed scene. The first camera defines the coordinate system of the world. The z-axis intersects the plane at \mathbf{o} and with distance d from the center of the first camera. The markers (red dots) define the bounding box of the region for which the BRDF is estimated.

5.4. Extract Texture

In order to estimate the BRDF, the outgoing radiance $L_o(\mathbf{x}, \boldsymbol{\omega}_o)$ has to be measured for each point on the surface. The radiance corresponds to a single ray originating from the surface at point \mathbf{x} and directed towards the optical center of the camera. Thus, the points on the surface are discretized and for each point on the grid, the radiance is extracted from the images captured by the cameras. Once the 3D world space coordinates of the surface points are known, the associated pixel coordinates in each image can be computed.

When camera poses and plane normal are known, the positions of the four marker points can be calculated. Thus, for each marker, rays are shot from the origins of the cameras through the points on the image plane. The 3D coordinates of the intersection point determine the 3D coordinates \mathbf{x}_i of the marker in world space.

Now, the normal $\mathbf{n} = (n_x, n_y, n_z)$ and four points \mathbf{x}_i of the plane are known, and for further calculations, the origin \mathbf{o} of the plane is defined as the point where the plane intersects the z-axis in world space. With one of the four points, the distance d of the plane to the origin of the world coordinate system can be calculated ($d = \mathbf{n} \cdot \mathbf{x}_o$). Then, the plane intersects the z-axis at the point $\mathbf{o} = (0, 0, d/n_z)^T$. In addition, the basis vectors of the plane \mathbf{u} and \mathbf{v} are generated based on the normal and the marker points. As mentioned earlier, the markers define the region that has to be analyzed. As the markers probably do not define a perfect rectangle, the bounding box of the markers on the plane is used as region boundaries.

From the pixels included in this bounding box, an image $I_k(x, y)$ is created that contains the radiance of the surface points measured from camera k . This is done by sampling the surface uniformly in directions \mathbf{u} , \mathbf{v} and identifying the corresponding pixel in the camera image. The pixel values are linearly interpolated if necessary. Additionally, for each pixel in $I_k(x, y)$ the direction of the ray from surface point to camera center is stored.

The result of this step is one image per camera with each pixel corresponding to the radiance from a single (discretized) surface point and its associated direction (see Fig. 5.3 for an example).



Figure 5.3.: Extracted textures of the plane from the captured images shown in Fig. 5.1.

5.5. Build Environment Map

In the scene, the incident lighting is represented by an environment map parameterized in latitude-longitude format (equirectangular projection, see Fig. 5.4). The incident illumination $L_i(\omega_i)$ is captured with a fish-eye lens, which measures the whole upper hemisphere in a single shot. In the prototypical implementation, it is assumed that the camera is placed onto the surface before taking the picture, thus ensuring a measurement that is as best as possible. In this case, the fish-eye’s optical axis points in the direction of the surface normal. Camera and lens are calibrated using OCamCalib v2.0 (Omnidirectional Camera Calibration Toolbox for Matlab) [SMS06]. For a calibrated setup, the library provides functions that convert between world space coordinates and camera coordinates.



Figure 5.4.: Full environment map in latitude-longitude format ($-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}$, $0 \leq \phi \leq 2\pi$).

For all synthetically generated images, a fish-eye lens with equisolid angle projection is simulated. This has the advantage that the solid angle subtended by all pixels is equal. The distance r of a point from the image center is calculated with the following formula, given the focal length f and the angle from the optical axis θ' :

$$r = 2f \sin \frac{\theta'}{2}. \quad (5.3)$$

The radius of the entire circular region in the image corresponds to an angle of $\theta' = 180^\circ$, leading to a focal length of $f = \frac{1}{\sqrt{2}}$.

For synthetic and real scenes, the fish-eye image is converted to the spherical environment map in the following way. First, the fish-eye image is downsampled to match the resolution of the environment map, i.e., for an environment map with size $w \times h$, the image is downsampled to $w/2 \times h$. Then, for each pixel (θ, ϕ) of the environment map, the corresponding pixel in the fish-eye image is determined by projecting the environment map pixel into the fish-eye image. The value is copied via point sampling. It is important that the fish-eye image is properly downsampled to nearly the same resolution as the environment map, because otherwise the simple point sampling could miss small light sources.

Generally, the environment is assumed to be distant, so the incident illumination is the same at each surface point. In other words, a ray from the camera hits a

surface point \mathbf{x}_i , but (for mirroring materials) is reflected from the center point \mathbf{x}_c of the surface. This assumption is approximately correct, especially if a small surface is located in a large environment. However, this is not always the case. If the environment can be approximated by a sphere and the size of the region on the plane as well as the radius of the sphere is known, a simple correction can improve the results. As illustrated in Fig. 5.5, instead of directly using direction \mathbf{d} , a ray originating at the surface point is intersected with the sphere from the environment. The corrected vector \mathbf{d}_c is then used to look up the incident illumination in the environment map.

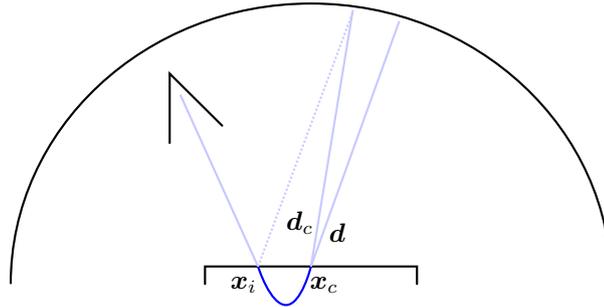


Figure 5.5.: Off-center reflection vector correction.

5.5.1. Handling Imperfect Calibration

The alignment of the environment is a critical factor that has large influence on the quality of the estimated BRDF. If the captured images show a reflection of the light source on the surface, it will be important that the reconstructed 3D scene allows to render the bright spot at exactly the same position, otherwise the estimated BRDF is too smooth. However, there are several sources of error, e.g., inaccurate homography decomposition or slightly misplaced camera while capturing the environment. Perfect calibration minimizes this errors at the cost of increased complexity. As it should be possible to use hand-held photographs from mobile devices, complex calibration is not possible. Instead, the alignment of the environment is improved by analyzing the contents of the images.

At first, highlights are extracted from the captured views and the corresponding light sources from the environment map. For the views, the algorithm described in Sec. 8.2.2 is used. For the environment map, a threshold is chosen based on the assumption that light sources are outliers in normal distributed data points. All pixels with an intensity greater than two standard deviations from the mean of the environment map are classified as light sources.

Afterwards, the surface is assumed to be a mirror and the scene is rendered with the environment map. This allows to examine the rendered views whether the mirrored

light source is at the same position as in the views captured with the camera. For this purpose, the dot product between the pixels previously classified as highlight on the surface or light source in the environment is computed, yielding a measure for similarity between rendered and observed views. A high similarity indicates that the environment is correctly aligned. The process is iterated with various values for the parameters influencing the alignment of the environment. Actually, each parameter that affects the scene has to be optimized. However, due to the great number of parameters, this is a time-consuming process. Therefore, it was chosen to only optimize the two coordinates that determine the center of the environment's sphere. Instead of being positioned directly in the center of the plane, it is moved on the plane in small steps. This corresponds to moving the camera on the plane when capturing the environment with the fish-eye lens.

Due to imperfect lenses and sensors, the intensities of the color channels can vary from one camera to another. This will become a problem if different cameras are used to capture views and environment map, e.g., back-facing and front-facing camera of the device. In order to adjust the colors, a point on a diffuse white material has to be specified. As the color at the point is known, it is possible to calculate a correction factor that is applied to all scenes captured with these cameras.

5.6. Convert Environment Map

When estimating normals and BRDFs, it is sometimes necessary to calculate the irradiance at diffuse surfaces. Sec. 2.7 describes irradiance environment maps and their spherical harmonics representation as a method to handle such problems with little computational effort. Therefore, in the final step of scene reconstruction, the environment map is transformed into the spherical harmonics basis.

The computation is straightforward and follows Eq. (2.35). For each color channel, the 9 coefficients L_{lm} are computed from the environment map $L(x_\theta, y_\phi)$ in the following way:

$$L_{lm} = \sum_{x=0}^{w-1} \sum_{y=0}^{h-1} L(x_\theta, y_\phi) Y_{lm}(x_\theta, y_\phi) \sin\left(\frac{\pi y_\theta}{h-1}\right) \frac{\pi}{h} \frac{2\pi}{w}, \quad (5.4)$$

where $Y_{lm}(x_\theta, y_\phi)$ are the basis functions in pixel coordinates.

6. Normal Estimation

This chapter deals with the problem of reconstructing normal maps from a single image and known illumination. The problem can be divided into two parts. At first, the image is separated into albedo, diffuse shading, and specular shading. Next, the diffuse shading can be used to estimate the normals of the surface. In the following, an algorithm to extract diffuse shading from an image is presented. Afterwards, an approach to estimate normals from diffuse shading is introduced.

6.1. Intrinsic Image Decomposition

As mentioned, normal reconstruction only works for images of objects with homogeneous diffuse BRDF, i.e., Lambertian objects. However, the BRDF of natural materials exhibits complex spatially varying diffuse and specular appearance. As the reflectance of such materials should be estimated in this work, this is clearly not sufficient. Thus, the image or radiance $I_i(x, y)$ has to be decomposed into its intrinsic images: albedo $\rho(x, y)$, diffuse shading or irradiance $E(x, y)$ and specularity $C_i(x, y)$ (cf. Eq. (4.2)). For RGB images, the color at each pixel is described by a vector. In order to simplify notation, the index i is dropped in the following description.

Extracting shading $E(x, y)$ is an ill-posed problem. In the following, an approach is presented that takes the specific circumstances into account in dealing with close-up images of materials on planar surfaces. This is in contrast to the approaches discussed in Sec. 3.3, where mostly images of 3D objects or complex natural scenes are considered.

Under the assumption that light sources and specular BRDF component are white, the decomposition problem is slightly easier. Shading and specularity are no longer vectors at a pixel, instead they become scalar values. For each RGB pixel, this yields $\mathbf{i} = e\boldsymbol{\rho} + c(1, 1, 1)$. \mathbf{i} denotes a pixel of $I(x, y) \in \{\mathbf{i}_{00} \dots \mathbf{i}_{mn}\}$, the same applies to e , $\boldsymbol{\rho}$ and c). Moreover, as the object is planar with a few high-frequency normals, shading information is mostly contained in the high frequencies of the image intensity. Furthermore, it is assumed that albedo changes always affect chroma and intensity of the image, but shading changes only affect the intensity.

With these assumptions, decomposition is approximatively accomplished by bilateral filtering [PKTD09]. In the first step, each pixel $\mathbf{i} = (i_r, i_g, i_b)$ is separated into

intensity i_t and chroma \mathbf{i}_c :

$$i_t = (i_r + i_g + i_b)/3 \quad (6.1)$$

$$\mathbf{i}_c = (i_r/(3i_t), i_g/(3i_t), i_b/(3i_t)) = (i_r/(3i_t), i_g/(3i_t), 1 - i_r/(3i_t) - i_g/(3i_t)). \quad (6.2)$$

Thereafter, a bilateral filter is used to remove details in $I_t(x, y) \in \{i_{t,00} \dots i_{t,mn}\}$ that are not present in $I_c(x, y) \in \{\mathbf{i}_{c,00} \dots \mathbf{i}_{c,mn}\}$. The filter blurs $I_t(x, y)$ spatially with a Gaussian function weighted by the distance between chromaticity. The spatial weighting between point $\mathbf{x} = (x, y)$ and $\mathbf{x}' = (x', y')$ is given as

$$b(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{2} \frac{|\mathbf{x} - \mathbf{x}'|^2}{\sigma_s^2}\right). \quad (6.3)$$

Similarly, the chroma values are weighted with

$$r(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{2} \frac{|I_c(\mathbf{x}) - I_c(\mathbf{x}')|^2}{\sigma_r^2}\right). \quad (6.4)$$

The filtering process results in an image $I_f(x, y) \in \{i_{f,00} \dots i_{f,mn}\}$, where regions with similar chromaticity are blurred, but edges occurring in chroma and intensity are preserved. It is now assumed that $I_f(x, y)$ contains low-frequency albedo intensity and specularity. By subtracting pixels i_f from i_t , the high-frequency components are obtained, and with the above assumptions, these components are related to shading from changes in normal direction:

$$\tilde{e} = i - i_f \quad (6.5)$$

$$\tilde{\rho} = i_f \mathbf{i}_c. \quad (6.6)$$

Images $\tilde{e}(x, y) \in \{\tilde{e}_{00} \dots \tilde{e}_{mn}\}$ and $\tilde{\rho}(x, y) \in \{\tilde{\rho}_{00} \dots \tilde{\rho}_{mn}\}$ are only crude approximations to $e(x, y)$ and $\rho(x, y)$. $\tilde{\rho}(x, y)$ includes specular highlights as well, a separate estimate for $c(x, y)$ is not generated. However, this does not influence normal reconstruction, as it only uses $e(x, y)$. Furthermore, the observed brightness generated from the light source is mostly contained in $\tilde{\rho}(x, y)$. Due to the subtraction, $\tilde{e}(x, y)$ only contains high-frequency irradiance information.

Removing highlights The extraction of $\tilde{e}(x, y)$ can be made more robust if specular highlights are removed before decomposition. If multiple images with varying highlight position are given as input, the specular component $C_i(x, y)$ can be eliminated by taking the minimum of all images $I_i(x, y)$ at each pixel. This generates a single image $\tilde{I}^d(x, y)$ that contains only the diffuse component:

$$\tilde{I}^d(x, y) = \min_i I_i(x, y). \quad (6.7)$$

$\tilde{I}^d(x, y)$ is only an approximation to the real diffuse shading and albedo image $I^d(x, y)$ that fails in regions with overlapping specular highlights.

Combination with normal reconstruction Normal reconstruction needs an image of a surface with homogeneous diffuse BRDF under arbitrary illumination. Although all low-frequency irradiance information is removed in $\tilde{e}(x, y)$, it can be used as input data. Normal reconstruction requires images of a homogeneous diffuse surface captured by a camera. This corresponds to radiance measurements per point. Given this input data, the algorithm will estimate albedo of the surface and irradiance per point. Using $\tilde{e}(x, y)$ is almost the same as using an image $I'(x, y)$ that shows the same surface as $I_i(x, y)$, but with small homogeneous albedo instead of the original colors (i.e., albedo of the surface in $I_i(x, y)$ is set to uniform gray). In conclusion, the prerequisites for normal reconstruction are fulfilled by $\tilde{e}(x, y)$, allowing it to be used as input.

Discussion and Evaluation The presented approach is simple and straightforward, but nevertheless provides plausible results, as shown in Fig. 6.1 ($\sigma_s = 16$, $\sigma_r = 0.05$). A single image of a surface with varying albedo and specular properties is rendered in the *pisa* environment (Fig. 6.3). The light sources are not purely white, but depending on the reflection properties, this has only minor consequences. The rendered image is used as input for the decomposition algorithm, which splits the image into intensity $I_t(x, y)$ and chroma $I_c(x, y)$ and estimates the diffuse shading $\tilde{e}(x, y)$ and albedo $\tilde{\rho}(x, y)$. The figure also shows the real albedo and a rendering of the surface with diffuse shading only, which can be compared to the output of the algorithm. As the brightness of $\tilde{e}(x, y)$ and $\tilde{\rho}(x, y)$ do not match the brightness of the renderings, the values are scaled for better illustration.

All surfaces are rendered with high-frequency normals, either noise-like (Fig. 6.2(a), (c), (d)) or with a regular pattern (Fig. 6.2(b)). The surfaces in Fig. 6.2(a) and Fig. 6.2(b) exhibit diffuse reflection only and the decomposition works very well. In Fig. 6.2(c), the blue colored parts do not have any normals and are highly specular, resulting in a large highlight. This is an example where the colored light source introduces artifacts in the shading image. As the material mirrors the light source, the observed color coincides with the color of the light source. This leads to a change in chromaticity and, thus, $I_t(x, y)$ is not fully blurred in the blue regions. In Fig. 6.2(d), these artifacts are barely visible, only the edges in the top left corner are slightly flawed.

Fig. 6.2 shows two examples of surfaces that violate the assumptions used in the decomposition. In Fig. 6.2(a), the chromaticity does not change at the edges of the bricks, so the bright color at the joints is not classified as albedo. In Fig. 6.2(b), a highly specular material is disturbed by normals. The material is rendered with the *ennis* environment (Fig. 6.3), where a very bright light source illuminates the surface from the right side. The high-frequency shading from the diffuse blue color is hardly visible in the image, instead the highlights create high-frequency changes in intensity and chroma too. Consequently, diffuse shading on the one hand and specular shading as well as albedo on the other hand can not be separated.

Real world examples are given in Fig. 6.4. Decomposition creates reasonable results for images with homogeneous albedo (*ingrain wallpaper, fabric, metal*), especially shading looks quite well. The estimated albedo contains some high-frequency artifacts (*fabric, metal*) or invalid low-frequency intensity variations (*ingrain wallpaper*). *Cork* has mixed variations in albedo intensity, chroma, and normals which cannot be decomposed without further information (e.g., multiple lighting directions). Details in *wood* mainly affect intensity, therefore the algorithm wrongly classifies them as shading.

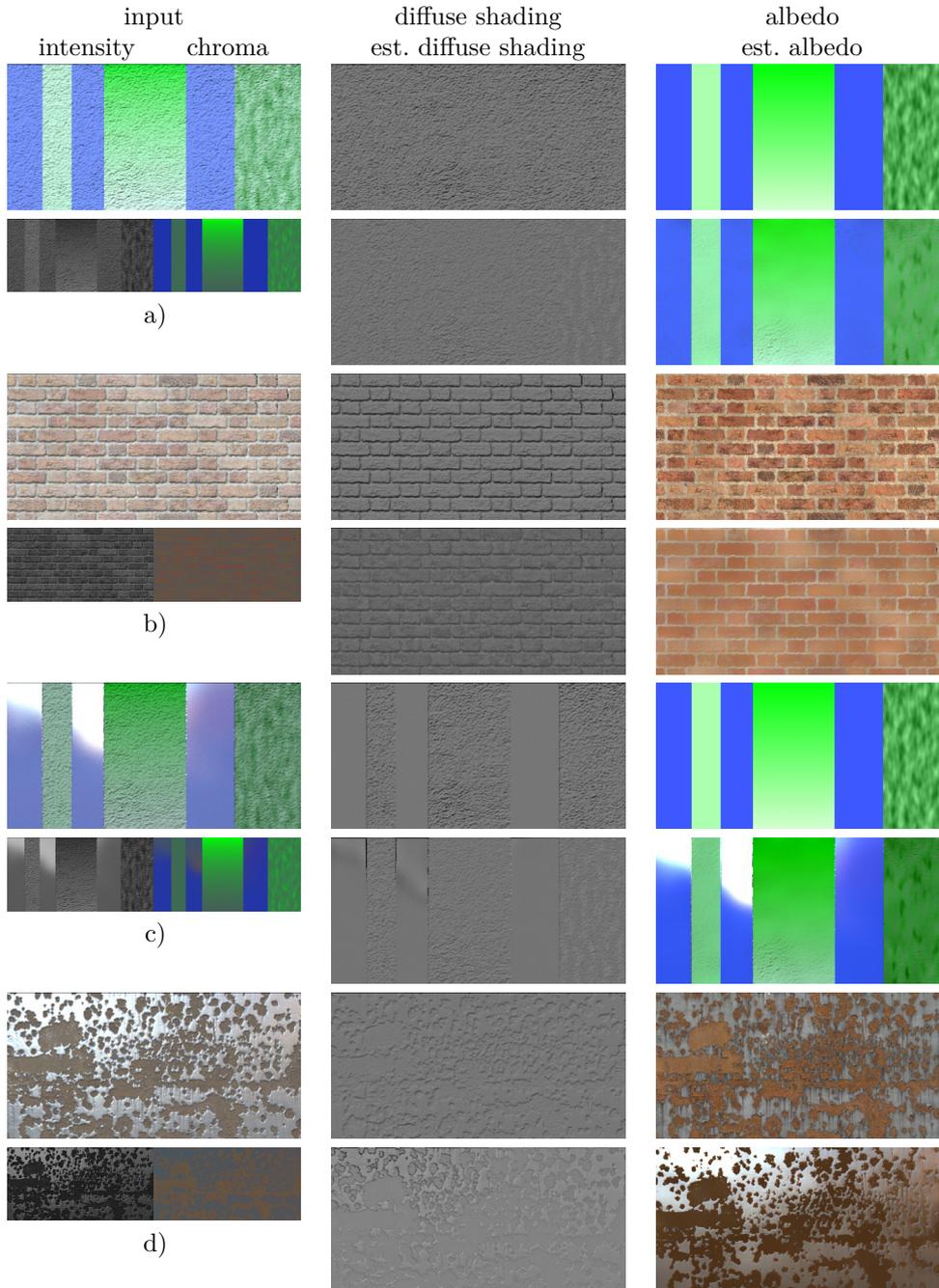


Figure 6.1.: Results from intrinsic image decomposition. The left column shows the rendered input image as well as its intensity and chroma. The algorithm then decomposes the image into diffuse shading and albedo (middle and right column, reference and estimation result shown on top of each other). Note that the estimated albedo contains specular highlights as well, as a separate estimate for specularity is not generated. Texture (d) is taken from <http://www.arroway-textures.com/> and slightly modified.

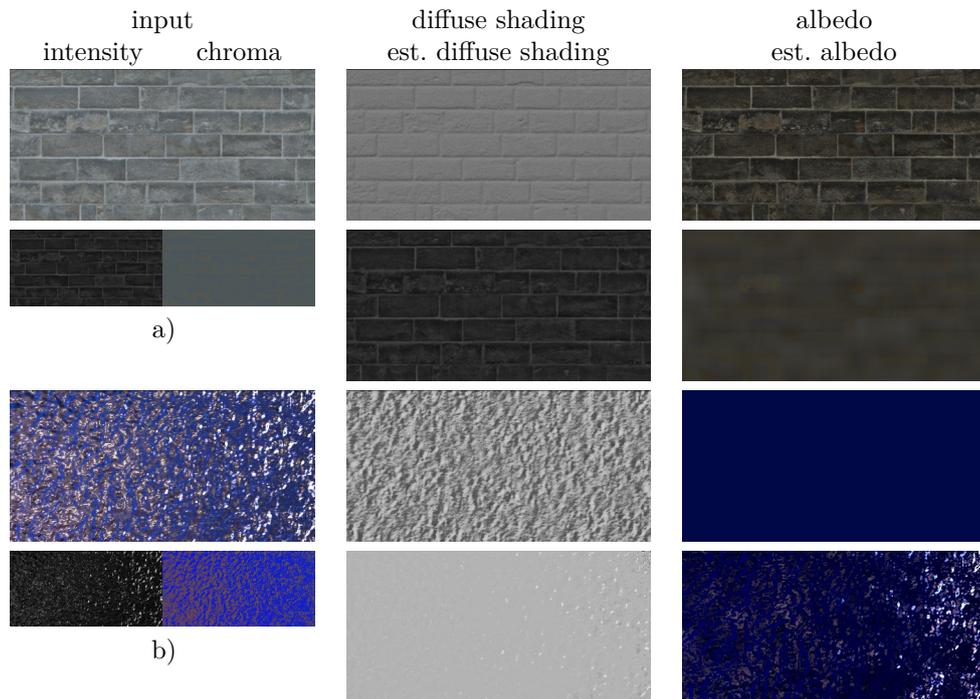


Figure 6.2.: Results from intrinsic image decomposition in case of failure. In the first row, a change in albedo only affects intensity, not chroma. In the second row, the bright colored light source is reflected from the highly specular material with noisy normals. Diffuse and specular shading interfere with each other, both consisting of high-frequency components. Texture (a) is taken from <http://www.arroway-textures.com/> and slightly modified.



Figure 6.3.: Environment maps used to render the surfaces in Fig. 6.1 and Fig. 6.2 (taken from <http://gl.ict.usc.edu/Data/HighResProbes/>).

6.1. Intrinsic Image Decomposition

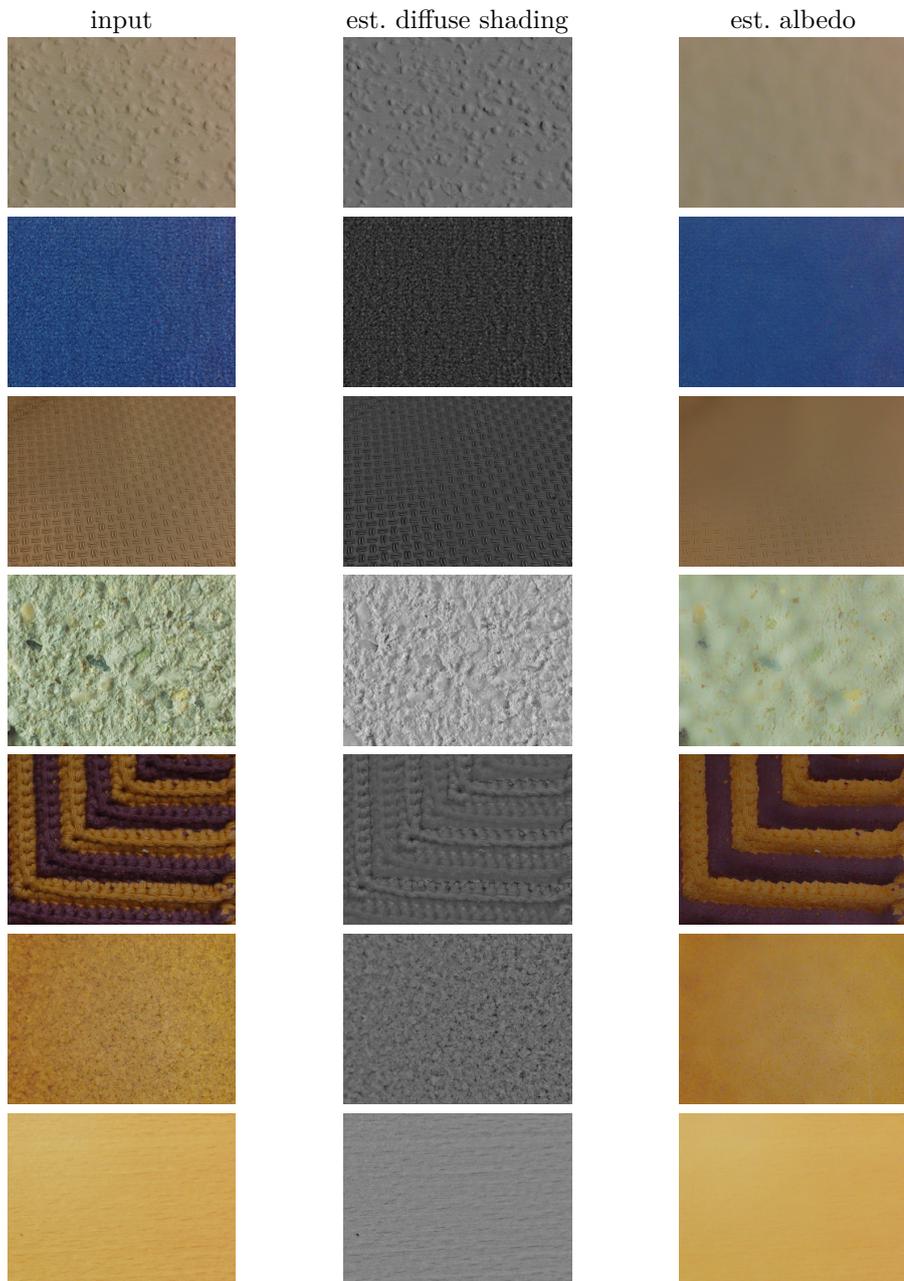


Figure 6.4.: Results from intrinsic image decomposition for real scenes. From top to bottom: *ingrain wallpaper*, *carpet*, *metal*, *stonewall*, *fabric*, *cork*, *wood*. Decomposition failed for *cork* and *wood*.

6.2. Estimating Normal Maps of Diffuse Surfaces

Estimating normals from images with static illumination represented by an environment map is an ill-posed problem. In this work, only normals that are visible as high-frequency details in the image are considered to achieve reasonable results. In the image, normals generally have two effects, depending on whether a surface point has a more diffuse or more specular appearance. Diffuse surface points become brighter if the angle between normal and incident lighting direction is small. In contrast, specular surface points exhibit a highlight if the normal lies half-way between viewing direction and incident light. Therefore, in order to perfectly reconstruct normals, the BRDF has to be known. Here, a homogeneous diffuse BRDF is assumed when reconstructing normals to simplify the problem.

The approach discussed in the following is based on the optimization methods in [WSTS08] and [IH89]. In contrast, instead of a single point-light source, the lighting from an environment map is considered using the spherical harmonics representation discussed in Sec. 2.7. Furthermore, the two scale reconstruction technique introduced in [DTPG11] is used: the surface is represented by two layers, coarse heights at the bottom and overlying perturbed normals for subtle details. First, normals are recovered on a smoothed image, then subtle details are added.

For objects with a homogeneous diffuse BRDF ($f_r(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = \frac{\rho}{\pi}$), the image formation with spherical harmonics lighting is described in Eq. (2.34) and Eq. (2.19). Using only 4 coefficients ($l \leq 1$), the equation becomes linear (with $\tilde{A}_l = \sqrt{\frac{4\pi}{2l+1}}$, $\mathbf{n}_l = (u, v, w)$ and Eqs. (2.22)-(2.25)):

$$L_o(\mathbf{n}_l) = \frac{\rho}{\pi} (\tilde{A}_0 L_{00} Y_{00}(\theta, \phi) + \tilde{A}_1 L_{1-1} Y_{1-1}(\theta, \phi) + \tilde{A}_1 L_{11} Y_{10}(\theta, \phi) + \tilde{A}_1 L_{11} Y_{11}(\theta, \phi)) \quad (6.8)$$

$$= \frac{\rho}{\pi} \left(\frac{\tilde{A}_0 L_{00}}{2\sqrt{\pi}} + \frac{\sqrt{3}\tilde{A}_1 L_{1-1}}{2\sqrt{\pi}} n_y + \frac{\sqrt{3}\tilde{A}_1 L_{10}}{2\sqrt{\pi}} n_z + \frac{\sqrt{3}\tilde{A}_1 L_{11}}{2\sqrt{\pi}} n_x \right) \quad (6.9)$$

$$= \frac{\rho}{\pi} (a_3 u + a_2 v + a_1 w + a_0). \quad (6.10)$$

The irradiance $E(x, y)$ at surface point (x, y) with normal

$$\mathbf{n}_l(x, y) = (u(x, y), v(x, y), w(x, y)) \quad (6.11)$$

is

$$E(x, y) = a_3 u(x, y) + a_2 v(x, y) + a_1 w(x, y) + a_0. \quad (6.12)$$

Initially, the surface albedo ρ is not known and has to be estimated before normal reconstruction to obtain $E(x, y)$. The surface is mainly planar with only a few high-frequency normals. The global orientation \mathbf{n} in world space is known from

homography decomposition. Therefore, averaging the radiance observed at all surface points approximately yields the radiance L_g of the global normal \mathbf{n} . Once a normal and its corresponding radiance are known, ρ can be calculated in the following way:

$$\rho = \pi \frac{L_g}{E(\mathbf{n})}. \quad (6.13)$$

With ρ , the irradiance $E(x, y)$ is known from the radiance $L(x, y)$ at each point:

$$E(x, y) = \frac{\pi}{\rho} L(x, y). \quad (6.14)$$

The normal can then be computed directly from the irradiances $E(x, y)$ by minimizing the following functional:

$$J(u, v, w) = \iint (a_3u + a_2v + a_1w + a_0 - i)^2 + \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2 + w_x^2 + w_y^2) dx dy \quad (6.15)$$

$$= \iint F(u, v, w, u_x, u_y, v_x, v_y, w_x, w_y, x, y) dx dy. \quad (6.16)$$

In this equation, $i = E(x, y)$, $u_x = \frac{\partial u(x, y)}{\partial x}$ and $u_y = \frac{\partial u(x, y)}{\partial y}$ are the partial derivatives of u , analogue to v_x, v_y, w_x, w_y . J consists of data and smoothness term. The data term (first term) ensures that the irradiance calculated from the estimated normals equals the observed irradiance i . The smoothness term (second term) produces a solution in which nearby normals do not change rapidly. The energy functional leads to Euler-Lagrange equations that are solved as described in Appendix A.1. Similar to [DTPG11], the algorithm operates on a Gaussian-filtered image (mask size 3×3 , $\sigma = 0.4$).

The algorithm requires meaningful initialization to facilitate fast convergence. From the scene reconstruction step, the global normal \mathbf{n} of the plane is already known. Thus, (u, v, w) can be initialized with this normal direction, and because the plane is almost flat, a solution is found after only a small number of iterations. The process is terminated after a fixed number of iterations or if the normals do not change anymore (threshold-based).

However, the normals estimated by this process are biased towards the brightest points in the environment map. This effect also occurs in [WSTS08], where the following solution is proposed. At first, a height field is reconstructed from the normals by solving a 2 dimensional Poisson equation. Afterwards, the height field is differentiated to obtain the final normals.

A height field determines the height $h(x, y)$ at each surface point. It is obtained from the normal map by minimizing the following energy functional:

$$J(h) = \iint \|\nabla h - q\|^2 dx dy \quad (6.17)$$

$$= \iint (h_x - q_x)^2 + (h_y - q_y)^2 dx dy, \quad (6.18)$$

where ∇ is the gradient operator and h denotes the 2 dimensional height field $h(x, y)$ with its derivatives h_x and h_y . q is a vector containing the functions that determine the relative height calculated directly from the normals. The functional is reformulated as Poisson equation and solved as described in Appendix A.2.

Once the height field is computed, it is back-converted to a normal map

$$\mathbf{n}_h(x, y) = (u_h(x, y), v_h(x, y), w_h(x, y)) \quad (6.19)$$

by calculating the gradient. At last, perturbed normals that model subtle details onto the height field are reconstructed by minimizing the following energy function at each surface point [DTPG11]:

$$J(u', v', w') = (a_3 u' + a_2 v' + a_1 w' + a_0 - i)^2 \quad (6.20)$$

$$+ \lambda((u' - u_h)^2 + (v' - v_h)^2 + (w' - w_h)^2). \quad (6.21)$$

This yields the final normal map $\mathbf{n}'_h(x, y) = (u'(x, y), v'(x, y), w'(x, y))$. λ is set to 0.001. Just like in Eq. 6.15, the first term minimizes the difference between observed irradiance and the irradiance calculated from the resulting normal map. The second term ensures that the result is similar to the normal \mathbf{n}_h calculated from the height field. The equation is overdetermined and is solved for each surface point independently by calculating the best fit in the least-squares sense, followed by normalization of the vectors to length 1.

Discussion and Evaluation Computing the normals with spherical harmonics has a few disadvantages. First of all, it is not guaranteed that the resulting normal vectors point towards the camera, even though this does not seem to be a problem, at least in the tests shown below. Moreover, lighting with spherical harmonics does not consider self-shadowing, even light from behind the plane can shine onto a surface point if the normal is tilted. The upper hemisphere related to a local normal is always fully visible, regardless of the orientation of the normal. However, as only a hemispherical environment map (upper hemisphere of the plane’s global normal) is captured with the fish-eye lens, the other hemisphere is black, not introducing any errors.

Fig. 6.5 shows recovered normal maps (color-coded) of a plane in four different lighting environments. The images only contain shading and intrinsic image decomposition is not necessary, because the plane is diffuse and has constant albedo. *Arealight-lt* exhibits a small area light source, thus the edges in the renderings are clearly visible. Note that this type of illumination can not be represented very well by spherical harmonics with 4 coefficients. *Smoothlight* shows very smooth lighting. Natural illumination from indoor and outdoor scenes is tested in *ennis* and *pisa*. Normal map *test* shows vertical, horizontal and diagonal edges. Normals similar to real-world materials are given with *stone* and *concrete*.

As mentioned, it is not possible to represent *arealight-lt* with only 4 coefficients in spherical harmonics. Nevertheless, the quality of reconstruction is similar to the other

scenes, indicating that the approximation is also sufficient for small light sources, often appearing in indoor scenes, e.g., from light bulbs. However, the rendering of normal map *test* shows that due to the position of the light source, diagonal edges have nearly the same intensity as the background, although the normal is different. This prevents a correct recovery of these edges, clearly visible in *test*, but not recognizable in *stone*. This problem is also visible in the other scenes, dependent on the lighting conditions. For example, in *pisa*, light comes mostly from the bright sky, so vertical edges are lost. The estimation of *stone* greatly suffers from this problem, as the vertical edges of the bricks are not visible anymore. A worst-case example is shown in Fig. 6.6. The light source is centered above the plane, consequently shading is faintly visible and independent of the edge's orientation. This problem is inherently given and can be solved, e.g., if multiple images under varying illumination are used (Photometric stereo) or if color information is exploited like in [JA11].

Results for real scenes are given in Fig. 6.7 and Fig. 6.8. In *env-1* (Fig. 6.7), the surface is illuminated from two light sources at similar location. *Env-2* consists of multiple light sources: fluorescent tubes at the ceiling and a bright light from the window. In both cases, the surface structure is clearly visible and the reconstructed normal map is plausible. *Env-3* suffers from the same problem as illustrated in Fig. 6.6. Reconstruction fails because the brightest light is centered above the surface. Fig. 6.8 shows results rendered from reconstructed normal maps.

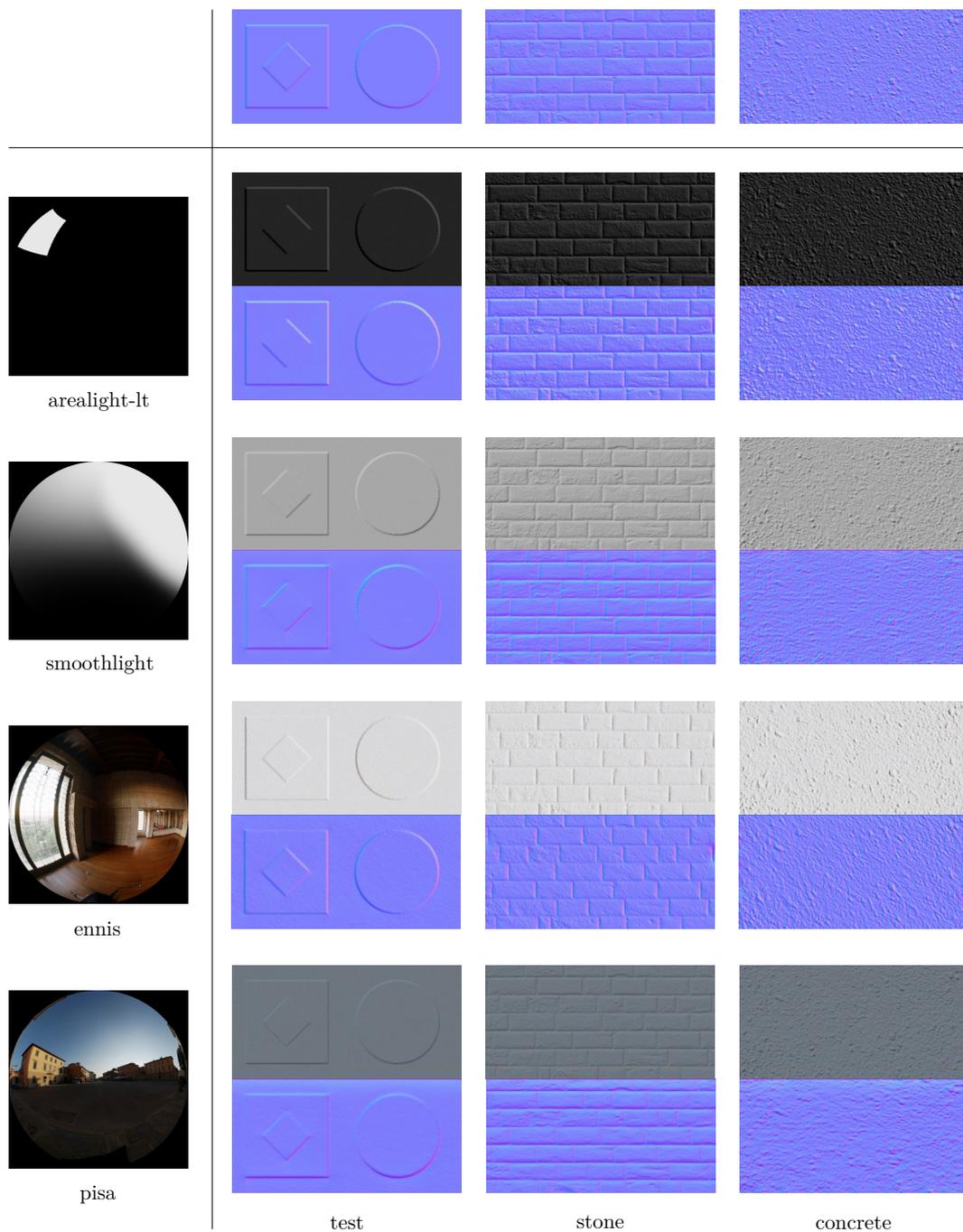


Figure 6.5.: Recovered normal maps in different lighting environments. The reference normal maps are displayed in the first row. The subsequent rows show the environment map (upper hemisphere of the plane), the rendering of the normal map used for estimation and the estimated normal map (*conrete* is taken from <http://www.arroway-textures.com/>).

6.2. Estimating Normal Maps of Diffuse Surfaces

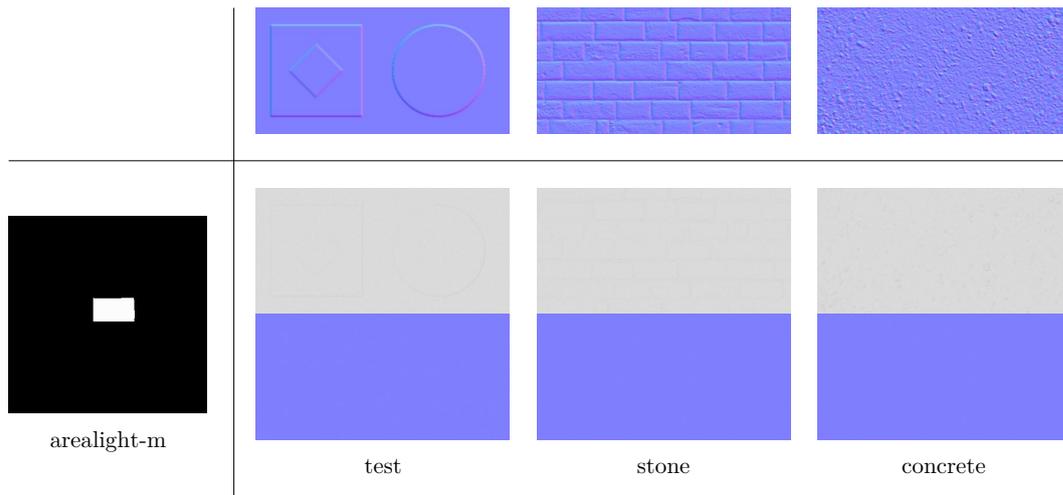


Figure 6.6.: Recovered normal maps for an area-light source centered above the plane. Normal reconstruction fails in this case, because shading is hardly visible in the images.

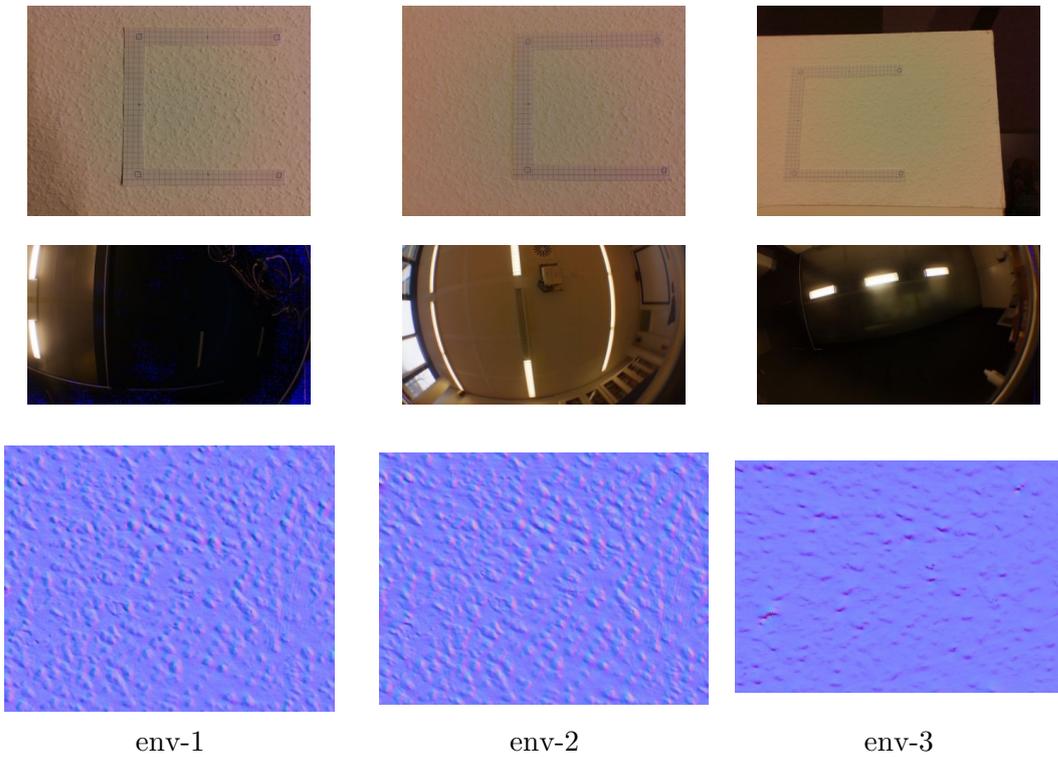


Figure 6.7.: Recovered normal maps from real scenes for different environments. The dominant light source in *env-1* and *env-2* is placed at the borders of the environment map, so the surface structure is clearly visible. In *env-3*, the light source is placed above the surface, which results in the same problems as in Fig. 6.6.

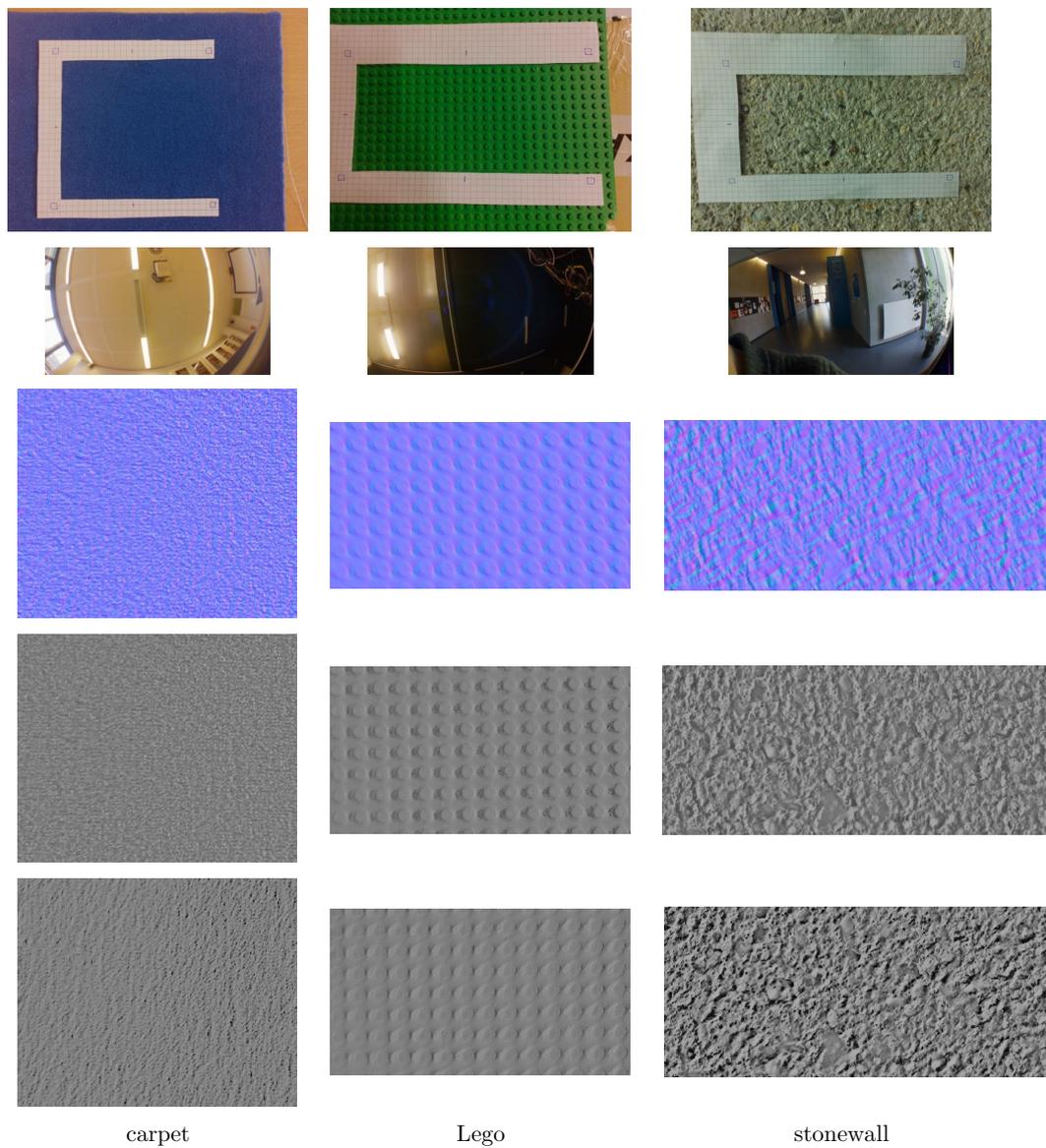


Figure 6.8.: Recovered normal maps from real scenes for different environments and surfaces. From top to bottom: one of the two input images, environment map, reconstructed normal map, normal map rendered with captured environment map (intensity only), normal map rendered under novel illumination (point light source at position $(-1, 1, -1)$).

7. BRDF Estimation

In this chapter, the problem of measuring a single BRDF of a homogeneous planar surface is examined. Given two views of the scene and the hemispherical environment, a BRDF model is fit to the measurements. At first, a BRDF model is presented that allows fitting using linear optimization with low memory requirements and short computation times. Afterwards, the estimation process is described in more detail and finally evaluated for different scenes.

7.1. A Linear BRDF Model

Analytic models for BRDFs are mostly non-linear, with one or more parameters controlling the shape of the specular lobe (i.e., the roughness of the surface). Fitting these models to measured data relies on non-linear optimization techniques that are sensitive to initialization and often find only a local minimum solution. A linear model does not suffer from these problems, as linear least-squares solvers can be used. The linear model is based on the assumption that a material is composed of several basis BRDFs with different weightings. This results in a linear combination of basis BRDFs, see Sec. 2.2.2 and Eq. (2.8). Analysis of measured BRDFs (e.g., from the MERL BRDF database) have shown that this assumption is valid. Matusik et al. performed Principal Component Analysis (PCA) on the dataset and discovered that only 45 principal components are necessary to represent each of the measured BRDFs [MPBM03a]. They have also shown that it is possible to represent new BRDFs as linear combination of the BRDFs in the original dataset [MPBM03b].

The presented work uses a linear model to estimate BRDFs by linear optimization. Instead of using measured data, the basis BRDFs are represented by the Ashikhmin-Shirley BRDF model [AS00]. This has the advantage that it is not necessary to store large amounts of data and it is possible to efficiently render materials with Monte-Carlo methods. The BRDF is then written as

$$f(\omega_i, \omega_o) = \frac{a_0}{\pi} + \sum_{l=1}^L a_l f_l(\omega_i, \omega_o). \quad (7.1)$$

The BRDF consists of a Lambertian term with albedo a_0 and L specular terms $f_l(\omega_i, \omega_o)$, weighted with a_l . According to the isotropic Ashikhmin-Shirley model,

the specular term is given as

$$f_l(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = \frac{m_l + 1}{8\pi} \frac{(\mathbf{n} \cdot \mathbf{h})^{m_l}}{(\boldsymbol{\omega}_i \cdot \mathbf{h}) \max((\mathbf{n} \cdot \boldsymbol{\omega}_i), (\mathbf{n} \cdot \boldsymbol{\omega}_o))} F((\mathbf{k} \cdot \mathbf{h})), \quad (7.2)$$

where \mathbf{n} is the surface normal, \mathbf{h} is the half-vector between $\boldsymbol{\omega}_i$ and $\boldsymbol{\omega}_o$, m_l is a Phong-like exponent to control the roughness, and $F(\cos \theta)$ is a function giving the Fresnel reflectance for incident angle θ . Using Schlick's approximation [Sch94], the Fresnel term is

$$F(R_l, \cos \theta) = R_l + (1 - R_l)(1 - \cos \theta)^5. \quad (7.3)$$

Thus, the shape of the specular lobe is parametrized by the two values m_l and R_l . To get a meaningful parametrization for the basis materials, the model is fitted to measured BRDFs from the MERL database. This has already been performed by Ngan et al. [NDM05], so their results can be used here.

However, if all 100 BRDFs of the MERL database are used, the estimation would have to find interpolation weights a_l for 100 materials. Therefore, it would be necessary to render the captured scene 100 times, resulting in long computation times. As previously mentioned, it has been shown that a much smaller number of basis BRDFs is sufficient for a decent approximation. In order to find out the minimal number of basis BRDFs, k-means clustering with $K = \{5, 10, 20, 30, 40, 50, 60\}$ is performed in the 2D parameter space (m_l, R_l) with the complete set of MERL BRDFs. Only lobes with $m_l > 5$ are used, as materials with smaller m_l are considered as fully diffuse and, thus, can be approximated by the diffuse term. The clustering results in $|K| = 7$ possible BRDF models

$$f^K(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = \frac{a_0}{\pi} + \sum_{l=1}^K a_l^K f_l^K(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) \quad (7.4)$$

from which to choose, with $f_l^K(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o)$ determining an Ashikhmin-Shirley BRDF using the parameters of the l -th cluster and its associated weighting a_l^K .

In order to choose the best model in terms of computational effort and approximation quality, each of the K models is fitted to each MERL BRDF. The fitting is performed following the approach of Ngan et. al [NDM05] by minimizing the following residual function for a linear BRDF with parameter vector \mathbf{a} :

$$E(\mathbf{a}) = \sqrt{\frac{\sum_{\boldsymbol{\omega}_i, \boldsymbol{\omega}_o} w (f^K(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) \cos \theta_i - m_j(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) \cos \theta_i)^2}{\sum_{\boldsymbol{\omega}_i, \boldsymbol{\omega}_o} w}}, \quad (7.5)$$

where $m_j(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o)$ is the j -th MERL BRDF and the BRDF samples are weighted according to uniform incoming radiance (w is the solid angle correction term). The function is evaluated on a grid of $90 \times 60 \times 90 \times 60$ ($\theta_i \times \phi_i \times \theta_o \times \phi_o$). Due

7.2. Estimating BRDF Weighting Coefficients from Images

to their unreliability in measurements, values with an incident or outgoing angle $\theta_{\{i,o\}} > 80^\circ$ are ignored. In contrast to [NDM05], the fitting is performed by non-negative linear least-squares minimization, ensuring that the fitting results in a non-negative BRDF. It is also necessary to down-weight (here: multiply with 0.01) values near mirror direction ($\theta_i \cong \theta_o \wedge \phi_i \cong \phi_o + \pi$), because the difference in the point-sampled analytic BRDF model and the measured values becomes high in this region, potentially dominating all other residuals. The down-weighting is only applied in the optimization, all errors shown in the following are calculated without down-weighting.

Results for the fit are shown in Fig. 7.1 and Fig. 7.2. The figure suggests that $K = 20$ basis BRDFs give a good trade-off between approximation quality and computational complexity. For $K \geq 40$ the result does not improve significantly. Note that the best fit in least-squares sense does not necessarily provide the best visual result.

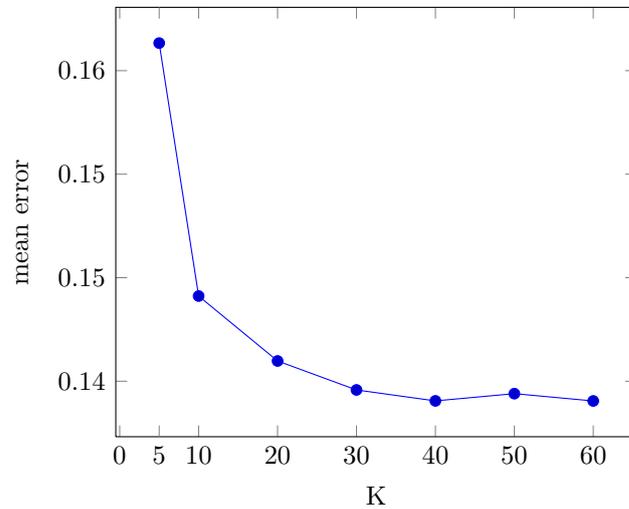


Figure 7.1.: Mean error for different number of basis functions. All MERL BRDFs are fitted to the linear BRDF model according to Eq. (7.5), independently for each color channel. The error is calculated for each channel and normalized to the maximum albedo of the BRDF (maximum albedo computed as in [NDM05]). Finally, the normalized error of all color channels is averaged.

7.2. Estimating BRDF Weighting Coefficients from Images

Estimating the linear BRDF from two views and an environment map means to determine the weighting coefficients for each of the basis BRDFs. The rendering

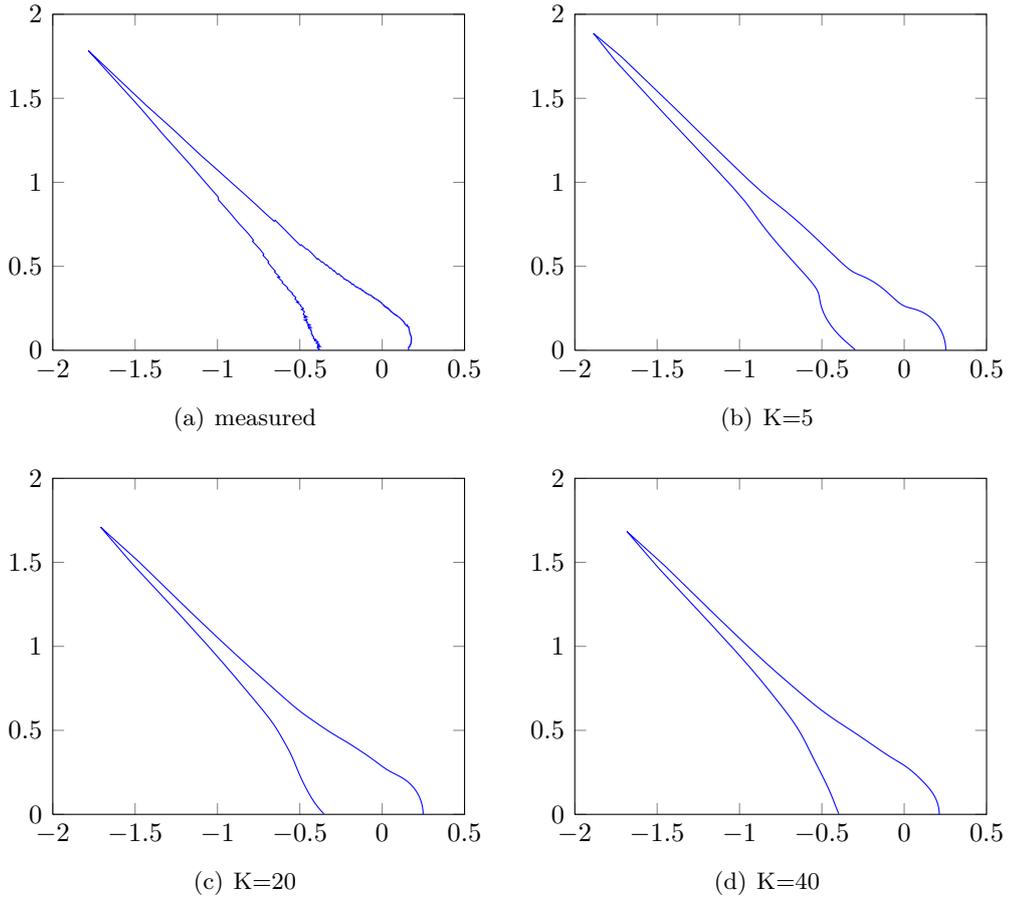


Figure 7.2.: Fitting results for BRDF *alum-bronze*, scaled by cubic root, $\theta_i = 45^\circ$, $\phi_i = 0^\circ$, $\phi_o = 180^\circ$

equation tells how the views were generated:

$$L_o(\mathbf{x}, \boldsymbol{\omega}_o) = \int_{\Omega_+} f^K(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) L_i(\mathbf{x}, \boldsymbol{\omega}_i) \cos \theta_i d\boldsymbol{\omega}_i. \quad (7.6)$$

Using parallel projection, a single view is a 2D slice of $L_o(\mathbf{x}, \boldsymbol{\omega}_o)$ for $\boldsymbol{\omega}_o = \text{constant}$. For a pinhole camera this is slightly more complex. The direction $\boldsymbol{\omega}_o$ of the outgoing radiance at surface point \mathbf{x} observed in the camera image is determined by the vector connecting \mathbf{x} with the projection center of the camera. Again, this results in a 2D slice of $L_o(\mathbf{x}, \boldsymbol{\omega}_o)$, but this time $\boldsymbol{\omega}_o$ is different for each pixel in a view. In both cases, each view contains a few scattered measurements of $L_o(\mathbf{x}, \boldsymbol{\omega}_o)$. Note that although an environment map is used for the incident illumination, L_i depends on \mathbf{x} as well. This is due to the corrections described in Sec. 5.5.

Plugging in $f^K(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o)$ and rearranging yields:

$$L_o(\mathbf{x}, \boldsymbol{\omega}_o) = \sum_{l=0}^K a_l \int_{\Omega_+} f_l^K(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) L_i(\mathbf{x}, \boldsymbol{\omega}_i) \cos \theta_i \, d\boldsymbol{\omega}_i, \quad (7.7)$$

with the diffuse component $f_o^K = \frac{1}{\pi}$.

Using linear least-squares optimization, the coefficients a_l can be determined. The quadratic difference between the measurements $L_o(\mathbf{x}, \boldsymbol{\omega}_o)$ and the weighted basis BRDFs has to be minimized:

$$E(\mathbf{a}) = \sum_{\substack{(\mathbf{x}, \boldsymbol{\omega}_o) \\ \in \text{observations}}} \left(L_o(\mathbf{x}, \boldsymbol{\omega}_o) - \sum_{l=0}^K a_l \int_{\Omega_+} f_l^K(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) L_i(\mathbf{x}, \boldsymbol{\omega}_i) \cos \theta_i \, d\boldsymbol{\omega}_i \right)^2. \quad (7.8)$$

This problem can be efficiently solved by precomputing the integral for each observation, resulting in a function $L_l(\mathbf{x}, \boldsymbol{\omega}_o)$ for each basis BRDF $f_l^K(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o)$. The error function becomes:

$$E(\mathbf{a}) = \sum_{\substack{(\mathbf{x}, \boldsymbol{\omega}_o) \\ \in \text{observations}}} \left(L_o(\mathbf{x}, \boldsymbol{\omega}_o) - \sum_{l=0}^K a_l L_l(\mathbf{x}, \boldsymbol{\omega}_o) \right)^2. \quad (7.9)$$

The equation describes an ordinary linear least-squares problem. The resulting linear BRDF is given as

$$f^K(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = \sum_{l=0}^K a_l^K f_l^K(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o). \quad (7.10)$$

To ensure non-negativity, the solution is computed via the non-negative linear least-squares solver described in Sec. 2.3. Thus, all coefficients a_l are either 0 or positive, based on the intuition of a BRDF that is composed of several specular lobes and a constant for the diffuse component.

What remains is the computation of $L_l(\mathbf{x}, \boldsymbol{\omega}_o)$. The region of integration Ω_+ as well as the cosine are dependent on the surface normal at point \mathbf{x} , so the precomputation has to be performed once for a new scene. At any point in each view and for each basis BRDF, the incident radiance has to be weighted with cosine, solid angle, and basis BRDF, afterwards integrated over the upper hemisphere (see Alg. 1). The following methods are used to perform this task.

Diffuse component The diffuse component is computed efficiently with the spherical harmonics representation of the environment map. This is easily performed through Eq. 2.19 and Eq. 2.34. Due to the use of spherical harmonics, the result is noise-free. However, as spherical harmonics coefficients are only calculated for the environment map seen from the central point on the plane, the corrections that would have to be applied to off-center surface points are ignored (see Sec. 5.5).

Algorithm 1 Integration of basis BRDFs and environment.

```

for all views  $v$  do
  for all surface points  $\mathbf{x}$  with ray direction  $\boldsymbol{\omega}_o$  in view  $v$  do
    calculate diffuse component  $L_o(\mathbf{x}, \boldsymbol{\omega}_o)$ 
    for all specular basis BRDFs  $f_l^K(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o)$  do
      calculate specular component  $L_l(\mathbf{x}, \boldsymbol{\omega}_o)$ 
    end for
  end for
end for

```

Specular component For the specular component, Monte-Carlo integration with importance sampling is used. The upper hemisphere is sampled according to the probability distribution of the Ashikhmin-Shirley BRDF [AS00], ensuring fast convergence with few samples. Here, the corrections for off-center surface points are considered to avoid large errors for highly specular BRDFs. The errors are mostly visible for materials that create a clear mirror image of the environment.

The use of spherical harmonics for the diffuse basis BRDF and importance sampling for the specular basis BRDFs guarantees efficient computation, considering running time as well as memory requirements.

7.3. Evaluation

7.3.1. Synthetic Scenes

An evaluation is performed by estimating the BRDF in different scenes that are composed of a plane rendered with various MERL BRDFs and illuminated by the environments shown in Fig. 7.3. The camera is placed at two different locations, creating two views of the plane, each with a size of 436×218 . The environments address different problems of the estimation process. *Smoothlight* is composed of low-frequency components only, similar to light coming from the sky. *Arealight* contains a single light source that is located at a position where its reflection on the plane is visible in both views. In contrast, the reflection of the light source in *arealight-single* is only visible in one of the views. Finally, a typical outdoor scene is given in the *pisa* environment.

Each scene is rendered with one of four materials from the MERL BRDF database [MPBM03a], ranging from highly specular (*green-acrylic*) to almost purely diffuse (*blue-rubber*). Monte-Carlo integration with cosine-distributed importance sampling is used to synthesize the images.

Fig. 7.5 lists some results of the estimation process. In the first column, one of the two input images is shown. The second column shows the plane from the same viewing

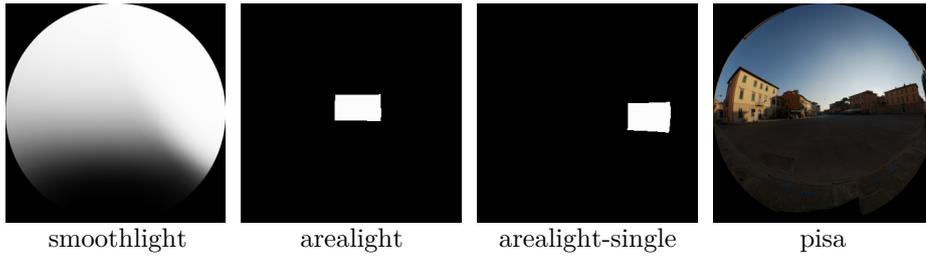


Figure 7.3.: Environments used to evaluate the results of the estimation. Each environment poses another challenge for the estimation.

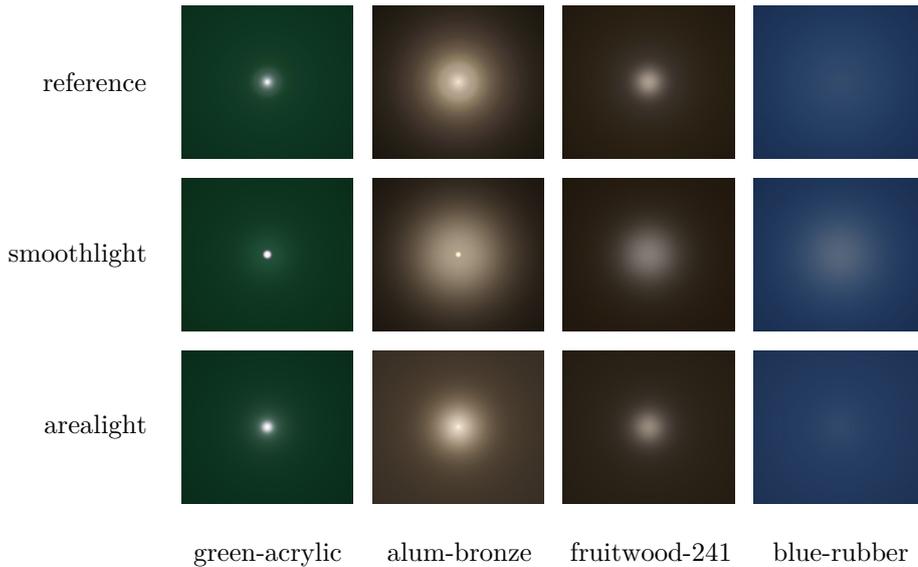


Figure 7.4.: Estimation results and reference for the scenes in Fig. 7.5.

angle, this time rendered with the estimated BRDF. The difference between both images, scaled by 10, is given in the last column. It is clearly noticeable that, as expected, the difference is much higher at regions with specular highlights, because the specular highlight has greater absolute values and the most information of a BRDF is contained in the specular part. The results for the *smoothlight* environment generally have small differences, as the smooth illumination does not produce any noticeable highlights. However, this does not imply that the reconstructed BRDF better approximates the real BRDF.

The error of the approximation is shown in Fig. 7.6(a). The estimated BRDF is compared to the real BRDF at each location in a grid of $90 \times 60 \times 90 \times 60$ ($\theta_i \times \phi_i \times \theta_o \times \phi_o$). The figure shows that the approximations of *blue-rubber* and *fruitwood-241* have smaller error compared to *alum-bronze* and *green-acrylic*. This is not surprising, as these BRDFs do not have specular lobes with high values that

cause large errors. Moreover, in each case, the environment apparently does not have great influence on the error of the estimation. Although there is no highlight visible if the scene is illuminated by *smoothlight*, the error is similar or smaller compared with the error in the other environments. This does not seem to coincide with the results shown in Fig. 7.4, where the estimated BRDF is applied to a plane and rendered under point-light illumination, resulting in a bright highlight in the center of the surface. It can be seen that the estimation result using *arealight* are better than the results using *smoothlight*. Plots of the BRDFs (Fig. 7.7) reveal the reason for this inconsistency. If grazing angles become smaller, BRDF values get higher. Consequently, the error is dominated by the differences at small grazing angles. However, the input images do not contain grazing angle effects, making the approximation very inaccurate. Furthermore, the images in Fig. 7.4 do not show grazing angle effects either. Ignoring these effects when calculating the error results in Fig. 7.6(b). The error when estimating the BRDF in *smoothlight* is now greater than in *arealight* for *alum-bronze*, *fruitwood-241*, and *blue-rubber*, complying with synthesized images under point-light illumination.

However, *green-acrylic* still results in a larger error with *arealight*. Again, this can be explained by Fig. 7.7. Clearly, when reconstructing the specular lobe with the *smoothlight* environment, the shape does not correspond to the reference BRDF, however, its length is similar. Because the values are comparatively high in the mirroring direction, the error is influenced more by the length of the lobe than by its shape. For *green-acrylic* and *smoothlight*, although the shape does not match, the length is more similar in *smoothlight*. Therefore, the error is smaller. In conclusion, it is not possible to reliably estimate the shape of a specular lobe if the illumination mainly contains low-frequency components. This coincides with the work of Ramamoorthi and Hanrahan who state that BRDF recovery is well-conditioned for high-frequency illumination and ill-conditioned for smooth lighting [RH01c].

The reconstructed shape is not perfect for *arealight* either, even for small $\theta_{\{i,o\}}$. Further experiments indicate that, to a certain extend, this issue is also related to the number of image pixels and samples used in Monte-Carlo integration. The problem is much less present for nearly diffuse BRDFs.

Returning to Fig. 7.6(b), if the highlight is only visible in one of the views (*arealight-single*), the error is similar, although less information about the shape of the lobe is observed in the images. The same can be observed for the *pisa* environment, indicating that real-world illumination places sufficient constraints for a decent solution.

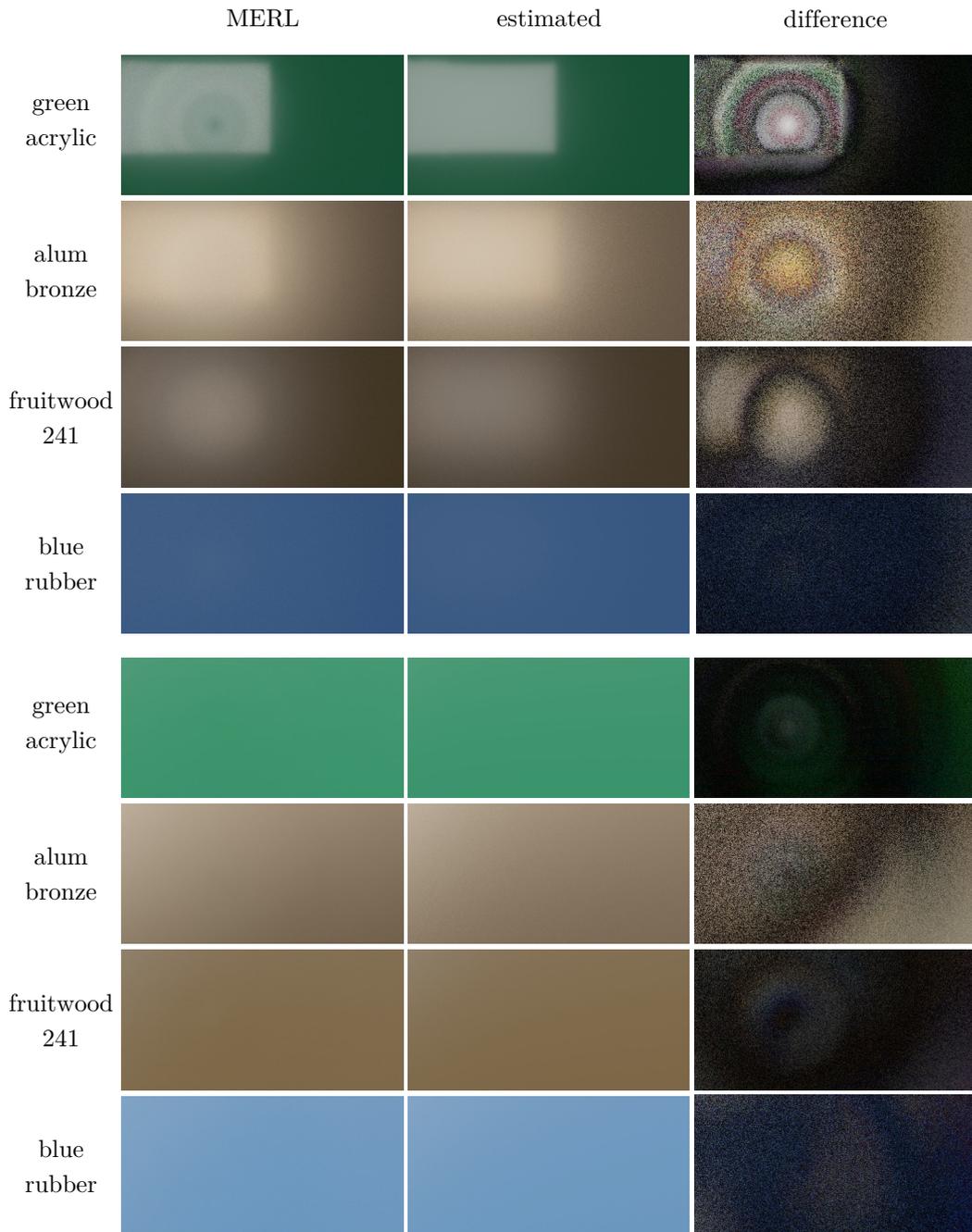
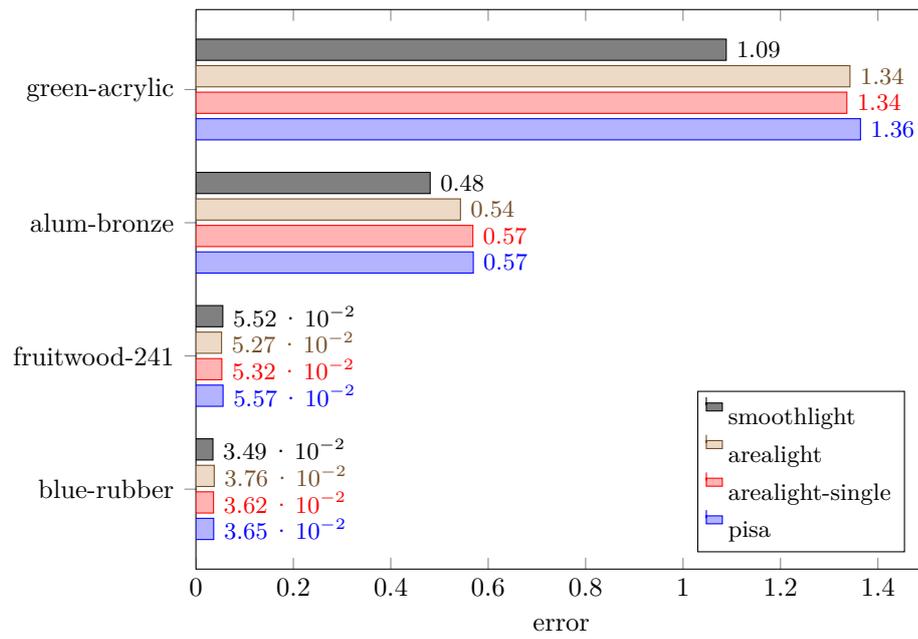
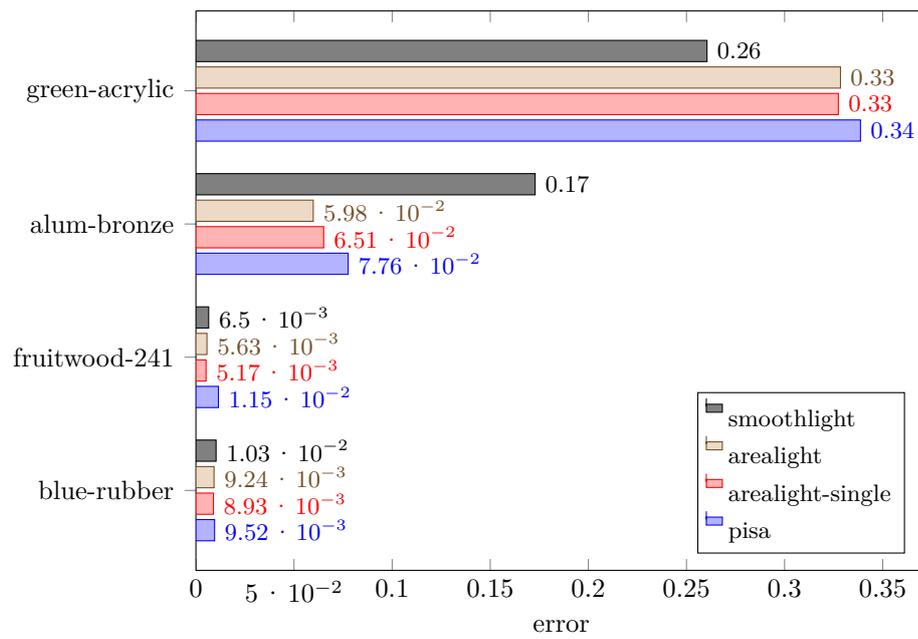


Figure 7.5.: Results of the estimation process for homogeneous surfaces. First column: one of the two views used as input images (rendered from MERL BRDFs). Second column: the same view rendered with the estimated BRDF. Third column: difference between estimation result and input image, scaled by a factor of 10. Environments *arealight* and *smoothlight* are used to illuminate the plane for the top and bottom group of images respectively.



(a)



(b)

Figure 7.6.: Error of estimated BRDFs according to Eq. (7.5) for different lighting environments. Data with incident or outgoing angle (a) $\theta_{\{i,o\}} > 80^\circ$ or (b) $\theta_{\{i,o\}} > 45^\circ$ is ignored.

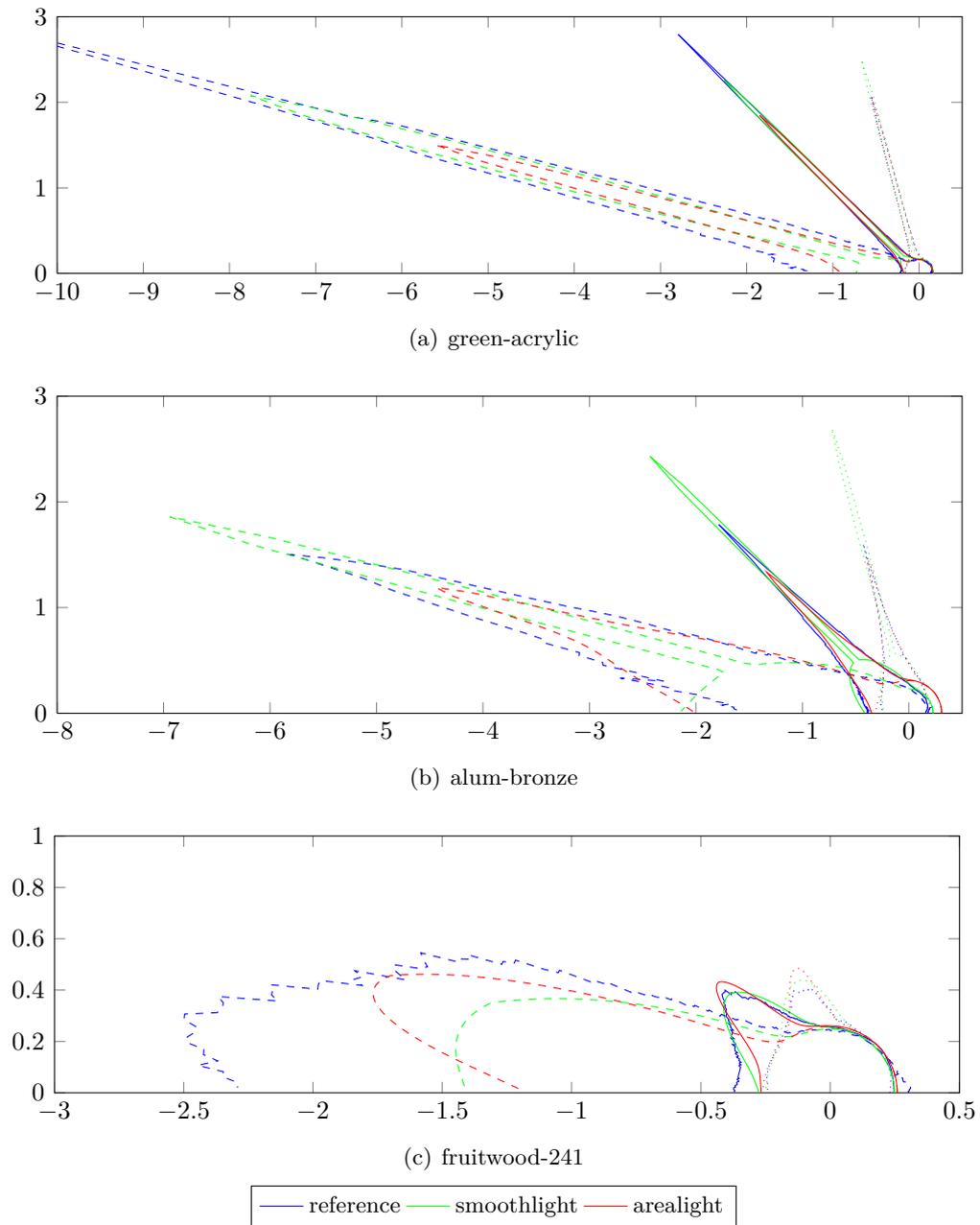


Figure 7.7.: Estimation results and reference for *green-acrylic*, *alum-bronze*, and *fruitwood-241*, scaled by cubic root, $\theta_i = \{15^\circ, 45^\circ, 75^\circ\}$, $\phi_i = 0^\circ$, $\phi_o = 180^\circ$

7.3.2. Real-World Scenes

Results of real scenes are shown in Figs. 7.8-7.14. Due to differences in surface, illumination, and view point, a large variety of problems are observable in the images. As noted earlier in this section, reconstruction quality depends on the reflection seen in the input images. Figs. 7.8-7.10 show a surface with small diffuse and high specular contributions. Under ideal conditions (direct reflection of a high-frequency light source; Fig. 7.8(a) and Fig. 7.10), the shape of the specular lobe is very similar. Without a reflection of the light source, the results are mixed: the BRDF is very blurry (Fig. 7.8(c)) or is otherwise distorted (Fig. 7.9(a)), similar to the behaviour observed with the *smoothlight* environment in Fig. 7.7. The same can be observed for another material in Fig. 7.11 and Fig. 7.12.

Although the shape of the specular lobe is similar in Fig. 7.8(a) and Fig. 7.10, the diffuse component is different. The reason is probably the differences in ambient light (color and intensity): in the former case, white light is coming from the window, mostly hitting the surface at small grazing angles. In the latter case, the only light source is the fluorescent lamp on the ceiling, so the scene is much darker and fully dependent on the color of the artificial light source. In this work, the influence of a light source's spectral power distribution on reconstruction quality will not be investigated further, due to the lack of reference materials and light sources. However, as Figs. 7.11-7.13 show, the problem is much less present when reconstruction materials with stronger diffuse components.

Another problem consists in the correct alignment of the environment reflection. Due to imperfect calibration, the scene geometry does not perfectly match reality, leading to slightly shifted reflections. This leads to blurred BRDFs as shown in Fig. 7.8(b). In Sec. 5.5.1, an image-based method is presented that tries to fix inaccuracies. However, the method can fail, leading to a more (Fig. 7.14(c)) or less (Fig. 7.9(b)) blurry result. Aligning the environment by hand as it was done in Fig. 7.9(c) improves the result.

Finally, Fig. 7.15 and Fig. 7.16 show the estimated BRDFs under novel illumination, i.e., a BRDF estimated in one environment is applied to another scene that shows the same material, but was captured in another environment. Again, a slight difference in diffuse color can be noted. Nevertheless, the shape of the highlights is similar. As expected, the error is smaller for diffuse BRDFs.

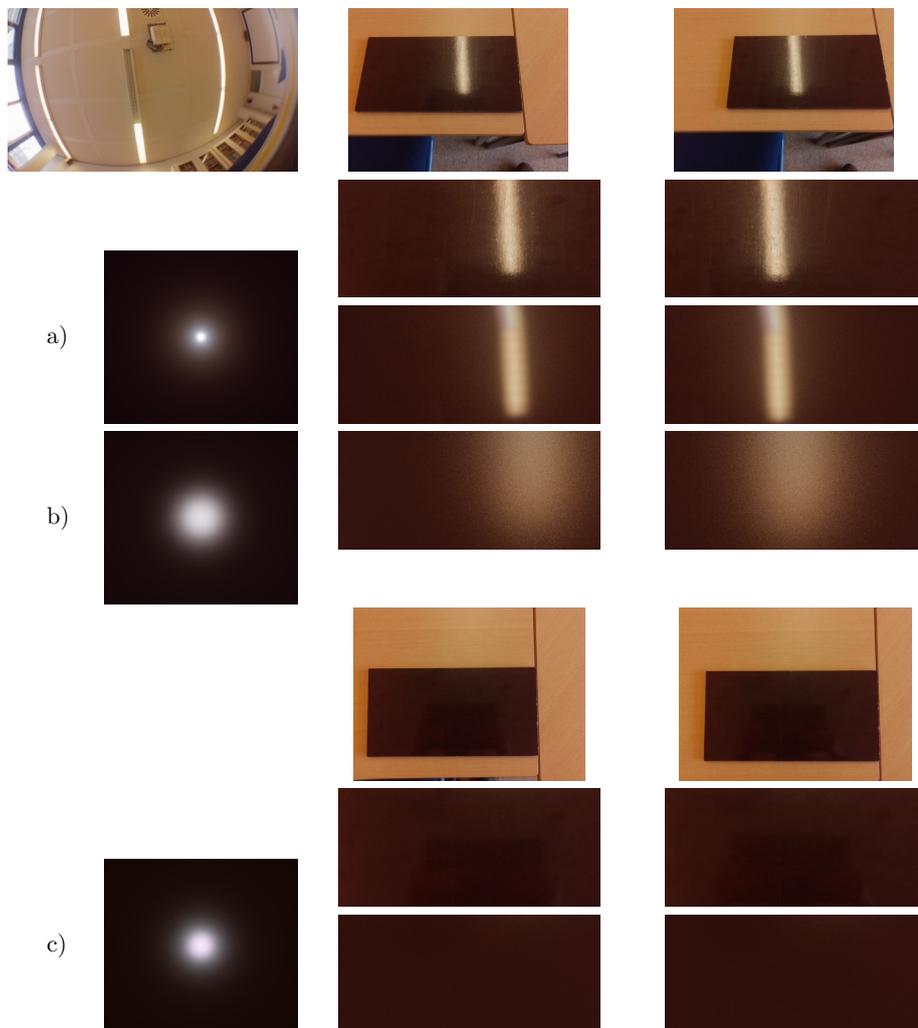


Figure 7.8.: Results for a dark surface with high specularity using two different sets of input images under the same illumination. From left to right and top to bottom: environment, first input image, second input image, close-up of surface in first input image, close-up of surface in second input image, estimated BRDF rendered with point light illumination, estimated BRDF rendered with same conditions as first input image, estimated BRDF rendered with same conditions as second input image. (a) Light source is reflected from surface. (b) Light source is reflected from surface, but automatic alignment of the environment is deactivated (Sec. 5.5.1). (c) No highlight visible.

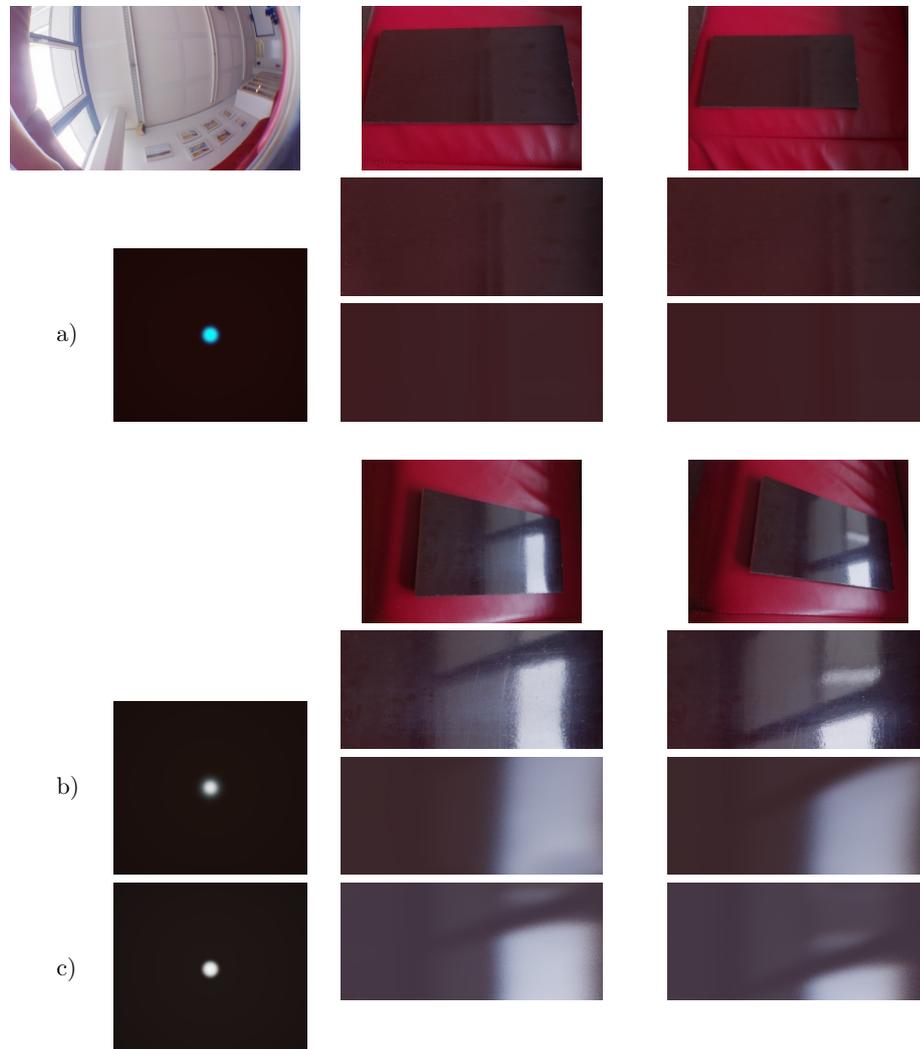


Figure 7.9.: Results for the same surface as in Fig. 7.8, but with different illumination. Light comes mostly from the window. (a) No highlight visible. (b) Highlight visible, but automatic alignment of the environment (Sec. 5.5.1) failed, leading to a slightly blurry BRDF. (c) Aligning the environment by hand improves the result.



Figure 7.10.: Results for the same surface as in Fig. 7.8, but with different illumination. Light comes from above only, there is little ambient light.

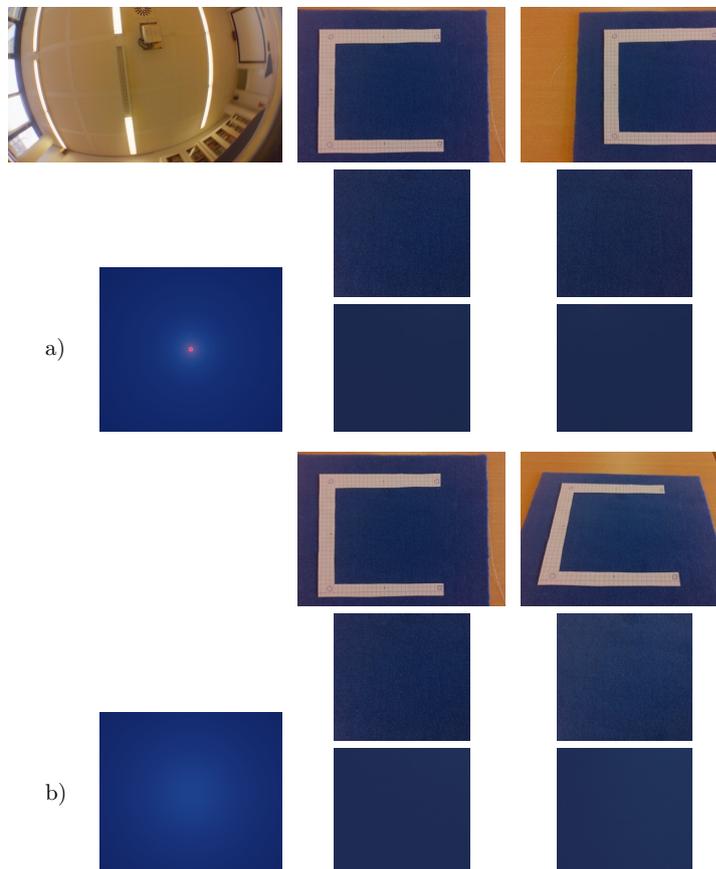


Figure 7.11.: Results for a mostly diffuse, blue surface (*carpet*). (a) Light is not reflected from the surface. (b) Light source is reflected in the right image (hardly visible because of the small specular component, cf. Fig. 7.8).

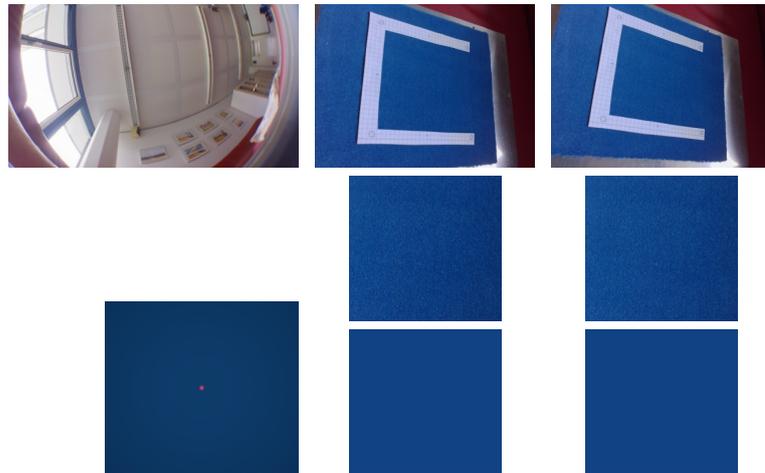


Figure 7.12.: Results for the same surface as in Fig. 7.11, but with different illumination. Light comes mostly from the window.



Figure 7.13.: Results for the same surface as in Fig. 7.11, but with different illumination. Light comes from above only, there is little ambient light.

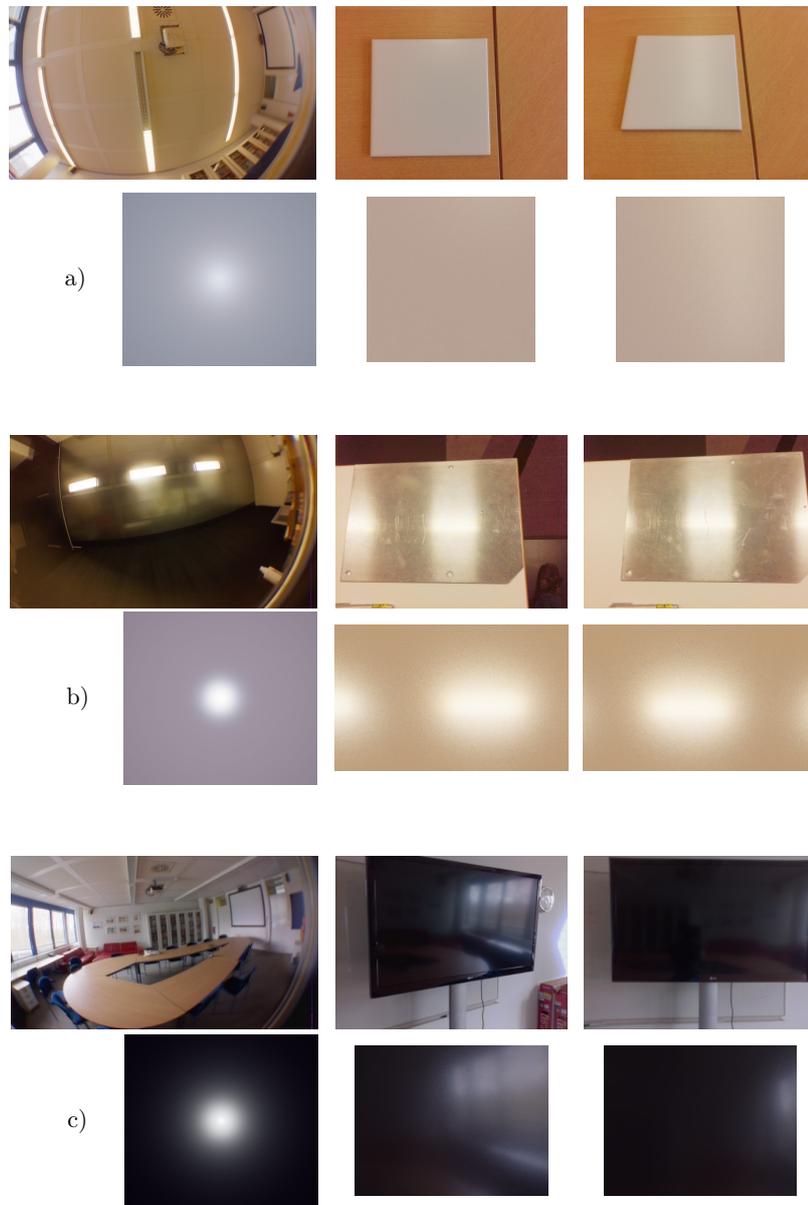


Figure 7.14.: Further results for different surfaces. (a) *tile* with moderate specular reflections. (b) *metal* with anisotropic behaviour that is currently not reproducible. (c) *tv* with mirror-like reflection that is slightly blurred, because of the complex and not perfectly aligned environment.

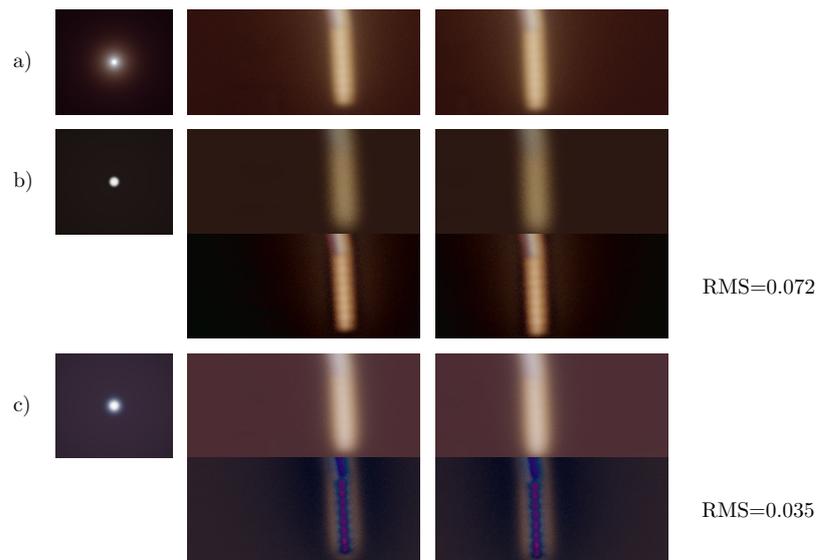


Figure 7.15.: Estimated BRDFs under novel illumination (environment of Fig. 7.8). From left to right and top to bottom: estimated BRDF rendered with point light illumination, estimated BRDF rendered with same conditions as first and second input image, differences between reference and relighted surface for first and second input image, RMS of difference. (a) Reference: BRDF of Fig. 7.8(a). (b) BRDF of Fig. 7.9(c). (c) BRDF of Fig. 7.10.

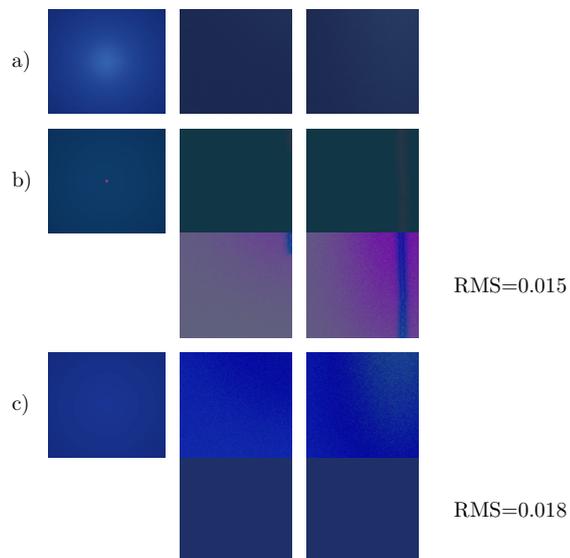


Figure 7.16.: Estimated BRDFs under novel illumination (environment of Fig. 7.11). Same arrangement as Fig. 7.15 (exception: differences are multiplied by 10). (a) Reference: BRDF of Fig. 7.11(b). (b) BRDF of Fig. 7.12. (c) BRDF of Fig. 7.10.

8. Estimating Spatially Varying BRDFs

Up to now, the reflection properties of the surface were homogeneous for all points. Thus, within a single image, a large number of samples corresponding to a single BRDF are observed. With the presented linear BRDF model, the reconstruction process results in an overdetermined system which can be solved in the least-squares sense. If, however, the reflection properties of a surface change from one point to another, i.e., the BRDF of the surface is spatially varying, the reconstruction gets more complicated.

In combination with the linear BRDF model described in Sec. 7.1, the weighting coefficients a_l^K for each basis BRDF become dependent on surface location \mathbf{x} , leading to the following spatially varying BRDF model with the weighting coefficient map $a_l^K(\mathbf{x})$:

$$f^K(\boldsymbol{\omega}_i, \mathbf{x}, \boldsymbol{\omega}_o) = \sum_{l=0}^K a_l^K(\mathbf{x}) f_l^K(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o). \quad (8.1)$$

With only two measurements per surface point, it is not possible to reliably reconstruct a single BRDF per point. A single surface point yields two observations, but there are K unknowns in the linear BRDF model, so the reconstruction is greatly undetermined. However, some points yield more information for the reconstruction process than others. If the observed color at a point changes drastically between two views, the surface is specular at this point. Certainly, if the observed color at a point stays the same between two views, this does not mean that the point is not specular at all. There are at least three possibilities:

1. The point is diffuse.
2. The point is specular, but the illumination does not produce a highlight at its location on the surface.
3. The point is specular and in both views there is a highlight visible at its location.

Therefore, it is necessary to establish some coherence between points. For example, points with similar color or surrounding texture could form a group in which all points share the same BRDF. It is also possible that the points only share some special properties, e.g., the shape of the specular lobe, instead of the full BRDF.

Depending on the surface’s appearance, one strategy might work better than the other.

In this chapter, three approaches to reconstruct spatially varying BRDFs are presented. Each approach is based on different assumptions, and thus each is capable of handling another surface type. In each case, advantages and disadvantages are discussed and examples for successful reconstructions as well as problematic cases are given.

8.1. Clustering Similar Points

In some cases, planar objects consist of few different materials that are uniquely assigned to points, i.e., each surface point corresponds to exactly one material. In this case, it is possible to partition the surface into groups of materials, and determine the BRDF of each group. The partitioning depends on some criteria, however, as reflections and highlights can change the color of a point significantly, a simple partitioning scheme based on color, intensity, or chromaticity is problematic. An example is given in Fig. 8.5(c). Therefore, the following approach combines a heuristic based on chroma with residual analysis of random models fitted to the observations.

In this approach, the surface consists of M BRDF models that are based on K basis BRDFs. Each model m has its own weighting coefficients $a_{l,m}^K$, $m = \{1, 2, \dots, M\}$. A function $s(\mathbf{x})$ assigns a model to each surface point, so $s(\mathbf{x}) \in \{1, 2, \dots, M\}$. The weighting coefficient map is given as

$$a_l^K(\mathbf{x}) = a_{l,s(\mathbf{x})}^K. \quad (8.2)$$

Now, the problem is to determine M , $a_{l,m}^K$, and $s(\mathbf{x})$. An algorithm to deal with this kind of problems is introduced in [ZK07], an overview is given in Sec. 2.4. The technique consists of two steps. At first, N model hypotheses are generated by randomly choosing data points to create a minimal sample set (MSS) and by fitting the models to the MSS (using non-negative linear least squares optimization as in the homogeneous case). Afterwards, the residuals of the fitted models are analyzed to choose the ones that best fit the data. Fig. 8.1 shows how this procedure is embedded within the system.

There are a few difficulties to consider when adapting the algorithm to BRDF estimation. In order to approximate BRDFs of materials, usually $K = 21$ basis BRDFs are needed in the linear BRDF model (see Sec. 7.1). Therefore, the MSS must contain at least 21 measurements. If, for example, the surface points can be partitioned into two equally-sized groups, estimating each individual BRDF means to deal with 50% outliers. Unfortunately, it is very unlikely that a MSS will entirely consist of inliers if 21 measurements are chosen randomly from the two views. Therefore, a large number of hypotheses have to be generated to ensure that at least one of them does not contain any outliers. Moreover, it is necessary that enough hypotheses are

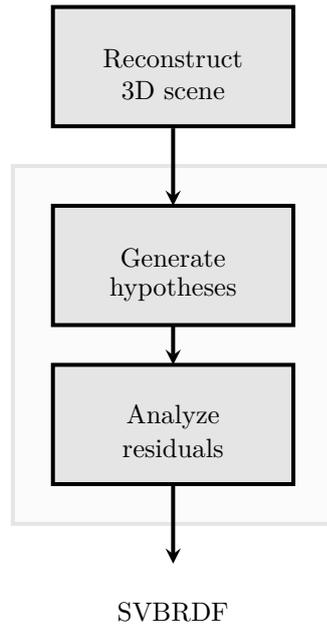


Figure 8.1.: System overview with clustering approach. BRDF estimation is performed in two steps: hypotheses generation and residual analysis.

outlier-free to separate them from hypotheses with outliers, so the problem gets even more difficult. Consequently, it is not feasible to choose data-points completely random, instead some heuristic has to be used to increase the probability that a MSS is only composed of inliers.

To form a model hypothesis, K measurements have to be chosen randomly. Ideally, all measurements correspond to the same BRDF. In the proposed approach, a heuristic based on chroma of surface points is used to increase the probability that each of the data points in the MSS corresponds to the same BRDF. Assuming purely white illumination and white specular BRDF colors, observed chroma stays the same at each point on the surface belonging to the same BRDF. This assumption is not valid anymore if the light or the specular BRDF is colored, nevertheless the chroma observed in highlighted regions varies smoothly (except for specular materials with very high exponents). This property can be used to build the selection algorithm.

Instead of randomly selecting K measurements and inserting all of them into the MSS, only data points that are presumably related to the same BRDF (according to a heuristic) are inserted. The first selected data point determines the BRDF of the new MSS, so all subsequently selected points have to be checked whether they correspond to this BRDF or not. Using two views, there are always two data points corresponding to the same surface point, so these two data points share the same BRDF. Therefore, the first criterion is that data points are always chosen in pairs, one data point for each view. Furthermore, for each subsequent data point P_k , a line

is traced from the starting point P_0 to P_k . If the line crosses the boundary of two adjacent regions with different BRDF (large chroma gradient), the chroma value before the crossing is stored. From now on, the line passes through a region with different BRDF and if P_k is reached, it is rejected. However, it is possible that the starting region can be entered again. At each subsequent boundary crossing, the stored chroma value is compared to the chroma in the entered region. If the values are equal, the region of the starting point is entered again. P_k is accepted if it is reached within the region of the starting point P_0 . Euclidean distance is used to compare chroma values. Because of the two views, two data points at the first surface point have to be compared to two data points at the second surface point. This is handled by calculating the Euclidean distances independently for the views and adding them afterwards. In Alg. 2, the algorithm is described in pseudo code.

Algorithm 2 Heuristic to choose random points for model hypothesis generation.

```

 $P_0 \leftarrow$  random point
 $prev\_chroma \leftarrow$  chroma at  $P_0$  in all views
 $k \leftarrow 1$ 
while  $k < K$  do
   $P_k \leftarrow$  random point
   $state \leftarrow$  INSIDE_STARTING_REGION
  for all points on line between  $P_0$  and  $P_k$  do
     $cur\_chroma \leftarrow$  chroma at  $P_k$  in all views
     $chroma\_diff \leftarrow \|cur\_chroma - prev\_chroma\|^2$ 
    if  $chroma\_diff > T_1$  then
      if  $state =$  INSIDE_STARTING_REGION then
         $last\_valid\_chroma = prev\_chroma$ 
         $state \leftarrow$  OUTSIDE_STARTING_REGION
      else
         $chroma\_diff \leftarrow \|last\_valid\_chroma - cur\_chroma\|^2$ 
        if  $chroma\_diff < T_2$  then
           $state \leftarrow$  INSIDE_STARTING_REGION
        end if
      end if
    end if
     $prev\_chroma \leftarrow cur\_chroma$ 
  end for
  if  $state =$  INSIDE_STARTING_REGION then
    accept  $P_k$ 
     $k \leftarrow k + 1$ 
  else
    reject  $P_k$ 
  end if
end while

```

Using this heuristic, only a small number of hypotheses is necessary, e.g., $N = 500$. Choosing an appropriate threshold for boundary detection is difficult, because if the threshold is too small, chroma changes due to highlights with high specular exponent will be classified as boundary. If it is too high, boundaries between regions with similar chroma will be missed. In this implementation, $T_1 = T_2 = 0.0005$ is used.

Adapting the work of [ZK07], the scene is rendered for each of the N model hypotheses and the residuals are calculated and accumulated at each point \mathbf{x} . The residual is defined as the Euclidean distance $d_{\mathbf{x}}^n$ between observed color at \mathbf{x} and estimated color, normalized to the observed color and clamped to $0 \leq d_{\mathbf{x}}^n \leq 20$, $n = \{1, 2, \dots, N\}$. In the analysis step, a residual histogram is generated at each point by dividing the interval $[0; \max_{x,n} d_{\mathbf{x}}^n]$ into B equally sized bins. Afterwards, modes are searched in the histograms. The median number of modes yields the number of models M . The model parameters $a_{l,m}^K$ corresponding to the peaks in the histograms are extracted and each surface point is assigned to one of the models, resulting in $s(\mathbf{x})$. In the original formulation, each observed data point is assigned to the nearest model. However, if the difference between the models is small and because the rendered basis BRDFs are noisy (due to few samples in Monte-Carlo integration), the result will be noisy. Therefore, the assignment takes the spatial neighborhood of the points into account. For each point, the difference between the point's neighborhood (3×3) and each model is used to determine the model to which the point belongs. After assignment, $a_{l,m}^K$ is estimated again with all points belonging to model m .

Discussion Reconstruction results are shown in Figs. 8.2-8.5. The final BRDF for each region is computed by solving an overdetermined linear system with each of the region's pixels. This is the same procedure as used in the homogeneous case (Sec. 7.2), but with less constraints. Consequently, the quality of the estimated BRDF is similar or worse, depending on the illumination and location of the region.

Due to random sampling, small regions can be missed, especially if a small number of hypotheses N with respect to the size of the surface (in pixels) is used. In this case, it is unlikely that the starting point lies in such a small region, consequently there are few hypotheses for the region's BRDF. This does not lead to a clear peak in the residual histogram. Therefore, no model is generated for the region in the final result. As a consequence, the approach is not suitable for surfaces that contain one or more small regions, at least if N is small.

The quality of segmentation is largely influenced by the filter used to smooth the residual histograms and the choice for the number of bins B . Both define the minimal allowed distance between the models, making it dependent on the input data. Images with noise demand for a smaller B . However, if B is too small, it will not be possible to differentiate between similar models. In scenes such as Fig. 8.2(c) it is difficult to find the best parameterization, as the BRDFs are very similar (smooth specular lobe, similar color). The results in Fig. 8.2 and Fig. 8.5 are created by smoothing

the histogram with $B = 200$ bins using a Gaussian filter with $\sigma = 0.8$ and $\sigma = 1.5$, respectively.

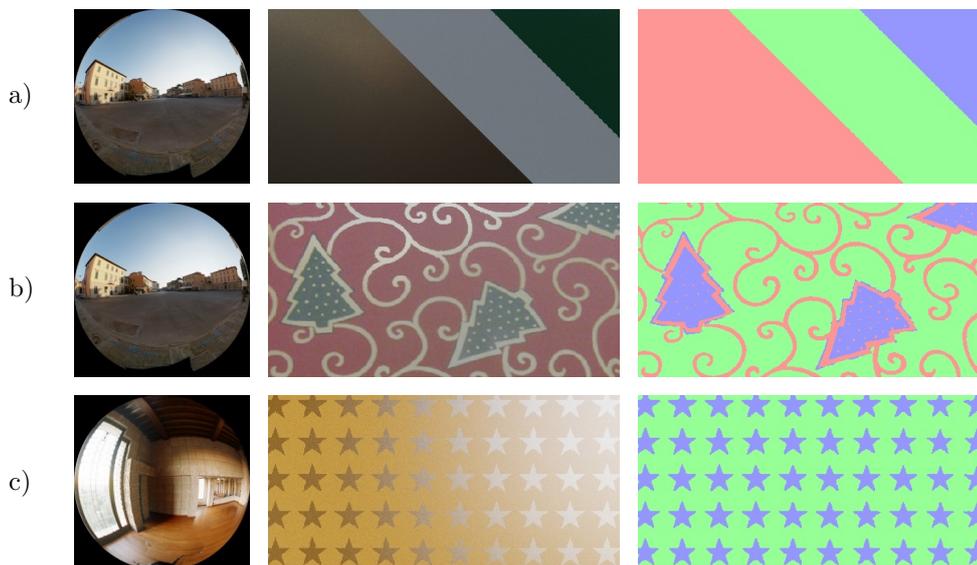


Figure 8.2.: Results of clustering approach for different scenes. From left to right: environment, one of the two input images, reconstructed segments (color-coded). Texture (b) taken from [DTPG11].

Furthermore, alignment errors (Sec. 5.5.1) are mostly visible at highlights from bright light sources, leading to large differences d_x^n in regions where a highlight should be visible according to the scene geometry, but actually is not present in the images. Therefore, the points in such regions form additional modes in the residual histogram, leading to an additional BRDF that is assigned only to these points. In the result this emerges as an additional (wrong) model for the points surrounding highlights.

The assignment of data points to the models is also problematic for surfaces with bumpy appearance (Fig. 8.4). In this example, normals are ignored, meaning that the variations due to normals are considered as noise. Using the same value for σ as in the other synthetic examples results in a wrong estimate for the number of segments. After adjusting σ slightly, the number of segments is correct, but the assignment of data points is still wrong. As described in Sec. 6, normal reconstruction does not work perfectly and BRDF estimation has to be able to deal with wrong normals. The influence of normal reconstruction on BRDF estimation is further discussed in Sec. 9.

Finally, many surfaces do not satisfy the assumptions in this approach. Smooth transitions between two BRDFs are not possible and textured materials can not be handled.

Summing up, the approach suffers mainly from two problems: difficult parameterization for robust results and inability to represent textured surfaces. These problems

are addressed in the following methods. Instead of fitting multiple models to the data, the approaches work image-based, i.e., the captured photos are separated into diffuse and specular components in order to make them relightable without the need to fit a model to each pixel.



Figure 8.3.: Surfaces of Fig. 8.2 under novel illumination.



Figure 8.4.: Results of clustering approach for a bumpy surface (from left to right: one of the two input images, reconstructed segments, reconstructed segments for adjusted σ).



Figure 8.5.: Results of clustering approach for a real-world scene. From left to right: environment, input images, reconstructed segments (color-coded) and surface under novel illumination.

8.2. Diffuse-Specular Separation

To overcome the problems of the clustering approach, especially difficult parameterization and limited support for textured surfaces, the following approaches separate the surface into diffuse and specular parts. The first assumes that the surface has a diffuse texture with per-pixel information for albedo and a single specular BRDF assigned to all pixels. The second approach additionally tries to build a specular map that clusters the surface into a region with glossy materials and a region with matte materials. The pixel in the specular map determines the assignment of each surface point to one of the regions. For each region, the specular lobe is estimated. The diffuse part of the BRDF is still calculated on a per-pixel basis.

For both approaches it is required that a BRDF model is used which allows to be split into diffuse and specular parts. This is the case for the linear model described in Sec. 7.1, but for some BRDFs the classification is uncertain, particularly if the BRDF is nearly diffuse except for a small directional dependence.

8.2.1. Handling Textured Surfaces

A large variety of surfaces are composed of a texture that alters the diffuse color at each point and a single specular BRDF that is shared by all pixels. In this case, the surface is usually composed of two layers: the bottom layer exhibits diffuse reflection properties with multiple colors, the top layer is transparent and highly specular.

The diffuse texture is described by the function $\rho(\mathbf{x})$ that assigns a color value to each surface point. The specular part of the BRDF consists of K weighting coefficients α_l^K for the basis BRDFs. As it is assumed that all points share the same specular properties, α_l^K does not depend on \mathbf{x} . In this approach, the weighting coefficient map is given as

$$a_l^K(\mathbf{x}) = \begin{cases} \rho(\mathbf{x}) & \text{for } l = 0 \\ \alpha_l^K & \text{for } l > 0 \end{cases} . \quad (8.3)$$

Diffuse and specular components are estimated iteratively. The complete system is depicted in Fig. 8.6.

Similar to Eq. (7.7) and with $L_l(\mathbf{x}, \boldsymbol{\omega}_o)$ from Eq. (7.9), the rendering equation in this approach looks as follows:

$$L_o(\mathbf{x}, \boldsymbol{\omega}_o) = \rho(\mathbf{x})L_0(\mathbf{x}, \boldsymbol{\omega}_o) + \sum_{l=1}^K \alpha_l^K L_l(\mathbf{x}, \boldsymbol{\omega}_o) . \quad (8.4)$$

When the unknowns $\rho(\mathbf{x})$ and α_l^K are combined in a single vector, the solution can again be computed via linear least-squares optimization. The system of equations is

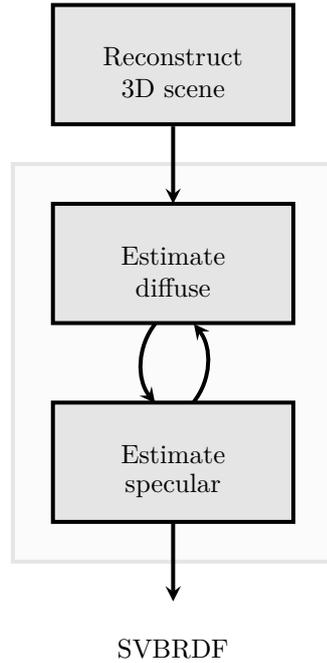


Figure 8.6.: System overview with first approach of diffuse-specular separation. The BRDF is estimated by iteratively updating the diffuse and the specular component. All points on the surface share the specular component, the diffuse component may vary.

overdetermined if at least two input images are given. However, the optimization involves computations with large sparse matrices, which has either large memory requirements or is complex to implement.

However, diffuse and specular coefficients can be estimated by alternating between solving for $\rho(\mathbf{x})$ and α_l^K . For simplicity, from now on the equation is discretized and written in matrix notation:

$$l_o = \rho L_0 + \alpha L_l. \quad (8.5)$$

In an iterative algorithm, ρ and α are estimated one after another by successively subtracting the current estimate for specular lobe and diffuse color, yielding the following update rules:

$$l_o - \alpha^{i-1} L_l = \rho^i L_0 \quad (8.6)$$

$$l_o - \rho^i L_0 = \alpha^{i+1} L_l. \quad (8.7)$$

ρ and α are computed using non-negative linear least-squares optimization, analog to the homogeneous case (Sec. 7.2). On initialization, $\alpha^0 = \mathbf{0}$. The algorithm terminates after a fixed number of iterations or if the residuals between two successive iterations do not decrease anymore (based on a threshold).

Discussion As all points of the surface contribute to the estimation of the specular reflection properties, it can be approximated nearly as well as in the homogeneous case. However, this is not the case for the diffuse information, which is calculated on a per-pixel basis. Therefore, it is very sensitive to noise and errors in calibration. If both views and the environment map are perfectly aligned and if the specular component is well approximated, the highlight will be removed nearly perfectly and the estimated BRDF matches the reference BRDF very well. Therefore, the approach generates good results for synthetic scenes (Fig. 8.7). As real-world scenes with small specular contributions are not sensitive to alignment errors either, the method works too (Fig. 8.8). Unfortunately, calibration is a major problem for materials with large specular component and bright highlights on the surface. In this case, the highlights are not removed completely from the diffuse texture or they are removed at the wrong location, leading to wrongly darkened regions (Fig. 8.10(a)). Still, the specular component of the BRDF is approximated sufficiently. If the surface is specular and bumpy, the same problem will occur. As normals are not known, highlights are not entirely removed from the diffuse texture (Fig. 8.10(b)). More details on the combination of normal reconstruction and BRDF estimation are given in Sec. 9.

In conclusion, the separation approach works successfully for materials with small specular contributions. It fails for highly specular materials, because the estimated diffuse component contains clearly visible artifacts from the specular part. Nevertheless, in some cases it is possible to remove these artifacts. As the problematic regions can be located on the surface, texture inpainting can be used to fix these regions. Less complicated, the minimum color value at each surface point can be used as diffuse contribution, because highlights usually increase the brightness at a point. The following method makes use of these facts.

8.2. Diffuse-Specular Separation

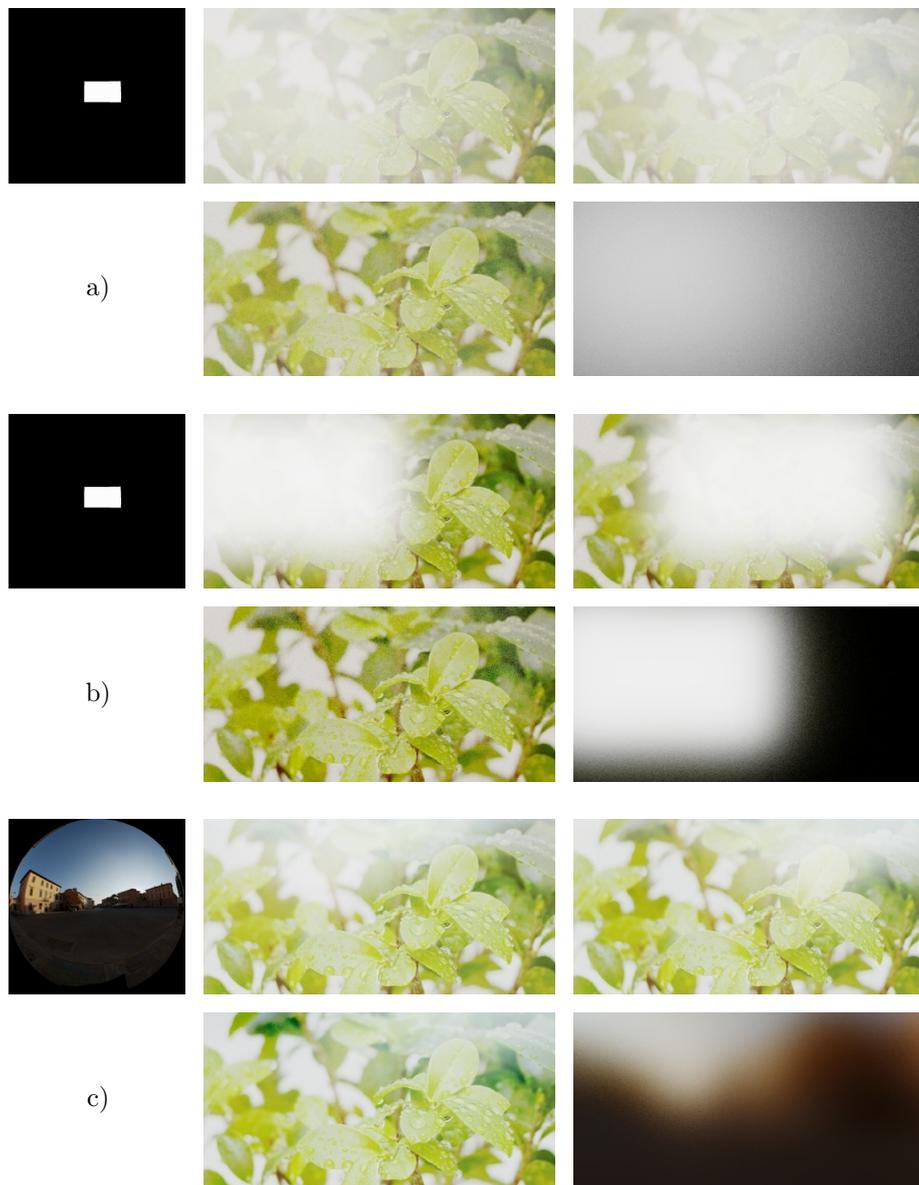


Figure 8.7.: Results of the diffuse-specular separation approach for synthetic scenes. From left to right and top to bottom: environment, first input image, second input image, estimated diffuse component, estimated specular component.

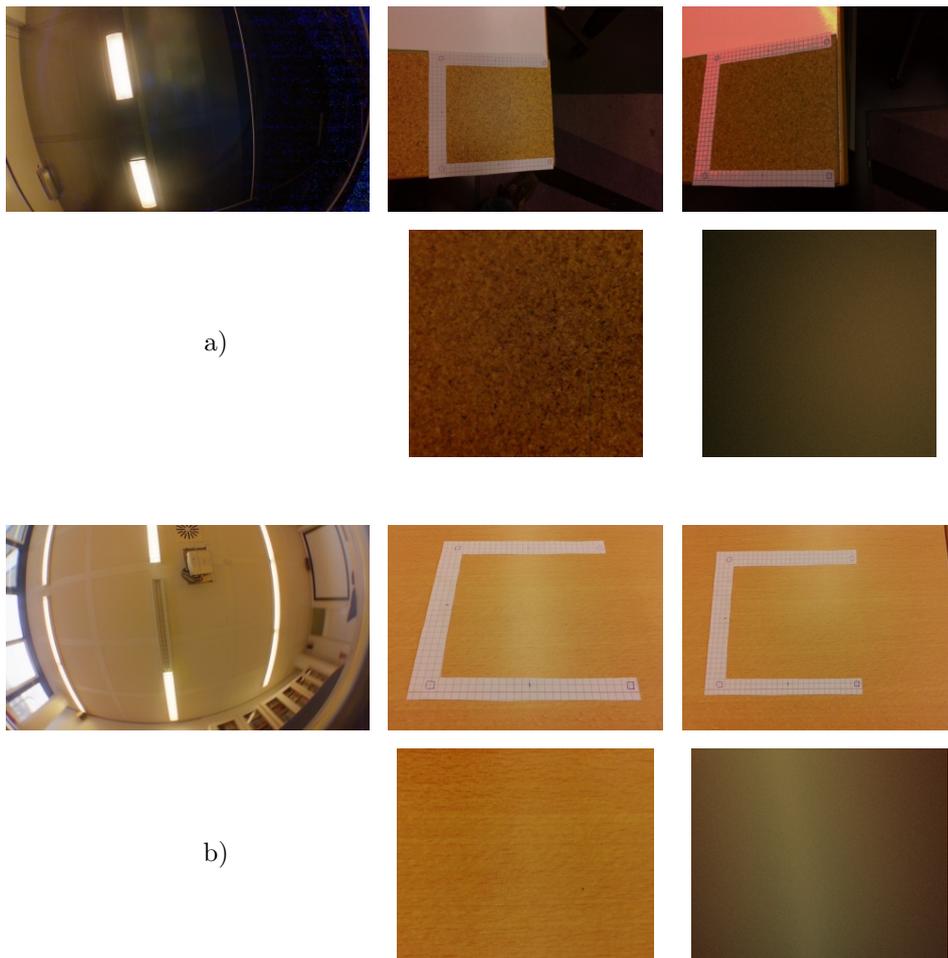


Figure 8.8.: Results of the diffuse-specular separation approach for real-world scenes (same arrangement as in Fig. 8.7).



Figure 8.9.: Surfaces from Fig. 8.8 rendered under novel illumination.

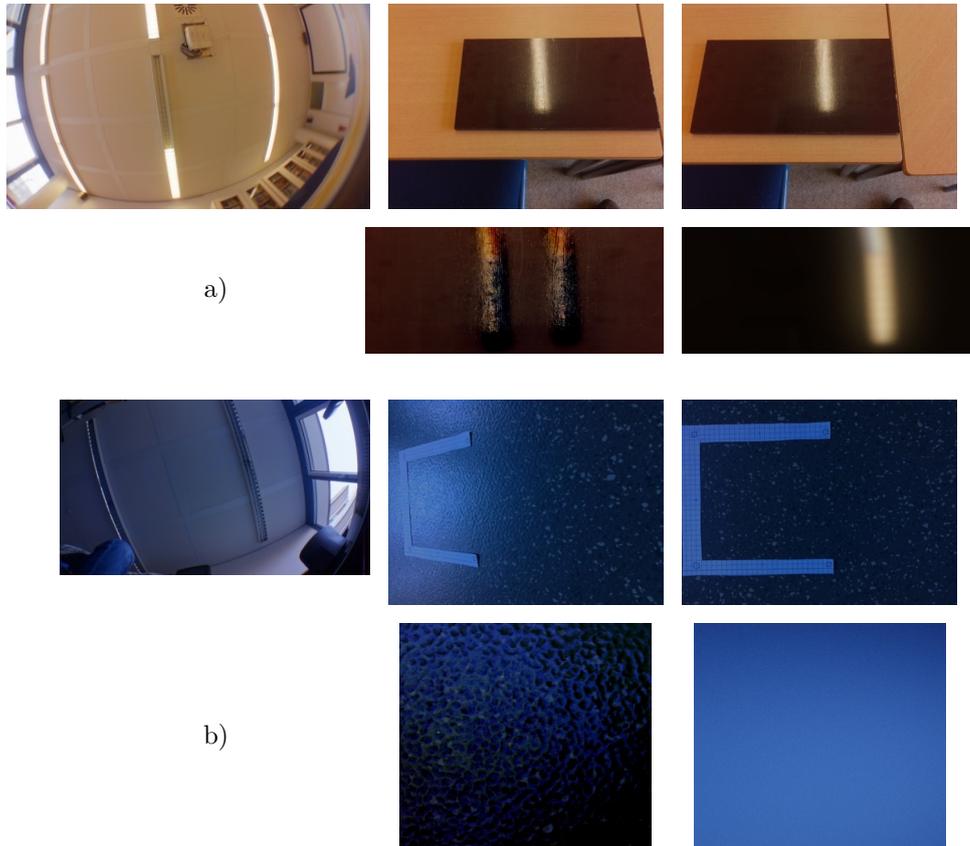


Figure 8.10.: Failure cases of the diffuse-specular separation (same arrangement as in Fig. 8.7). (a) Strong highlights are not entirely removed in the diffuse texture. (b) Highlights on a bumpy surface cannot be removed from the diffuse texture without an exact normal map.

8.2.2. Combining Separation and Clustering

Analyzing the difference image between both views yields information about specular properties of some points. If the point changes its color, it is specular (see Sec. 8). However, the difference image does not contain any information for points that do not change its color, they could be diffuse or specular. The following approach is based on this observation. Each point is a weighted combination of two groups, a group with almost diffuse reflection properties and a group with diffuse and highly specular reflection properties (i.e., matte and glossy regions). Furthermore, the disadvantages of the former method regarding calibration errors are avoided. Therefore, it is better suited for real-world scenes, although captured images have to satisfy additional requirements.

Each of the two groups is represented by the weighting coefficient map given in Eq. (8.3). The final BRDF per point is a weighted combination of the two groups, determined by the specular map $w(\mathbf{x})$. In consequence, the combined weighting coefficient map becomes

$$a_l^K(\mathbf{x}) = (1 - w(\mathbf{x}))a_{l,d}^K(\mathbf{x}) + w(\mathbf{x})a_{l,s}^K(\mathbf{x}) \quad (8.8)$$

$$= \begin{cases} (1 - w(\mathbf{x}))\rho_d(\mathbf{x}) + w(\mathbf{x})\rho_s(\mathbf{x}) & \text{for } l = 0 \\ (1 - w(\mathbf{x}))\alpha_{l,d}^K + w(\mathbf{x})\alpha_{l,s}^K & \text{for } l > 0 \end{cases}, \quad (8.9)$$

where $a_{l,d}^K(\mathbf{x})$ is the weighting coefficient map of the matte regions and $a_{l,s}^K(\mathbf{x})$ is the weighting coefficient map of the glossy regions. In other words, the surface is represented by a diffuse per-pixel color and two regions with small and large specular contributions. This means that two specular lobes have to be estimated. The per-pixel weighting $w(\mathbf{x})$ interpolates between these two lobes. The diffuse color is also weighted by $w(\mathbf{x})$, because it is separately estimated in matte and glossy regions.

Fig. 8.11 gives an overview on the system. First, the specular map $w(\mathbf{x})$ is generated. Next, a BRDF with a spatially varying diffuse and a homogeneous specular component is estimated for each region. The method to estimate the BRDF is different for the two groups. Matte regions are handled with the separation approach described in Sec. 8.2.1. Glossy regions are handled similar, but the problems of the separation approach are avoided by locating and removing highlights independently of the specular estimation. Finally, the specular map is used to blend the separate parts at each point.

Estimating the specular map The specular map is estimated by finding image regions for which it is very likely that they belong either to a matte or to a glossy material. Typically, only a few points can be safely assigned to one of the groups. The remaining points are assigned by examining similarities in albedo.

At first, glossy points are identified. Specular materials can be detected by analyzing the differences between the surface points $I_i(\mathbf{x})$ extracted from the input images.

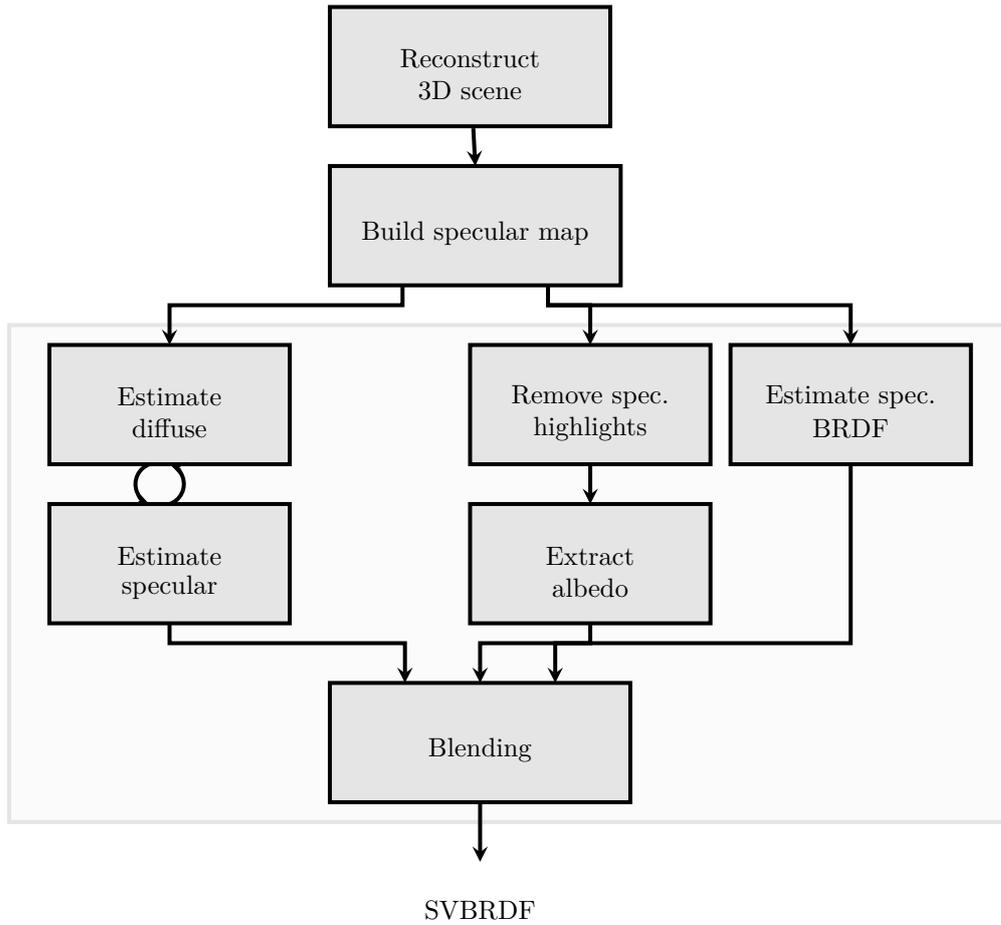


Figure 8.11.: System overview with combination of clustering and separation approach. The surface is clustered into a region containing matte materials and a region containing glossy materials. Each pixel gets a weighting coefficient that describes its assignment to one of the regions (specular map).

The varying camera position lets the specular highlights change their position, which can be identified in the differences. The detection begins by calculating the minimum and maximum intensity at each pixel:

$$I_i^t(\mathbf{x}) = \frac{1}{3} \left(I_i^r(\mathbf{x}) + I_i^g(\mathbf{x}) + I_i^b(\mathbf{x}) \right) \quad (8.10)$$

$$I_{min}^t(\mathbf{x}) = \min_i I_i^t(\mathbf{x}) \quad (8.11)$$

$$I_{max}^t(\mathbf{x}) = \max_i I_i^t(\mathbf{x}). \quad (8.12)$$

$I_i^t(\mathbf{x})$ determines the intensity of surface point \mathbf{x} observed in the i -th input image with the three color channels $I_i^{\{r,g,b\}}(\mathbf{x})$. Highlights are detected by thresholding the relative difference between $I_{min}(\mathbf{x})$ and $I_{max}(\mathbf{x})$. If the intensity of a point doubles,

it will be classified as highlight:

$$I_h(\mathbf{x}) = \begin{cases} 1 & \text{if } \frac{I_{max}^t(\mathbf{x})}{I_{min}^t(\mathbf{x})} > 2 \\ 0 & \text{otherwise} \end{cases} . \quad (8.13)$$

Noise is removed by eroding $I_h(\mathbf{x})$ with a disk-shaped structuring element of size 3×3 . The remaining points are assigned to the group of mixed materials.

In the next step, matte regions are searched. For this purpose, an initial estimation for the specular BRDF is computed with the points of glossy materials found so far. As the diffuse component can vary from point to point, the technique described in Sec. 8.2.1 is used in this step. Afterwards, using the specular component of this BRDF, the surface is rendered from the same camera position as in the original views. These images will be denoted by $R_i(\mathbf{x})$. Analog to the computation of $I_{\{min,max\}}^t(\mathbf{x})$, minimum and maximum of $R_i(\mathbf{x})$ are generated, yielding $R_{\{min,max\}}^t(\mathbf{x})$.

The key point in this strategy is the following: if the observed surface was entirely built of specular materials, it would look like the just rendered views. Thus, the rendered views have to be compared with the observations. To get the views and the observations comparable, only specularity is considered, extracted by using differences in intensity:

$$I_{diff}^t(\mathbf{x}) = I_{max}^t(\mathbf{x}) - I_{min}^t(\mathbf{x}) \quad (8.14)$$

$$R_{diff}^t(\mathbf{x}) = R_{max}^t(\mathbf{x}) - R_{min}^t(\mathbf{x}) . \quad (8.15)$$

By comparing $R_{diff}^t(\mathbf{x})$ and $I_{diff}^t(\mathbf{x})$, it is possible to identify regions that should be glossy but actually are matte. The comparison is simply performed by subtracting $I_{diff}^t(\mathbf{x})$ from $R_{diff}^t(\mathbf{x})$ and thresholding the result. Threshold T must assure that only surface points with bright highlights are considered:

$$D^t(\mathbf{x}) = \begin{cases} 1 & \text{if } R_{diff}^t(\mathbf{x}) - I_{diff}^t(\mathbf{x}) > T \\ 0 & \text{otherwise} \end{cases} . \quad (8.16)$$

At this stage, $D^t(\mathbf{x})$ is 1 at matte points and $I_h(\mathbf{x})$ is 1 at glossy points. The remaining points which are not classified so far are assigned to one of the groups by similarity in albedo color. For this purpose, an initial estimate of albedo is generated by computing the minimum color value $I_{\{min\}}^{\{r,g,b\}}(\mathbf{x})$ at each point, assuming that highlights always increase the point's color. Note that this is not the true albedo, because irradiance would have to be removed from the color value to obtain the actual albedo. Afterwards, a process inspired by the work of [DTPG11] is used to group the points. Albedo for pixels in $D^t(\mathbf{x})$ and $I_h(\mathbf{x})$ are inserted into two sets D and I , respectively. It can happen that the sets contain outliers, i.e., a value is included in both sets. Outliers are removed by checking each entry whether 60% of its 1000 nearest neighbors are contained in the same or in the other set. If the

neighbors are contained in the other set, the value is rated as outlier and is removed from the set. The square norm is used as similarity measure. Finally, the assignment of the remaining points is performed by determining whether D or I contains the nearest neighbor based on Euclidean distance. The specular map $w(\mathbf{x})$ is 0 at matte points and $0 < s \leq 1$ at glossy points. If the nearest neighbor to the point is in I and the Euclidean distance in chromaticity between the point and the neighbor is c , then $s = 1 - \sqrt{c}$.

Reconstructing the matte region The spatially varying BRDF of the matte region is estimated using the separation approach presented in Sec. 8.2.1. In the optimization process, the observation of each surface point is weighted with the corresponding point in the specular map with $(1 - w(\mathbf{x}))^4$. This ensures that only matte points contribute to the result. Furthermore, the linear BRDF model is restricted to broad specular lobes in order to avoid problems like depicted in Fig. 7.11. As a side effect, this decreases the computation time, because fewer weighting coefficients have to be estimated. The result of this step is $\rho_d(\mathbf{x})$ and $\alpha_{l,d}^K$.

Reconstructing the glossy region It is not possible to reconstruct glossy surfaces using only the separation of Sec. 8.2.1, as bright highlights are not fully removed from the diffuse component. Therefore, diffuse and specular components of glossy regions are extracted independently by different methods.

The diffuse component is reconstructed by locating and removing specular highlights directly from the input images. Most environments can be described by smooth ambient light and few bright spots from light sources. Usually, the light sources create bright highlights on the surface. Assuming that highlights always increase the brightness at a point, computing the minimum color at each surface point results in a specular-free surface $I_{min}(\mathbf{x})$. It is, however, necessary that highlights do not overlap in the captured images. This has to be ensured when taking the photos. This simple highlight removal technique is also described in Sec. 6.1.

The minimum $I_{min}(\mathbf{x})$ contains radiance measurements per point. In order to obtain albedo, the irradiance has to be removed. As the illumination is known, the irradiance $E(\mathbf{x})$ for each surface point (\mathbf{x}) can be easily computed (Eq. (2.34)). Then, $\hat{\rho}_s(\mathbf{x})$ becomes

$$\rho_s(\mathbf{x}) = \pi \frac{I_{min}(\mathbf{x})}{E(\mathbf{x})}. \quad (8.17)$$

The specular component of the BRDF in the glossy region is again computed by separation (Sec. 8.2.1), although the method produces diffuse and specular components (i.e., the diffuse component is discarded). Again, in the optimization the surface points are weighted according to the specular map with $w(\mathbf{x})^4$. This step results in $\alpha_{l,s}^K$.



Figure 8.12.: Results of the combined separation and clustering approach for a real-world scene. From left to right and top to bottom: environment, input images, estimated specular map and surface under novel illumination (*ennis* and point-light illumination).

Combining the estimations In the last step, the four parts are blended according to $w(\mathbf{x})$ to form the final spatially varying BRDF. At each surface point, $a_l^K(\mathbf{x})$ is computed by linearly interpolating between $a_{l,d}^K(\mathbf{x})$ and $a_{l,s}^K(\mathbf{x})$. As $w(\mathbf{x})$ is 0 in the matte region and between 0 and 1 in the glossy region, the matte BRDF is assigned to the matte region and a mixture of both BRDFs is assigned to the glossy region.

Discussion A major difference between this approach and the previous (Sec. 8.2.1) is that albedo is extracted directly from images in glossy regions. This should prevent artifacts from highlights that can not be removed perfectly, because of imperfect calibration or inaccurately approximated specular lobes. However, during image capturing it has to be ensured that each surface point is visible at least once without highlight. There are environments and materials for which it is not possible to fulfill this requirement, so this approach can not be used in all cases.

Like in the approach presented in Sec. 8.2.1, the number of specular lobes is limited, in this case to one matte and one glossy lobe. By further analyzing the difference images, it may be possible to separate points by the shape of the lobe. However, this is a challenging problem, mainly due to noise and other factors that degrade the quality of the difference images, for example, inaccurately aligned textures extracted from the surface in the two views (due to errors in homography estimation, HDR image merging, etc.).

Figs. 8.12-8.14 show results of the approach for real and synthetic scenes. The surface

8.2. Diffuse-Specular Separation

in Fig. 8.13(a) contains some perturbed normals, which produce details that are lost in the result, as normal reconstruction is not used in this case. Fig. 8.13(b) shows a surface with very similar colored diffuse and specular regions. Nevertheless, only a few small points are wrongly assigned to the glossy region. However, applying filters to remove these outliers would also remove the small details in Fig. 8.13(a).

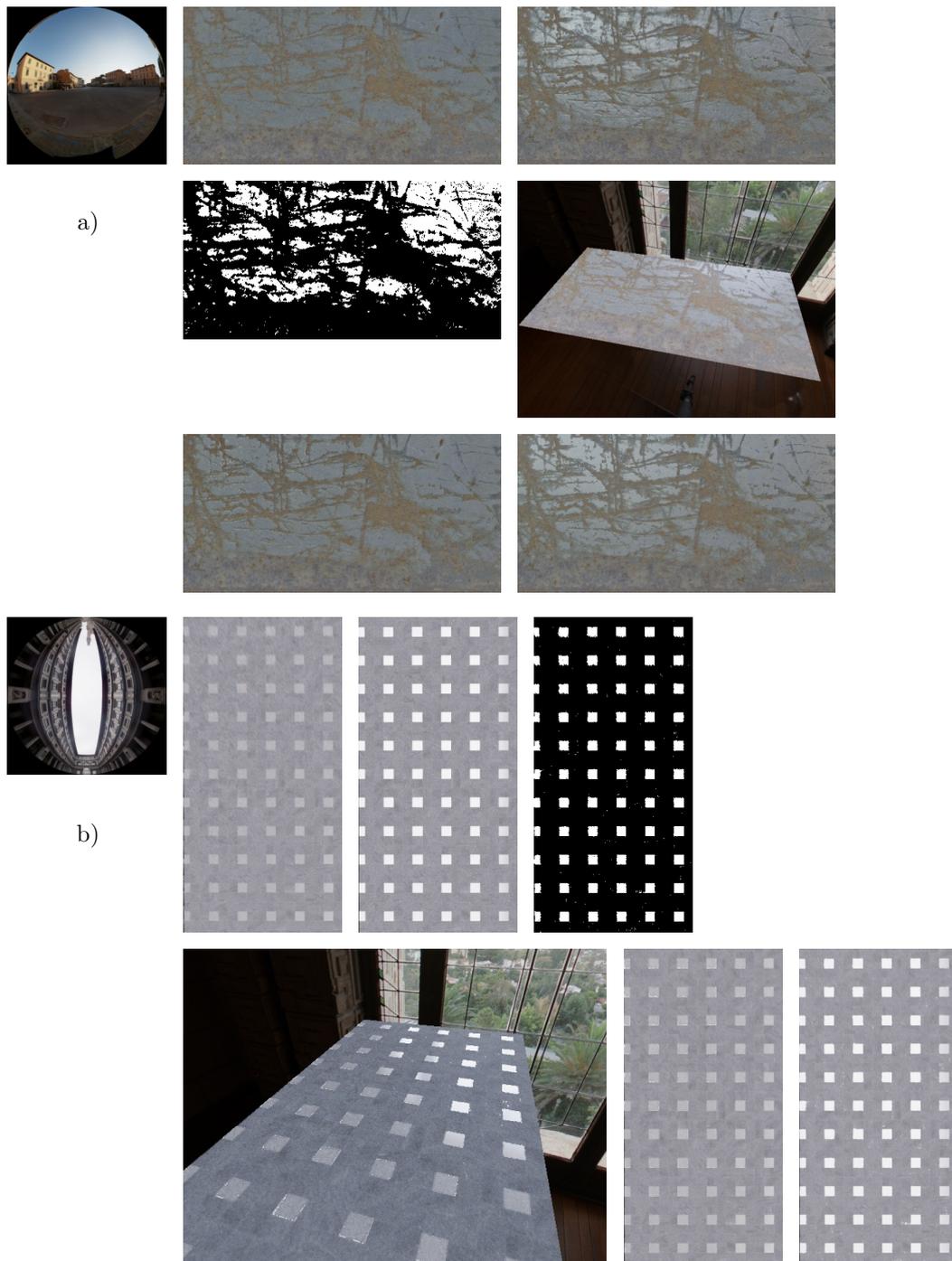


Figure 8.13.: Results of the combined separation and clustering approach for synthetic scenes. From left to right and top to bottom: environment, input image 1, input image 2, estimated specular map, result rendered under novel illumination, result rendered under same illumination as input images. Note that the first input image always shows the surface without highlights, so the albedo can be separated from specular reflections.

8.2. Diffuse-Specular Separation

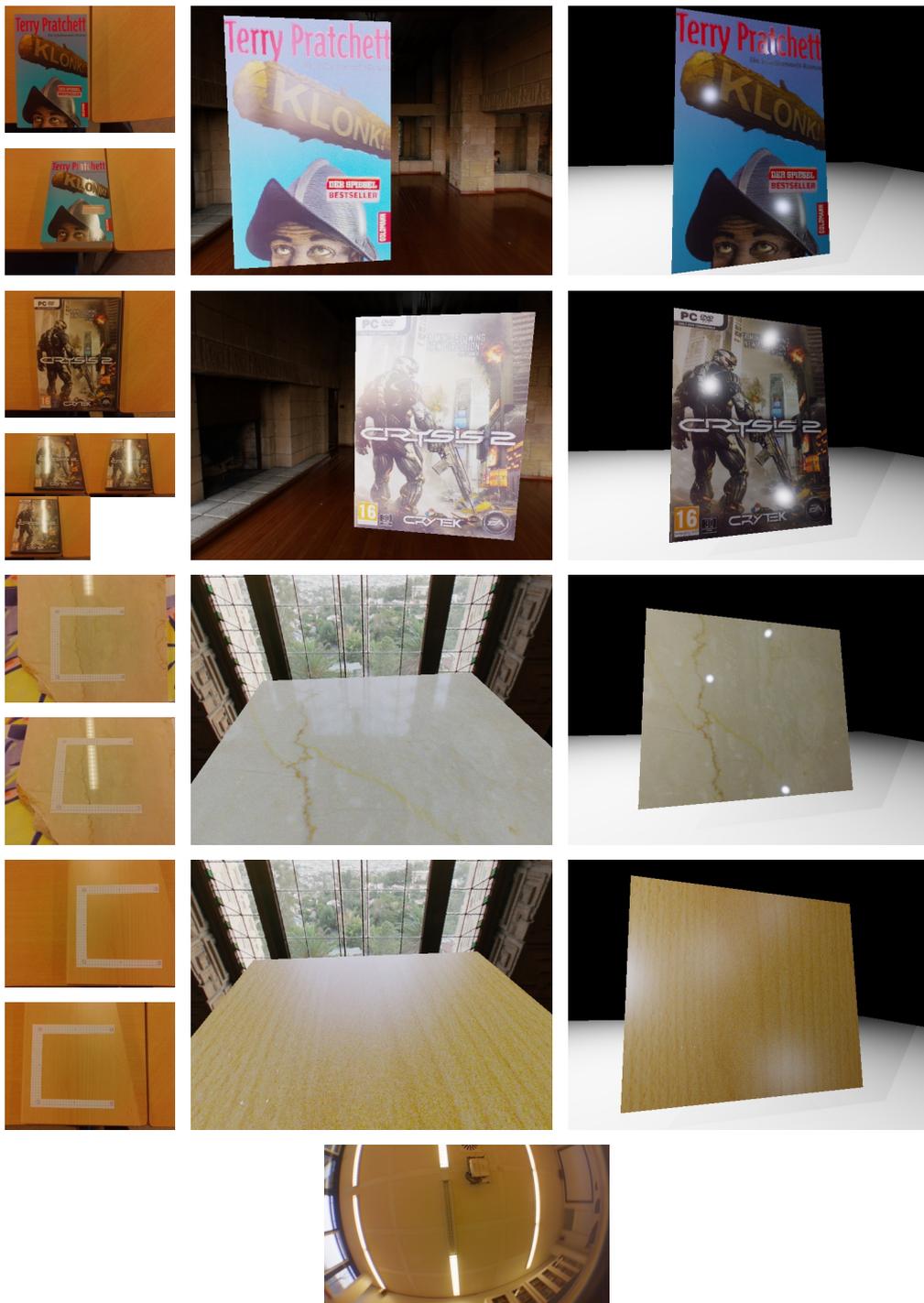


Figure 8.14.: Results of the combined separation and clustering approach for real-world scene taken under the illumination in the bottom row. From left to right: input images, relighting with *ennis* environment, relighting with point-light sources.

9. Combining Normal and BRDF Estimation

In the previous chapters, algorithms for normal reconstruction and BRDF estimation were presented independently. Now, these algorithms are combined to reconstruct normal map and BRDF of the surface. In the following, some problems that occur due to the combination of the algorithms are discussed and shown in various examples.

BRDF estimation provides the best results if the normal map is known. However, in order to reconstruct the normal map, the BRDF has to be known, leading to a circular dependency. In this work, this dependency is avoided by using intrinsic image decomposition to extract diffuse shading and recovering the normal map only from diffuse information. As demonstrated in Sec. 7.3, BRDF estimation works best if at least a single highlight is visible (i.e., a strong high-frequency light source is reflected from the surface to the camera). However, highlights degrade the performance of normal reconstruction, because highlights from colored light sources change the observed chromaticity. This introduces artifacts in diffuse shading generated by the intrinsic image decomposition algorithm (see Fig. 6.1(c)). Generally, intrinsic image decomposition does not work well on specular materials with high-frequency normals, as the decomposition algorithm can not distinguish between diffuse and specular shading if both consist only of high-frequency components. Nevertheless, even if the decomposition would be perfect, there are still materials for which normals cannot be reconstructed sufficiently well from diffuse information only due to their small diffuse contribution (e.g., mirror-like materials).

To sum up, intrinsic image decomposition and normal reconstruction produce normal maps that are corrupted for specular materials, especially at regions with highlights. However, these regions contain the most information about the BRDF. Therefore, incorrect normals probably degrade the performance of BRDF estimation significantly and it may be better to assume a flat surface instead of using the recovered normal map. In the following, the combination of BRDF and normal estimation is analyzed for surfaces with homogeneous BRDF. Afterwards, the implications for the different approaches to handle spatially varying reflectance are described.

9.1. Homogeneous Materials

A reliable reconstruction of normals and BRDFs is only possible if the light comes from a specific direction and creates highlights on the surface. For example, using *arealight-ll* of Fig. 6.5 and placing the camera in front of the plane would not allow to reconstruct the BRDF, because the light source does not create a highlight on the surface. In contrast, if environment *arealight* in Fig. 7.3 is used, it is not possible to reconstruct normals, because the light source is centered above the plane (cf. Fig. 6.6).

The environments used in the following satisfy both requirements (Fig. 9.1(a)). *Smoothlight* is a combination of previously used environments (Fig. 6.5 and Fig. 7.3). The former contains a smooth component to support normal reconstruction and a green area-light in the center for BRDF estimation, the latter is again the *pisa* environment. The normal maps are shown in Fig. 9.1(b). Normal map *test* is flat and contains a few edges. *Metal* (rusty metal) is flat too, but there are much more edges than in *test*. In contrast, normal map *concrete* generates a bumpy surface.

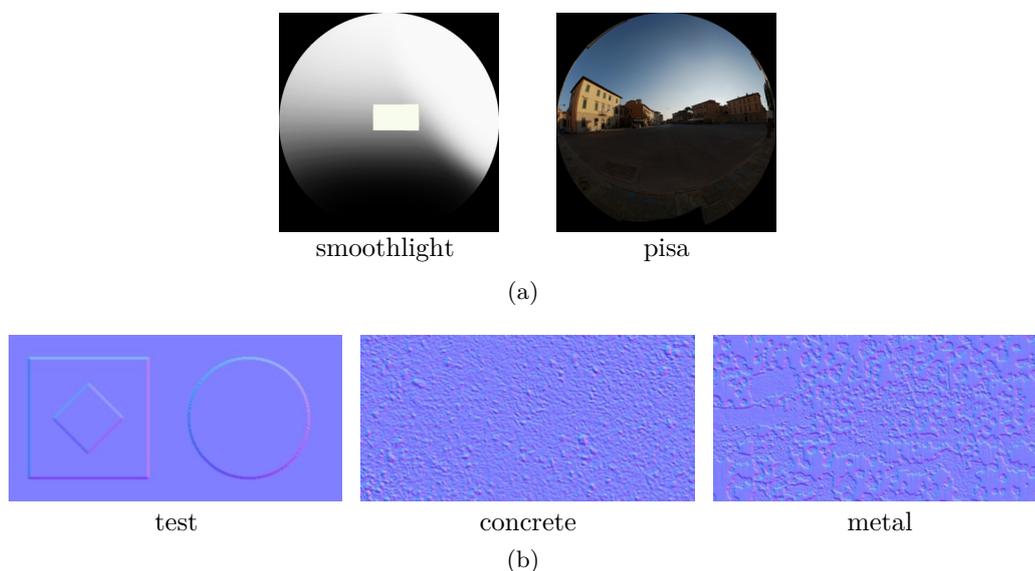


Figure 9.1.: Environments and normal maps used in reconstruction.

Each scene is rendered from two viewpoints, just like in Sec. 7.3. The reconstruction starts by extracting shading from rendered images as described in Sec. 6.1. The extracted shading is used to reconstruct normal maps (Sec. 6.2). Finally, the BRDF of the surface is estimated in two different ways:

1. Use reconstructed normal map in BRDF estimation.
2. Ignore reconstructed normal map in BRDF estimation, i.e., assume that the surface is flat.

If normal reconstruction was perfect, the best choice would be option 1. However, as mentioned above, normal reconstruction is problematic especially in specular regions. Normals that lead to small displacements of surface points generate blurry specular highlights modelled by the roughness parameter of the BRDF. The presented reconstruction approach can not handle such small displacements. Larger displacements become visible in the image, but due to specularity it is still difficult to reconstruct the corresponding normals. In option 2, these unreliable normals are ignored, which means to still represent them by the roughness parameter. As a result, the BRDF is smoothed and normals can be discarded, as they are contained in the roughness parameter. The surface looks as if it is seen from far away. In practice, when seen from smaller distances, the result sometimes looks better if normals are used to render the surface. In the following, option 1 and 2 are compared and in each case the surface is rendered with the reconstructed normal map.

Fig. 9.2 shows the error of the estimated BRDFs. Overall, it is similar to the errors measured for surfaces without normals (Fig. 7.6). Therefore, as expected, the error is small for diffuse BRDFs (*fruitwood-241* and *blue-rubber*). Likewise, the error for specular BRDFs is much greater. Plots of *alum-bronze* in Fig. 9.3 show that the reason is the shortening of the specular lobe, which is a result of the blurring described above. The effect is stronger for *metal*, because *metal* contains more edges than *test*. Surprisingly, the error is greater if the reconstructed normal map is used in the estimation.

The reason can be found in Fig. 9.4, where the reconstructed normal maps are shown. Especially in normal map *test*, flat parts of the surface become slightly deformed. Therefore, edges as well as flat regions are not reliable in the reconstructed normal map. In contrast, if the surface is assumed to be flat, only normals at edges are wrong. However, whether the normal map is used in the estimation or not, there is not much difference in the rendered images (Fig. 9.5).

Summing up, estimating BRDFs of specular surfaces with normals is difficult, as good normal maps are necessary to estimate the specular appearance. Furthermore, errors in normal maps are more visible for specular than for diffuse surfaces (Fig. 9.4(a) and Fig. 9.4(c)). Finally, for nearly flat surfaces it seems to be better to assume that the entire surface is flat when estimating the BRDF.

In order to improve the results of the BRDF estimation, an iterative approach to estimate normal map and BRDF one after another could be used as a solution for this problem. However, this would only improve normals in regions where highlights have been observed, and consequently there has to be some concept to propagate information from these normals to other regions of the image.

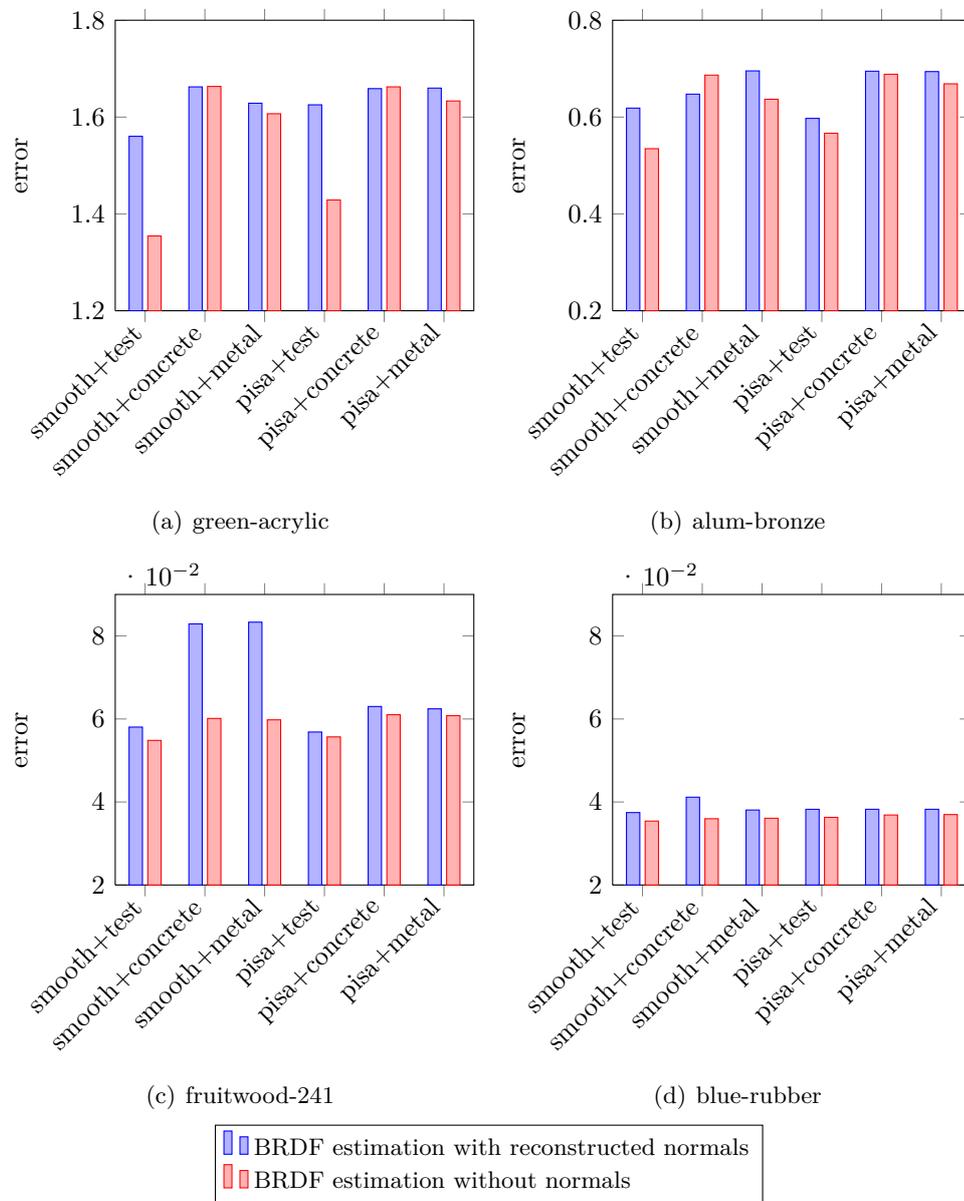


Figure 9.2.: Error of the estimated BRDF for different lighting environments and normal maps, computed as in Fig. 7.6(a) with $\theta_{\{i,o\}} > 80^\circ$. Results are similar if $\theta_{\{i,o\}} > 45^\circ$.

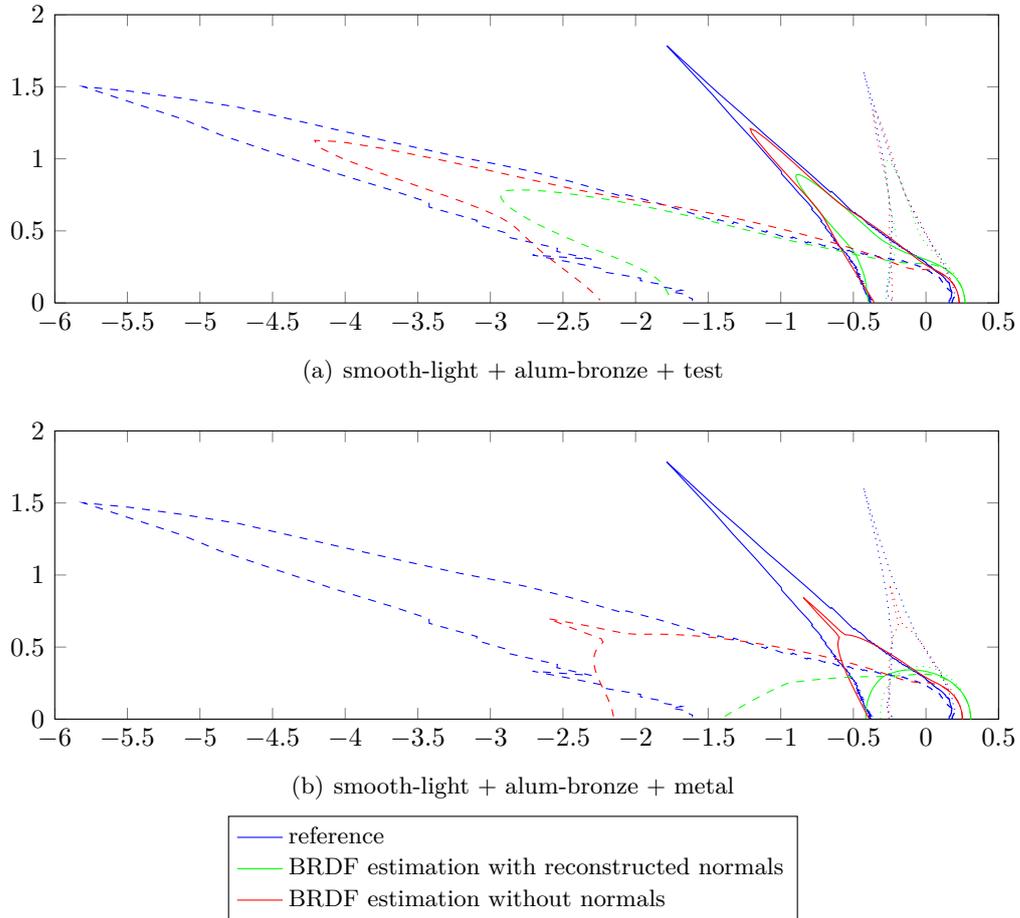


Figure 9.3.: Plots for BRDFs estimated with and without normal map, scaled by cubic root, $\theta_i = \{15^\circ, 45^\circ, 75^\circ\}$, $\phi_i = 0^\circ$, $\phi_o = 180^\circ$

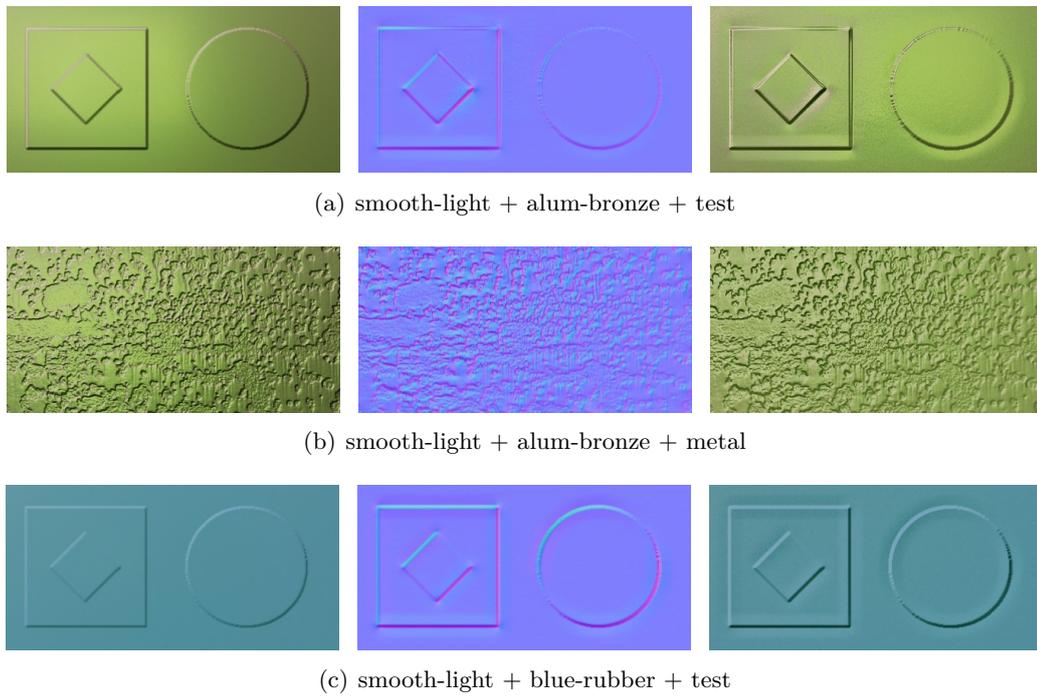


Figure 9.4.: Results for various scenes. From left to right: one of the two input images, reconstructed normal map, rendered surface with reconstructed normal map and estimated BRDF. The normal map was used in the estimation.

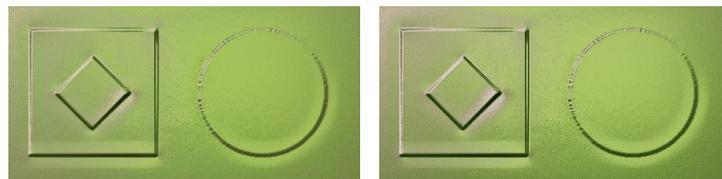


Figure 9.5.: Rendering results for BRDF estimation with reconstructed normals (left) and BRDF estimation without normals (right).

9.2. Spatially Varying Materials

In the previous section, it is shown that the estimated normal map is too inaccurate for a reliable reconstruction of the BRDF. This is a large problem in the clustering approach (Sec. 8.1), as it is very sensitive to outliers caused by normals or noise (refer to Fig. 8.4 for an example). Therefore, diffuse shading has to be removed before BRDF estimation without altering specular shading, otherwise the recovered specular lobe will be corrupted. Because a purely image based solution like in intrinsic image decomposition is difficult to achieve without further information about the specular BRDF component, this is not further investigated in this work.

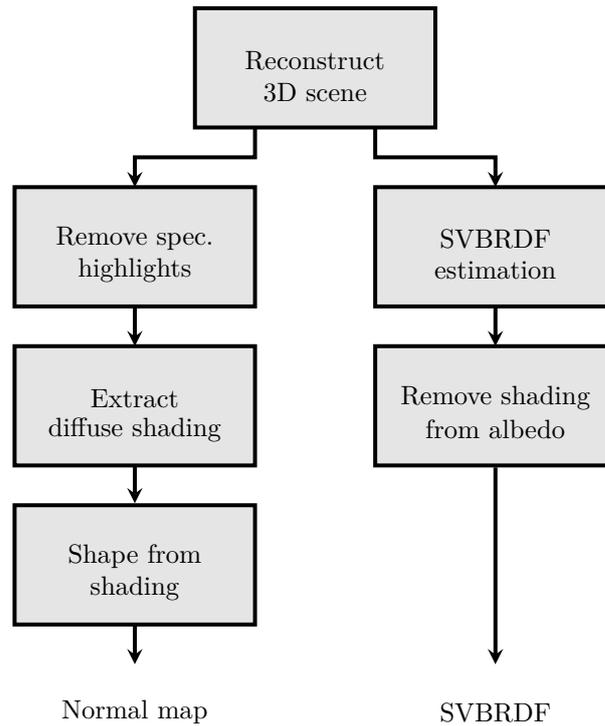


Figure 9.6.: Combining the separation approach for estimating spatially varying BRDFs with normal estimation. The BRDF is estimated by assuming a flat surface, i.e., the estimated normal map is not considered. Therefore, diffuse shading due to normals has to be removed from albedo.

In contrast, the integration of normal estimation into the separation approach described in Sec. 8.2 is easier, because it can deal with outliers from perturbed normals. Without normals, the approach separates the observed surface into albedo and coefficients for the specular lobe. If, however, the surface is perturbed by normals, the per-pixel diffuse shading from the normals is modulated onto albedo. The intrinsic image decomposition algorithm described in Sec. 6.1 is used to remove the modulated shading from albedo. Normal estimation works the same as in case of homogeneous

materials.

Fig. 9.6 shows the main idea of the method: normals and spatially varying BRDF are estimated independently. As before, normals are reconstructed by decomposing the image into its intrinsic images and applying the normal estimation algorithm to the extracted diffuse shading. Because the surface is assumed to be flat when estimating the BRDF, diffuse shading is not removed from albedo. Therefore, the intrinsic image decomposition algorithm is called once again to accomplish this task. Results for real scenes are given in Fig. 9.7 and Fig. 9.8. The spatially varying BRDF is estimated by the approach described in Sec. 8.2.2.



Figure 9.7.: Results of BRDF estimation and normal reconstruction. The approach described in Sec. 8.2.2 is used to estimate the spatially varying BRDF. From left to right and top to bottom: input images, relighting with *ennis* environment, relighting with point-light sources, environment where the input images were taken, estimated normal map.

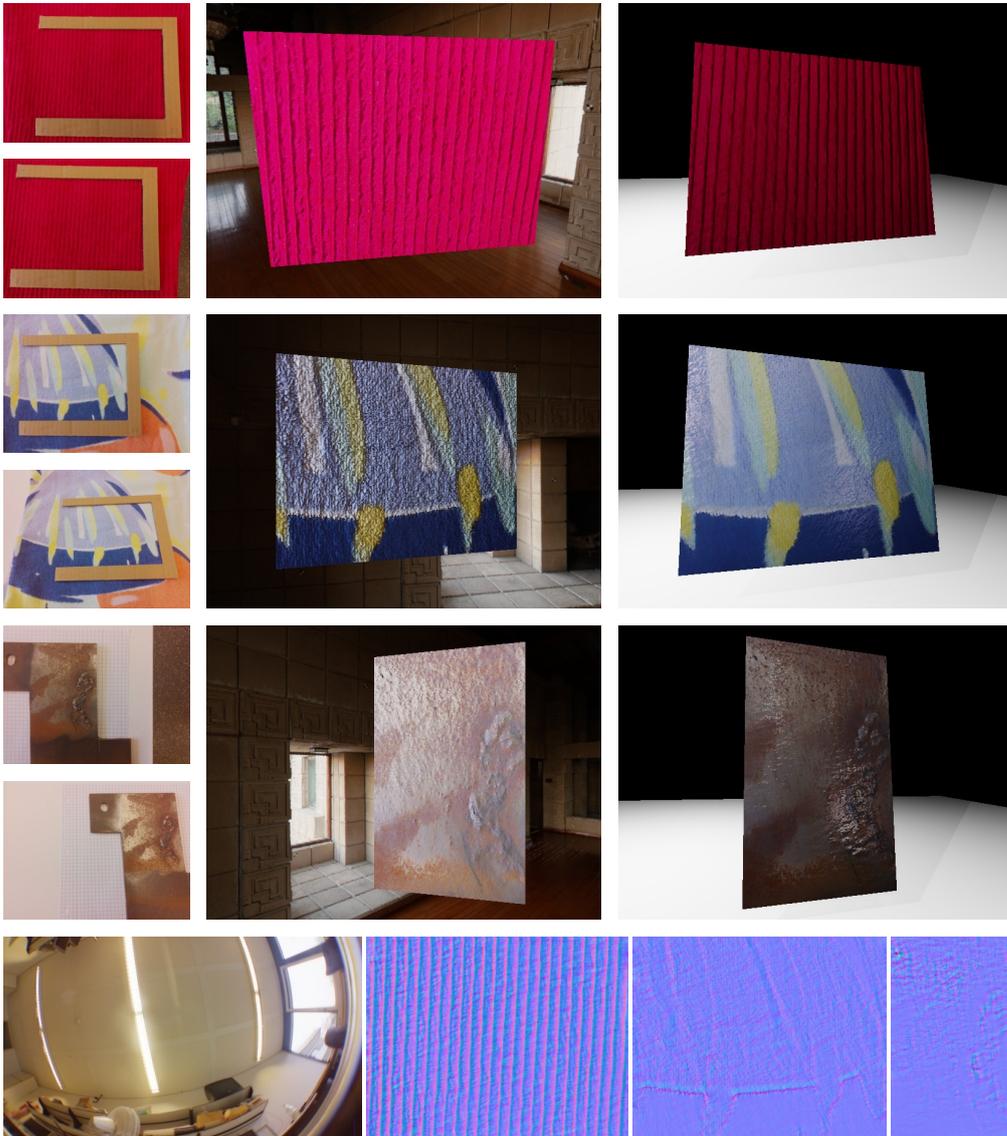


Figure 9.8.: Results of BRDF estimation and normal reconstruction. The approach described in Sec. 8.2.2 is used to estimate the spatially varying BRDF. From left to right: input images, relighting with *ennis* environment, relighting with point-light sources. Last row: environment where the input images were taken and estimated normal maps.

10. Conclusion

In this work, the normal map and the spatially varying bidirectional reflectance distribution function of almost planar surfaces were estimated from three images. Two images show the plane from different camera positions. In order to capture the incoming illumination within the third image, the camera is equipped with a fish-eye lens. The acquisition process is simple and focuses on the use of mobile devices. In this case, perfect calibration is not possible. Therefore, it was shown how to deal with imperfect calibration in order to get reasonable results.

The presented method to estimate normals uses diffuse shading information contained in the images. An algorithm was introduced that allows to separate images of textured surfaces into its intrinsic images. However, the problem is ill-posed and a correct decomposition is difficult to achieve. Thus, there are some surfaces for which the decomposition fails. Diffuse shading is used to reconstruct a normal map. The presented shape from shading approach reconstructs normals from almost planar surfaces. Although arbitrary illumination can be used, there are environments that work better than others. It is important that diffuse shading is clearly visible in the image, otherwise the algorithm will fail.

BRDF estimation is based on a linear BRDF model, which means that the BRDF is represented as a linear combination of basis materials. Fitting is performed via non-negative linear least-squares optimization by taking all surface points into account. It was shown that it is possible to generate a decent approximation of the real BRDF from two images if highlights are seen on the surface. More precisely, the light sources have to contain high frequencies and must be reflected from the surface into the camera. If this condition is not fulfilled, the shape of the specular lobe will not be estimated correctly. Furthermore, it was shown that including estimated normal maps in BRDF estimation does not always improve the quality of the result, because the estimated normal maps are too inaccurate for this.

Finally, the system was extended to handle spatially varying BRDFs. An approach that clusters the points based on residuals of randomly generated models was developed and evaluated. However, it is difficult to find a parameterization that generates robust results in a large variety of scenes. Furthermore, it is not possible to represent textured surfaces by a few clusters. Therefore, a method that separately estimates the diffuse and the specular BRDF component was introduced. In this case, a diffuse texture with per-pixel color information and a single specular lobe for the whole surface is extracted. However, it turned out that the estimated specular component is

not good enough to separate it from the diffuse texture in real images. This problem can be solved if one of the images is free of highlights, because it only contains diffuse contributions. Based on differences in the input images, the approach was extended to separate the surface into matte and glossy regions. In this case, besides per-pixel albedo, two specular lobes are extracted from the surface. By combining this technique with normal reconstruction, a system was created that is capable of estimating the normal map, the albedo, and two specular BRDF lobes for glossy and matte regions.

In future work, it would be interesting to incorporate information from the estimated BRDF into normal reconstruction. This should improve the quality of the normal map, which in turn can be used to estimate better BRDFs. By combining both steps into a single algorithm, it should be possible to optimize normals and BRDFs iteratively. Nevertheless, the quality of the estimation still depends on the illumination, which also means that there is no highlight in some scenes or highlights are only visible in small regions. Therefore, the normal map can only be refined in these regions. In order to become independent of the environment, one or more images with activated flash could be taken. In this case it is useful that the position of the flash in respect to the optical center of the camera is known.

A. Iterative Solutions for Normal Estimation

A.1. Initial Normal Map

The initial normal map is obtained by solving the following Euler-Lagrange equations (refer to Sec. 6.2 for details on notation):

$$\frac{\partial F}{\partial u} - \frac{d}{dx} \frac{\partial F}{\partial u_x} - \frac{d}{dy} \frac{\partial F}{\partial u_y} = 0 \quad (\text{A.1})$$

$$\frac{\partial F}{\partial v} - \frac{d}{dx} \frac{\partial F}{\partial v_x} - \frac{d}{dy} \frac{\partial F}{\partial v_y} = 0 \quad (\text{A.2})$$

$$\frac{\partial F}{\partial w} - \frac{d}{dx} \frac{\partial F}{\partial w_x} - \frac{d}{dy} \frac{\partial F}{\partial w_y} = 0. \quad (\text{A.3})$$

Forming the partial derivatives and plugging them in, the equations become:

$$(a_3u + a_2v + a_1w + a_0 - i)a_3 - \mu\Delta u = 0 \quad (\text{A.4})$$

$$(a_3u + a_2v + a_1w + a_0 - i)a_2 - \mu\Delta v = 0 \quad (\text{A.5})$$

$$(a_3u + a_2v + a_1w + a_0 - i)a_1 - \mu\Delta w = 0, \quad (\text{A.6})$$

where Δ is the Laplace operator ($\Delta f = f_{xx} + f_{yy}$). A discrete approximation of Δ based on a 2D binomial filter is given by the filter mask

$$\Delta = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 2 & -12 & 2 \\ 1 & 2 & 1 \end{bmatrix}. \quad (\text{A.7})$$

This mask can be separated into two parts, a mask that computes the local average B and the identity I :

$$\Delta = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 2 & -12 & 2 \\ 1 & 2 & 1 \end{bmatrix} = 3 \left(\underbrace{\frac{1}{12} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & 2 \\ 1 & 2 & 1 \end{bmatrix}}_B - \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_I \right). \quad (\text{A.8})$$

Appendix A. Iterative Solutions for Normal Estimation

Applied to u , $\Delta u = 3(\bar{u} - u)$, with $\bar{u} = B * u$ and $u = I * u$. This is similar to the approach by Horn and Schunck [HS81] who used it to determine the optical flow. With this approximation, Eqs. (A.4)-(A.6) become ($\mu' = 3\mu$):

$$(a_3u + a_2v + a_1w + a_0 - i)a_3 - \mu'(\bar{u} - u) = 0 \quad (\text{A.9})$$

$$(a_3u + a_2v + a_1w + a_0 - i)a_2 - \mu'(\bar{v} - v) = 0 \quad (\text{A.10})$$

$$(a_3u + a_2v + a_1w + a_0 - i)a_1 - \mu'(\bar{w} - w) = 0. \quad (\text{A.11})$$

In order to build an iterative algorithm to solve the system at each surface point, the difference of the normal (u, v, w) from the local average $(\bar{u}, \bar{v}, \bar{w})$ is estimated:

$$u - \bar{u} = -a_3\tilde{b} \quad (\text{A.12})$$

$$v - \bar{v} = -a_2\tilde{b} \quad (\text{A.13})$$

$$w - \bar{w} = -a_1\tilde{b} \quad (\text{A.14})$$

$$\tilde{b} = \frac{a_3\bar{u} + a_2\bar{v} + a_1\bar{w} - i + a_0}{\mu' + a_1^2 + a_2^2 + a_3^2}. \quad (\text{A.15})$$

Finally, the normals in step $k + 1$ are obtained by adding this difference iteratively to the local average of step k :

$$u^{k+1} = \bar{u}^k - a_3\tilde{b}^k \quad (\text{A.16})$$

$$v^{k+1} = \bar{v}^k - a_2\tilde{b}^k \quad (\text{A.17})$$

$$w^{k+1} = \bar{w}^k - a_1\tilde{b}^k. \quad (\text{A.18})$$

After each step, the vectors are normalized to ensure a length of 1.

A.2. Height Map

Using the relative height calculated directly from the normals

$$q(x, y) = \begin{bmatrix} \frac{u(x,y)}{w(x,y)} \\ \frac{v(x,y)}{w(x,y)} \end{bmatrix} = \begin{bmatrix} q_x(x, y) \\ q_y(x, y) \end{bmatrix}, \quad (\text{A.19})$$

the height map is reconstructed by minimizing the following Euler-Lagrange equation (refer to Sec. 6.2 for details on notation):

$$\frac{\partial F}{\partial h} - \frac{d}{dx} \frac{\partial F}{\partial h_x} - \frac{d}{dy} \frac{\partial F}{\partial h_y} = 0. \quad (\text{A.20})$$

Inserting the derivatives yields:

$$-h_{xx} - h_{yy} + \frac{dq_x}{dx} + \frac{dq_y}{dy} = 0. \quad (\text{A.21})$$

Because

$$\frac{dq_x}{dx} + \frac{dq_y}{dy} = \left(\frac{d}{dx} \right) \cdot \begin{pmatrix} q_x \\ q_y \end{pmatrix} = \nabla \cdot q, \quad (\text{A.22})$$

this can be reformulated as Poisson equation:

$$\Delta h = \nabla \cdot q. \quad (\text{A.23})$$

If the Laplace operator is simply approximated by

$$\Delta = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad (\text{A.24})$$

in discrete form the equation is given as

$$\begin{aligned} & h(x-1, y) + h(x+1, y) + h(x, y-1) + h(x, y+1) - 4h(x, y) \\ &= \frac{u(x, y)}{w(x, y)} + \frac{v(x, y)}{w(x, y)} - \frac{u(x-1, y)}{w(x-1, y)} - \frac{v(x, y-1)}{w(x, y-1)} \end{aligned} \quad (\text{A.25})$$

and rearranged for an iterative solution

$$\begin{aligned} h^{k+1}(x, y) = & \frac{1}{4} \left(h^k(x-1, y) + h^k(x+1, y) + h^k(x, y-1) + h^k(x, y+1) \right. \\ & \left. - \frac{u(x, y)}{w(x, y)} - \frac{v(x, y)}{w(x, y)} + \frac{u(x-1, y)}{w(x-1, y)} + \frac{v(x, y-1)}{w(x, y-1)} \right). \end{aligned} \quad (\text{A.26})$$

Bibliography

- [AS00] Michael Ashikhmin and Peter Shirley. An anisotropic phong BRDF model. *Journal of Graphics Tools*, 5(2):25–32, February 2000.
- [Bli77] James F. Blinn. Models of light reflection for computer synthesized pictures. *SIGGRAPH Computer Graphics*, 11(2):192–198, July 1977.
- [BN76] James F. Blinn and Martin E. Newell. Texture and reflection in computer generated images. *Communications of the ACM*, 19(10):542–547, October 1976.
- [Bra00] Gary Bradski. The OpenCV library. *Dr. Dobb's Journal of Software Tools*, 2000.
- [CT82] Robert L. Cook and Kenneth E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics*, 1(1):7–24, January 1982.
- [D'A] Pablo D'Angelo. Hugin - panorama photo stitcher. <http://hugin.sourceforge.net/>, April 22, 2012.
- [DTG⁺04] Paul Debevec, Chris Tchou, Andrew Gardner, Tim Hawkins, Charis Poullis, Jessi Stumpfel, Andrew Jones, Nathaniel Yun, Per Einarsson, Therese Lundgren, Marcos Fajardo, and Philippe Martinez. Estimating surface reflectance properties of a complex scene under captured natural illumination. Technical Report ICT-TR-06.2004, University of Southern California, 2004.
- [DTPG11] Yue Dong, Xin Tong, Fabio Pellacini, and Baining Guo. AppGen: interactive material modeling from a single image. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, pages 146:1–146:10, 2011.
- [FDB92] Brian V. Funt, Mark S. Drew, and Michael Brockington. Recovering shading from color images. In *Proceedings of the Second European Conference on Computer Vision*, pages 124–132, 1992.
- [GCHS10] Dan B. Goldman, Brian Curless, Aaron Hertzmann, and Steven M. Seitz. Shape and Spatially-Varying BRDFs from photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):1060–1071, June 2010.

Bibliography

- [GJAF09] R. Grosse, M. K Johnson, E. H Adelson, and W. T Freeman. Ground truth dataset and baseline evaluations for intrinsic image algorithms. In *IEEE International Conference on Computer Vision*, pages 2335–2342, October 2009.
- [HFB⁺09] Tom Haber, Christian Fuchs, Philippe Bekaer, Hans-Peter Seidel, Michael Goesele, and Hendrik P. A. Lensch. Relighting objects from image collections. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 627–634, June 2009.
- [Hor86] Berthold K. P. Horn. *Robot Vision*. The MIT Press, 1986.
- [HS81] Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.
- [ICG86] David S. Immel, Michael F. Cohen, and Donald P. Greenberg. A radiosity method for non-diffuse environments. *SIGGRAPH Computer Graphics*, 20(4):133–142, August 1986.
- [IH89] Katsushi Ikeuchi and Berthold K. P. Horn. Numerical shape from shading and occluding boundaries. In Berthold K. P. Horn, editor, *Shape from shading*, pages 245–299. MIT Press, Cambridge, MA, USA, 1989.
- [JA11] Micah K. Johnson and Edward H. Adelson. Shape estimation in natural illumination. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2553–2560, June 2011.
- [LFTG97] Eric P. F. Lafortune, Sing-Choong Foo, Kenneth E. Torrance, and Donald P. Greenberg. Non-linear approximation of reflectance functions. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 117–126, 1997.
- [LH74] Charles L. Lawson and Richard J. Hanson. *Solving least squares problems*. Prentice-Hall, Englewood Cliffs, NJ, USA, 1974.
- [LKG⁺03] Hendrik P. A. Lensch, Jan Kautz, Michael Goesele, Wolfgang Heidrich, and Hans-Peter Seidel. Image-based reconstruction of spatial appearance and geometric detail. *ACM Transactions on Graphics*, 22(2):234–257, April 2003.
- [Low99] David G. Lowe. Object recognition from local Scale-Invariant features. In *Proceedings of the 7th IEEE International Conference on Computer Vision*, pages 1150–1157, 1999.
- [MKMS07] Rafał Mantiuk, Grzegorz Krawczyk, Radosław Mantiuk, and Hans-Peter Seidel. High dynamic range imaging pipeline: Perception-motivated representation of visual content. In Bernice E. Rogowitz, Thrasyvou-

- los N. Pappas, and Scott J. Daly, editors, *Proc. of IS&T/SPIE's Human Vision and Electronic Imaging*, February 2007.
- [MPBM03a] Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan. A data-driven reflectance model. *ACM Transactions on Graphics*, 22(3):759–769, June 2003.
- [MPBM03b] Wojciech Matusik, Hanspeter Pfister, Matthew Brand, and Leonard McMillan. Efficient isotropic BRDF measurement. In *Proceedings of the 14th Eurographics workshop on Rendering*, pages 241–247, 2003.
- [MV07] Ezio Malis and Manuel Vargas. Deeper understanding of the homography decomposition for vision-based control. Technical Report RR-6303, INRIA, 2007.
- [MWL⁺99] Stephen R. Marschner, Stephen H. Westin, Eric P. Lafortune, Kenneth E. Torrance, and Donald P. Greenberg. Image-Based BRDF measurement including human skin. In *Proceedings of the 10th Eurographics Workshop on Rendering*, pages 139–152, June 1999.
- [MWLT00] Stephen R. Marschner, Stephen H. Westin, Eric P. F. Lafortune, and Kenneth E. Torrance. Image-Based bidirectional reflectance distribution function measurement. *Applied Optics*, 39:2592–2600, 2000.
- [NDM05] Addy Ngan, Frédo Durand, and Wojciech Matusik. Experimental analysis of BRDF models. In *Proceedings of the Eurographics Symposium on Rendering*, pages 117–226, 2005.
- [NRH⁺77] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis. Geometrical considerations and nomenclature for reflectance. *Final Report National Bureau of Standards, Washington, DC. Inst. for Basic Standards*, October 1977.
- [NRH04] Ren Ng, Ravi Ramamoorthi, and Pat Hanrahan. Triple product wavelet integrals for all-frequency relighting. *ACM Transactions on Graphics*, 23(3):477–487, August 2004.
- [PF05] Emmanuel Prados and Olivier Faugeras. Shape from shading: a well-posed problem? In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 870–877, June 2005.
- [PKTD09] Sylvain Paris, Pierre Kornprobst, JackTumblin Tumblin, and Frédo Durand. Bilateral filtering: Theory and applications. *Foundations and Trends in Computer Graphics and Vision*, 4(1):1–75, 2009.
- [RH01a] Ramamoorthi and P. Hanrahan. On the relationship between radiance and irradiance: determining the illumination from images of a convex lambertian object. *Journal of the Optical Society of America A*, 18(10):2448–2459, October 2001.

Bibliography

- [RH01b] Ravi Ramamoorthi and Pat Hanrahan. An efficient representation for irradiance environment maps. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 497–500, 2001.
- [RH01c] Ravi Ramamoorthi and Pat Hanrahan. A signal-processing framework for inverse rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 117–128, 2001.
- [RHP⁺10] Erik Reinhard, Wolfgang Heidrich, Sumanta Pattanaik, Paul Debevec, Greg Ward, and Karol Myszkowski. *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting*. Morgan Kaufmann, Burlington, MA, USA, 2nd edition, 2010.
- [RZ10] Fabiano Romeiro and Todd Zickler. Inferring reflectance under real-world illumination. Technical Report TR-10-10, Harvard School of Engineering and Applied Sciences, 2010.
- [Sch94] Christophe Schlick. An inexpensive BRDF model for physically-based rendering. *Computer Graphics Forum*, 13(3):233–246, August 1994.
- [SM09] Peter Shirley and Steve Marschner. *Fundamentals of Computer Graphics*. A K Peters, Natick, MA, USA, 3rd edition, 2009.
- [SMS06] Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart. A toolbox for easily calibrating omnidirectional cameras. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5695–5701, October 2006.
- [STL08] Li Shen, Ping Tan, and Stephen Lin. Intrinsic image decomposition with non-local texture cues. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, June 2008.
- [TFA05] Marshall F. Tappen, William T. Freeman, and Edward H. Adelson. Recovering intrinsic images from a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(9):1459–1472, September 2005.
- [TPP12] Alejandro Troccoli, Dawid Pajak, and Kari Pulli. FCam for multiple cameras. In *SPIE Electronic Imaging: Multimedia on Mobile Images*, January 2012.
- [VBW08] Oliver Vogel, Michael Breuß, and Joachim Weickert. Perspective shape from shading with Non-Lambertian reflectance. In *Proceedings of the 30th DAGM symposium on Pattern Recognition*, pages 517–526, 2008.
- [Vea98] Eric Veach. *Robust monte carlo methods for light transport simulation*. PhD thesis, Stanford University, Stanford, CA, USA, 1998.

- [War92] Gregory J. Ward. Measuring and modeling anisotropic reflection. *SIGGRAPH Computer Graphics*, 26(2):265–272, July 1992.
- [WLL⁺08] Tim Weyrich, Jason Lawrence, Hendrik Lensch, Szymon Rusinkiewicz, and Todd Zickler. Principles of appearance acquisition and representation. In *ACM SIGGRAPH 2008 classes*, page 80:1–80:119, 2008.
- [Woo80] Robert J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 19(1):139–144, January 1980.
- [WSB⁺98] D. Rod White, Peter Saunders, Stuart J. Bonsey, John van de Ven, and Hamish Edgar. Reflectometer for measuring the bidirectional reflectance of rough surfaces. *Applied Optics*, 37:3450–3454, 1998.
- [WSTS08] Tai-Pang Wu, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Interactive normal reconstruction from a single image. *ACM Transactions on Graphics*, 27(5):119:1–119:9, December 2008.
- [WWHL07] R. Peter Weistroffer, Kristen R. Walcott, Greg Humphreys, and Jason Lawrence. Efficient basis decomposition for scattered reflectance data. In *Proceedings of the Eurographics Symposium on Rendering*, June 2007.
- [YDMH99] Yizhou Yu, Paul Debevec, Jitendra Malik, and Tim Hawkins. Inverse global illumination: recovering reflectance models of real scenes from photographs. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 215–224, 1999.
- [ZK07] Wei Zhang and Jana Kosecká. Nonparametric estimation of multiple structures with outliers. In *Proceedings of the 2005/2006 international conference on Dynamical vision*, pages 60–74, 2007.
- [ZTCS99] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape from shading: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):690–706, August 1999.