DOCTORAATSPROEFSCHRIFT

2010 | School voor Informatietechnologie
Kennistechnologie, Informatica, Wiskunde, ICT

Mesostructure Acquisition With Planar Illuminants

Proefschrift voorgelegd tot het behalen van de graad van
Doctor in de Wetenschappen: Informatica, te verdedigen door:

Yannick FRANCKEN

Promotor: prof. dr. Philippe Bekaert
Copromotor: prof. dr. Frank Van Reeth

Maastricht University

universiteit hasselt

D/2010/2451/6

Acknowledgments

Finishing a Ph.D dissertation is not something you achieve on your own. Therefore I would like to thank everyone who directly or indirectly contributed to this work. Thank you all!

In particular, I want to thank my supervisor Prof. dr. Philippe Bekaert, for providing me with an excellent research environment and giving me the research freedom I desired. Getting the opportunity to really do what I like to do, is very enjoyable.

I am also grateful to the members of my Ph.D. committee and jury who reviewed this dissertation and provided me with valuable comments: Prof. dr. Philippe Bekaert, Dr. Marco Tarini, Prof. dr. ir. Philip Dutré, Prof. dr. Frank Van Reeth, Prof. dr. Karin Coninx, Prof. dr. Marc Gyssens.

As a Ph.D student you travel the world, presenting and discussing new research ideas. This was a great opportunity to meet interesting researchers, visit foreign countries for the first time, and last but not least, try the local cuisine. Thank you Roger, for taking care of the financial aspects. Thank you Ingrid, for always getting everything booked, even in overbooked city centers a few days before departure!

Before visiting conferences, of course a lot of hard work was involved to get the research done and get the paper accepted. Since the success of this labour often depended on the help of many colleagues from our graphics team, I want to thank all the members and former members (alphabetically listed): Codruta Ancuti, Cosmin Ancuti, Koen Beets, Tom Cuypers, Bert De Decker, Fabian Di Fiore, Maarten Dumont, Jan Fransens, Karel Frederix, Mark Gerrits, Patrik Goorts, Tom Haber, Chris Hermans, Erik Hubo, Steven Maesen, Johan Nulens, Tom Mertens, Sammy Rogmans, Johannes Taelman, Tom Van Laerhoven, Cedric Vanaken and William Van Haevre.

With some of them, I have cooperated quite intensively and repeatedly. Many ideas have been created, discussed and practically realized together with Tom Cuypers. Tom Mertens has shared his research and presentation skills by teaching me how to work out and present ideas by means of papers as well as presentations. Chris Hermans, Cedric Vanaken, Mark Gerrits and Johan Huysmans have taken the effort for getting through several exhausting proofreading sessions. Tom Haber, Bert de Decker, Johannes Taelman and Luc Claessen always provided me with instant hardware support.

I have experienced good times working on my Ph.D at the EDM. Walking down the corridors getting another coffee often resulted in interesting discussions with whoever I met, talking about whatever we thought of at that moment. For instance, I remember conversations with Peter Vandoren ranging from the misleading reduced weight of candy bars to explaining our latest research developments. I am very thankful to the management of the EDM for providing such a pleasant place for doing a Ph.D.

Last but not least I want to thank my girlfriend Tanja Van den Eede, my family and friends for keeping me always motivated in periods when I was stuck with my research.

Diepenbeek, February 2010.

Abstract

In this dissertation we propose several techniques for acquiring local surface orientation and reflectance properties of small-scale surface details, or the surface *mesostructure*. Our primary focus is on increasing the applicability by providing an *efficient*, *easy to implement and execute* approach, employing solely *off-the-shelf* hardware components.

Basically, our setup consists of a regular digital still camera and a computer screen functioning as a planar illuminant. Light patterns are displayed on the screen, illuminating the surface to be scanned, and reflections are captured by the camera. The recorded images are then processed, yielding a digital representation of the relief and reflectance of the scanned surface.

In order to correctly process the acquired input images, geometric information of the position and orientation of the screen with respect to the camera has to be available. Therefore, we propose two different calibration methods employing a spherical mirror in order to make the screen visible to the camera. The first approach is efficient in terms of the number of calibration images, whereas the second method is more accurate and more efficient in terms of the number of manual sphere displacements.

When the setup is calibrated, the material placed in front of the screen-camera setup can be illuminated by specific light patterns. We propose the use of Gray code patterns to efficiently scan the surface orientation as well as the reflectance of specular materials. Gradient patterns are used to obtain relief information of both specular and diffuse surfaces. We point out that Gray code patterns perform especially well for highly specular materials, whereas gradient patterns are more suited for glossy and diffuse surfaces.

In practice, materials are often a combination of both a specular and a diffuse reflection component. We propose the use of an LCD screen to handle such cases, since specular and diffuse separation can robustly be achieved by cross polarization of the linearly polarized light emitted by the LCD. Only a simple polarizing filter has to be placed in front of the camera, blocking and unblocking specular reflections by rotating the filter a 90 degrees. The separated components can then be processed by the appropriate algorithm.

As both the geometric relief and the reflectance properties strongly influence a surface's appearance, we propose a method to also analyze the gloss level of the specular reflection component in addition to the geometric relief. This is achieved by a straightforward extension of the Gray code based normal acquisition method.

The presented results show that a simple and inexpensive computer screen and digital camera can be transformed into a mesostructure acquisition system, yielding high quality scans. We believe that our system has possible applications in several areas, such as computer games, computer aided design and industrial inspection.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

1.1 Problem Statement

During the last few decades, computers have become increasingly important for performing a wide variety of tasks. One of these tasks consists of generating images of virtual scenes. Nowadays, convincing rendering techniques are employed in many applications, such as computer games. Even the production of photorealistic images has become common practice, for example during movie post-production. Fast and/or accurate rendering techniques have been developed for this purpose, approximating or accurately simulating light transport within the virtual world.

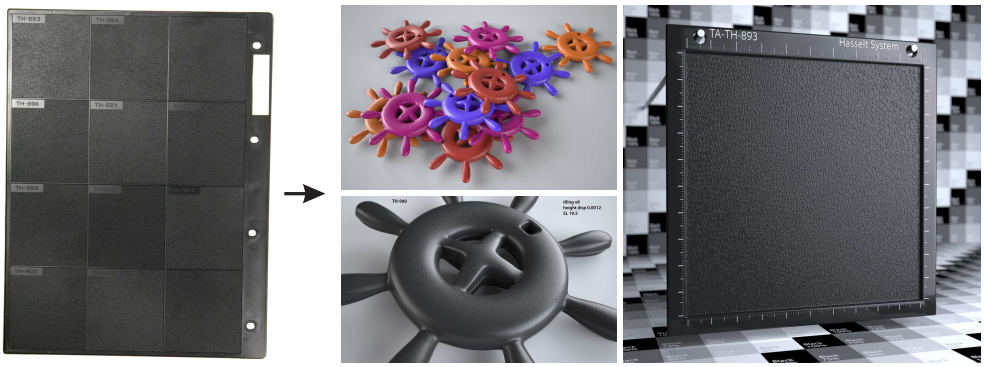
However, even if light interaction is simulated in a physically correct manner, scene data still has to be provided in the form of a 3D model. If the input scene data does not contain small-scale surface details such as scratches, imperfections, etc, the scene will be perceived as unrealistic. Manually modeling the world at such a level of detail can be a tedious task, suggesting the use of automatic 3D scanning methods.

Throughout the years, many techniques have been proposed to digitize the world around us. These techniques typically capture either (a) the light in the scene, (b) the geometry, or (c) the reflectance properties, or any combination of these. In this dissertation we will focus only on capturing the geometry and reflectance properties.

Although numerous geometry acquisition techniques exist, typically they are only able to acquire *global* shape information rather than *local* small-scale relief properties (figure 1.1). In practice, local surface deviations with respect to the global surface are often still manually modeled by designers, instead of automatically measured using real-world sample surfaces. The same holds for reflectance properties, which are often manually created using graphics shaders.

Even though several techniques exist for scanning local surface details as well as reflectance properties, users tend to stick to their manual approach. One of the reasons for this is the complexity of the currently available methods. Many approaches require special purpose setups, containing exotic hardware components, time consuming calibration procedures, difficult implementations and scanning procedures etc.

In this dissertation, the goal is to make small-scale appearance acquisition available to the public, bridging the gap between current research and practical usage.



(a) Real picture (b) Virtual rendering

Figure 1.1: A digitized version of a detailed real-world plastic sample sheet (a) is used for image synthesis (b).

This is achieved by presenting an *efficient*, *easy to implement and to execute* approach, employing solely *off-the-shelf hardware components* consisting of a regular still camera and a computer screen that functions as a planar illuminant.

1.2 Contributions

In this dissertation we present a number of contributions regarding relief and glossiness acquisition using a digital still camera and a planar light source. Part of this work has been presented [Francken 07a, Francken 07c, Francken 08a, Francken 08d, Francken 08c, Francken 08b, Francken 09c, Francken 09a, Francken 09b] at various conferences.

The main contributions of this work are:

Setup calibration: We present two different approaches to geometrically calibrate the screen with respect to the camera using a spherical mirror.

Relief acquisition: A relief acquisition method for specular, glossy and diffuse surfaces is presented and evaluated. We propose the use of a computer screen emitting Gray code and gradient patterns, illuminating a mesostructure surface, in order to efficiently reconstruct fine-scale surface relief.

Glossiness acquisition: Per pixel reflectance properties are measured by extending the Gray code based relief acquisition method.

1.3 Applications

The acquisition of small-scale surface and/or reflectance properties has applications within many different areas.

Entertainment: 3D computer games have become omnipresent during the last decade. Due to the ever increasing hardware support for real-time realistic rendering, techniques such as bump/normal and displacement mapping have become common in 3D games engines, by the use of shaders. The work presented in this dissertation aims to provide these shaders with the required data. Another large entertainment business making use of captured 3D models, is the movie industry. In order to increase the perceived realism of movie scenes by adding surface irregularities, the presented techniques can be employed to capture this level of detail.

Cultural heritage: Digital models of historical artefacts provide a way to preserve them without the risk of degradation. Detailed scans are desirable, as details will dissapear fast, compared to their coarse structure. The proposed methods can be used to scan such surface details.

Computer aided design / modeling: Often, in the early stages of production processes, prototypes are created to evaluate the look or other features of the product thus far. For instance, during the design phase of the shape and fine-scale relief of a car's dashboard, prototypes are created. In order to avoid long and expensive processes to create individual real-world prototypes, digital prototyping has become popular. Several of the techniques proposed in this thesis are useful to create digital libraries of fine-scale surface patterns, such as dashboard plastics, to be used for computer aided design / modeling.

Quality control for manufacturing industry: Manufacturers often want to perform automated quality control of produced objects, e.g. checking a painted car for reflection deviations due to orange skin artefacts, verifying the surface roughness of milled metal objects, or controlling the sharpness of diamond

edges. Such quality control examples can be accomplished using the investigated approaches within this work.

Medical applications: 3D scanners are employed in this field in order to analyse the functioning of particular body parts, to plan surgeries, or create appropriate prostheses. Techniques proposed in this dissertation can efficiently create detailed scans of skin, nails and eyes.

Augmented and mixed reality: Augmenting the world by adding/mixing virtual data with real data has recently become very popular. To provide a seamless "mix", realistic small-scale surface data has to be integrated.

1.4 Overview of the Dissertation

In this section a chapter by chapter content outline is given.

Chapter 1: Introduction This introductory chapter.

Chapter 2: Background Relevant background information about the acquisition and representation of small-scale surface details is covered here.

Part I: Setup This part will focus on building and calibrating the proposed setup.

Chapter 3: Hardware In this chapter, the involved hardware components are discussed. What are the advantages and drawbacks of using a screen as planar illuminant and digital still camera as light sensor?

Chapter 4: Calibration This chapter demonstrates different methods to calibrate a screen-camera setup. It describes how the setup parameters are defined, which mirror type that is suggested, and what the expected precision of the proposed methods will be.

Part II: Relief and Glossiness Acquisition The second part will focus on the actual surface acquisition approaches using the hardware proposed in part I.

Chapter 5: Relief Acquisition This chapter provides different methods to acquire local surface orientation, employing specific light patterns displayed on the screen, captured by the camera, and processed to a normal map by the computer.

Chapter 6: Glossiness Acquisition An extension of the previously presented normal acquisition is described here, which measures additional local reflectance information.

Chapter 7: Conclusions The conclusions chapter contains a short summary of the dissertation, and some interesting future directions will be pointed out.

A: Scientific Contributions and Publications: In appendix A, a list of contributions and publications is given.

B: Samenvatting (Dutch Summary) Appendix B contains a Dutch summary of this thesis.

Chapter 2

Background

Contents

In this chapter, we will give an overview of small-scale surface representations as well as acquisition.

In order to avoid that smooth global surfaces are subdivided into very small primitives to obtain a high quality rendering, local surface detail is traditionally *mapped* onto the global surfaces. This mapping establishes a 1-to-1 relationship between 2D texture elements (texels) and the coarse 3D surface. We will discuss approaches that model the *reflectance* (section 2.1) and the geometric *relief* (section 2.2) of the surface texture. The acquisition is described in section 2.3.

2.1 Reflectance

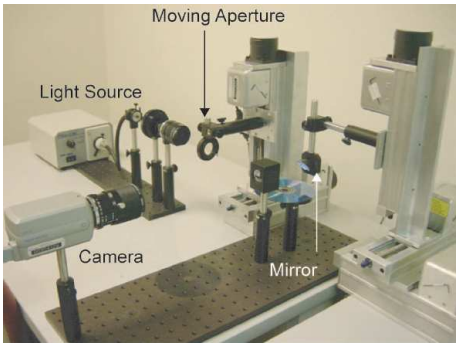
The general way to model the reflectance, or the microstructure, of a surface is by describing the relation between the incident and exitant light for all possible light direction combinations, for all texels. This six-dimensional function is referred to as the Bidirectional Texture Function [Dana 99], which can be seen as a spatially varying BRDF with integrated light conditions:

$$BTF_{r,g,b}(\theta_i, \phi_i, \theta_e, \phi_e, u, v) \tag{2.1}$$

The incoming light (θ_i, ϕ_i), exitant light (θ_e, ϕ_e), the spatial texture location (u, v) and the wavelength (r, g, b) in general all influence the amount of light transport. However, in practice many techniques reduce the number of parameters for rendering efficiency, reducing the acquisition time, avoiding material dependent redundancy, or preferred simplicity. In the remaining part of this section we will discuss some popular mesostructure reflectance formats.

2.1.1 Bidirectional Texture Function

The bidirectional texture function is a very general, but high dimensional format. Although many compression schemes have been introduced to reduce the amount of data, the acquisition still requires a special hardware setup including moving parts as each texel has to be illuminated and observed under all possible directions. In figure 2.1(a) a proposed setup of Wang and Dana [Wang 06] is depicted. BTFs can also be synthesized instead of captured, as described by Tong et al. [Tong 02].



(a) Bidirectional Texture Function setup of Wang and Dana [Wang 06]

(b) Polynomial Texture Map setup of Malzbender et al. [Malzbender 01].

Figure 2.1: Mesostructure reflectance acquisition setups.

2.1.2 Polynomial Texture Maps

A Polynomial Texture Map (PTM) is a 4D version of the BTF, where the exitant directions are fixed:

$$PTM_{r,g,b}(\theta_i, \phi_i, u, v) \tag{2.2}$$

Hence, no view dependent light changes like specular highlights can be captured. Other effects such as self-shadowing and interreflections are nicely preserved. The PTM is stored as a number of per texel coefficients of a biquadratic polynomial.

An example of acquisition setup is given in figure 2.1(b). Due to the fixed viewing direction and the use of multiple light sources, no moving parts are involved.

2.1.3 Regular Texture Maps

In practice, mesostructure reflectance is often approximated using more simplified models. Nowadays, most computer graphics applications make use of only a limited number of texture maps, containing information such as diffuse color, specular color, glossiness and local surface deviations. These texture maps function as the parameters of the reflection model (e.g. Phong [Blinn 77]). Since current graphics hardware is designed to work in this fashion, very efficient rendering can easily be achieved. Although no specific texture formats are strictly required due to the flexibil-

ity of current shading languages, we will discuss some standard texture map formats traditionally used in graphics applications.

If surface materials are modeled as a combination of a diffuse and specular component, the necessary parameters can be stored in a *diffuse*, *specular* and *gloss* map. The diffuse map contains the material's diffuse albedo required for Lambertian reflections, the specular map defines the brightness of specular reflections, and the gloss map contains the shininess or glossiness of the material. Intuitively, the intensity of a specularity is defined by the specular map, where its size is defined in the gloss map. As stated before, many other approaches are possible. Examples include models for specific materials such as human skin [Tsumura 03] or wood [Marschner 05].

Traditionally, in order to obtain the required texture maps, a manual approach is applied, sometimes based on photographs under uniform white illumination or procedurally generated images [Perlin 85].

2.2 Relief

The appearance of an image surface point does not only depend on the material's reflectance properties, but also on the local surface orientation. Therefore it is beneficial to store local surface perturbations with respect to the global surface as well. Several commonly used representations exist, which we will categorize into two separate groups, the *height* based and *orientation* based relief representations.

2.2.1 Height Based Relief

Height based relief maps are functions $h(u,v)$ assigning a height value to each texel (u,v). They are used for rendering techniques such as bump mapping, displacement mapping and parallax mapping. An example is given in figure 2.2, where the height $h(u,v)$ of an individual bump is plotted for illustrating the difference between the different mesostructure representations.

2.2.2 Orientation Based Relief

This type of relief stores local surface *orientation*. The following concrete orientation based representations are often used:

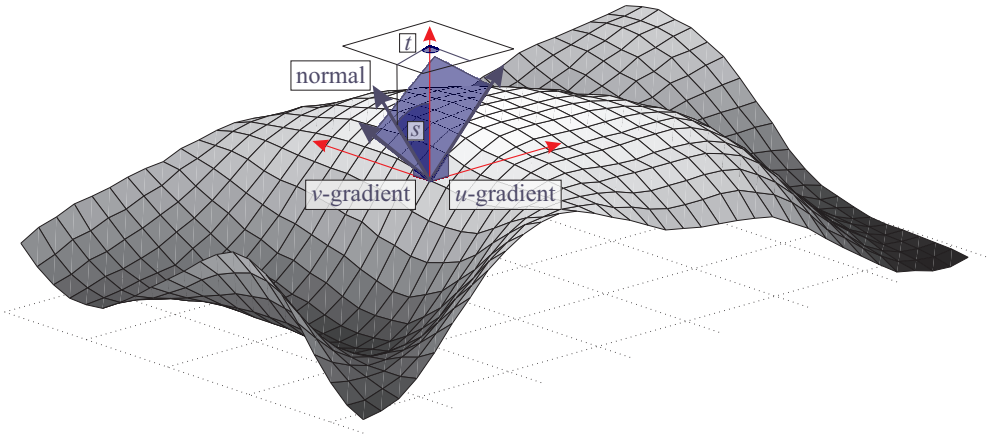


Figure 2.2: Zoomed view on a single surface bump. Local surface orientation for a texel is indicated by the surface normal, $u-$ and v-gradient and the slant s and tilt t.

Surface normals: for each texel a surface normal $n(u,v) = (n_x, n_y, n_z)$ is stored. It allows for straightforward light reflection calculations.

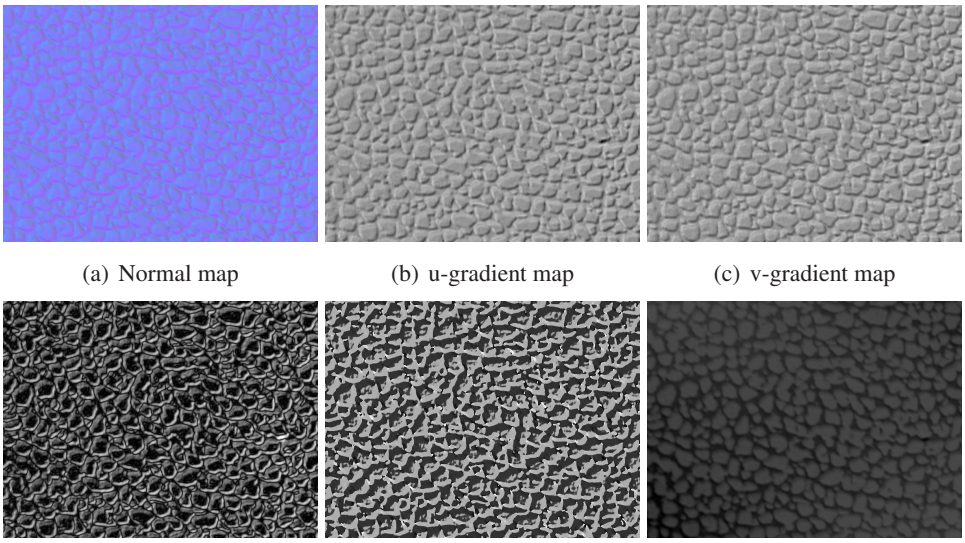
Gradients: the discrete partial derivatives $\frac{\partial h}{\partial u}$ and $\frac{\partial h}{\partial v}$ are stored for each texel. This representation is often used to convert the local orientation into a local height value.

Slant-tilt: two angles s (slant) and t (tilt) define the orientation. This representation is also mainly used for converting orientations into height values, or to define surface orientation with respect to a viewer (red vertical vector in figure 2.2).

The main disadvantage of orientation based approaches is that light occlusions / visibility cannot directly be handled.

2.2.3 Converting Height from/to Orientation Based Relief

The preferred relief format depends on the application. However, the acquired relief format tends to strongly depend on the underlying acquisition technique used. Hence a conversion between height and orientation based relief is needed. An example of a scanned dashboard plastic stored as height and orientation based relief maps is given in figure 2.3.



(a) Normal map (b) u-gradient map (c) v-gradient map

(d) Slant map (e) Tilt map (f) Height map

Figure 2.3: Relief maps.

Converting height values into orientation values is a straightforward procedure. By calculating the discrete partial derivatives $\frac{\partial h}{\partial u}$ and $\frac{\partial h}{\partial v}$ of the height function $h(u,v)$ local surface orientation is obtained. These $u-$ and $v-$gradients can afterwards be transformed into other formats, such as slant-tilt or normal values.

The transformation of orientation values into height values is more complicated. The idea is that a surface has to be found, satisfying the condition that at each coordinate (u,v) the derived orientation of the found height map has to equal the given orientation. In practice this is a hard problem as there does not always exist a perfectly suitable surface due to discrete sampling, acquisition errors, and quantization errors. These errors can yield to the non-integrability of the surface normals, and thus approximations have to be made. Most practical methods make assumptions, such as smoothness, and fit a surface by minimizing a predefined error function on the reconstructed orientation deviations. Depending on the used technique, different height maps might be obtained [Agrawal 05]. There exist a number of proposed techniques, which fall outside the scope of this dissertation [Frankot 88, Nehab 05, Agrawal 05, Kovesi 05].

2.3 Acquisition

In this section, the concept of vision based mesostructure acquisition will be introduced. Concrete approaches will be discussed later in the thesis text.

The shared idea among vision based mesostructure acquisition techniques is that a surface is illuminated by one or more light sources while a fixed camera captures one or more images of the subject. The captured light reflections depend on how the surface reacts on the incident light. More specifically, it depends on the *reflectance* and *relief* of the surface. Hence, given one or more images under known illumination, and making some assumptions on the material properties, reflectance or relief can be determined. For example, assuming point light sources at a large distance from the object and a Lambertian reflectance model, Horn [Horn 75] and Woodham [Woodham 80] were able to obtain local surface orientation taking a shape-from-shading or photometric stereo approach.

As will be discussed later, many extensions and alternative approaches have been introduced to allow for scanning different types of materials. However, since relief details are only small-scale surface deviations, most techniques acquire local surface orientation instead of directly measuring height maps. As current graphics applications mostly require height maps to be used for displacement mapping executed on graphics hardware, orientation based formats often have to be converted into height maps.

Part I

Setup

Chapter 3

Hardware

Contents

In this chapter, we will present the employed hardware setup for relief and glossiness acquisition. The principal hardware components involved are a digital still camera functioning as a light sensor and a regular computer screen functioning as a planar illuminant. As this setup will be used for optical measurements, we will concentrate on their radiometric properties.

3.1 Digital Still Camera as Light Sensor

During the last decade, digital still cameras have become ubiquitous. Inexpensive off-the-shelf single-lens reflex (SLR) cameras are already able to capture high resolution images retaining a sufficient quality to be used as physical optical measurement devices. However, a mathematical camera model (see chapter 4) and a relation between the scene radiance and the corresponding pixel intensity have to be established.

The function relating radiance to brightness is called the *camera response function*. Often this function is nonlinear, while most photometric methods require a linear response. In many imaging devices, this nonlinearity is intentional to mimic the response of film, the human visual system, or even to create aesthetic effects. Classical deviations result from white balancing, gamma correction and automatic gain control. Fortunately, typical sensors, such as CCD or CMOS, respond linearly, and cameras often allow the user to read this 12- or 16-bit raw sensor data. If this is not the case, the nonlinear response function is usually measured and compensated for. However, depending on the shape of the response curve, some of the $n-$bit color resolution is lost in the process.

An overview of the image formation pipeline is given in figure 3.1. Notice that two different types of transformations can be distinguished: effects due to camera *optics* or *electronics*. We discuss them separately.

3.1.1 Optics

The effect of lens optics is well-established research area [Horn 86, Mitsunaga 99, Wilburn 08]. The function relating image scene radiance $L(t)$ and irradiance $E(t)$ is given by:

$$E(t) = L(t)\frac{\pi}{4}\left(\frac{d}{h}\right)^2 \cos^4(\phi) \tag{3.1}$$

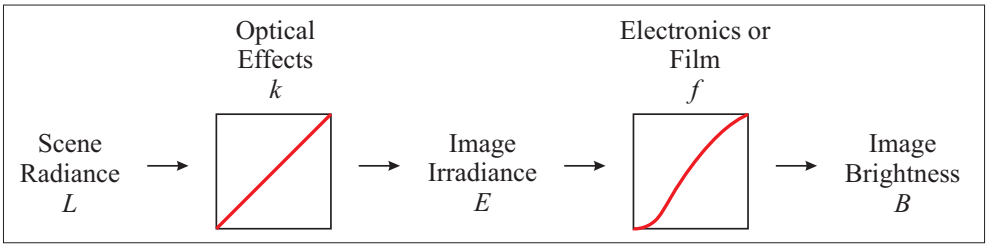


Figure 3.1: Image acquisition pipeline.

where d is the aperture diameter, h the focal length, and ϕ is the angle subtended by the principal ray from the imaging axis. When these constant parameters are grouped into a single constant $k = \frac{\pi}{4}\left(\frac{d}{h}\right)\cos^4(\phi)$, this relation is clearly linear:

$$E(t) = kL(t) \tag{3.2}$$

As image sensors are integrating devices, the total integrated irradiance for an exposure from time t_0 to time t_1 is given by:

$$I = \int_{t_0}^{t_1} kL(t)\,dt \tag{3.3}$$

3.1.2 Electronics

The nonlinear transform performed by the electronics is modeled as a single function f. There exist many different approaches to measure and model the nonlinear radiometric response functions [Healey 94, Debevec 97, Mitsunaga 99, Tsin 01, Grossberg 04, Wilburn 08]. The shared idea among most techniques is the recording of a number of images under different exposure times.

For the results presented in this dissertation, the previous approaches are straightforwardly applied to assert that the response curve of our camera is linear. A computer screen displaying a completely white image is recorded under different exposure times and, except from a limited amount of sensor noise, a linear response was measured.

	CRT	LCD	Plasma
Viewing angle	A	C	A
Brightness	C	A	A-C*
Low black level	A	C	B
Polarization	n/a	A	n/a
Contrast ratio	A	C	A-C*
High resolution	B	A	B
Manoeuvrability	C	A	A

Table 3.1: Comparison of CRT, LCD and plasma display technologies. A stands for excellent, B for good and C for fair performance. A-C* means the performance varies because of the Average Picture Level.

3.2 Screen as Planar Illuminant

The continuously increasing quality and decreasing prices of display technology make the use of easily and precisely controllable planar illuminants widely available in the form of regular computer screens. Even though this is a rectangular planar light source, only approximating a hemispherical one, the benefits outweigh the disadvantages.

Alternatively, a projector illuminating the front (or back) of a diffuser, can replace a computer screen [Schechner 03]. This type of setup allows for creating large and even curved screens, at a reasonably low cost. However, we have opted to employ regular displays as, for several parts in the dissertation, linearly polarized light has to be emitted. This cannot be straightfowardly achieved with a single projector and a regular diffuser.

Nowadays there are three common types of display technologies: CRT, LCD and plasma. An overview of their relevant properties is summarized in table 3.1. Notice that LCD screens emit linearly polarized light by default because of the underlying technology, which will be exploited in our applications.

When no polarized light is required, a CRT screen will be employed. The plasma display technology does not offer any exclusive advantages and the luminance of individual pixels cannot be straightforwardly controlled as they depend on the Average Picture Level (APL).

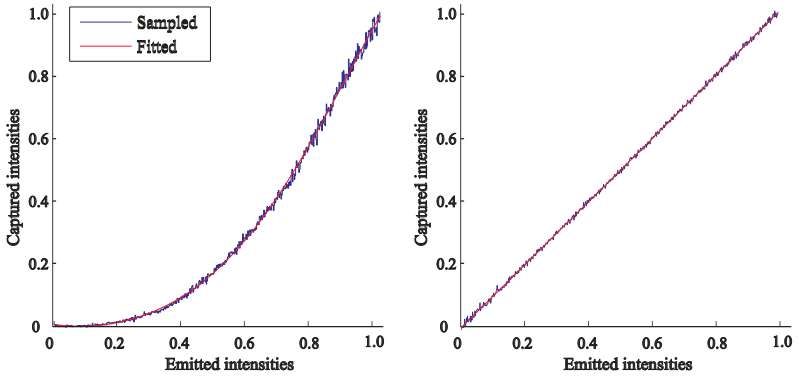


Figure 3.2: Captured versus emitted intensities measured with Canon EOS 400D camera. Left: a nonlinear response by directly capturing displayed linearly increasing grey intensities. Right: temporal black/white integration approach. Except from a small amount of sensor noise, the response is clearly linear.

The quality of displayed images on a computer screen strongly depends on a number of factors. The relevant factors will now briefly be discussed.

3.2.1 Intensity/Color Response

Screen settings, such as contrast, brightness and gamma, have a major influence on the displayed image intensity (figure 3.2 (left)). As a result, the emitted screen pixel radiance $L(t)$ at time t cannot straightforwardly be controlled. This suggests the necessity for a radiometric calibration step [Lamond 09]. Instead, we propose an alternative approach avoiding any response dependent calibration, using concepts borrowed from *camera response curve* measurement [Wilburn 08].

If a sensor with a linear response is exposed to the radiance $L(t)$ of a pixel during a period t_0 to t_1, the integrated irradiance I is:

$$I = \int_{t_0}^{t_1} kL(t)\,dt \tag{3.4}$$

Although $L(t)$ is not precisely known in our case, it is possible to minimize or maximize its output by displaying either a black or white pixel color, respectively L_{black} or L_{white}. This yields the following equations:

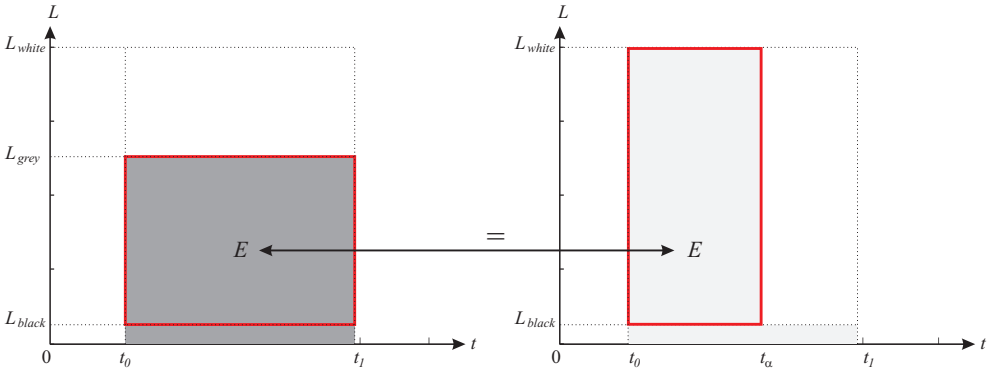


Figure 3.3: Simulating grey intensities L_{grey} by temporally integrating black/white intensities L_{black} and L_{white}.

$$I_{black} = \int_{t_0}^{t_1} kL_{black}(t)\,dt \tag{3.5}$$

$$I_{white} = \int_{t_0}^{t_1} kL_{white}(t)\,dt \tag{3.6}$$

As we work with long sensor exposure times $(t_1 - t_0)$ compared to the screen's refresh rate, L_{black} and L_{white} can reasonably assumed to be constant over time [Wilburn 08]. The goal is now to create a linearly controllable pixel radiance I_α, with $\alpha \in [0,1]$. This is obtained by exposing the image partly with L_{black} and partly with L_{white}. The image irradiance is then:

$$I_\alpha = \int_{t_0}^{t_\alpha} kL_{white}(t)\,dt + \int_{t_\alpha}^{t_1} kL_{black}(t)\,dt \tag{3.7}$$

$$= \alpha I_{white} + (1-\alpha) I_{black} \tag{3.8}$$

This idea is illustrated in figure 3.3. Notice that when the so-called black level L_{black} is not sufficiently dark for certain applications, a dark image can be recorded and later be subtracted from the recorded images. Concrete measurements that illustrate the linearity of our approach with respect to the naive approach, are depicted in figure 3.2.

(a) (b) (c)

Figure 3.4: Exaggerated screen response “aberrations”. The figures show (a) angular, (b) spatial and (c) temporal aberrations. (a, b) are LCD displays, where (c) is a CRT.

3.2.2 Angular Dependence

Certain displays (mainly LCDs) suffer from a drop in brightness when a pixel is observed from a “large” angle with respect to the screen’s surface normal (see figure 3.4 (a)). As this effect is generally considered to be spatially uniform, the angular response can be measured by recording an individual pixel under different angles. In our work we deal with the angular dependence by (a) employing a screen which only slightly exhibits this problem, and (b) placing the subject at a sufficient distance from the screen. Of course this is only required in cases where the method is sensitive to this dependence.

3.2.3 Spatial and Temporal Dependence

Although spatial (figure 3.4 (b)) and temporal (figure 3.4 (c)) dependent brightness variations are negligible for a majority of applications, it is worth mentioning that pixels might have different responses at other screen pixel locations or at a different moment in time. The former effect is due to small aberrations in the backlight , the latter because of the difference in latency between the redrawing moment of different pixels. Problems due to this latency are only present in cases of very short exposure times (less than a second). As the exposure time in our application consists of multiple seconds, we do not experience any such problems.

3.3 Box Construction as Dark Room

When employing a screen-camera setup for optical measurements, it is desirable that the illumination is fully controllable. In order to remove uncontrollable stray light, we simply use a box covered with a matte, black and thus light absorbing cloth. Furthermore, high albedo or highly specular materials inside the box are avoided where possible, in order to prevent contamination of the measurements by having disturbing interreflections [Goesele 00].

3.4 Conclusions

There are a number of factors we need to take into account when employing computer screens as light sources. In practice however, most of them can either be calibrated for or they can simply be avoided. Fortunately, the majority of these disadvantageous factors will probably decrease by the continuously improving technology, as these negative properties are also not desirable for regular usage. Current SLR cameras can be employed straightforwardly, as long as the raw sensor data can be accessed.

Chapter 4

Calibration

Contents

In this chapter, we will discuss different approaches to geometrically calibrate a setup consisting of a camera, a planar illuminant and an approximately planar subject. Since a computer screen is employed as planar illuminant, this means that for every screen pixel location we want to know the 3D location in Euclidean space (e.g. expressed in millimeters or inches) with respect to the camera. The same holds for each subject's location visible by the camera.

4.1 Setup Parameters

The geometric properties of the setup can be described by a number of parameters. The collection of these parameters will be explained in this section.

4.1.1 Camera Parameters

A traditional digital camera can be seen as a device projecting 3D points onto a 2D image plane. Mathematically, this is described by a single matrix transform combined with a lens distortion function. In this work, we will only focus on pinhole camera models, excluding affine camera models and other less common models such as pushbroom cameras.

Projection Matrix

When assuming there is no lens distortion, the relation between a 3D point $\boldsymbol{X}$ and its corresponding 2D projected pixel $\boldsymbol{x}$ is defined by a projective transform P as follows: $\boldsymbol{x} = P\boldsymbol{X}$. This projection matrix P contains the camera related parameters K, the position $\boldsymbol{C}$, and orientation R of the camera coordinate frame. More specifically, $P = KR[I|-\boldsymbol{C}]$, where I is a 3×3 identity matrix. The matrix K contains the so-called *internal* or *intrinsic* camera parameters, whereas $R[I|-\boldsymbol{C}]$ contains the *external* or *extrinsic* parameters. The intrinsic matrix K describes sensor and lens dependent image formation properties, as will be explained later.

Since we work with a single camera in our setup, we are free to let the camera center define the origin of the Euclidean coordinate system, and to let the optical axis define the $\boldsymbol{Z}$-axis, as shown in figure 4.1. Therefore, in our case P can be simplified to $P = K[I|\mathbf{0}]$, where $\mathbf{0}$ is a 3×1 matrix containing zeroes. Consequently, only the internal camera parameters K will matter in our case.

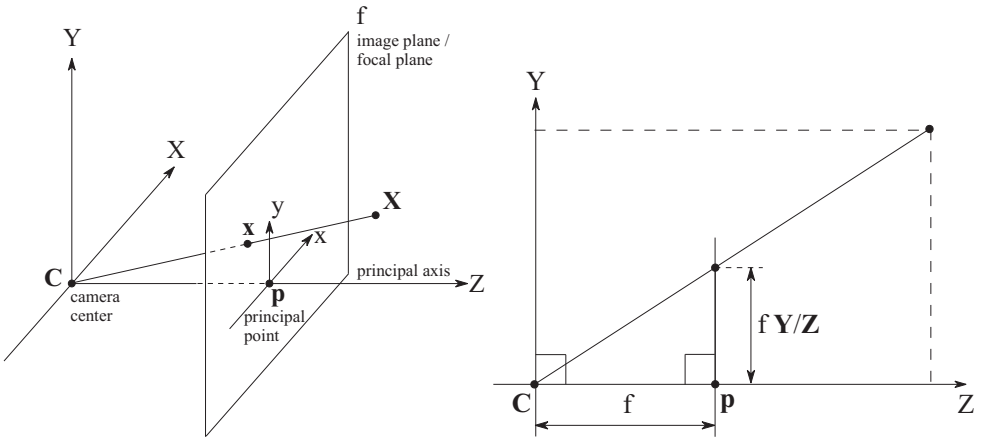


Figure 4.1: Pinhole camera geometry.

The K-matrix is constructed as follows:

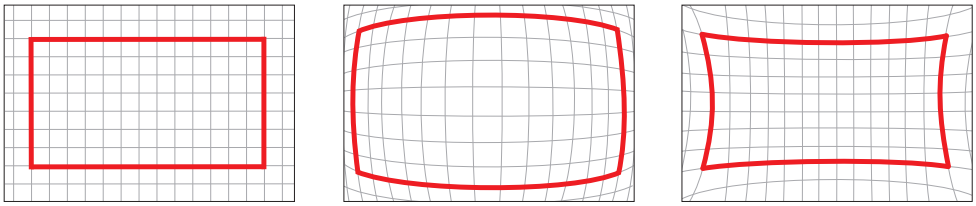
$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

where the parameters are defined as:

- α_x, α_y represent the focal plane distance divided by the physical width (respectively height) of the pixels on the sensor. Note that the value for α_x and α_y might be different, since the sensor might consist of non-square pixels.
- x_0 and y_0 define the principal point in terms of pixel dimensions. This is the point on the image plane which is at the base of the perpendicular from the center of the lens.
- s is the skew parameter, describing the skewness of pixels. Usually this value will be equal to 0, as pixels are rectangular.

Distortion Model

In practice, lenses often suffer from a number distortions including radial distortions, chromatic aberrations, vignetting etc. For the camera and lenses we use, only radial



(a) no distortion (b) barrel distortion (c) pincushion distortion

Figure 4.2: Image of a red rectangle under different three types of radial lens distortions.

lens distortions are a real issue for our purposes. Fortunately, this form of aberrations can easily be compensated for.

Although there exist lenses exhibiting more complex radial distortion patterns, typically a lens with a certain focal length can be categorized in either one of the following groups (see figure 4.2):

No distortion: high quality lenses often exhibit no or a negligible amount of radial distortion (figure 4.2 (a)).

Barrel distortion: this type of radial distortion is typically present when using wide angle lenses (figure 4.2 (b)).

Pincushion distortion: this mostly appears when employing large focal length lenses (figure 4.2 (c)).

Radial distortions can be regarded as displacements applied to the theoretical pixel locations. These displacements depend on the distance r with respect to the principle point and can be modeled by a distortion function $L(r)$. The relation is then:

$$\begin{pmatrix} x_d \\ y_d \end{pmatrix} = L(r) \begin{pmatrix} x \\ y \end{pmatrix}$$

where:

- (x_d, y_d) is the actual image position after radial distortion, relative with respect to the distortion center which is assumed to coincide with the principal point.

- (x, y) is the theoretical undistorted projected pixel, also relative to the distortion center.
- r is the radial distance $\sqrt{x^2 + y^2}$ from the distortion center.
- $L(r)$ represents the distortion factor and is function of the distance r.

The function $L(r)$ is defined for positive values r as the following Taylor expansion:

$$L(r) = 1 + \kappa_1 r + \kappa_2 r^2 + \kappa_3 r^3 + \dots$$

In practice, a limited set (two or three) of parameters $\kappa_1, \kappa_2, \kappa_3, \dots$ suffices to describe the distortion required for undoing this unwanted lens effects .

Obtaining Camera Parameters

Obtaining camera parameters is a well-understood field and outside the scope of this dissertation, we will refer the interested reader to the work of Hartley and Zisserman [Hartley 04].

In our work, we make use of traditional calibration tools which employ a checkerboard calibration pattern. Both the internal camera parameters and a distortion function are constructed from a limited number of images of this pattern under different angles and positions [Bouguet 06].

4.1.2 Subject's Supporting Plane Parameters

As our sample will always be globally planar, we will approximate its coarse geometry by its supporting plane. The supporting plane is defined by the parameters $\boldsymbol{\pi}$ as follows:

$$\boldsymbol{\pi}^T \boldsymbol{X} = 0$$

$\boldsymbol{X}$ is a 1×4 homogeneous 3D point lying on the plane $\boldsymbol{\pi}$, where $\boldsymbol{\pi}$ is a also 1×4 vector. Similar to the previous section, these plane parameters can be found using common calibration tools making use of a checkerboard pattern placed at the position of the subject [Hartley 04, Bouguet 06].

4.1.3 Illuminant Parameters

When all camera and subject related parameters are defined, only the illuminant's parameters are not covered yet. In this dissertation we will use two different styles of defining the position of the illuminant with respect to the camera, because both are used in literature.

The first, and straightforward, method is simply specifying the four 3D corner points of the screens $\boldsymbol{X}^{ul}_{screen}$, $\boldsymbol{X}^{ur}_{screen}$, $\boldsymbol{X}^{bl}_{screen}$ and $\boldsymbol{X}^{br}_{screen}$. A 3D screen point is then found after bilinear interpolation of the four screen corners.

The second approach of storing this position is by a similarity transform M. Given a screen pixel $\boldsymbol{u}$, the corresponding 3D point $\boldsymbol{X}$ is found as follows: $\boldsymbol{X} = M\boldsymbol{u}$.

If the screen is directly visible to the camera, the calibration process is straightforward: displaying a checkerboard pattern on the screen, standard calibration tools can be used to locate the screen plane [Hartley 04, Bouguet 06]. This procedure is very similar to the one presented in section 4.1.2.

In our case, the camera and screen are facing a similar directon (figure 4.3). A mirroring surface is required in order to render the screen visible to the camera [Tarini 05, Bonfort 06, Funk 07, Nitschke 09]. In this work, we will only focus on this case, presenting two novel approaches. Since this is a more involving task, it will be explained more carefully in sections 4.2 and 4.3.

Introducing a mirror turns our setup into a so-called catadioptric system. Catadioptric systems are combinations of reflective (cataoptric) and refractive (dioptric) objects, or as in this dissertation, a mirror and lens combination which will be used to geometrically calibrate the screen and camera. Depending on the type (plane/-sphere/hyperbolic/...) and placement (central/off principal axis) of the mirror and the lens, a single or multi-viewpoint catadioptric camera is created [Baker 99, Geyer 01]. Notice that in the case of a multi-viewpoint system, it is possible to reconstruct 3D geometry from a single image because of the viewpoints created by the mirror reflections [Kuthirummal 06, Lanman 06]. In the following sections only two major mirror types will be discussed, being a planar and a spherical one. The former produces a single viewpoint system, whereas the latter produces a multi-viewpoint system.

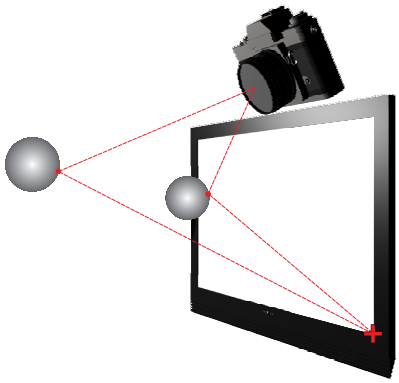


Figure 4.3: Setup calibration of a computer screen and camera facing a similar direction. Intersecting the reflection rays of the red feature yields the 3D position of the screen corner.

4.2 Calibration with a Planar Mirror

In this section, two different planar mirror calibration methods are explained. The fundamental difference between these approaches is in the detection of the plane equation of the mirror.

4.2.1 Mirror Detection

Funk and Yang [Funk 07] determine the screen's position and orientation with respect to the camera using a planar surface mirror. In order to determine the location of the mirror, a calibration pattern is attached to the mirror. The camera proceeds to capture this calibration pattern, in addition to the reflected image of an additional calibration pattern emitted by the screen. The screen coordinates can then be determined by mirroring back the reflected screen pattern over the planar mirror.

Bonfort et al. [Bonfort 06] present an alternative approach using a planar mirror, but instead of using a calibration pattern to identify the mirroring plane, they employ a mirroring hard disk platter with known interior and exterior radii. The projections of these circular boundaries yield two ellipses, which provide sufficient constraints to identify the mirroring plane.

4.2.2 Calibration Pattern

The type of calibration pattern used is rather irrelevant, as long as it correctly specifies a sufficient number of illuminant locations. Funk and Yang [Funk 07] employ checkerboard patterns, allowing for using traditional calibration tools. Bonfort et al. [Bonfort 06] make use of coded illumination patterns instead yielding a higher number of labeled locations but requiring a more specialized calibration method.

4.3 Calibration with a Spherical Mirror

In this dissertation we present two novel calibration methods employing a spherical mirror instead of a planar one. First the procedure will be described, then the results will be discussed and compared.

Our proposed algorithms consist of two major stages: (a) locating the position of the spherical mirror, using only the camera image of the sphere and the radius, and (b) computing the 3D location of the screen surface. The first stage is applied to two or more images with different sphere locations and these results are passed on together to the second stage.

4.3.1 Mirror Detection

In order to estimate the Euclidean world coordinates of the spherical mirror, we first need to locate its image in camera coordinates. More precisely, the image contour is sufficient for our purposes. This, combined with the intrinsic camera parameters is sufficient to locate the spherical mirror.

Contour Detection

The contour of a projected sphere on the image plane of a camera will be an ellipse. Hence, given a set of 2D edge pixels $\{\boldsymbol{x}^i\}$, an ellipse has to be fitted. This set of pixels can either be sparse when a manual selection is applied, or dense when an automatic background subtraction procedure is performed. This background subtraction is then followed by a set of morphological operations (erosion, dilation and subtraction) that provide us with the actual contour, eliminating any unwanted pixel noise and bridging minor lapses.

Given the resulting pixel set $\{\boldsymbol{x}^i\}$, we want to locate the ellipse that provides the best fit through these data points. The equation of a general conic in homogeneous coordinates is $\boldsymbol{x}^T C\boldsymbol{x} = 0$, where C is of the form

$$C = \begin{bmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{bmatrix} \tag{4.1}$$

However, as the ellipse contour is the result of a sphere projected onto the image plane, this poses an additional constraint on our conic equation: $b = 0$. As standard ellipse detection algorithms do not take this into account, we choose to apply a tailored RANSAC-based [Fischler 81] approach, in which we can instantiate the model from a four point sample. We rewrite our conic equation as

$$\begin{bmatrix} (x_1^i)^2 & (x_2^i)^2 & x_1^i & x_2^i & 1 \end{bmatrix} \boldsymbol{c} = 0 \tag{4.2}$$

where $\boldsymbol{c} = (a,c,d,e,f)^T$ is the conic C represented as a 5-vector. After the proper data normalization [Hartley 97], the data points are stacked in a 4×5 matrix. The conic for the associated model can then be determined as the null space of this system, by means of Singular Value Decomposition (SVD). During the RANSAC search, conics that do not represent ellipses with a low eccentricity are automatically rejected. After the search has completed, a final least-squares solution from the stacked matrix of all inliers of the found model is computed.

Sphere Detection

After the sphere contour in the camera image is located, we need to determine its location in our reference coordinate system $[I|\mathbf{0}]$.

The back-projection of a conic C results in a degenerate quadric [Hartley 04], the right circular cone Q with apex $(0,0,0,1)^T$.

$$\begin{aligned} Q &= P^T C P & (4.3) \\ &= (K[I|\mathbf{0}])^T C (K[I|\mathbf{0}]) & (4.4) \end{aligned}$$

In order to reduce the problem to a manageable form, we align the cone with the Z-axis of the coordinate system, the viewing direction of the camera. Unlike the

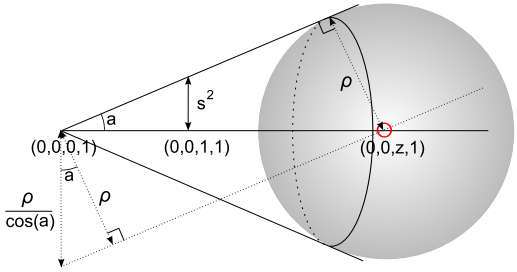


Figure 4.4: Sphere localization (reduced form). A sphere with a known radius is fitted into a right circular cone. The cone is defined by the camera position and the contour of the projected spherical mirror.

approach of Shiu and Ahmad [Shiu 89], our change of coordinate system is achieved by only a simple rotation R.

$$Q' = R^T Q R \tag{4.5}$$

After the transformations mentioned above, the obtained matrix Q' is of the following form:

$$Q' = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -s^2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.6}$$

The projection angle of the cone center is now given by $\arctan(s^2)$ (figure 4.4). The distance z from the sphere center to the apex of the cone can now be determined, assuming the physical sphere radius ρ is known.

$$z = \frac{\rho}{s\cos\left(\arctan s^2\right)} \tag{4.7}$$

$$= \rho\frac{\sqrt{1+s^2}}{s} \tag{4.8}$$

Once the distance z is determined, we are able to locate the position of the sphere center c in our reference coordinate frame.

$$c = z\,\frac{K^{-1}\left[x\ y\ 1\right]^T}{\left\|K^{-1}\left[x\ y\ 1\right]^T\right\|} \tag{4.9}$$

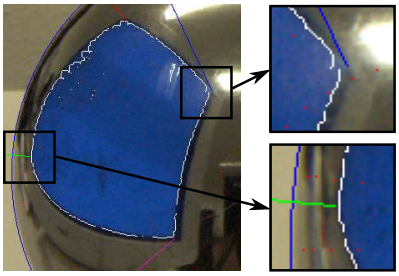


Figure 4.5: (left) Detection of barely visible corners; (top right) Corners do not lie on detected edges (white) due to a specular highlight; (bottom right) Nearly indistinguishable corner.

where $[x\ y\ 1]^T$ is the center of the detected contour, and we reduce the viewing direction vector to its unit length.

4.3.2 Calibration from Corner Reflections

Locating the spherical mirror in a frame allows for the computation of the corresponding reflection vectors for each of the pixels within the sphere contour. After the image locations of the four screen corners are detected, a set of reflection vectors intersect in their three-dimensional coordinates. Combining two (or more) sets of such vectors associated with different mirror positions, an accurate estimate of these screen corners can be calculated. This method is presented by Tarini et al. [Tarini 05].

However, accurately locating the screen corners in camera coordinate space is not a trivial task. Due to the nonlinearity of the four reflected screen edges and possible distortions in the contour detection due to specular highlights, commonly used corner detectors such as the Harris detector [Harris 88] are unable to properly locate the required screen corners. Even manual corner labeling can be difficult, depending on the relative positioning of the hardware in the setup (figure 4.5). Therefore, we will present alternative approaches that take into account the screen edges and the interior pixels.

4.3.3 Calibration from Edge Reflections

In order to facilitate corner detection, the contour pixels are transformed to a better suited coordinate space (figure 4.6).

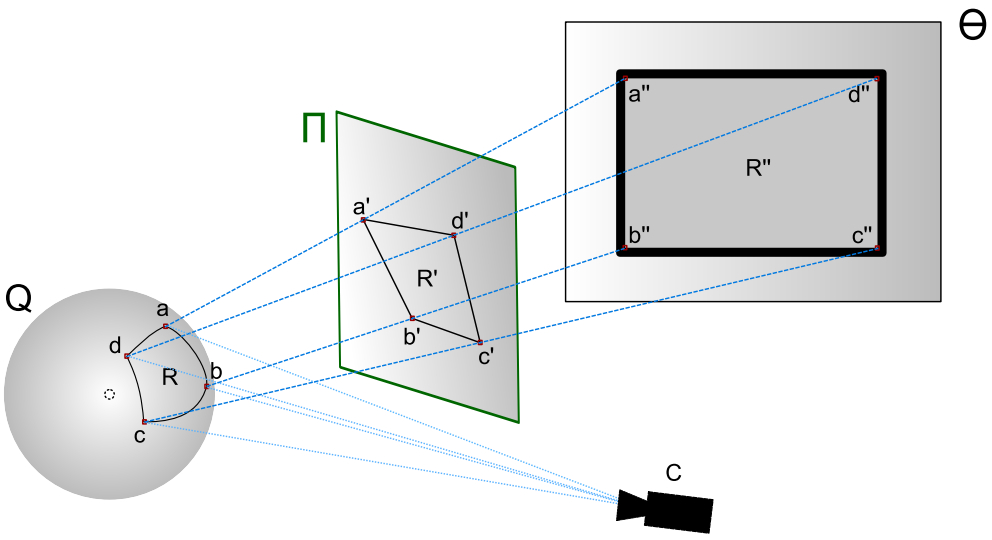


Figure 4.6: Visualization of the 2D corner detection in alternate space.

2D Screen Corner Estimation

Letting the screen emit a constant luminance value in a single color channel allows for the detection of its corresponding pixels in the sphere contour (figure 4.5). We perform a set of morphological operations to extract the edge of the screen, similar to our approach for the sphere detection. Then we will convert the curved edges into straight ones by working in an alternate coordinate space.

Alternate Coordinate Space In camera coordinate space, we label a set of three pixels $\{\boldsymbol{p}^i_{cam}\}$. These pixels are chosen from the four - or when two of these points collide, three - intersections of the screen contour with its bounding box: $\{(x,y) \mid x = min_x \vee y = min_y \vee x = max_x \vee y = max_y\}$. While clockwise traversing the contour, we label the selected points respectively as $\boldsymbol{p}^1_{cam}$, $\boldsymbol{p}^0_{cam}$, and $\boldsymbol{p}^2_{cam}$. As the position and radius of the spherical mirror $(\boldsymbol{c}, \rho)$ are known, we compute the corresponding intersections $\{\boldsymbol{P}^i_s\}$ of this sphere with the back-projected lines from $\{\boldsymbol{p}^i_{cam}\}$. Because the sphere normals in these points are also known, we can compute the resulting reflection directions $\{\vec{\boldsymbol{R}}^i\}$. Finally, we locate the points $\{\boldsymbol{P}^i_{pl}\}$:

$$\boldsymbol{P}^i_{pl} = \boldsymbol{P}^i_s + k\vec{\boldsymbol{R}}^i \tag{4.10}$$

where k is a positive constant.

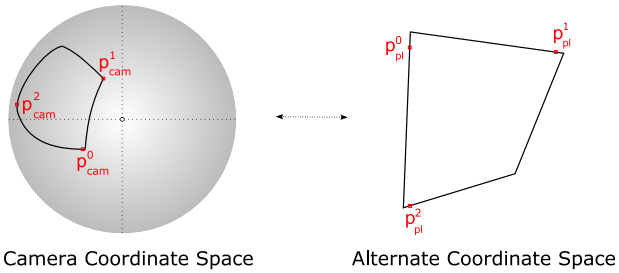


Figure 4.7: By projecting the reflections of the screen contour on an appropriately parameterized plane, we reduce the problem of corner detection in camera space to line detection and intersection in an alternate space.

We will now use the plane through these three points as the basis of our alternate coordinate space. For each screen edge pixel, the intersection of the corresponding reflection rays with the computed plane is determined. If a reflection ray is parameterized as $\boldsymbol{X}_s^i + t\vec{\boldsymbol{R}}^i$, with $\boldsymbol{X}_s^i$ a point on the sphere and $\vec{\boldsymbol{R}}^i$ the associated reflection direction, the following equation provides us with a new parametrization.

$$\begin{bmatrix} t \\ u \\ v \end{bmatrix} = \begin{bmatrix} -\vec{\boldsymbol{R}}^i & (\boldsymbol{P}_{pl}^1 - \boldsymbol{P}_{pl}^0) & (\boldsymbol{P}_{pl}^2 - \boldsymbol{P}_{pl}^0) \end{bmatrix}^{-1} \left[\boldsymbol{X}_s^i - \boldsymbol{P}_{pl}^0\right] \tag{4.11}$$

Line Detection As can be seen in figures 4.6 and 4.7, this new parametrization has reduced the problem of locating the screen corners to line detection and the choice of appropriate intersection points. The four screen edges are detected using RANSAC and a final optimization step. We then compute the six intersection points, and automatically choose the four non-collinear points from this set.

Original Coordinate Space In order to map the coordinates of the screen corners back to camera space, we first need to compute the 3D coordinates of the corners (u,v) on the parameterized plane:

$$\boldsymbol{X}_{pl} = \boldsymbol{P}_{pl}^0 + u(\boldsymbol{P}_{pl}^1 - \boldsymbol{P}_{pl}^0) + v(\boldsymbol{P}_{pl}^2 - \boldsymbol{P}_{pl}^0) \tag{4.12}$$

Given a point $\boldsymbol{X}_{pl}$, camera center $\boldsymbol{O}$ and spherical mirror $(\boldsymbol{c}, \rho)$, the point $\boldsymbol{X}_s = \rho(\boldsymbol{N} - \boldsymbol{O}) + \boldsymbol{C}$ on the mirror that reflects the ray through $\boldsymbol{X}_{pl}$ into the camera is

uniquely defined by the reflection equation.

$$\frac{\boldsymbol{X}_{pl}-\boldsymbol{X}_s}{\|\boldsymbol{X}_{pl}-\boldsymbol{X}_s\|}\cdot\boldsymbol{N}=\frac{\boldsymbol{O}-\boldsymbol{X}_s}{\|\boldsymbol{O}-\boldsymbol{X}_s\|}\cdot\boldsymbol{N} \tag{4.13}$$

The unknown parameter in the equation above is the reflection normal $\boldsymbol{N}$. If we parameterize this vector as a normalized weighted sum of the vectors $\boldsymbol{X}_{pl}-\boldsymbol{C}$ and $\boldsymbol{O}-\boldsymbol{C}$,

$$\boldsymbol{N}(t)=\frac{t\frac{\boldsymbol{X}_{pl}-\boldsymbol{C}}{\|\boldsymbol{X}_{pl}-\boldsymbol{C}\|}+(1-t)\frac{\boldsymbol{O}-\boldsymbol{C}}{\|\boldsymbol{O}-\boldsymbol{C}\|}}{\left\|t\frac{\boldsymbol{X}_{pl}-\boldsymbol{C}}{\|\boldsymbol{X}_{pl}-\boldsymbol{C}\|}+(1-t)\frac{\boldsymbol{O}-\boldsymbol{C}}{\|\boldsymbol{O}-\boldsymbol{C}\|}\right\|} \tag{4.14}$$

the three-dimensional normal $\boldsymbol{N}$ is defined by a single scalar $t\in[0,1]$. We estimate the correct value of t by iteratively minimizing the energy function

$$E(t)=\left\|\left[\frac{\boldsymbol{X}_{pl}-\boldsymbol{X}_s(t)}{\|\boldsymbol{X}_{pl}-\boldsymbol{X}_s(t)\|}-\frac{\boldsymbol{O}-\boldsymbol{X}_s(t)}{\|\boldsymbol{O}-\boldsymbol{X}_s(t)\|}\right]^T\boldsymbol{N}(t)\right\| \tag{4.15}$$

We initiate the process with a value of $t=\frac{1}{2}$. This initiation is already a reasonably good estimation, as it the approximation is correct up to the neglected sphere's radius. Using this initial value, the algorithm quickly converges to a global minimum.

The camera pixel coordinates x_c of the wanted screen corners are now given by the equation

$$x_c=[K|\mathbf{0}]\boldsymbol{X}_s \tag{4.16}$$

An example of the accuracy of these reprojected corners is illustrated in figure 4.5.

3D Screen Corner Estimation

As the screen corners are now located for each frame in camera coordinates, we can combine all sets of reflection rays associated with each individual corner into a least-squares problem. Next, we will describe the solution for the set of reflection rays associated with a single corner, looking for their common intersection point.

Single Corner Estimation The reflection ray associated with frame i can be parameterized as $l^i:\boldsymbol{X}_s^i+t\vec{\boldsymbol{R}}^i$. The distance $d(\boldsymbol{X},l^i)$ from a point $\boldsymbol{X}$ to a line l^i is

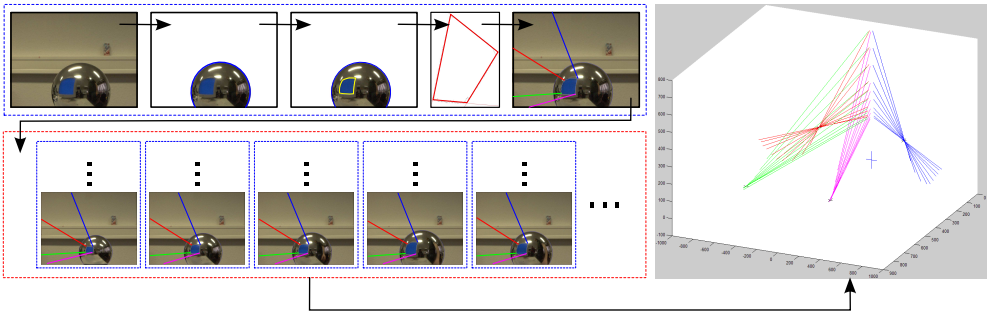


Figure 4.8: An overview of our calibration pipeline, using images from a real-world data set.

defined as follows:

$$d(\boldsymbol{X}, l^i) = \frac{\|\vec{\boldsymbol{R}}^i \times (\boldsymbol{X}_s^i - \boldsymbol{X})\|}{\|\vec{\boldsymbol{R}}^i\|} \tag{4.17}$$

$$= \left\| \left(\frac{[\vec{\boldsymbol{R}}^i]_\times}{\|\vec{\boldsymbol{R}}^i\|} \right) x - \left(\frac{[\vec{\boldsymbol{R}}^i]_\times}{\|\vec{\boldsymbol{R}}^i\|} \boldsymbol{X}_s^i \right) \right\| \tag{4.18}$$

If we formulate the problem of finding the common intersection point of all lines l^i as finding the point $\boldsymbol{X}$ that minimizes the distance $d(\boldsymbol{X}, l^i)$ for all i, then we reduce the problem to a least-squares minimization of the form $\|A\boldsymbol{X} - b\|$. Problems of this form can be solved by using the normal equations $(A^T A)x = A^T b$. If $A^T A$ is invertible, the solution to such a problem is $\boldsymbol{X} = (A^T A)^{-1} A^T b$.

Global Pixel Localization Once the position of the individual screen corners is computed, every pixel $(u, v) | u, v \in [0, 1]$ located on the screen surface can be mapped onto their three-dimensional Euclidean coordinates $\boldsymbol{X}$ by bilinear interpolation.

$$\boldsymbol{X} = \begin{bmatrix} 1-u \\ u \end{bmatrix}^T \begin{bmatrix} \boldsymbol{X}_{screen}^{ul} & \boldsymbol{X}_{screen}^{ur} \\ \boldsymbol{X}_{screen}^{bl} & \boldsymbol{X}_{screen}^{br} \end{bmatrix} \begin{bmatrix} 1-v \\ v \end{bmatrix} \tag{4.19}$$

An overview of the entire procedure is depicted in figure 4.8.

4.3.4 Calibration from Full Surface Reflections

In this approach, we will also employ the localized spherical mirror to render the screen visible to the camera. However, instead of only taking into account the reflected screen corners and edges, we will let every single screen pixel contribute to

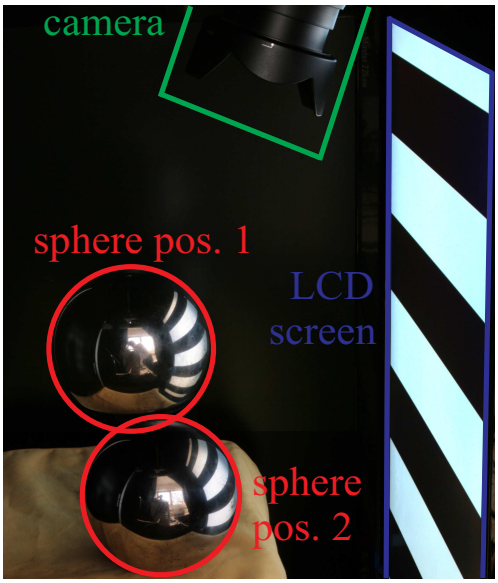


Figure 4.9: The camera records Gray code patterns reflected off a spherical mirror. This procedure is executed for two different sphere locations.

the estimated solution by encoding their positions. This reduces the required number of manual sphere placements. The setup is depicted in figure 4.9

The processing pipeline consists of three separate modules. First, for each set of images associated with a single sphere position, we estimate the 3D location of the spherical mirror as described in section 4.2.1. Second, we establish the bijective function between the camera pixels and the corresponding screen pixels. Finally, we estimate the optimal 3D screen location by minimizing the geometric error between the reprojected 2D screen pixels and the computed reflected ray intersections. The last two modules will be discussed in more detail in this section.

Screen Pixel Labeling

We start explaining how Gray codes can provide us with a unique labeling for the screen pixels, as well as how it facilitates our ability to distinguish between screen reflections among different pixels.

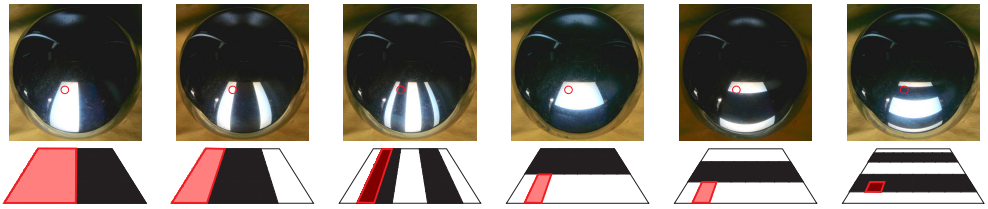
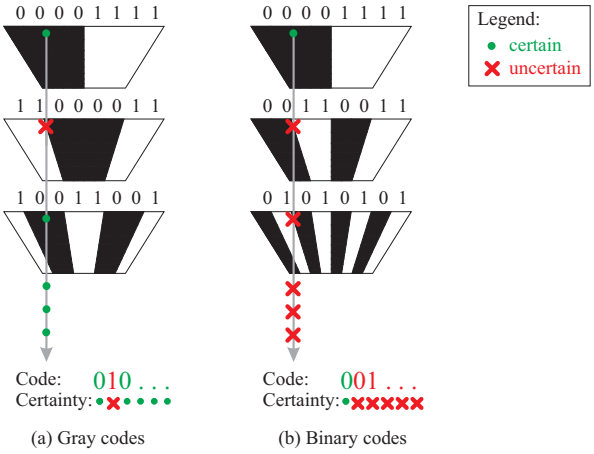


Figure 4.10: Gray codes uniquely identify matching camera pixels (top row) and screen pixels (bottom row). The patterns are refined step by step, narrowing down the possible regions illuminating the camera pixel. At the finest pattern level, the only remaining possibility will be the reflected pixel.

Gray codes As we want to reconstruct the screen's position with respect to the camera, the reflected screen pixels have to be matched to their reflections on the sphere. In order to uniquely identify each individual reflected pixel, we encode its 2D position on the screen using Gray code illumination patterns. This type of patterns has been used before to match surface locations for stereo [Sato 87, Scharstein 03] and allows for robustly and efficiently encoding ten thousands of screen positions with only tens of images. An example of Gray code illumination is given in figure 4.10.

In order to simplify the explanation of the functioning of Gray code patterns, we initially assume we deal with the more intuitive, but less robust, binary codes. The basic idea is to assign a unique increasing number to each pixel from left to right, and top to bottom. For instance, for a screen resolution of 1024×768 pixels, labels range from 0 (left) to 1023 (right), and 0 (top) to 767 (bottom). For now, let us focus only on the left to right encoding as the top to bottom encoding works analogously. When each unique label's binary representation is sequentially displayed as n black (0) and white (1) pixels, n binary patterns such as depicted in figure 4.11(b) will be generated.

For detecting the reflected codes, we need to establish for each reflected screen pixel, and each pattern, if it reflects either a white (1) or black (0) screen region. In order to increase the robustness of this decision, we also employ their complementary illumination patterns. The brightest pixel then gets a 1 assigned, the other a 0. The composed bit-code now only needs to be converted back from binary to decimal representation and the reflected screen position is found.



(a) Gray codes (b) Binary codes

Figure 4.11: Gray code patterns (left) versus binary patterns (right). In case of Gray code patterns, a reflection can point toward a pattern edge for only a single pattern refinement level as the black/white edges will be moved each level. This is not the case for binary patterns. Once a reflection points toward a binary pattern edge, it will fall on an edge for every subsequent pattern yielding an uncertain code detection.

Clearly, n-bit codes can represent 2^n different labels. Hence, only $\log_2(n)$ patterns are needed to encode n different labels/locations. More specifically, for a $m \times n$ resolution screen, $2\left(\log_2(m)+\log_2(n)\right)$ patterns are necessary when complement patterns are included.

In practice however, we use Gray codes instead of binary codes. As can be seen in figure 4.11, these patterns are very similar except from a small transformation. In a perfect situation, where a reflection points clearly to either a black or white region, Gray codes and binary code patterns perform equally well. However, in practice, when small detection errors occur because of camera noise, reflecting an edge of a pattern etc, Gray codes perform better [Boyer 87, Salvi 04]. There are two important benefits for using Gray code patterns:

- A reflection can only point to a pattern edge at a single refinement level. This is not the case for binary patterns. Once a reflection points toward a binary pattern edge, it will fall on an edge for every subsequent pattern yielding an uncertain code detection. This is illustrated both theoretically (figure 4.11) and practically (figure 4.12).

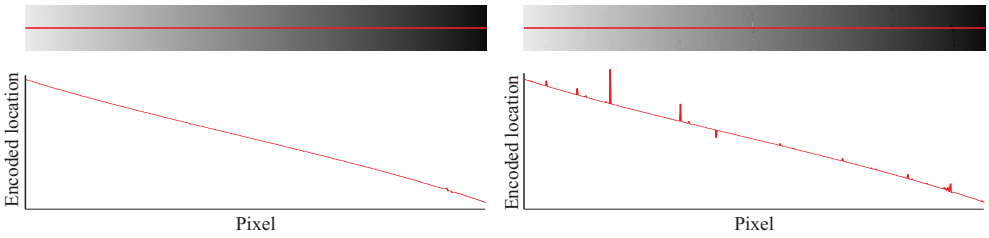


Figure 4.12: Practical example of Gray code patterns (left) versus binary patterns (right). Overall, the resolved reflected screen positions are equal, except from the errors on the pattern edges when binary patterns are used.

- The Hamming distance between two subsequent codes (or the number of code positions for which the corresponding symbols are different) is always one. This can be seen by comparing the code of two subsequent numbers from table 4.1. For example, take 7 and 8 in binary representation, which is 0111 and 1000 respectively. In Gray code representation, their codes are 0100 and 1100. Hence the Hamming distance for the binary code is 4, as all 4 bits have to be flipped to make the codes equal, while the distance is only 1 for the Gray codes.

When using Gray codes, the detected codes can be transformed in the more directly applicable binary representation. This conversion is nothing more than the application of a number of bitwise operations and can therefore efficiently be handled.

Reflection Mask Using the Gray code patterns, we will get a code for every single camera pixel. Of course, only the reflections of the screen have to be extracted for use in the next step of our algorithm. A naive method to accomplish this would consist of illuminating the sphere with a complete black and a complete white pattern, subtract the first captured image from the second one, and threshold the result. This way reflections of alternative constant light sources would be filtered out. However, this is not optimal as there may still be bright interreflections.

In our method we present a more sophisticated masking technique that selects only sufficiently specular screen reflections. It is inspired by our glossiness acquisition method, which will employ Gray codes in a similar setup (see chapter 6). In

Decimal	Gray codes	Binary codes
0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1
2	0 0 1 1	0 0 1 0
3	0 0 1 0	0 0 1 1
4	0 1 1 0	0 1 0 0
5	0 1 1 1	0 1 0 1
6	0 1 0 1	0 1 1 0
7	0 1 0 0	0 1 1 1
8	1 1 0 0	1 0 0 0
9	1 1 0 1	1 0 0 1
10	1 1 1 1	1 0 1 0
11	1 1 1 0	1 0 1 1
12	1 0 1 0	1 1 0 0
13	1 0 1 1	1 1 0 1
14	1 0 0 1	1 1 1 0
15	1 0 0 0	1 1 1 1

Table 4.1: Comparison of the structure of decimal, Gray and binary codes. Notice the one Hamming distance between subsequent Gray codes, and the often larger distance for standard binary codes.

order to limit our mask to screen reflections only, we will observe the intensity differences between their complements for each pattern refinement level. As has been shown in the work of Ramamoorthi et al. [Ramamoorthi 01], reflections can be seen as a convolution of the incoming light pattern and the BRDF kernel of the material. Because of this property, the stripe pattern will be blurred away when a certain pattern refinement level has been reached, as the BRDF kernel will be larger than the pattern stripes . Thus, the reflection of a pattern and its complement tend to converge at a certain pattern refinement level, which means that their intensity differences will converge to zero. This is the key idea behind the masking algorithm, which will be combined with the information from the mask already obtained from the background subtraction step.

First, we determine the per pixel intensity differences by subtracting the complement pattern from the original one. This is followed by determining the pattern

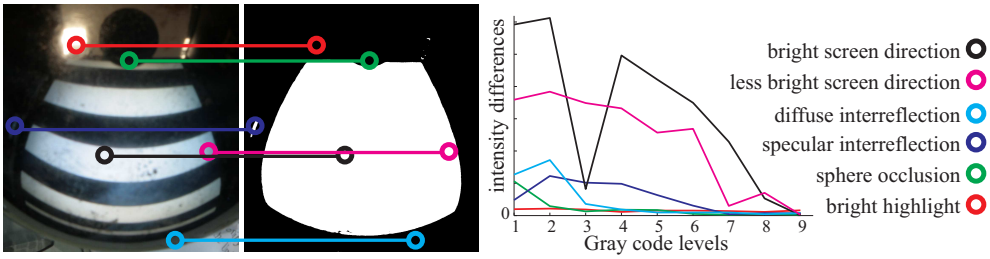


Figure 4.13: The mask indicates whether or not a pixel is a screen reflection. Left: input image. Middle: generated mask. Right: plot of the intensity differences between scene reflections under normal and complement illumination patterns.

refinement level at which all the absolute values of the intensity differences stay bellow a certain threshold. Intuitively, we now have a number indicating the size of the BRDF kernel, where a low number indicates a large kernel (diffuse reflections) and a high number indicates a narrow kernel (specular reflection). Simply thresholding this "glossiness" number gives a robust estimate of the useful screen reflections.

The previous procedure is illustrated in figure 4.13. Our method performs well in the presence of alternative constant light sources, high albedo diffusers, and under all viewing angles with respect to the LCD screen. Our method cannot deal with specular interreflections, so we avoid other shiny objects beside the spherical mirror in our setup.

In figure 4.14 we show a combination of the obtained reflection mask and detected screen positions of the (extended) data set depicted in figure 4.10.

3D Reconstruction

In the final phase of our algorithm, our goal is to estimate the optimal 3D screen location with respect to the camera. For the remainder of this section, we will assume the position of the sphere to be known.

Reflected Ray Intersections In the previous phases, we have established a bijective function between each screen pixel $\boldsymbol{X}$ and its observed camera pixel for each set of images associated with sphere position i. If we use the camera's internal parameters to backproject this camera pixel, we can locate its intersection point $\boldsymbol{P}_i$ on sphere

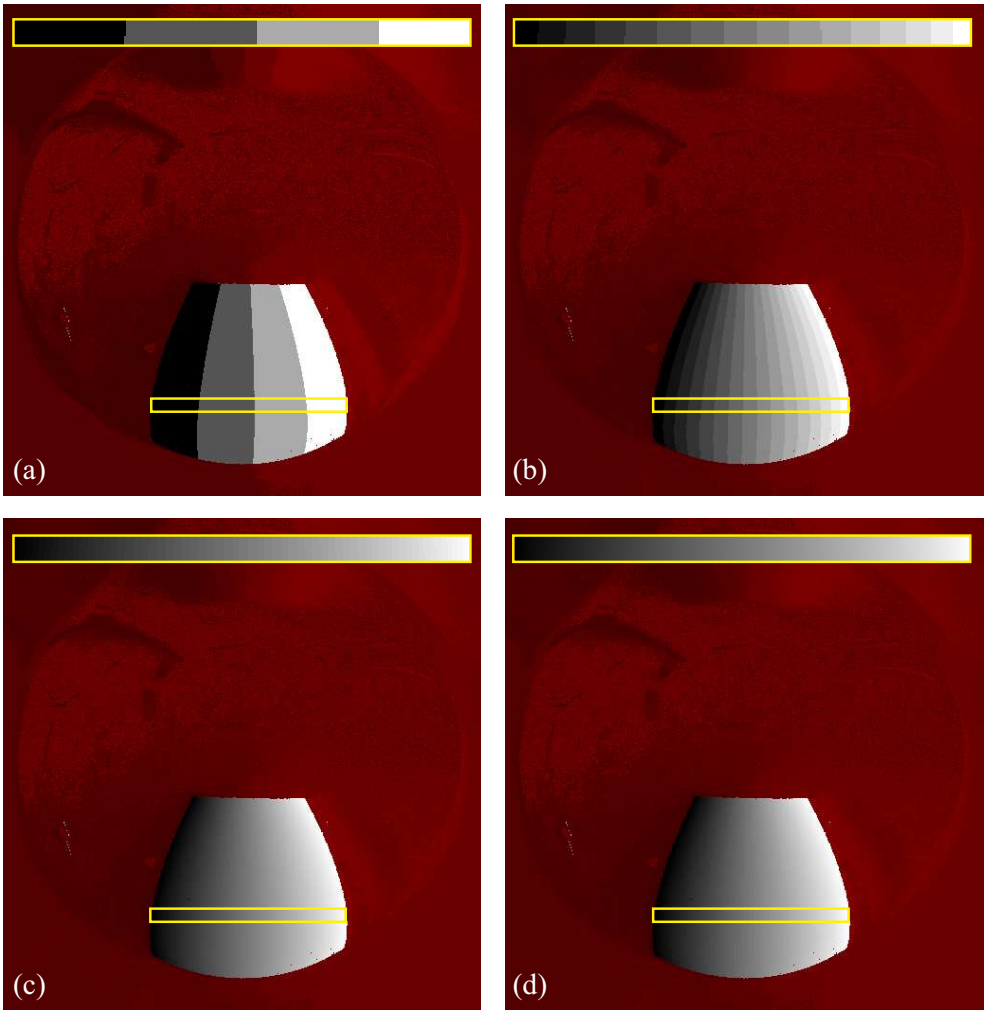


Figure 4.14: Detected screen positions for different pattern refinement levels, blended with the obtained reflection mask. Decoded horizontal 2D screen positions are expressed relative to the screen dimensions. Number of vertical light patterns used: (a) two, (b) four, (c) six and (d) eight.

i. Once this point is known, we can compute the associated reflected ray $l_i : \boldsymbol{P}_i + t\vec{\boldsymbol{R}}_i$, where $\vec{\boldsymbol{R}}_i$ stands for the reflection direction. Combining the information from all reflected rays associated with a single Gray code / screen pixel, we estimate the 3D position for each individual screen pixel $\boldsymbol{X}$, minimizing the combined point-line distance $d(\boldsymbol{X}, l^i)$, defined and solved as proposed in section 4.3.3.

Plane Estimation Once we have a point cloud of estimated 3D screen pixel locations, the next step is to enforce the constraint that all pixels are part of the same planar illuminant. As such, we need to estimate the 3D plane $\boldsymbol{\pi} = (a,b,c,d)^T$ which minimizes the point-plane distance $d(\boldsymbol{\pi}, \boldsymbol{X}) = (\boldsymbol{\pi} \cdot \boldsymbol{X})/|\boldsymbol{\pi}|$. We initiate our estimation with a RANSAC search for a good initial parameter guess, followed by Levenberg-Marquardt minimization with distance measure $d(\boldsymbol{\pi}, \boldsymbol{X})$.

Grid Estimation Once we have a rough estimate of the plane on which the screen should be located, the next step consists of finding the exact 3D coordinates for each screen pixel. In this final step of the 3D reconstruction, we use the final constraint for our estimation problem: the pixels are part of a rigid grid structure. As such, there exists a 2D-to-3D similarity transformation M, which maps each 2D screen pixel location $\boldsymbol{u}$ as close as possible to its 3D counterpart $\boldsymbol{X}$. Assuming we can estimate this optimal M for each given plane $\boldsymbol{\pi}$, this gives rise to a second Levenberg-Marquardt plane estimation routine, minimizing the point-point distance $d(M\boldsymbol{u}, \boldsymbol{X})$.

Similarity Transformation M In order to guarantee good numerical stability for the estimation of M, we have to perform numerical pre-conditioning on both point sets [Hartley 97]. As we want to apply the same normalization procedure on both $\{\boldsymbol{u}\}$ and $\{\boldsymbol{X}\}$, we compute the similarity transformation in a 2D reference frame. For this purpose, we first transfer the 3D plane $\boldsymbol{\pi}$ to the $\boldsymbol{XY}$-plane using transformation T_{3D}.

$$T_{3D} = \begin{bmatrix} R_3 & -R_3\bar{\boldsymbol{X}} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \tag{4.20}$$

In this equation, $\bar{\boldsymbol{X}}$ is the mean vector of the 3D point cloud, and R_3 rotates plane normal $\boldsymbol{N}_{\boldsymbol{\pi}}$ to plane normal $\boldsymbol{N}_{\boldsymbol{XY}}$. Note that the transformation moves the mean $\bar{\boldsymbol{X}}$ to the origin, facilitating subsequent normalization procedures. This transformation

is followed by a projection onto $\boldsymbol{XY}$-plane coordinates.

$$P_{\boldsymbol{XY}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.21}$$

Finally, after this transformation, we rescale the resulting point set $\{\boldsymbol{x}''\}$ to map the average distance $\overline{\|\boldsymbol{x}''\|}$ to $\sqrt{2}$.

$$S_{3D} = \begin{bmatrix} \sqrt{2}/\overline{\|\boldsymbol{x}''\|} & 0 & 0 \\ 0 & \sqrt{2}/\overline{\|\boldsymbol{x}''\|} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.22}$$

In a similar fashion, we need to normalize the 2D screen pixel set. First we need to convert the screen resolution (eg. 1280 $\times$ 1024 pixels) into a physical resolution (eg. 474 $\times$ 297 mm).

$$S_{res} = \begin{bmatrix} w_{phys}/w_{res} & 0 & 0 \\ 0 & h_{phys}/h_{res} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.23}$$

After this aspect ratio correction, we relocate the new point set's mean $\overline{u'}$ to the origin.

$$T_{2D} = \begin{bmatrix} 1 & 0 & -\overline{\boldsymbol{u}'}_x \\ 0 & 1 & -\overline{\boldsymbol{u}'}_y \\ 0 & 0 & 1 \end{bmatrix} \tag{4.24}$$

As finally, we normalize the 2D $\{\boldsymbol{u}''\}$ coordinates, in a similar fashion to the normalization we performed on the 3D $\{\boldsymbol{x}''\}$ coordinates.

$$S_{2D} = \begin{bmatrix} \sqrt{2}/\overline{\|\boldsymbol{u}''\|} & 0 & 0 \\ 0 & \sqrt{2}/\overline{\|\boldsymbol{u}''\|} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.25}$$

After we have transformed both $\{\boldsymbol{X}\}$ and $\{\boldsymbol{u}\}$ to a common reference frame, we can employ standard techniques to estimate the 2D-to-2D similarity transformation H that minimizes the geometric error $d(H\boldsymbol{u}''', \boldsymbol{x}''')$. After this minimization, we can concatenate all matrices to find our final 2D-to-3D similarity transformation.

$$M = (S_{3D} P_{\boldsymbol{XY}} T_{3D})^{-1} H S_{2D} T_{2D} S_{res} \tag{4.26}$$

As such, all the obtained calibration information is compressed in a single matrix M, allowing for the direct transformation of 2D screen pixels into 3D locations by a simple matrix multiplication: $\boldsymbol{X} = M\boldsymbol{u}$.

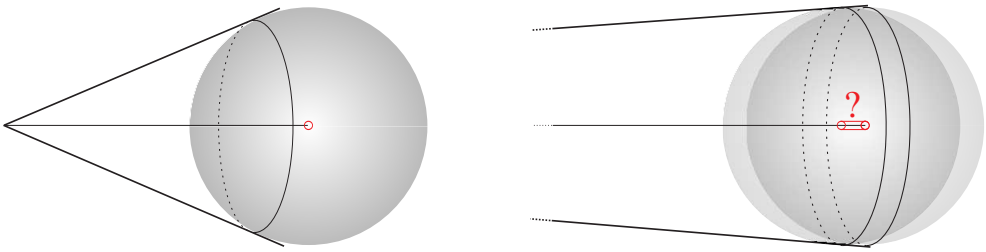


Figure 4.15: Illustration of a weak perspective projection due to a large camera-sphere distance. The sphere location is robustly detected if the sphere is close to the camera (left) whilst the precision decreases as the camera-sphere distance increases (right).

4.4 Results

Every phase of the presented calibration pipeline will be subjected to a short error analysis, based on synthetic data rendered using the POV-Ray [POV-Ray 09] ray tracer and / or real input images. These synthetic data sets are provided with exact camera, screen and sphere positions, making it possible to calculate the geometric error between the measurements and the exact data.

4.4.1 Locating the Spherical Mirror

During our first set of experiments, we used a data set of 25 synthetic images, displaying a 50 millimeter sphere at varying positions. In all cases, our observations agreed with the predicted error values of the sphere: an increase in sphere depth lead to a similar increase in error. This is to be expected, as an increasing camera and sphere distance weakens the perspective nature of the projection by the camera. A graphical representation of this problem is given in figure 4.15.

In addition we checked a data set of real-world images to verify the robustness of the ellipse detection algorithm used in this phase. An example of such a detection is shown in figure 4.16.

Figure 4.16: Locating the spherical mirror: robust ellipse detection.

4.4.2 Locating the Screen

Edges

During the second set of experiments, we used four data sets of 10 synthetic images, using the scale defined in the previous experiments (by the 50mm sphere). Figure 4.17 shows the geometric error of our estimated world coordinates versus the number of frames used to calculate this estimate. The geometric error is defined as the sum of the distances between the ground truth and reconstructed corners. Several things are apparent from the displayed results:

- It is recommended to use more than the minimum of two frames to perform the calibration in order to avoid unnecessary errors.

- After a certain number of images are inserted into the pipeline (in our example 8 or 9), the quality of our estimate seems to converge. After this point has been reached, adding new images does not seem to improve the quality of our results.

- As can be seen by the error plot of data set 2, it is possible to achieve good results using very few images. This implies that an intelligent choice of sphere positions (e.g. close to the camera, respecting a minimum distance w.r.t. each other) may facilitate convergence.

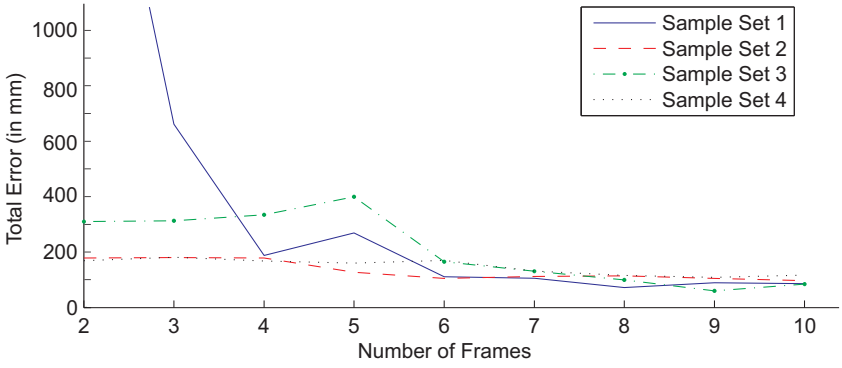


Figure 4.17: Geometric error plot. The sum of the distances between the ground truth and reconstructed corners is plotted as function of the number of frames used for calibration. Four random subsets of samples were used.

Full Surface

Initially, we will determine the relation between the number of pattern refinements and the error of the produced calibration. Afterwards, we will analyze the influence of the sphere location on the quality of the results. Results from both real-world and virtual data sets are provided, and are compared to the previous method.

In order to determine the correctness of the estimated screen positions, we perform a ground truth evaluation. The accuracy of the produced solution is measured by the sum of squared distances between the estimated screen corners and their corresponding ground truth values.

Pattern Refinements The accuracy of our method is largely dependent on the number of estimated ray-ray intersections. As the number of pattern refinements is directly related to the number of possible intersection points, an increase in pattern refinements will result in a higher accuracy. This is illustrated in figure 4.18, where we have plotted the error in function of the pattern refinement level. At a certain point in the function, the extra refinements will no longer yield any improvements. On the contrary, the results can even start to deteriorate as the least significant bits of the detected code are no longer distinguishable and are thus “randomly” assigned, lowering the chance to find exactly this code in other images.

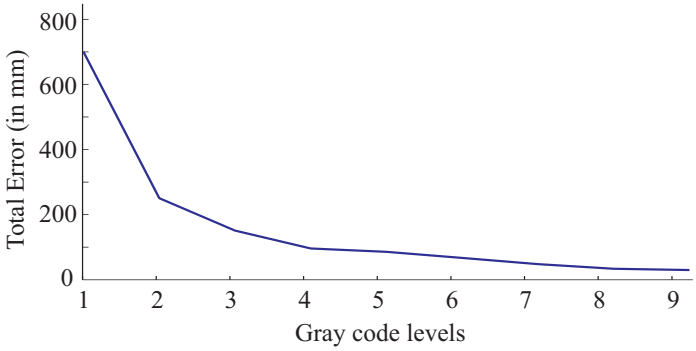


Figure 4.18: Error in function of the maximum applied pattern refinement level.

Sphere Placement The quality of our results not only depends on the chosen pattern refinement level, but also on the number of sphere displacements as well as the positioning of the sphere. From our experiments, we have concluded that only two sphere locations are required, as this case produces the lowest error values. In contrast to previous methods, the error does not decrease and converge when adding extra, less optimal, sphere locations. As can be seen in table 4.2, where we compare the error values of two sphere locations in terms of their distance to the camera, the sphere locations closest to the camera produce the best results. Additionally, in order to maximize the accuracy of the sphere location detection algorithm, we place the center of the sphere on the principal axis of the camera. Finally, a minimum distance between two different locations is preserved, avoiding degenerate configurations.

Side-by-side Error Comparison

A side-by-side comparison between the proposed methods is also performed. As it turns out, the full surface method's error is approximately 7 to 8 times lower compared to the edge method, and we only require 2 different sphere locations instead of 6 or 7. However, the reduced number of manual sphere displacements is at the cost of requiring almost 5 times as many input images as the previous methods.

Since no ground truth is available for our real-world setup, we have measured the vertical and horizontal edges of the LCD screen. In order to verify the accuracy of our method for real-world data, our error metric consist of a simple comparison of the measured screen edge lengths and the distances between the estimated screen

Far														.901
												.856	.986	.860
											.704	.704		.826
												.729		.622
									.374	.487		.557		
								.435	.397	.470	.495	.551		
							.317	.346	.404	.434		.469	.545	.668
						.306	.320		.342	.384	.502	.503	.484	.549
					.245	.246	.278	.306	.374	.369	.434	.467	.555	.494
				.189		.232		.301	.278		.372	.429	.461	
↑			.156	.170	.179	.223	.240	.255	.304	.321	.340	.370	.427	
		.107	.123	.138	.162	.178	.194	.216	.241	.274	.274		.360	.366
Near	.075	.085	.099	.111	.130	.138	.163	.178	.202	.217	.224	.264	.341	.279
	Near	→												Far

Table 4.2: Error for different combinations of 2 out of 14 sphere locations. The spheres all lie on the principal axis on evenly spaced distances from the camera. Spheres near the camera yield the lowest error. Missing values are due to a low number of Gray code correspondences between both image sets, resulting in an insufficient number of estimated 3D screen pixels.

corners. For a screen of 474$\times$ 297 (measured in millimeters), we compute dimensions of 478.7 $\times$ 300.4, suggesting an accuracy up to a few millimeters. The edge method produces dimensions of 449.8 $\times$ 286.7, suggesting an accuracy up to a few centimeters.

4.5 Conclusions

We have presented two novel automatic methods for screen-camera calibration, based on the use of a single moving spherical mirror. A ground truth evaluation has shown that both algorithms can be performed within practical error bounds. The edge method requires a low number of input images but a considerable amount of sphere movements. The full surface method requires a higher number of input images, but only a single sphere displacement (or two different sphere locations). The overall error is lower for the latter technique, because of the increased number of data samples.

As cameras are often focussed on the scene in front of the setup, and not on the reflected screen, we currently need to refocus the camera. This is not desirable since we then alter the internal camera parameters implying a recalibration. Therefore, we are now looking for alternative pattern sequences which are more insensitive to blur due to an out of focus camera.

In the near future, we will also compare the use of Gray code patterns to other codification methods in order to limit the number of required recordings. Specifically, we expect the use of gradient patterns [Ma 07] can reduce the number of input images from 30-35 to less than 10, reducing the calibration time even more.

Part II

Relief and Glossiness Acquisition

Chapter 5

Relief Acquisition

Contents

5.1 Introduction

There exist a wide variety of methods for scanning 3D geometry. Most of these methods focus on acquiring the global shape of an object, but ignore small-scale details such as texture or skin. However, if we want to convincingly reproduce real-world objects, this so-called *mesostructure level* cannot be ignored. In this chapter we present techniques for scanning mesostructures, using inexpensive off-the-shelf hardware. Each technique will be most suited for a specific level of glossiness, ranging from purely diffuse to highly specular materials.

Our methods produce a normal map of the scanned surface. Normals can be transformed into the original 3D shape of the surface [Frankot 88], or they can be combined with the output of a global shape acquisition method [Nehab 05]. In computer graphics, normals can be added as texture maps to enrich 3D models with relief, and rendered directly using graphics hardware.

Currently, high quality methods for accurately acquiring normal maps exist, but these methods often require a delicate and/or expensive setup [Wang 06, Ma 07, Johnson 09]. Alternatively, there are also easy to use methods demanding no specialized hardware, but these typically yield low(er) quality scans [Rushmeier 97] or they are time-consuming procedures when high quality is needed [Chen 06, Holroyd 08]. Hence, there is still need for a method that is *simple and fast*, yet yields *high quality* results. The word *simple* in this context means not only easily executable but also easily implementable. This is due to the fact that no special hardware is involved and presented algorithms can easily be transformed into practical program code.

5.2 Related Work

A large body of work deals with recovering shape of real-world objects. We distinguish two trends in computer vision literature. First, stereo matching algorithms recover depth maps using triangulation, by observing a scene from two (or more) views [Scharstein 02]. Second, photometric stereo [Woodham 80] computes surface normals from a sequence of illumination directions, while observing the scene from a single viewpoint. Our technique can be classified in the latter category.

Geometry acquisition methods usually assume that the observed materials are perfectly diffuse. However, techniques have been developed that focus on more

"difficult" materials, in particular specular materials. The effect of weak specular reflection can be filtered out in order to apply techniques that assume a diffuse material. This can be realized with cross-polarization [Wolff 89a, Wolff 89b, Nayar 97, Umeyama 04, Ma 07, Chen 07], color transformations [Schlüns 93, Mallick 05] or controlled illumination [Lamond 09]. More general photometric stereo techniques can deal with arbitrary BRDFs [Zickler 02, Goldman 05, Hertzmann 05, Wu 05]. Highly specular surfaces are particularly challenging, so specialized techniques have been developed [Ikeuchi 81]. These techniques usually focus on recovering global shape [Sanderson 88, Nayar 90, Halstead 96, Zheng 00, Tarini 05, Bonfort 06, Mallick 05, Miyazaki 07, Adato 07, Kutulakos 08]. Some of these techniques analyze reflections of computer-generated patterns [Tarini 05, Bonfort 06], but require that the surface is a perfect mirror. Local surface orientation can also be analyzed from specular highlights [Sanderson 88, Healey 88, Chen 06, Holroyd 08], and this does not necessarily require that the surface is an ideal reflector. As specular highlights are barely affected by subsurface light transport, this makes them an excellent cue to recover transparent, translucent and low albedo surfaces. In this dissertation we propose three techniques, where each technique performs optimally for either Lambertian, glossy or highly specular surfaces.

Techniques have also been introduced specifically for recovering small-scale surface details, in the form of relief (height) maps or normal maps, assuming various types of materials. Rushmeier et al. [Rushmeier 97] acquired normal maps from Lambertian surfaces using photometric stereo [Woodham 80]. Hernandez et al. [Hernández 07] apply a multispectral photometric stereo approach for scanning detailed shape bends and wrinkles of deforming, homogeneous, surfaces. Yu and Chang [Yu 05] obtained relief maps by analyzing cast shadows. Chung and Jia [Chung 08] acquire surface normals of glossy surfaces by employing cast shadows as an additional cue to optimally recover BRDF parameters. Malzbender et al. [Malzbender 01] presented an alternative technique to obtain polynomial texture maps (PTMs) of non-specular surfaces, which can be converted to normal maps. The method employs a specialized setup which captures about 50 images under different illumination. Another interesting scanning method is presented by Han and Perlin [Han 03] and uses a kaleidoscope to generate views from multiple directions to obtain a bidirectional texture function (BTF). BTFs can directly be transformed into a relief representation [Neubeck 05]. Paterson et al. [Paterson 05] use a digital still camera and a mounted flash to scan partly specular and diffuse mesostructures by

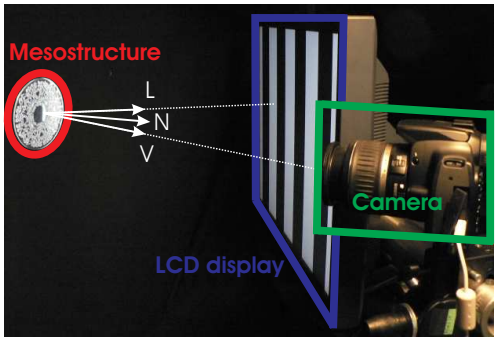


Figure 5.1: Our setup consists of a digital reflex camera, LCD screen and mesostructure.

interpreting the reflected flash light. Morris and Kutulakos [Morris 07] reconstructed exterior surfaces of inhomogeneous transparent scenes by capturing a large set of images from one or more viewpoints while moving a proximal light source. Wang and Dana [Wang 06] obtain relief textures from specularities using a specialized hardware setup consisting of a camera, a parabolic mirror, a beam splitter and an illumination source with corresponding lens. A more practical technique is proposed by Chen et al. [Chen 06], which utilizes a manually moved light source and a fixed video camera. Unfortunately, many input images are needed when high angular resolution normal maps are required. A faster method is proposed by Ma et al. [Ma 07], where they use an extended gradient light source requiring only 4 to 8 captured images at the cost of having a more delicate and expensive setup. Their method utilizes a construction containing a spherical array of polarized light sources. Although traditional specular surface acquisition methods approximate the centroid of the reflection direction by taking the brightest reflection point, Holroyd et al. [Holroyd 08] do not make this assumption. Unfortunately, their method requires a considerable amount of recordings. In the recent work of Johnson and Adelson [Johnson 09], “difficult” materials are avoided by pressing the object surface against a painted transparent elastomer. This creates a temporary imprint of the surface, but with known BRDF.

In our work we propose a system suited for highly specular, glossy, and purely diffuse surfaces. These methods require only a very limited number of images and make use of inexpensive and off-the-shelf hardware components including a digital still camera and a computer screen.

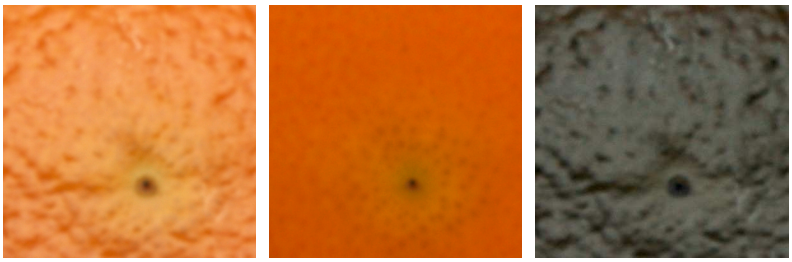


Figure 5.2: Diffuse and specular (left) separated into diffuse only (middle) and specular only (right) by the use of cross polarization.

Currently computer screens are more and more often employed as a controlled planar illuminant, for example for the purpose of environment matting [Zongker 99]. Recently screen-camera setups have also been introduced as a device for normal map acquisition [Tarini 05, Clark 06, Funk 07, Morris 07, Nehab 08]. Our work falls within this category, with a primary focus on acquisition efficiency.

5.3 Separating Diffuse from Specular Reflections

We scan mesostructures based on available specular or diffuse reflection information. However, typical materials are not purely specular or diffuse, but rather a combination of specular and diffuse components [Umeyama 04]. Therefore, we want to be able to separate diffuse from specular reflections, which is accomplished by exploiting light polarization.

The idea behind this is that the polarization direction of specularly reflected polarized light remains approximately unaltered, whereas the diffuse reflection randomizes this direction. Hence, by recording an object illuminated by the vertically polarized light of the LCD screen, after aligning the filter in front of the camera with the polarization, half of the diffuse component together with the specular reflections are captured. Rotating the filter 90 degrees to a perpendicular direction results in the capture of only the other half of the diffuse light, this time without the specular component. By subtracting the latter from the former, the specular reflections are easy to isolate (figure 5.2). More information on this topic can be found in the literature [Wolff 89a, Wolff 89b, Nayar 97, Umeyama 04, Iizuka 02].

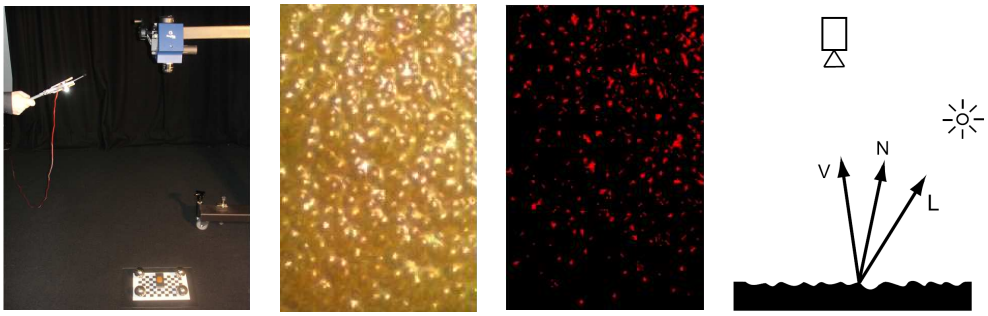


Figure 5.3: Overview mesostructure from specularity method. Image courtesy of Chen et al. [Chen 06]. Left to right: (1) surface illuminated by moving point light source. (2) Recorded image. (3) Detected specularities. (4) Recovered surface normal.

5.4 Gray Code Patterns

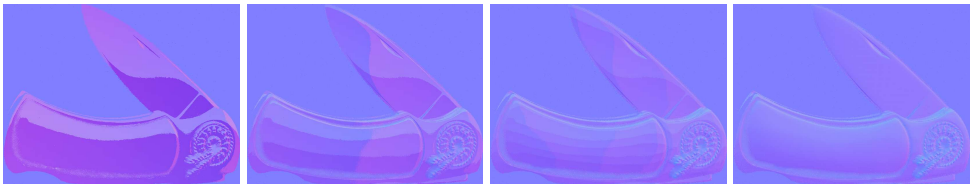
In this section, we detail our method for scanning predominantly specular surfaces from specularly reflected Gray code patterns.

5.4.1 Motivation

Inferring normals from specularities is fairly straightforward. An overview of Chen et al.'s [Chen 06] method to achieve this, is given in figure 5.3. First, the specular mesostructure is illuminated by a moving, bright, white, distant point light source, while being observed by the camera. Then, for each frame, the specular highlights are detected by applying a thresholding step. For each detected highlight, the surface normal $\vec{N}$ is reconstructed by computing the halfway vector between the viewing vector $\vec{V}$ and the light vector $\vec{L}$.

The accuracy of the solution will improve as more light directions are taken into account. We refer to the amount of different light directions as the *angular sampling rate*. Figure 5.4 shows the importance of having a high angular sampling rate. Many samples are needed to recover subtle surface details. When using a low sampling rate, certain normals will be missed (which show up as quantization artifacts).

We attain a high angular sampling rate by using more than one light at a time (figure 5.5). As will be explained in section 5.4.3, our technique samples n different light



(a) 4 dirs, 3 images (b) 16 dirs, 5 images (c) 64 dirs, 7 images (d) 4096 dirs, 13 images

Figure 5.4: Normal map reconstruction quality depends on the number of sampled light source directions, or angular sampling rate.

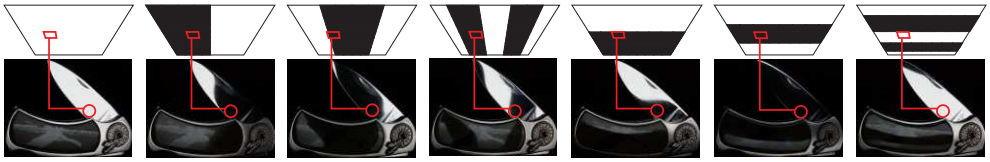


Figure 5.5: Top: floodlit image and Gray code patterns; bottom: corresponding images. Each location on the monitor (red square) is directly reflected (red circle). By decoding the Gray code patterns, we can recover the illumination direction, and consequently, the surface normal.

source directions, with $O(\log_2 n)$ images. In contrast, Chen et al.'s mesostructure-from-specularity technique requires n images for n directions [Chen 06]. In order to reach a sufficiently high sampling rate, Chen et al. sample continuously using a video camera. However, video often suffers from poor image quality. Because our method requires a relatively low number of images, it becomes practical to employ a digital still camera. Therefore, the quality of the results improves significantly, as still cameras have better optics and larger sensors, and can handle longer exposure times.

5.4.2 Recovering Normal Maps from Specularities

Our goal is to find a normal map based on specularities [Wang 06, Chen 06, Ma 07]. To this end, we assume that the surface is an ideal reflector. If we know that illumination arrives from direction $\vec{L}$ and is reflected in direction $\vec{V}$, the normal must be:

$$\vec{N} = \frac{1}{\|\vec{L}+\vec{V}\|}(\vec{L}+\vec{V}), \tag{5.1}$$

Subsequently, for each specularity in the image, we use equation 5.1 to infer the corresponding normal, as done by Chen et al. [Chen 06]. Therefore we reasonably assume the mesostructure to be approximately planar and furthermore require a geometrically calibrated screen-camera-mesostructure setup (see chapter 4).

Equation 5.1 is only an approximation because a surface is never perfectly flat (at the microscopic level). Consequently, light is usually scattered in a small cone around the ideal reflection direction, and highlights become less clearly defined. For instance, glossy materials like plastic exhibit such behavior. We will discuss how this affects our results in section 5.4.5. For the sake of simplicity, we also ignore specular interreflection.

5.4.3 Efficient Acquisition

A high angular sampling rate is required in order to recover all possible normal orientations (see figure 5.4), which we obtain by sampling more than one light direction per image. We therefore require an array of light sources, where every source can be switched on/off individually. This array could be a computer monitor [Zongker 99, Tarini 05, Clark 06, Bonfort 06, Funk 07, Nehab 08], a (hemi)spherical rig fitted with lights [Sanderson 88, Malzbender 01, Ma 07], or even an unstructured set of controllable lights . In our implementation, we employ an LCD monitor (figure 5.1), which provides us with a high-resolution, regularly-spaced grid of light sources.

Under the assumption of ideal reflection described in the previous section, we know that a specularity can only be caused by exactly one light direction, which in turn corresponds to a location on the LCD monitor. If this location is known, we can simply compute the local normal using equation 5.1. A naive solution would be to iterate through all lights, enabling them one at a time, and observing which pixels contain highlights [Chen 06]. Unfortunately, this becomes impractical for thousands of light sources.

Instead, we identify monitor locations using Gray code patterns as described in chapter 4. The only difference is that the codes are now detected using cross polarization instead of complement patterns, as many mesostructure surfaces will also have a diffuse component that has to be removed. For an $m \times n$ resolution encoding,

$2\,(\log_2(m)+\log_2(n)+1)$ patterns are required, $\log_2(m)+\log_2(n)+1$ for the combination of a specular and diffuse component, and $\log_2(m)+\log_2(n)+1$ for the diffuse only component. Notice that an additional floodlit pattern ($\ldots+1$) is employed to enable each light source at least once. Notice also that we can reduce the number of recordings by a factor of 2 by not recording the diffuse only images in case the material to be scanned is perfectly specular.

5.4.4 Specularity Detection

If the surface has a negligible albedo and is homogeneous, specularities can easily be detected using global thresholding [Umeyama 04, Chen 06]. However, these assumptions do not hold in practice.

Heterogeneous objects reflect light depending on the spatial location, which makes it hard to determine a global threshold that can be applied to all pixels. We therefore normalize a pixel's intensity w.r.t. the maximal intensity value at that pixel's location. This maximum can be found by emitting an additional floodlit pattern (all lights enabled). After normalization, we can simply apply global thresholding like before. See figure 5.11 for examples of heterogeneous surfaces ("graphics card" and "wallet").

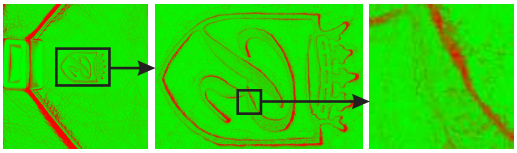
Even with normalization, specularity detection can be very sensitive to the chosen threshold, in particular for high albedo materials such as human skin (figure 5.10). We therefore isolate the specular component in the input images using cross polarization as described in section 5.3.

5.4.5 Limitations

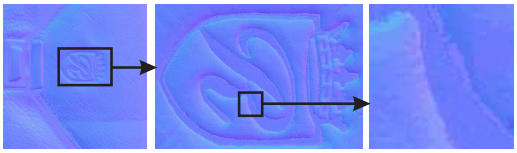
There are two factors that might have a negative influence on the reconstruction results, namely occlusion and glossiness. This section describes how they affect our results. Note that these issues also occur with previous mesostructure-from-specularity methods [Wang 06, Chen 06, Ma 07].

Occlusion

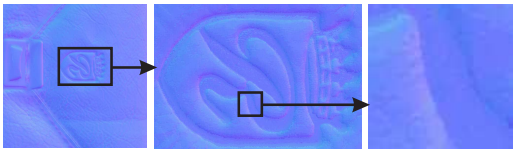
Self-shadowing occurs in deep grooves and pits, which produces meaningless information. However, we can easily detect these regions using a floodlit pattern (fig-



(a) Confidence map (green: high conf, red: low conf).



(b) Original normal map.



(c) Normal map with filled gaps.

Figure 5.6: Uncertain regions are filled in by interpolating normals from the nearest valid neighbors.

ure 5.5). We assign a confidence value to each pixel, chosen proportional to a pixel's intensity, and fill low confidence areas by interpolating normals from the nearest valid neighbors (figure 5.6).

Glossiness

Our method becomes less accurate for glossier (or rougher) materials. This is visually demonstrated in figure 5.7 and verified on a ground truth comparison in figure 5.9. For the latter figure, we generated images of a mesostructure using a photorealistic renderer (PBRT, [Pharr 04]) under our Gray coded illumination patterns. We employed a physically-based reflectance model [Torrance 67], which allows for controlling the glossiness. It should be noted that the error increases for more gloss. Including interpolated normals of less confident pixels tends to add an overall increase to the numerical error. Fortunately, the interpolation also tends to produce smooth results, and thus does not introduce visually distracting artifacts.

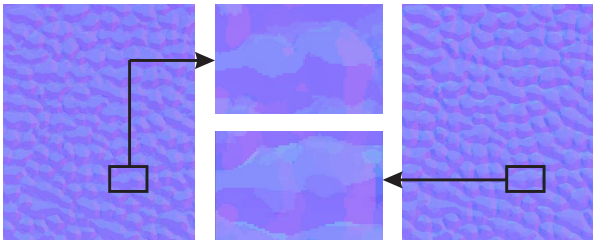


Figure 5.7: Comparison between results from a specular and glossy surface. The left image shows the result of capturing a glossy plastic surface. In the right image, the same surface was acquired after having applied a fine layer of oil in order to increase specularity. If the surface is more specular, finer normal variations are detected, whereas the glossy version looks more quantized.

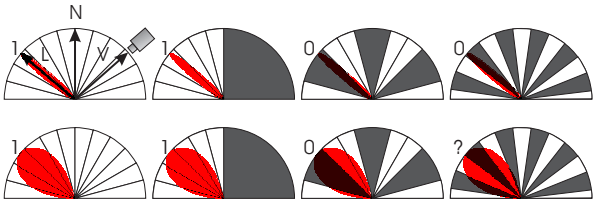


Figure 5.8: Illustration of precision issues associated with glossy materials. A specular BRDF directly picks up the Gray code patterns, whereas glossy BRDFs blurs the patterns. For fine patterns, this convolution makes specularity detection ambiguous, which means that no more information can be gained after a certain refinement level.

The reason why glossier materials result in less accurate normal maps is illustrated in figure 5.8. It shows how Gray codes are interpreted for a very specular material (narrow specular lobe) and for a more glossy object (broader lobe). As the Gray code patterns become more refined, it becomes harder to distinguish between the black and white parts of the pattern, which limits the attainable angular sampling rate. We therefore stop refinement when the sum of absolute differences (SAD) between consecutive refinements drops below a threshold. Even if a material is glossy, we can still infer a reasonable normal map, albeit with less angular detail (figure 5.7,figure 5.9).

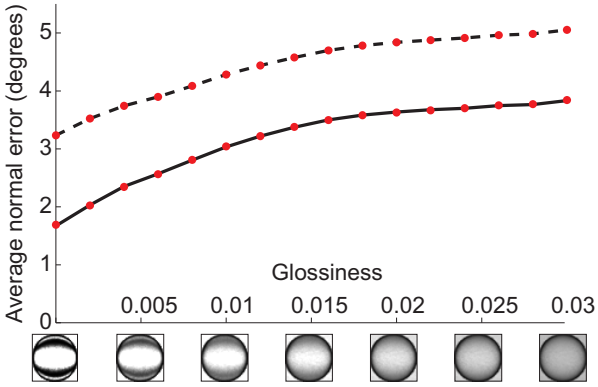
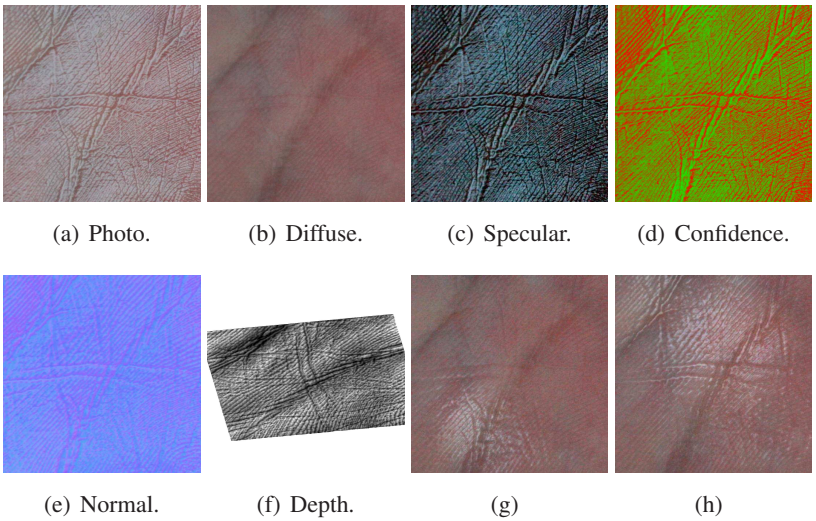


Figure 5.9: Error analysis for glossy materials on a synthetic example. The average angular error of our reconstruction is plotted in function of the glossiness. The solid curve shows the error measured on confident pixels only. The dashed curve shows the error when taking into account all pixels, including interpolated ones.

5.4.6 Results

Our results were generated with a regular 1280×1024 LCD screen, a digital SLR camera (Canon EOS 400D with default EF-S 18-55mm, f/3.5-5.6 kit lens) and a linear polarizing filter. The camera, screen and mesostructure are calibrated with respect to each other. The screen and camera are facing a similar direction, and hence are geometrically calibrated using a (spherical) mirror to make the screen visible to the camera as described in chapter 4. The mesostructure is attached to a supporting surface with position and orientation determined using a calibration pattern [Bouguet 06].

Assuming the surface is in perfect focus, and depending on the mesostructure's placement, we have a spatial resolution of approximately 40 normal pixels per millimeter. The largest number of patterns we used in our setup is 15 (one floodlit, 7 horizontal, 7 vertical). The angular resolution then is approximately 1.12 normals per degree. The recoverable normal angles range from -22.5 to 22.5 degrees w.r.t. the supporting surface's normal. Using a larger screen, or multiple screens, can improve this range. Even though the normal range might seem limited, it suffices for the mesostructures we scanned. We always obtained a visually plausible reconstruction. Moreover, reflected light from grazing angles is often blocked anyway due to self-occlusions.



(a) Photo. (b) Diffuse. (c) Specular. (d) Confidence.

(e) Normal. (f) Depth. (g) (h)

Figure 5.10: Acquired maps from oiled human palm skin (massage oil). (a) photo, (b) diffuse map, (c) specular map, (d) confidence map (green: high confidence, red: low confidence), (e) normal map, (f) depth map and (g,h) renderings.

material	#directions	#images	polarization
graphics card	256	18	yes
metal coaster	1024	11	no
wallet	1024	11	no
rough glass	16384	30	yes

Table 5.1: Number of images used in figure 5.11, as function of the number of sampled light directions. Polarization is used for high albedo surfaces, which doubles number of images (accounted for in **#images**-column).

Figure 5.10 shows the different maps that we obtain from an acquired surface. In the first place, our method returns a normal map (with confidence), which can be turned into a depth map [Frankot 88]. As a byproduct of the polarization-based separation discussed in section 5.4.4, we also infer a diffuse and specular map, which were computed from the floodlit illumination pattern. These maps can easily be used as textures in graphics applications, and rendered in real-time using graphics hardware. In figure 5.11 we show different normal maps and renderings, for a variety of materials: metal (“coaster”, “wallet”, “graphics card”), plastic (“graphics card”),

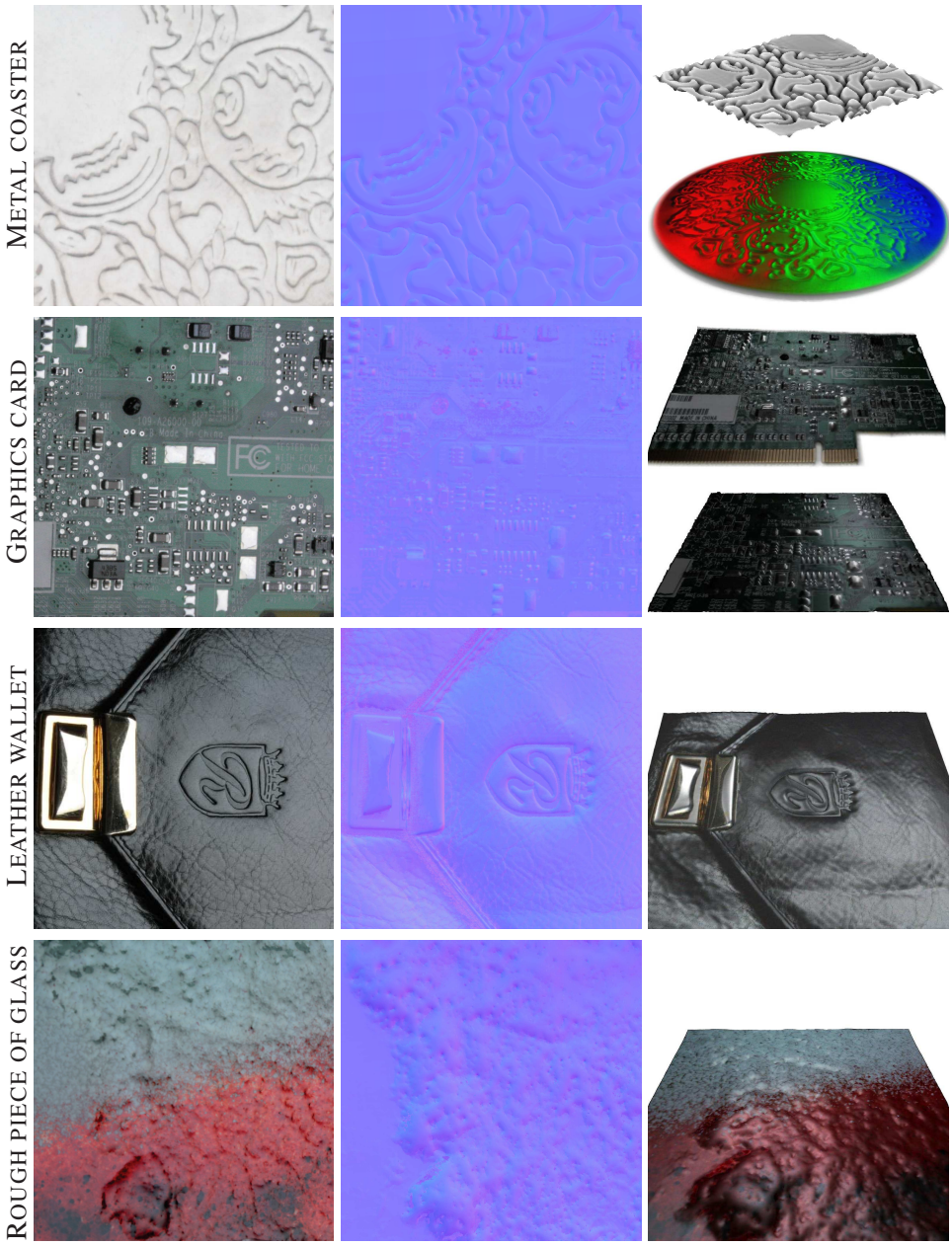


Figure 5.11: Results. Left: input images, middle: normal maps, right: renderings.

semi-transparent (“rough glass”), organic (“leather”) and heterogeneous (“graphics card”, “wallet”) materials. For each of the recovered surfaces, the angular sampling rate and number of acquired images, are given in table 5.1.

5.5 Gradient Patterns

The next two methods also produce normal maps of scanned surfaces, using an LCD screen as illuminant and a digital camera to capture the reflections (figure 5.12). However gradient patterns are used instead of Gray code patterns. Because of the structure of the patterns, gradients can be employed to infer normals from specular as well as diffuse reflections. First we will focus on diffuse materials, and later specular materials will be investigated.

5.5.1 LCD Screen as a Gradient Illuminant

Before we can start acquiring data, we are required to know the exact location of the setup’s three essential elements (display, camera, and object) as well as to have accurate control over the amount of light emitted by our illuminant. Techniques to achieve this are given in chapter 4.

The primary component we need is a set of patterns from which we can derive the scanned surface normal maps. Once the necessary parameters describing the setup configuration are found, the set of gradient patterns can be defined as they strongly depend on the object’s placement with respect to the display. For the sake of simplicity, we assume the object center lies on the perpendicular axis through the screen center.

We employ four different patterns (figure 5.13(a-d)): three patterns represent gradients in the three dimensions, and one fully lit pattern is used to compensate for the inability to emit ’negative’ light intensities. We can think of these patterns as windows to globes around the scanned object, covered with a linear gradient pattern in one of the three dimensions (figure 5.14). This is achieved by projective mapping of the gradient spheres onto the window plane (which is centered around the y-axis), producing the gradients shown in figure 5.14 (a). However, should we make use of the gradients in this form, we would not be making use of the complete intensity spectrum available on our LCD screen. Therefore we transform the patterns in a more

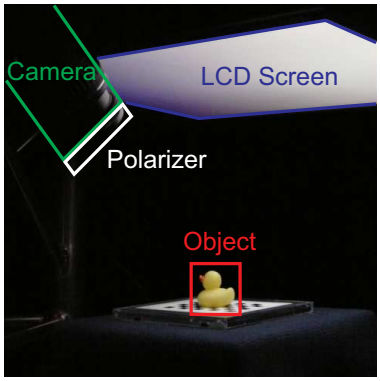


Figure 5.12: Our setup, consisting of an LCD screen and a camera with a linear polarizer filter. The scanned object is located on a planar pattern.

appropriate form, depicted in figure 5.14 (b). So after applying this intensity transformations, using the symbols shown in figure 5.14, we can describe our patterns by the following equations:

$$P_x(\vec{\omega}) = \frac{1}{2}\left(\frac{\omega_x}{\sin(\sigma_w)}+1\right) \tag{5.2}$$

$$P_y(\vec{\omega}) = \frac{\omega_y-\cos(\sigma_w)\cos(\sigma_h)}{1-\cos(\sigma_w)\cos(\sigma_h)} \tag{5.3}$$

$$P_z(\vec{\omega}) = \frac{1}{2}\left(\frac{\omega_z}{\sin(\sigma_h)}+1\right) \tag{5.4}$$

$$P_c(\vec{\omega}) = 1 \tag{5.5}$$

where $\vec{\omega}=(\omega_x,\omega_y,\omega_z)$ represents the normalized incident illumination direction, and $(2\sigma_w,2\sigma_h)$ is the window's width and height expressed in radians.

5.5.2 Diffuse Reflection

In this section we will apply the previously defined four gradient patterns to obtain local surface orientation of *diffuse* surfaces. Therefore we assume diffuse only images are captured from the illuminated scene. In addition, we assume the size of the object is small compared to the object-display distance so we can approximate the object by a point, simplifying calculations.

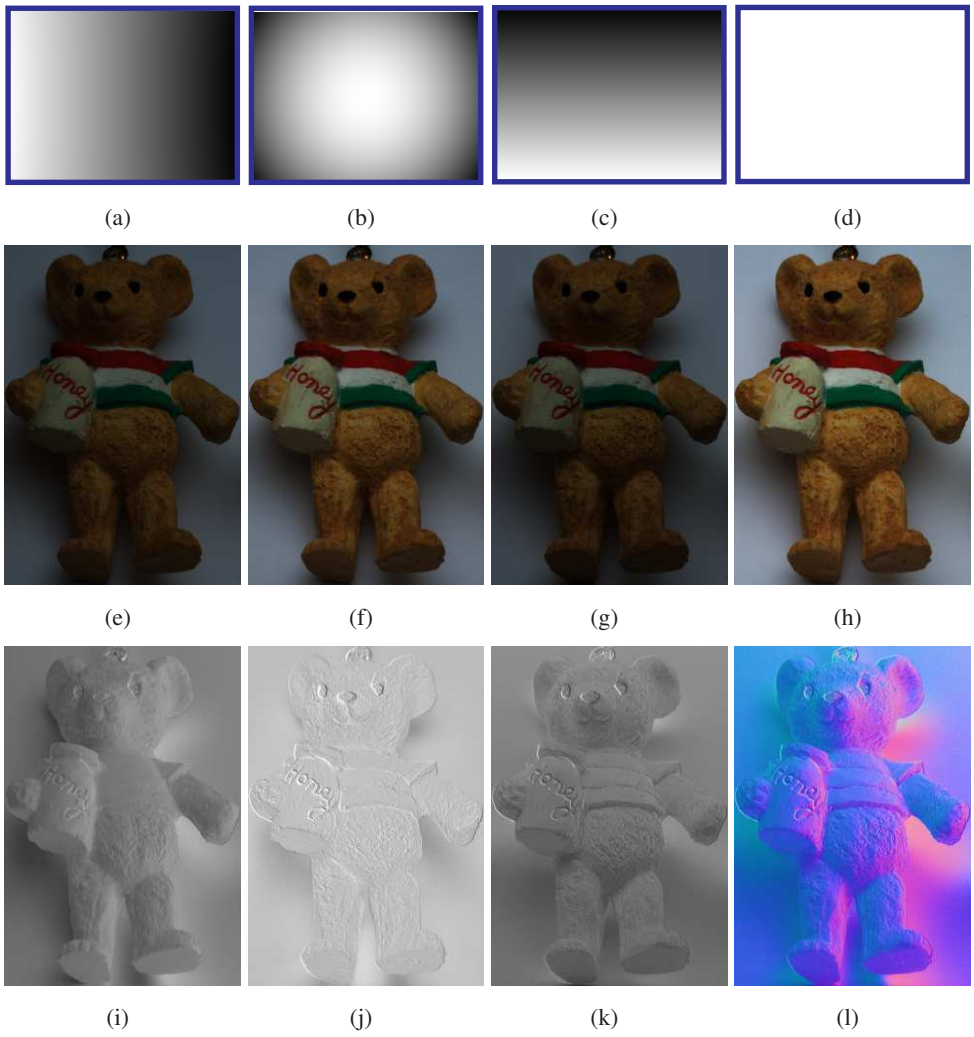


Figure 5.13: (a-d) The four illumination patterns P_i emitted by our LCD. (e-h) An object illuminated by these patterns. (i-k) Ratio images of (e-g) divided by the uniformly lit image (h). (l) Surface normal map estimate derived from (i-k) using rgb values to indicate surface normal coordinates.

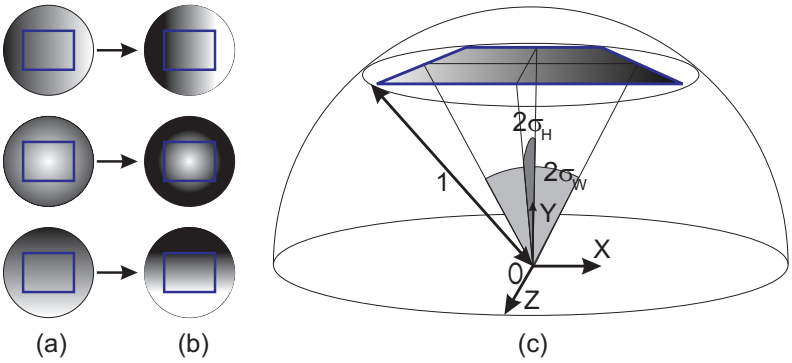


Figure 5.14: Spherical projection. (a) screen as a naive window to the virtual spherical area light source (top view). (b) scaled illumination patterns for better intensity distribution (top view). (c) schematic representation of the setup (perspective view).

The diffuse/Lambertian BRDF over incident illumination $\vec{\omega}$ and normal $\vec{n}$ is defined by the equation $R(\vec{\omega},\vec{n}) = \rho_d F(\vec{\omega},\vec{n})$, where $F(\vec{\omega},\vec{n})$ is the foreshortening factor $\max(\vec{\omega}\cdot\vec{n},0)$ and ρ_d is the diffuse albedo. The observed reflectance $L_i(\vec{v})$ from a view direction $\vec{v}$ under illumination pattern P_i is defined by the following equation:

$$L_i(\vec{v}) = \int_\Omega P_i(\vec{\omega}) R(\vec{\omega},\vec{n})\, d\vec{\omega} \tag{5.6}$$

where Ω is the set of all possible incident illumination vectors defined by the window $(2\sigma_w, 2\sigma_h)$. Considering our constant illumination pattern, equation (5.6) simplifies to:

$$L_c(\vec{v}) = \int_\Omega R(\vec{\omega},\vec{n})\, d\vec{\omega} \tag{5.7}$$

The observed reflectance of the gradient patterns can thus be defined in terms of observed illumination using the constant pattern:

$$\begin{aligned} L_x(\vec{v}) &= \tfrac{\rho_d}{2\sin(\sigma_w)} \textstyle\int_\Omega \omega_x F(\vec{\omega},\vec{n})\, d\vec{\omega} + \tfrac{L_c(\vec{v})}{2} \\ L_z(\vec{v}) &= \tfrac{\rho_d}{2\sin(\sigma_h)} \textstyle\int_\Omega \omega_z F(\vec{\omega},\vec{n})\, d\vec{\omega} + \tfrac{L_c(\vec{v})}{2} \\ L_y(\vec{v}) &= I_y + \left(\tfrac{\cos(\sigma_w)\cos(\sigma_h)}{\cos(\sigma_w)\cos(\sigma_h)-1}\right) L_c(\vec{v}) \end{aligned} \tag{5.8}$$

where I_y represents the corresponding integral derived from equation (5.3). In order to properly expand the integral expressions I_i, we transform the incident illumi-

nation vectors $\vec{\omega}$ to their spherical coordinates $(\theta,\phi) \in [\frac{\pi}{2}-\sigma_w, \frac{\pi}{2}+\sigma_w] \times [\frac{\pi}{2}-\sigma_h, \frac{\pi}{2}+\sigma_h]$. The integral expressions I_i can now be expanded to:

$$\begin{aligned} I_x &= \frac{\sin(\sigma_h)(\cos(\sigma_h)^2+2)(2\sigma_w-\sin(2\sigma_w))}{6\sin(\sigma_w)}\rho_d n_x \\ I_y &= \frac{\sin(\sigma_h)(cos(\sigma_h)^2+2)(2\sigma_w+\sin(2\sigma_w))}{3-3\cos(\sigma_w)\cos(\sigma_h)}\rho_d n_y \\ I_z &= \frac{3\sin(\sigma_h)^2\sigma_w}{2}\rho_d n_z \end{aligned} \tag{5.9}$$

Combining equations (5.8) and (5.9), we can recover the surface normal $\vec{n} = (n_x, n_y, n_z)$ up to an unknown scale factor (the inverse diffuse albedo $\frac{1}{\rho_d}$), which can be removed by normalization:

$$\begin{aligned} \rho_d n_x &= c_x\left(L_x(\vec{v}) - \frac{1}{2}L_c(\vec{v})\right) \\ \rho_d n_y &= c_y\left(L_y(\vec{v}) - \frac{\cos(\sigma_w)\cos(\sigma_h)}{\cos(\sigma_w)\cos(\sigma_h)-1}L_c(\vec{v})\right) \\ \rho_d n_z &= c_z\left(L_z(\vec{v}) - \frac{1}{2}L_c(\vec{v})\right) \end{aligned} \tag{5.10}$$

The constants c_x, c_y and c_z are only dependent on the known calibration parameters σ_w and σ_h, so they need to be computed only once. A schematic overview of the normal and albedo map recovering procedure is depicted in figure 5.15. For each pixel, the desired information is determined by simply linearly combining the input images. As a result, our method requires very few operations, making it very easy to implement.

Discussion

We have created an experimental setup to verify the results on several real-world examples. The setup consists of a 19 inch LCD screen, a digital reflex camera (Canon EOS 400D), and a linear polarizer filter. Their relative positioning is depicted in figure 5.12. In figure 5.16 we show examples of captured images of Lambertian surfaces, the associated normal maps and novel synthesized views under different lighting circumstances and viewpoints.

When dealing with diffuse or slightly specular objects, our technique produces convincing results. However, there are two cases where our technique encounters

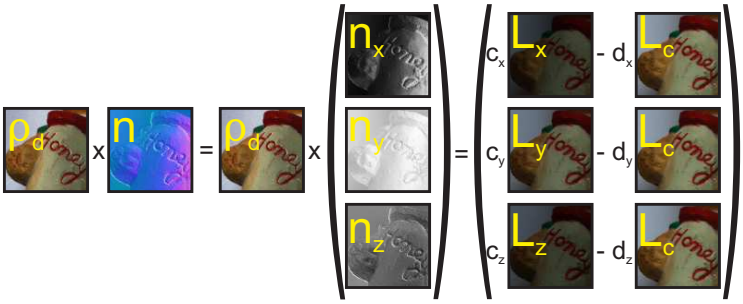


Figure 5.15: The albedo ρ_d and normal information n is determined by linearly combining the input images. The factors c and d are known setup parameters.

difficulties. The first case occurs when we are dealing with highly specular materials with a low albedo. When this occurs, there simply is not enough reflected diffuse light to compute an accurate normal, which results in noisy normals. This case is illustrated by figure 5.16(l) (the owl's eyes). The second case occurs when we are dealing with objects with a high amount of interreflections or self-shadowing. In our equations, we assume that all intensities emitted by our planar illuminant are received at every point of the scanned surface, and that all received intensities are from this source only. Interreflections or self-shadowing effects invalidate these assumptions, resulting in an erroneous region of the produced normal map. An instance of self-shadowing is illustrated by figure 5.16(n) (the bear's feet).

5.5.3 Specular Reflection

In this section we will explain how gradient illumination patterns displayed on an LCD screen allow us to extract surface normal information of specular materials. Again, we therefore assume light polarization is exploited to separate diffuse from specular reflection, as only specular information is required by the method. Finally, we will describe the relevant implementation concepts.

Normal From Specularity

As explained in section 5.4.2, a surface normal can easily be reconstructed given a detected specularity together with the corresponding light source, camera, and mesostructure position. This technique is well-known [Chen 06, Sanderson 88] and extensions have been proposed to improve its efficiency using controllable ex-

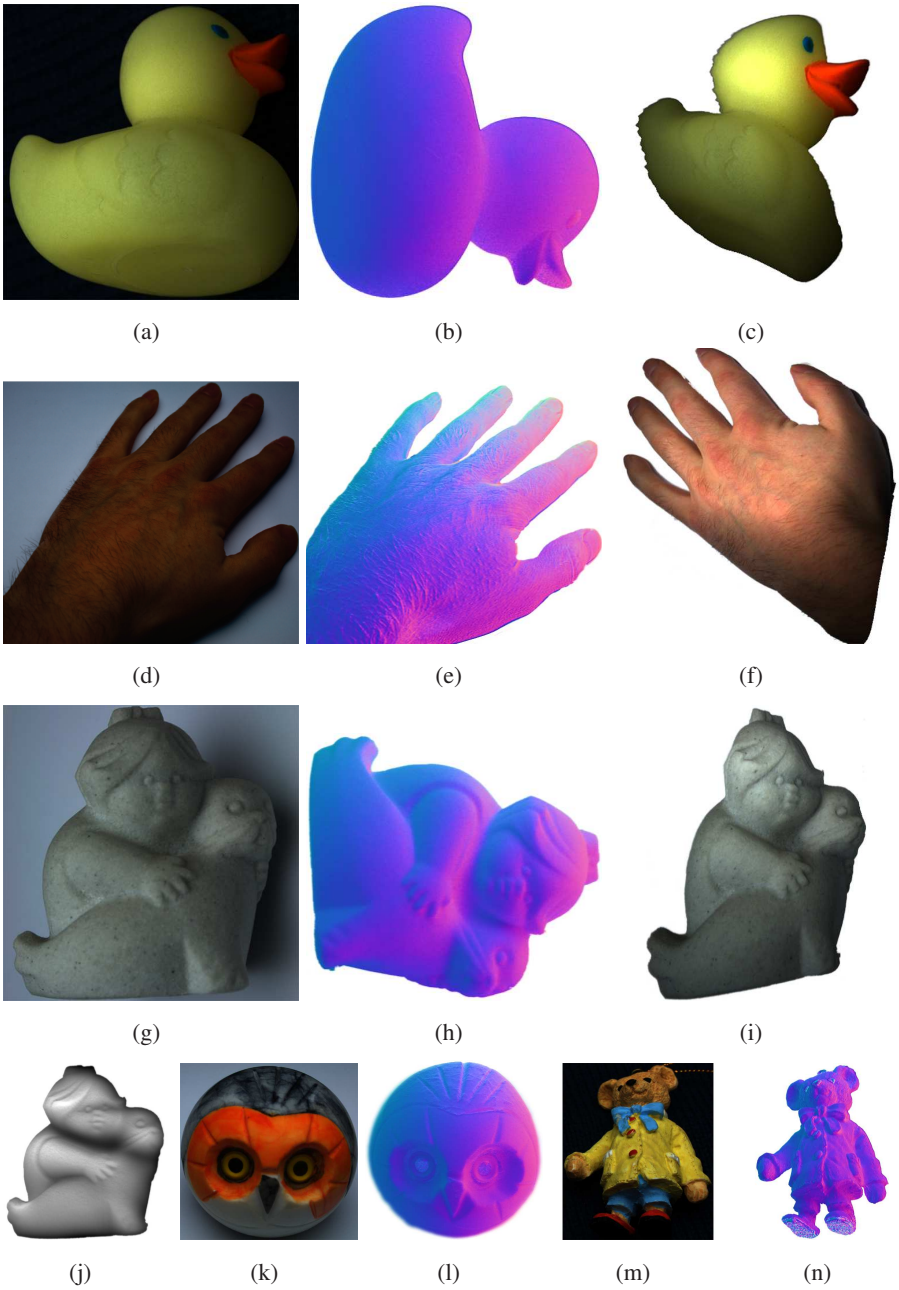


Figure 5.16: (a-i) Three examples of captured Lambertian images, their associated normal map, and the associated relighted and rotated surface. (j) Depth map. (k-n) Two examples of instances where our method can experience difficulties.

tended light sources [Ma 07]. Our technique is most similar to the method of Ma et al. [Ma 07], but requires only a simple and cheap setup instead. Additionally, we require less image captures.

Reflection Direction from Gradients

In this section we will explain how we efficiently detect the specular reflection direction for each pixel. The main idea is that a vertical and horizontal gradient pattern are displayed on the screen, and that we find out to which pixel (region) the reflection vector points to "pick" that intensity. We will derive the resulting formulas from this modification to the work of Ma et al. [Ma 07], making the technique applicable to a screen-camera setup.

In order to find for each pixel the reflection direction to extract a normal map, the object to be captured is illuminated by the three patterns P_i with $i \in \{x,z,c\}$. The reflected amount of light L_i in the viewing direction $\vec{v}$ is given by:

$$L_i(\vec{v}) = \int_\Omega P_i(\vec{\omega}) S(\vec{r},\vec{\omega}) F(\vec{\omega},\vec{n})\, d\vec{\omega} \tag{5.11}$$

where $\vec{r} = 2(\vec{r}\cdot\vec{v})\vec{n} - \vec{v}$. S represents the specular symmetric lobe, and $F(\vec{\omega},\vec{n}) = max(\vec{\omega}\cdot\vec{n},0)$ is the foreshortening factor. Notice that we assume that there is no interreflection, nor self-shadowing and that the recording is executed in a dark room without any stray light.

First we will execute a coordinate transform T to align $\vec{r}$ with $\vec{z} = [0,0,1]$ to facilitate further derivations. The transform T is computed by rotating the integration domain Ω to Ω'. The corresponding matrix is defined as $T = [\vec{s},\vec{t},\vec{r}]^t$ where $\vec{s},\vec{t}$ and $\vec{r}$ are orthogonal vectors with respect to each other.

The functions S and F are rotationally invariant because they only depend on the *angles* between the argument vectors. This yields $S(\vec{r},\vec{\omega}) = S(\vec{r}',\vec{\omega}')$ and $F(\vec{\omega},\vec{n}) = F(\vec{\omega}',\vec{n}')$ where the added apostrophes refer to the rotated versions of the vectors in the original integration domain. The pattern P_i is not invariant to rotation, so T has to be taken into account as follows: $P(\vec{\omega}) = P(T^{-1}\vec{\omega}') = P(T^t\vec{\omega}')$. As such, the equation is now written as:

$$L_i(\vec{v}) = \int_{\Omega'} P_i(T^t\vec{\omega}') S(\vec{z},\vec{\omega}') F(\vec{\omega}',\vec{n}')\, d\vec{\omega}' \tag{5.12}$$

For the remainder of this section we will assume a narrow specular lobe S. Therefore $F(\vec{\omega}', \vec{n}')$ can reasonably be assumed constant (c_F) in the small solid angle of S. This assumption breaks down in the case of grazing angles, but this is not a problem in practice because of the structure of our setup.

Further simplifications will depend on the illumination pattern P_i that is used. We derive L_x for the x-gradient, which is analogous to that of L_z. After filling in P_x, isolating some constants and splitting the integral, we obtain:

$$\begin{aligned} L_x(\vec{v}) &= \frac{c_F}{2\sin(\sigma_w)} \int_{\Omega'} (\omega_s' s_x + \omega_t' t_x + \omega_r' r_x) S(\vec{z}, \vec{\omega}')\, d\vec{\omega}' \\ &\quad + \frac{c_F}{2} \int_{\Omega'} S(\vec{z}, \vec{\omega}')\, d\vec{\omega}' \end{aligned} \tag{5.13}$$

Again by taking into account the narrowness of S, we can state that $(\omega_s' s_x + \omega_t' t_x) S(\vec{z}, \vec{\omega}') \approx 0$. This can easily be seen from the fact that $\vec{\omega}$ has almost to be aligned to have some response, meaning that r_x is almost equal to 1 and so s_x and t_x are negligible. Analogously the simplified integral $\int_{\Omega'} \omega_r' S(\vec{z}, \vec{\omega}')\, d\vec{\omega}'$ can be approximated by $\int_{\Omega'} S(\vec{z}, \vec{\omega}')\, d\vec{\omega}'$. After applying these steps to both the horizontal and vertical patterns we obtain the following straightforward equations:

$$L_x(\vec{v}) = \frac{L_c(\vec{v})}{2} \left(\frac{r_x}{\sin(\sigma_w)} + 1 \right) \tag{5.14}$$

$$L_z(\vec{v}) = \frac{L_c(\vec{v})}{2} \left(\frac{r_z}{\sin(\sigma_h)} + 1 \right) \tag{5.15}$$

Notice that L_c is the image taken under constant (or flood lit) illumination:

$$L_c(\vec{v}) = \int_{\Omega} P_c(\vec{\omega}) S(\vec{r}, \vec{\omega}) F(\vec{\omega}, \vec{n})\, d\vec{\omega}' \tag{5.16}$$

$$= c_F \int_{\Omega'} S(\vec{z}, \vec{\omega}')\, d\vec{\omega}' \tag{5.17}$$

This factor is necessary because of the structure of the patterns due to the impossibility of emitting negative light.

As all the required parameters for equation (5.14) and (5.15) are given except from r_x and r_z, and knowing that r is a normalized vector, the reflection direction r is found.

Implementation

In practice the proposed procedure allows for a straightforward and efficient implementation. For generating our results we have created a proof of concept implementation in Matlab as well as an optimized C++ version. The important concepts of the method are summarized in a five-step process:

1. Record object illuminated by P_x, P_y and P_c under **b**locking (L_x^b, L_z^b and L_c^b) and **n**on-**b**locking (L_x^{nb}, L_z^{nb} and L_c^{nb}) polarizing filter orientations

2. Determine specular images L_x, L_z and L_c:

$$L_x = L_x^{nb} - L_x^b \tag{5.18}$$
$$L_z = L_z^{nb} - L_z^b \tag{5.19}$$
$$L_c = L_c^{nb} - L_c^b \tag{5.20}$$

3. Determine ratio images $R_x = \frac{L_x}{L_c}$ and $R_z = \frac{L_z}{L_c}$

4. For each pixel, find $\vec{r}$:

$$r_x = \sin(\sigma_w)(2R_x - 1) \tag{5.21}$$
$$r_z = \sin(\sigma_h)(2R_z - 1) \tag{5.22}$$
$$r_y = \sqrt{1 - r_x^2 - r_z^2} \tag{5.23}$$

5. For each pixel, calculate the normal $\vec{n}$ (halfway vector between viewing direction $\vec{v}$ and reflection direction $\vec{r}$)

$$\vec{n} = \frac{\vec{r} + \vec{v}}{||\vec{r} + \vec{v}||} \tag{5.24}$$

The overview shows the simplicity of the method, where only six input photographs and a few simple image operations suffice to obtain a normal map. Notice that in practice for "pure" specular materials, even three image recordings could suffice because then the diffuse images will be black. A graphical overview of our algorithm is depicted in figure 5.17.

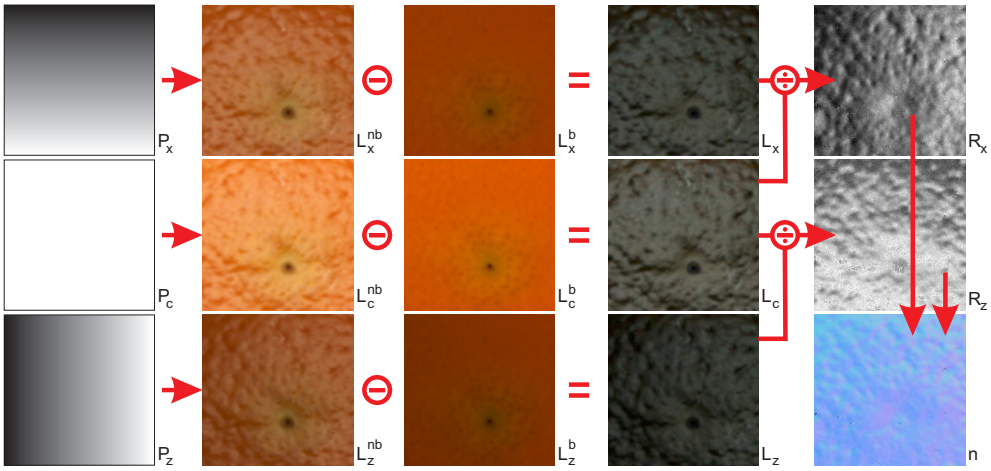


Figure 5.17: Procedure overview. From left to right, a piece of orange skin is illuminated by patterns P_i under both blocking and non-blocking polarization directions, yielding recordings L_i^b and L_i^{nb}. Then the specular images $L_i = L_i^{nb} - L_i^b$ are computed allowing for calculating the ratio images R_x and R_z. These are finally converted into the normal map n.

technique	# image captures	cheap/simple
[Holroyd 08]	> 200	yes
[Morris 07]	> 200	yes
[Wang 06]	> 200	no
[Chen 06]	$35 - 200$	yes
[Malzbender 01]	50	no
[Francken 08c]	$10 - 30$	yes
[Paterson 05]	$8 - 15$	yes
[Ma 07]	$4 - 8$	no
Our approach	$3 - 6$	yes
[Johnson 09]	1	no

Table 5.2: Comparison of mesostructure reconstruction methods. For each method is given how many input images that are approximately required, and if the scanning procedure can be executed easily/efficiently employing a simple setup.

Figure 5.18: A picture of a glossy plastic button and its corresponding normal map.

Results

In this section our presented method will be evaluated on real as well as synthetically generated data sets. The real data is captured using a Canon EOS 400D camera, a low-cost linear polarizing filter and a standard 19 inch LCD screen. For the evaluation on real data, we compared our results with the Gray code technique. The generated photorealistic synthetic data is rendered using the physically based renderer PBRT [Pharr 04]. We apply the Cook-Torrance reflectance model with different parameter settings. Miscellaneous scans of different real-world mesostructures are depicted in figure 5.20.

Efficiency First we will evaluate the efficiency of our method by comparing to relevant previous work. We will focus on the number of image acquisitions necessary to scan a moderate mesostructure. An overview is given in table 5.2. Clearly our method has a high performance when considering the number of required input images, while maintaining a pleasing quality, as will be shown later. This due to the fact that the acquisition time of our technique is *independent* of the number of light sources we want to sample (N). We only need $O(1)$ inputs compared to $O(logN)$, which is a considerable speedup for a large number of inputs. If for example 15000 samples are desired, we still need only 6 input images, where the Gray code approach needs about 30.

Glossiness Theoretically our technique works only for perfect reflectors (see section 5.5.3). However, in practice it seems to behave well for glossier materials as well. An example of a normal map scan of a glossy plastic button is given in figure 5.18.

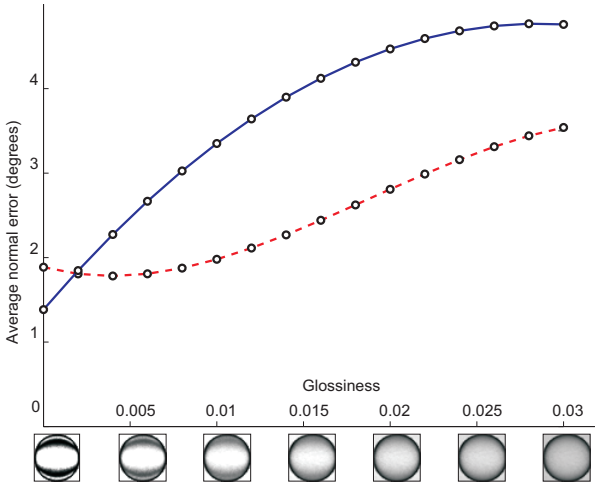


Figure 5.19: Error comparison for glossy materials on synthetic examples under gradient and Gray coded illumination patterns. The average angular error of the normals is plotted in function of the glossiness. The dashed curve shows the error of our gradient method, where the solid curve shows the error of Gray coded patterns.

In order to analyse the loss of quality as function of the increasing glossiness (or roughness), we have provided a ground truth evaluation on photorealistic synthetic data. In addition, a comparative evaluation is made for the use of Gray coded patterns. Figure 5.19 shows a plot of the average angular normal error with respect to the ground truth, for both the gradient and Gray coded illumination. This figure clearly shows that, except from extremely specular materials, overall the gradient patterns outperform the Gray coded ones. This can be explained from the fact that broader specular lobes cause a more extensive pattern convolution, which more or less keeps approximating the wanted center of the specular lobe (which is the reflection direction). This is not the case for Gray code patterns, where the convolution renders the patterns at a certain refinement level totally useless.

In conclusion we note that, where in our approach the normal maps will become blurrier for glossier materials, in the Gray code approach they will become noisier, which is typically worse. Noise in a normal map may cause visually distracting artefacts when the normal map is used for relighting or for generating a depth map. A blurrier normal map just over-smooths the surface, which is less disrupting.

Sensitivity to Noise The main drawback of the use of gradient patterns instead of black/white binary patterns is that they are more sensitive to sensor noise. However, the rather limited amount of input images allows for longer exposure times, avoiding deviating normals due to noise. Still, it remains a more sensitive process.

Occlusion Self-shadowing, or light that cannot reach the surface because it is occluded by the surface itself, causes dark regions where we cannot gather any information from. However, this is a general disadvantage of all shape from reflection methods, and not especially of this method. The only thing we do about it is build up a confidence map based on the per pixel darkness of the fully lit image (L_c). This tells us how useful every pixel is. We take this into account when cleaning up normal maps (bridge minor lapses, remove noise etc.).

5.6 Conclusions

We have presented efficient acquisition methods for scanning mesostructure surfaces in the form of a normal map, only making use of off-the-shelf hardware components, namely a digital camera, an LCD screen and a linear polarizing filter. Whilst current methods often require specialized hardware setups or need a high number of input images, ours only needs a low-cost setup, while retaining a similar quality. Depending on the light patterns employed, 3 to 6 or 11 to 30 input images suffice for acquiring a high quality normal map of specular, glossy or diffuse surfaces. Separation between diffuse and specular/glossy reflections is achieved by cross-polarization of the emitted linearly polarized light by the LCD display.

Each presented method is suited for scanning a different class of materials. As pointed out by our ground truth evaluation on photorealistically rendered imagery, Gray codes perform well for highly specular surfaces, whereas gradients are suited for more glossy surfaces. An additional advantage of gradient patterns is the ability to obtain surface normals from Lambertian reflections. The results are convincing, as shown on our real-world examples.

All three presented methods can easily be implemented and efficiently be executed. This is because only basic image processing operations are involved, all working on a per pixel basis, allowing for straightforward parallellization.

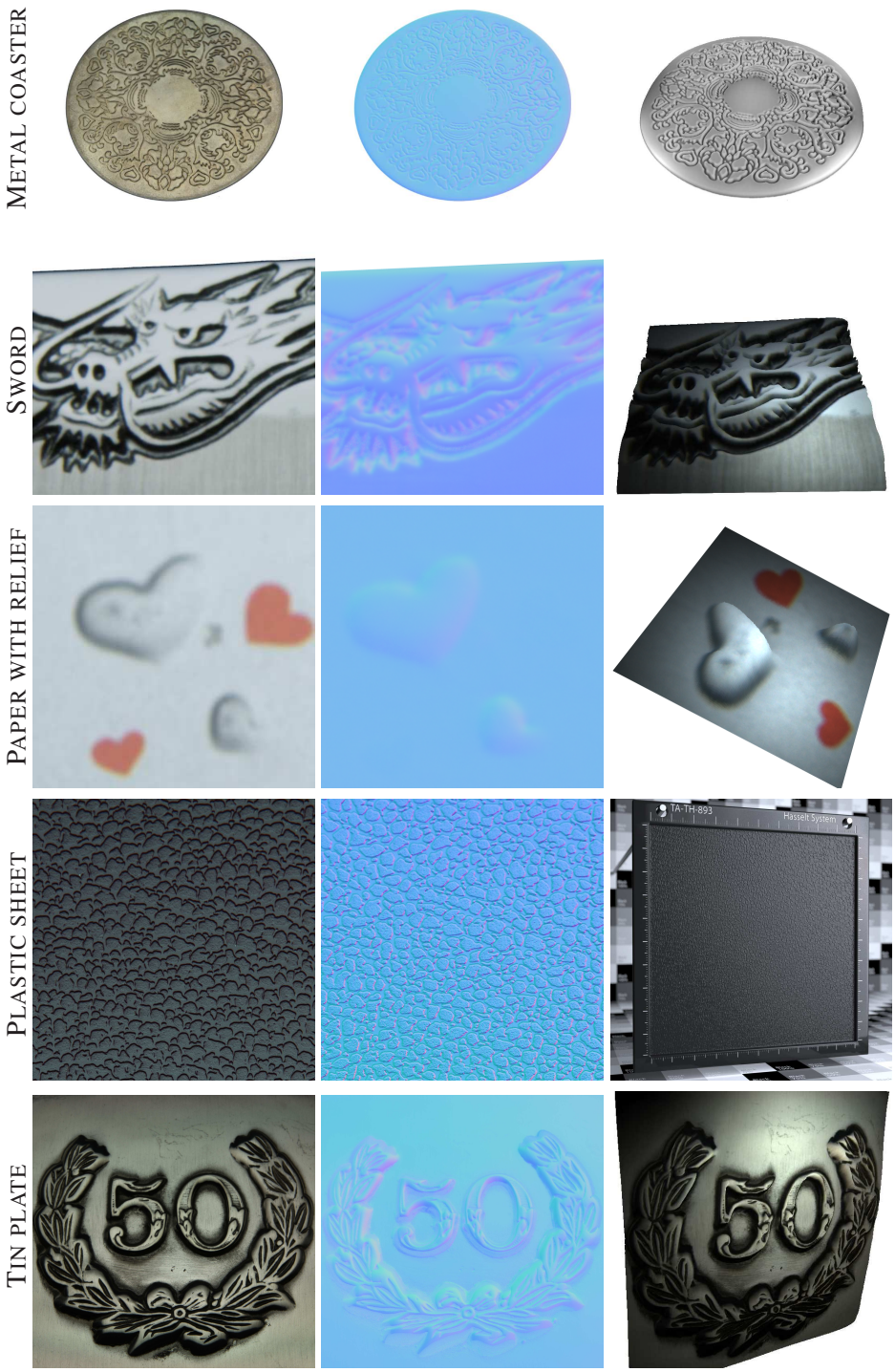


Figure 5.20: Results. Left: input images, Middle: normal maps, Right: renderings.

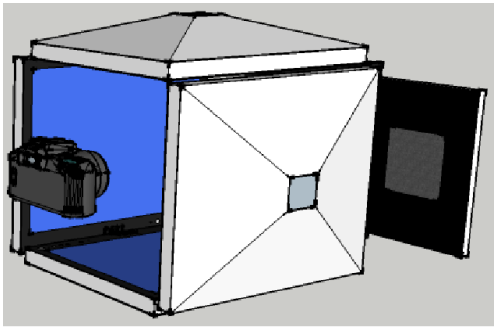


Figure 5.21: Future setup. Multiple screens can increase the range of scannable normals.

5.7 Future Work

Our current setup limits the range of normals that can be acquired. A larger monitor, or multiple monitors (figure 5.21), would alleviate this problem. Another possibility would be to illuminate the mesostructure from different angles by using a turntable, and then to register and fuse the results.

It would be interesting to develop an hybrid method that incorporates highly specular, glossy as well as diffuse components for computing a normal map of heterogeneous materials. The distinct methods yield distinct normal maps for the same object, but how can they be merged into a single, and more correct, normal map?

Chapter 6

Glossiness Acquisition

Contents

In the previous chapter we have discussed how local surface orientation, or the *mesostructure level*, can be obtained using a planar illuminant and a camera. However, for rendering purposes, an estimate of the per pixel reflectance, or the *microstructure level*, is often desirable for heterogenous surfaces. In this chapter, we propose an extension of the Gray code normal acquisition approach, in order to allow for extracting approximate glossiness information from the available input images.

6.1 Introduction

Acquiring spatially varying reflectance usually requires a complicated hardware setup, which measures the Bidirectional Reflectance Distribution Function (BRDF) [Nicodemus 77] at each spatial location. This is a four dimensional function describing the surface's response given the exitant (light) and incident (observation) direction. Our method is much simpler and cheaper. Even though we assume a simplified BRDF model, our technique is able to reproduce the mesostructure's appearance faithfully.

Our primary goal is to provide an efficient, easily applicable, and sufficiently accurate method to acquire glossiness information. We build on the Gray code relief acquisition method, which acquires normal maps by analyzing the reflection of Gray code patterns. These patterns are successively refined in this process to uniquely label screen positions. The key idea in this chapter is that this refinement also allows us to measure the shininess for each spatial location, resulting in a gloss map.

6.2 Related Work

Numerous representations exist for storing either modeled or captured BRDFs [He 91, Ashikhmin 00, Ashikmin 00, He 92]. As storing individual data samples of densely sampled BRDFs is memory inefficient, often approximating models are fitted through the large data collection. This is either achieved by fitting an analytical model [Ward 92, Lafortune 97, Lensch 01, Gardner 03, Ngan 05], or projecting the data to polynomials [Koenderink 96, Malzbender 01], spherical harmonics [Westin 92, Ramamoorthi 02, Basri 03] or wavelet bases [Lalonde 97, Ng 03]. For the sake of simplicity as well as compatibility with known tools, in our work we will employ a simple analytical Phong model [Blinn 77] where the glossiness is represented by a single exponent parameter.

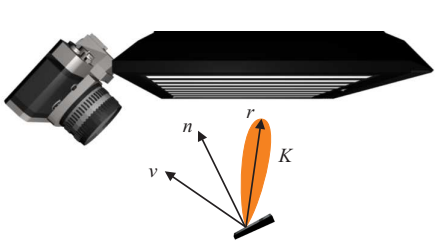
Previous methods tend to focus mainly on improving the quality of the measured BRDF, and less on acquisition speed and practical usability for a large class of users. Often, very specialized setups or long and tedious procedures are required. As we focus on increasing the wide applicability, rather than improving the quality of recent BRDF methods, an approximate glossiness acquisition suffices for our purposes.

The most closely related approach was presented by Gosh et al. [Ghosh 09]. They estimate roughness as well as anisotropy from second order spherical gradient illumination. Their approach makes use of a specialized hardware setup. In our work we take an alternative approach, as we want to avoid non-trivial hardware requirements. We achieve this by using a screen-camera setup consisting of off-the-shelf and omnipresent hardware components.

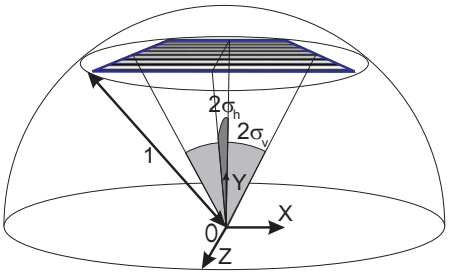
In our work, we start from our Gray code based mesostructure acquisition system and show that glossiness information can easily be extracted from the already available data necessary for shape reconstruction. The original method only has to be slightly modified by replacing the polarization based specular-diffuse separation with the use of pattern complements instead. No extra data is required, and besides LCD screens, also non-polarization based illuminants such as CRT screens can now be employed.

6.3 Acquiring Surface Gloss

Acquiring local surface orientation and glossiness is achieved by placing the target object in front of a CRT or LCD monitor which acts as a light source, and recording the corresponding images using a camera. As in our normal map acquisition technique, we display stepwise refining vertical and horizontal Gray code patterns. We also display each pattern's complement in order to robustly separate diffuse from specular reflection. The specular reflections then efficiently encode discrete spatial screen coordinates in a bit-wise fashion. In a geometrically calibrated setup, this allows for estimating the ideal reflection direction for each pixel. This enables us to estimate the surface normal $\vec{n}$ by taking the halfway vector between the reflection vector $\vec{r}$ and viewing vector $\vec{v}$, as depicted in figure 6.1 (a). In this section, we will extend this system by performing an additional glossiness analysis step.



(a) Camera captures light emitted by the screen and reflected off a specular/glossy mesostructure. Normal $\vec{n}$ and reflection kernel K are obtained from observing the reflected Gray codes.



(b) A screen modeled as a rectangular window on virtual surrounding hemispherical light source.

Figure 6.1: Glossiness acquisition setup.

6.3.1 Overview

In order to extract glossiness information from the recorded mesostructure taken under Gray code illumination, we require some additional illumination patterns. More specifically, complements of the original Gray code patterns are introduced. Fortunately, these render the use of polarization based separation redundant so the number of required patterns does not increase. This is due to the fact that specular highlights are considered much stronger than diffuse reflections [Umeyama 04, Chen 06] and hence a binary decision (white or black reflection) can robustly be made by comparing the pixels illuminated by the pattern and the pattern's complement.

As indicated by the grey area in figure 6.2 (b), after a certain number of pattern refinements, no extra information will be gained as the intensity differences between reflected patterns and their complements will converge to zero. We analyze this convergence process to obtain glossiness information. Without requiring additional input images, we are now able to obtain a per pixel shininess coefficient as well as a surface normal.

The more pattern refinements that can be discerned, the more specular the material will be, and vice versa. This is the case because glossy reflections blur the reflected incoming light pattern. More precisely, the reflected pattern is convolved with a BRDF kernel around the ideal reflection direction [Ramamoorthi 01]. The number of refinements thus is proportional to the shininess of the material. The size

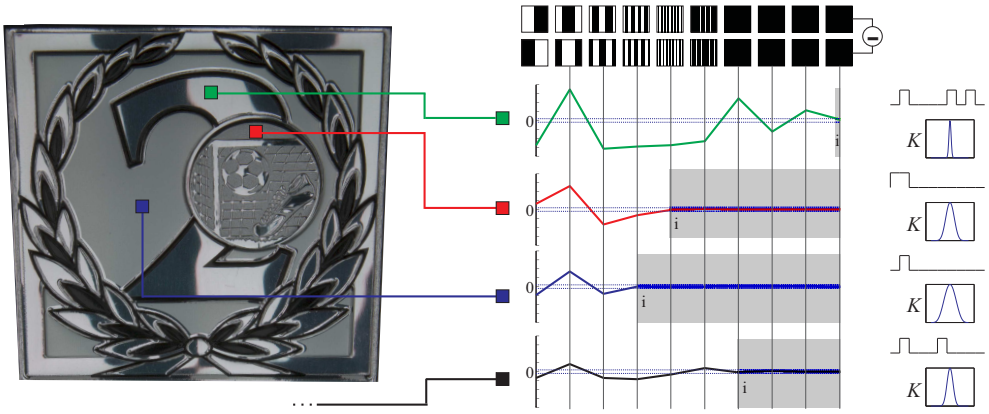


Figure 6.2: Acquisition pipeline. (a) mesostructure, (b) intensity differences as function of pattern refinement level, (c) detected normal codes and Phong kernels.

(or narrowness) and shape of the kernel is defined by the specular lobe of the BRDF. For the sake of simplicity as well as compatibility with known tools, we assume a Phong reflection model. This symmetric lobe is then described by a single exponent value n which is stored in the gloss map.

6.3.2 Theory

We will now formalize the concept proposed in the previous section. Therefore a model will be build that describes the captured radiance L of an imaged surface point, observed from a direction $\vec{v}$, illuminated by a given light pattern P. The equation is given by:

$$L(\vec{v}) = \int_{\Omega} P(\vec{\omega}) \left[R_d(\vec{\omega}, \vec{n}) + R_s(\vec{r}, \vec{\omega}, \vec{n}) \right] d\vec{\omega} \tag{6.1}$$

The following assumptions are made before applying this equation for determining the gloss level:

Specular + diffuse: The imaged surface is assumed to be a combination of a specular component $R_s(\vec{r}, \vec{\omega}, \vec{n})$ and the diffuse component $R_d(\vec{\omega}, \vec{n})$, where $\vec{\omega}$ is the incoming light direction, $\vec{n}$ the surface normal and $\vec{r}$ the specular reflection vector depending on the observation direction $\vec{v}$.

Distant hemispherical illumination: The mesostructure is assumed to be a point in front of the center of the screen, illuminated by a rectangular part of the hemisphere $\Omega = [\frac{\pi}{2} - \sigma_v, \frac{\pi}{2} + \sigma_v] \times [\frac{\pi}{2} - \sigma_h, \frac{\pi}{2} + \sigma_h]$ (figure 6.1 (b))

Interreflections and occlusions: Both interreflections as occlusions are ignored for reasons of simplicity.

Under uniform illumination, where $P_u(\vec{\omega}) = 1$ for each incoming light direction ω, the equation can be simplified.

$$\begin{aligned} L_u(\vec{v}) &= \int_\Omega R_d(\vec{\omega},\vec{n})\, d\vec{\omega} + \int_\Omega R_s(\vec{r},\vec{\omega},\vec{n})\, d\vec{\omega} && (6.2)\\ &= L_d + L_s(\vec{v}) && (6.3) \end{aligned}$$

As we use Gray code patterns, we will define the patterns P_i in terms of the pattern refinement level i. For each incoming light direction $\vec{\omega} \in \Omega$, the pattern $P_i(\vec{\omega})$ is either 0 or 1. The precise pattern definition for vertical patterns P_i^v and horizontal patterns P_i^h are given in equation (6.4) and (6.5), where $(\theta, \phi) \in \Omega$. Notice that the Gray code patterns are basically modeled as a phase shifted ($\frac{1}{4}$ of the period) square wave in the vertical or horizontal interval $[\frac{\pi}{2} - \sigma, \frac{\pi}{2} + \sigma]$. Each pattern refinement from i to $i+1$ the frequency of the wave doubles as $\frac{i^{(i+1)-2}}{i^{i-2}} = 2$.

$$P_i^v(\vec{\omega}) = \frac{1}{2}\Psi\left(\frac{2^{i-2}(\theta - \frac{\pi}{2} + \sigma_v)}{2\sigma_v} + \frac{1}{4}\right) + \frac{1}{2} \quad (6.4)$$

$$P_i^h(\vec{\omega}) = \frac{1}{2}\Psi\left(\frac{2^{i-2}(\phi - \frac{\pi}{2} + \sigma_h)}{2\sigma_h} + \frac{1}{4}\right) + \frac{1}{2} \quad (6.5)$$

The integer function Ψ is defined as

$$\Psi(x) = \begin{cases} +1 & \text{if } x - \lfloor x \rfloor \in [0, 0.5) \\ -1 & \text{if } x - \lfloor x \rfloor \in [0.5, 1) \end{cases} \quad (6.6)$$

Also the complements of the patterns need to be defined. They are referred to as $P_i^{c,v}$ and $P_i^{c,h}$.

$$\begin{aligned} P_i^{c,v}(\vec{\omega}) &= 1 - P_i^v(\vec{\omega}) && (6.7)\\ P_i^{c,h}(\vec{\omega}) &= 1 - P_i^h(\vec{\omega}) && (6.8) \end{aligned}$$

The captured radiance can now be modeled applying the previous definitions. The remainder part of this section will focus on the use of horizontal Gray code patterns only. However, an analogous derivation can be done for vertical patterns.

$$L_i(\vec{v}) = \frac{1}{2}\left[L_d + L_s(\vec{v}) + \int_\Omega \Psi\left(\frac{2^{i-2}(\phi - \frac{\pi}{2} + \sigma_h)}{2\sigma_h} + \frac{1}{4}\right) R_d(\vec{\omega},\vec{n})\, d\vec{\omega} + \int_\Omega \Psi\left(\frac{2^{i-2}(\phi - \frac{\pi}{2} + \sigma_h)}{2\sigma_h} + \frac{1}{4}\right) R_s(\vec{r},\vec{\omega},\vec{n})\, d\vec{\omega}\right] \tag{6.9}$$

If the frequency of the pattern i is sufficiently large, the Lambertian term is approximately zero, as shown by Lamond et al. [Lamond 09]. The underlying reason for this is that the Lambertian reflection can be seen as an applied low frequency convolution filter blurring away the high frequency pattern. Hence the following form can be obtained:

$$L_i(\vec{v}) = \frac{1}{2}\left[L_d + L_s(\vec{v}) + \int_\Omega \Psi\left(\frac{2^{i-2}(\phi - \frac{\pi}{2} + \sigma_h)}{2\sigma_h} + \frac{1}{4}\right) R_s(\vec{r},\vec{\omega},\vec{n})\, d\vec{\omega}\right] \tag{6.10}$$

When the pattern frequency is high compared to the size of the specular lobe, the same holds for the specular term, meaning that it also converges to zero. Hence, the wider the specular lobe, the faster this term converges to zero. As the same reasoning applies for the pattern complement P_i^c, the difference between the radiance of a scene illuminated by P_i and P_i^c converges to zero after a certain pattern refinement level i:

$$|L_i(\vec{v}) - L_i^c(\vec{v})| = \frac{1}{2}\,|L_d + L_s(\vec{v}) - L_d^c - L_s^c(\vec{v})| = 0 \tag{6.11}$$

Concretely, the smallest pattern number i has to found, such that for *all* the subsequent patterns $j \geq i$ the intensity differences $\left|L_j(\vec{v}) - L_j^c(\vec{v})\right|$ drop below a given threshold (figure 6.2 (b)). When i is found, it is converted into a corresponding Phong kernel $K(\omega) = \cos^n(\omega)$ (figure 6.2 (c)). Therefore we propose a simple heuristic which takes into account the following constraint: The surface area $\int_0^{\pi/2} K(\omega) d\omega$ under the kernel K has to halve if the assigned i increases one level. We have empirically established that this relation can be well-approximated by a simple exponential function:

$$n = 4^{(i-1)} \tag{6.12}$$

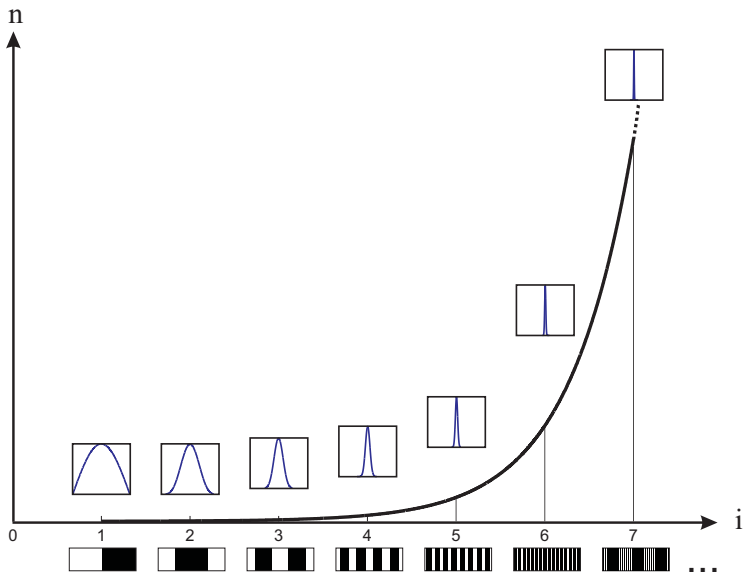


Figure 6.3: Relation between pattern refinement level i and the gloss level n.

This relation is illustrated in figure 6.3. Note we assign no n value when $i = 0$ as the material is then meant to be perfectly diffuse.

It should be noted that this kernel fitting is only approximate because of the limited number of input images we require and the reflection model we employ. However, as the focus of our work lies more on the efficiency and easy applicability than on pure accuracy, it yields sufficiently precise results, as can be seen in the next section.

6.4 Results and Discussion

We have created a proof of concept implementation of the described procedure. The setup we employed consists of a 19 inch LCD monitor and a Canon EOS 400D camera. Experiments were done on different specular materials including plastics, leather, metals, glass and polished marble. For all our results 40 input images were recorded, 10 for each direction plus their complements.

Results on real-world examples are illustrated in figure 6.4. Column (a) shows the acquired normal maps stored in the red, green and blue color channel. Column (b)

contains the gloss maps. The gloss values range from black to white. Black values indicate diffuse reflections, white values represent highly specular reflections, and intermediate grey values represent glossy reflections. The results show for example that different metal coatings yield different gloss values (top row), the glass of the watch is more specular than the plastics (middle row), and scratches on the wallet's hasp make it locally less specular (bottom row). However, also notice in this image that self-shadowed regions in the pores and grooves of the leather are mistakenly classified as non-specular since the occlusion assumptions did not hold. Column (c) shows virtual renderings of the scanned surfaces under point light illumination, taking into account the displayed normal and gloss maps as well as regular texture maps.

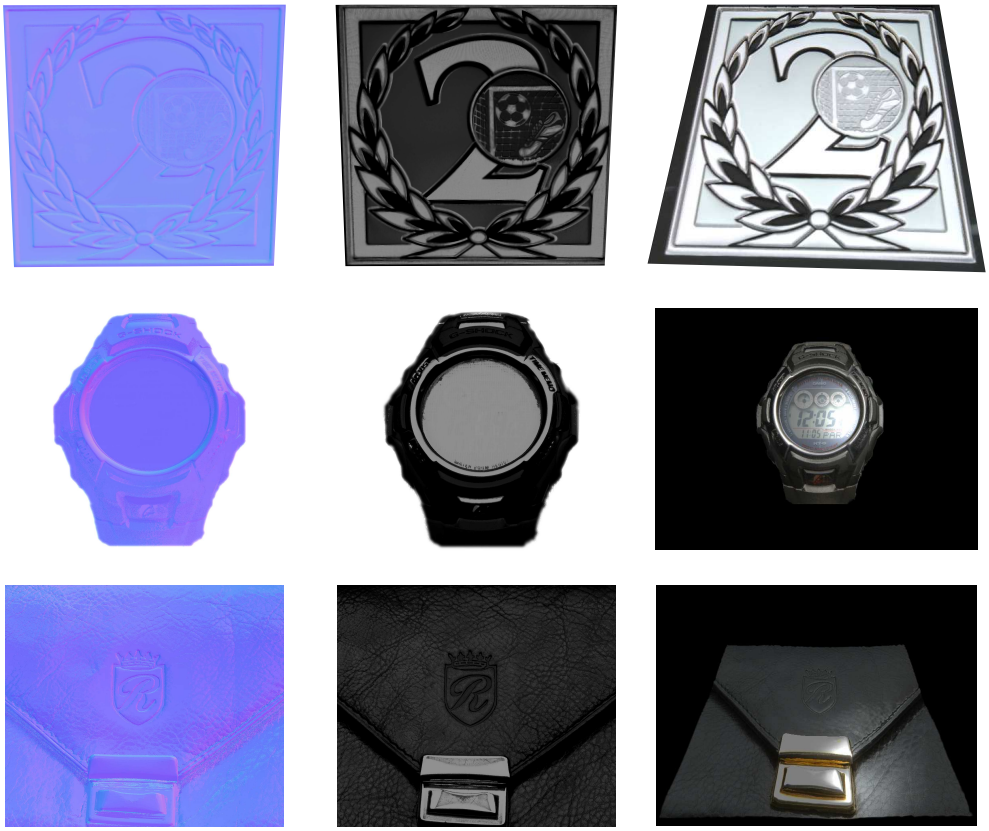
Besides inherent problems of the relief acquisition, such as light occlusions, the main limitation is the small number of available per pixel samples due to efficient binary encoding. Only p possible gloss levels can be distinguished using our technique, where p is the number of patterns used (in our case 10). In addition, as the kernel width is directly dependent on the exponentially decreasing pattern's stripe width, only a few possible kernels can be assigned to diffuse materials. A more dense kernel size distribution may be desirable in this case.

6.5 Conclusions

In this chapter we have illustrated how a straightforward extension of a Gray code based normal scanning can provide us with a very simple reflectance approximation in the form of a single Phong exponent. However, taking into account these approximate gloss maps, in addition to traditional texture and normal maps, tends to considerably improve rerenderings of heterogenous materials.

6.6 Future Work

Improvements are possible regarding the convolution kernel approximations. Currently we are looking into recovering more general BRDFs by adding extra and more optimal patterns to allow for a more precise kernel fitting. Furthermore we believe this work can function as basis for an integrated normal map acquisition system, where the type of pattern/method depends on which mesostructure regions are processed.



(a) (b) (c)

Figure 6.4: Results. (a) normal maps obtained from detected codes, (b) gloss maps containing Phong exponents, (c) renderings.

Chapter 7

Conclusions

Contents

7.1 Summary

We have proposed methods for acquiring local surface orientation and reflectance properties of small-scale surface details, or the surface's *mesostructure*. The primary focus was on increasing the applicability by providing an *efficient*, *easy to implement and execute* approach, employing *off-the-shelf* hardware components.

Our setup consisted of a regular digital still camera and a computer screen as planar illuminant. Light patterns were displayed on the screen, illuminating the surface to be scanned, and reflections were captured by the camera. The recorded images were then processed yielding a digital representation of the relief and reflectance of the scanned surface.

The presented results showed that a simple and inexpensive screen and camera can be turned into a mesostructure acquisition system yielding high quality scans. We believe our system has possible applications in several areas, such as computer games, computer aided design and industrial inspection.

7.1.1 Setup Calibration

We have presented two novel automatic methods for calibrating the screen-camera setup, using a single moving spherical mirror. A ground truth evaluation has shown that both algorithms can be performed within practical error bounds. The edge method requires a low number of input images, but a considerable amount of sphere movements. The full surface method, however, requires a higher number of input images but only a single sphere displacement (two different sphere locations) suffices. The overall error is lower for the latter technique because of the increased number of data samples.

7.1.2 Relief Acquisition

Whilst current methods often require specialized hardware setups or need a high number of input images, ours only needs a low-cost setup, while retaining a similar quality. Depending on the type of light patterns employed, 3 to 6 or 11 to 30 input images suffice for acquiring a high quality normal map of specular, glossy or diffuse surfaces. Separation between diffuse and specular/glossy reflections is achieved by cross-polarization of the emitted linearly polarized light by the LCD display.

Each presented scanning method is suited for scanning a different class of materials. As pointed out by our ground truth evaluation on photorealistically rendered imagery, Gray codes perform well for highly specular surfaces, whereas gradients are suited for more glossy surfaces. An additional advantage of gradient patterns is the ability to obtain surface normals from Lambertian reflections. The results are convincing, as shown on our real-world examples.

All three presented methods can easily be implemented and executed efficiently. This is because only basic image processing are involved, all working on a per pixel basis, allowing for straightforward parallellization.

7.1.3 Glossiness Acquisition

As both the geometric relief and the reflectance properties strongly influence a surface's appearance, we propose a method to also analyze the gloss level of the specular reflection component. This is achieved by a straightforward extension of the Gray code based normal acquisition method. Taking into account the obtained gloss maps in addition to traditional texture and normal maps tends to considerably improve rerenderings of heterogenous materials.

7.2 Future Work

In this section some interesting future directions will be given.

7.2.1 Setup Calibration

As cameras are often focussed on the scene in front of the setup, and not on the reflected screen, we currently need to refocus the camera. This is not desirable since we then alter the internal camera parameters implying a recalibration. Therefore, we are now looking for alternative pattern sequences which are more insensitive to blur due to an out of focus camera.

In the near future, we will also compare the use of Gray code patterns to other codification methods in order to limit the number of required recordings. Specifically, we expect the use of gradient patterns [Ma 07] can reduce the number of input images from 30-35 to less than 10, reducing the calibration time even more.

7.2.2 Relief Acquisition

Our current setup limits the range of normals that can be acquired. A larger monitor, or multiple monitors (figure 5.21), would alleviate this problem. Another possibility would be to illuminate the mesostructure from different angles by using a turntable, and then to register and fuse the results.

It would be interesting to develop an hybrid method that incorporates highly specular, glossy as well as diffuse components for computing a normal map of heterogeneous materials. The distinct methods yield distinct normal maps for the same object, but how can they be merged into a single, and more correct, normal map?

7.2.3 Glossiness Acquisition

Improvements are possible regarding the convolution kernel approximations. Currently we are looking into recovering more general BRDFs by adding extra and more optimal patterns to allow for a more precise kernel fitting. Furthermore we believe this work can function as basis for an integrated normal acquisition system, where the type of pattern/method depends on which mesostructure regions are processed.

Appendices

Appendix A

Scientific Contributions and Publications

The following list of publications, presented at scientific conferences, contains work that is part of this dissertation:

[Francken 07a] *Yannick Francken, Chris Hermans & Philippe Bekaert.* Screen-Camera Calibration using a Spherical Mirror. *In Proceedings of Canadian Conference on Computer and Robot Vision, pages 11–20, 2007*

[Francken 07c] *Yannick Francken, Tom Mertens, Jo Gielis & Philippe Bekaert.* Mesostructure from specularity using coded illumination. *In SIGGRAPH '07: ACM SIGGRAPH 2007 sketches, page 73, New York, NY, USA, 2007. ACM Press*

[Francken 08c] *Yannick Francken, Tom Cuypers, Tom Mertens, Jo Gielis & Philippe Bekaert.* High Quality Mesostructure Acquisition Using Specularities. *In Proceedings of Conference on Computer Vision and Pattern Recognition, pages 1–7. IEEE, 2008*

[Francken 08d] *Yannick Francken, Chris Hermans, Tom Cuypers & Philippe Bekaert.* Fast Normal Map Acquisition Using an LCD Screen Emitting Gradient Patterns. *In Proceedings of Canadian Conference on Computer and Robot Vision, pages 189–195. IEEE Computer Society, 28-30 May 2008*

[Francken 08a] *Yannick Francken, Tom Cuypers & Philippe Bekaert.* Mesostructure from Specularity Using Gradient Illumination. *In International Workshop on Projector-Camera Systems, pages 1–7, New York, NY, USA, 2008. ACM*

[Francken 08b] *Yannick Francken, Tom Cuypers, Tom Mertens & Philippe Bekaert.* Gloss and Normal Map Acquisition Using Gray Codes. *In SIGGRAPH ASIA sketches, New York, NY, USA, 2008. ACM Press*

[Francken 09c] *Yannick Francken, Chris Hermans & Philippe Bekaert.* Screen-Camera Calibration Using Gray Codes. *In Proceedings of Canadian Conference on Computer and Robot Vision. IEEE Computer Society, 25-27 May 2009*

[Francken 09a] *Yannick Francken, Tom Cuypers, Tom Mertens & Philippe Bekaert.* Gloss and Normal Map Acquisition of Mesostructures Using Gray Codes. *In Proceedings of International Symposium on Visual Computing. Springer, 2009*

[Francken 09b] *Yannick Francken, Tom Cuypers, Tom Mertens, Chris Hermans & Philippe Bekaert.* Mesostructures Acquisition Using Planar Illuminants. *In Proceedings of International 3D Stereo Film and Technology Festival, 2009*

The next work is not part of this dissertation:

[Francken 07b] *Yannick Francken, Johan Huysmans & Philippe Bekaert.* Sharing visual information in virtual environments using projective texture mapping. *In International conference on Advances in computer entertainment technology, pages 135–138, New York, NY, USA, 2007. ACM Press*

[Cuypers 08b] *Tom Cuypers, Cedric Vanacken, Yannick Francken, Frank Van Reeth & Philippe Bekaert.* A Multi-Camera Framework for Interactive Videogames. *In International Joint Conference on Computer Vision and Computer Graphics Theory and Applications, pages 443–449. INSTICC - Institute for Systems and Technologies of Information, Control and Communication, 2008*

[Cuypers 08a] *Tom Cuypers, Yannick Francken & Philippe Bekaert.* Shadow Multiplexing for Single Camera Visual Hull Reconstruction. *In International Workshop on Projector-Camera Systems, 2008*

[Cuypers 09c] *Tom Cuypers, Yannick Francken, Cedric Vanaken, Frank Van Reeth & Philippe Bekaert.* Smartphone Localization on Interactive Surfaces Using the Built-in Camera. *In International Workshop on Projector-Camera Systems, 2009*

[Cuypers 09b] *Tom Cuypers, Yannick Francken, Johannes Taelman & Philippe Bekaert.* Shadow Multiplexing for Real-Time Silhouette Extraction. *In International Workshop on Projector-Camera Systems, 2009*

[Hermans 09b] *Chris Hermans, Yannick Francken, Tom Cuypers & Philippe Bekaert.* Depth from Sliding Projections. *In Proceedings of Conference on Computer Vision and Pattern Recognition, 2009*

[Hermans 09a] *Chris Hermans, Yannick Francken, Tom Cuypers & Philippe Bekaert.* Depth from Encoded Sliding Projections. *In Proceedings of International Symposium on Visual Computing. Springer, 2009*

[Hermans 09c] *Chris Hermans, Yannick Francken, Tom Cuypers & Philippe Bekaert.* Mobile Structured Light. *In Proceedings of International 3D Stereo Film and Technology Festival, 2009*

[Cuypers 09a] *Tom Cuypers, Tanja Van den Eede, Stijn Ligot, Yannick Francken, Chris Hermans, Frans Arickx & Philippe Bekaert.* StereoWiision: Stereo Vision with Wiimotes. *In Proceedings of International 3D Stereo Film and Technology Festival, 2009*

Bijlage B

Samenvatting (Dutch Summary)

Gedurende de laatste decennia hebben computers steeds een belangrijkere rol ingenomen bij het uitvoeren van een breed scala aan taken. Eén van deze taken is het weergeven van virtuele scènes op een overtuigende manier. De zogenaamde rendertechnieken die dit mogelijk maken, worden binnen verscheidene toepassingsgebieden, zoals computerspelletjes, gebruikt. Zelfs het genereren van foto-realistische beelden is ondertussen zeer gebruikelijk, zoals bijvoorbeeld in de post-productie van films. Snelle en/of nauwkeurige rendertechnieken, die lichttransport binnen de virtuele wereld benaderen of nauwkeurig simuleren, werden door de jaren heen ontwikkeld en verfijnd.

Zelfs indien lichtinteractie op een fysiek correcte wijze gesimuleerd wordt, zijn de uiteindelijke resultaten steeds afhankelijk van het verstrekte 3D model. Als de ingevoerde scène geen kleinschalige oppervlakte details zoals krassen, onvolkomenheden, enz. bevat, zal deze als onrealistisch worden opgevat. Het handmatig modelleren van deze "details" kan een tijdsintensieve taak zijn. Dit wijst erop dat het gebruik van automatische scanmethoden wenselijk is.

Hoewel tal van geometrie acquisitie-technieken bestaan, zijn ze meestal enkel in staat om globale oppervlakte informatie te verwerven, waardoor lokale kleinschalige eigenschappen van het reliëf verdwijnen. In de praktijk worden lokale afwijkingen ten opzichte van het globale oppervlak nog al te vaak handmatig gemodelleerd door ontwerpers, in plaats van deze uit de reële wereld in te scannen. Hetzelfde geldt

voor reflectie-eigenschappen, die vaak handmatig worden nagemaakt met behulp van zogenaamde “shaders” op de grafische kaart. Eén van de redenen hiervoor is de complexiteit van de huidige beschikbare methoden. Vele aanpakken vereisen speciale opstellingen, met exotische hardware componenten, tijdrovende kalibratie procedures, moeilijke implementaties en scanprocedures.

In dit proefschrift is het doel om het scannen van kleinschalige oppervlakte eigenschappen beschikbaar te maken voor het grote publiek en zo de kloof tussen huidig onderzoek en praktisch gebruik te dichten. Dit wordt bereikt door een aanpak voor te stellen die *efficiënt* is en *eenvoudig* kan worden geïmplementeerd en uitgevoerd. Daarbij zijn enkel *alomvertegenwoordigde hardware componenten* vereist. Onze opstelling bestaat uit een gewone digitale fotocamera en een computerscherm dat fungeert als een vlakke lichtbron. Lichtpatronen, die het te scannen oppervlak belichten, worden weergegeven op het scherm en de reflecties worden opgenomen door de camera. De opgenomen beelden worden vervolgens verwerkt tot een digitale representatie van het oppervlakte reliëf alsook de reflectie-eigenschappen van het gescande oppervlak.

Met het verwezenlijken van het voorgenoemde systeem, hebben we wetenschappelijke bijdragen geleverd in drie subdomeinen, namelijk (a) kalibratie, (b) opmeten van reliëf, (c) opmeten van glans. Deze zullen nu afzonderlijk besproken worden.

We hebben twee nieuwe methoden gepresenteerd voor het automatisch kalibreren van de scherm-camera opstelling. Beide methoden maken gebruik van een enkele, manueel te verplaatsen, sferische spiegel. Uit de nauwkeurigheidsevaluatie is gebleken dat beide algoritmen kunnen worden uitgevoerd binnen een praktische foutmarge. De eerste methode vereist slechts een klein aantal invoerbeelden, maar wel een aanzienlijke hoeveelheid manuele interventies voor het verplaatsen van de spiegel. De tweede methode vereist echter een groter aantal invoerbeelden, maar slechts één enkele spiegel verplaatsing is vereist (of dus twee verschillende sfeer locaties). De nauwkeurigheid hiervan is ook beter vanwege de grotere hoeveelheid aan beschikbare gegevens die zich in de opnames bevinden.

Voor het opmeten van het reliëf van het oppervlak, volstaan 3 tot 6 of 11 tot 30 invoerbeelden voor het verkrijgen van een normaalmap van hoge kwaliteit van speculaire, glossy of diffuse oppervlakken. Dit aantal is afhankelijk van de aard van de gebruikte lichtpatronen, namelijk Gray codes of gradiënten. Het scheiden van

diffuse en speculaire/glossy reflecties is vereist en wordt verkregen door middel van cross-polarisatie van het lineair gepolariseerd licht van de LCD-display.

Elke gepresenteerde scanmethode is geschikt voor het scannen van een andere klasse van materialen. Zoals blijkt uit de kwaliteitsevaluatie op fotorealistisch gerenderde beelden, presteren Gray codes goed voor sterk spiegelende oppervlakken, terwijl gradiënten beter geschikt zijn voor meer glossy oppervlakken. Een bijkomend voordeel van gradiënt patronen is de mogelijkheid om ook oppervlakte normalen te verkrijgen uit Lambertiaanse (diffuse) reflecties. De resultaten zijn overtuigend, zoals getoond werd op voorbeeldscans van objecten uit de reële wereld. De gepresenteerde methoden kunnen op een eenvoudige manier worden geïmplementeerd en efficiënt worden uitgevoerd. Dit komt omdat alleen elementaire beeldverwerkingsoperaties nodig zijn, die allen werken op een per pixel basis, waardoor deze ook eenvoudig kunnen worden geparallelliseerd.

Aangezien zowel het geometrische reliëf als de reflectie-eigenschappen het uiterlijk van een oppervlak sterk beïnvloeden, stellen we ook een methode voor om het glansniveau van speculaire reflecties te bepalen. Dit wordt bereikt door een eenvoudige uitbreiding van de Gray code gebaseerde acquisitie methode. Het in rekening brengen van het verkregen glansniveau, naast de traditionele kleur en reliëf informatie, verbetert weergaven van heterogene materialen aanzienlijk.

De gepresenteerde resultaten binnen deze dissertatie tonen aan dat een eenvoudige en goedkope scherm-camera combinatie kan worden omgezet in een systeem dat hoge kwaliteit scans van mesostructuren oplevert. Wij geloven dat ons systeem toepasbaar is in verschillende domeinen, zoals computerspelletjes, computer-aided design en industriële inspectie.

Bibliography

[Adato 07] Yair Adato, Yuriy Vasilyev, Ohad Ben-Shahar & Todd Zickler. *Toward a Theory of Shape from Specular Flow*. In Proceedings of International Conference on Computer Vision, 2007.

[Agrawal 05] Amit Agrawal, Rama Chellappa & Ramesh Raskar. *An Algebraic Approach to Surface Reconstruction from Gradient Fields*. In International Conference on Computer Vision, pages 174–181, Washington, DC, USA, 2005. IEEE Computer Society.

[Ashikhmin 00] Michael Ashikhmin & Peter Shirley. *An anisotropic phong BRDF model*. Journal of Graphics Tools, vol. 5, no. 2, pages 25–32, 2000.

[Ashikmin 00] Michael Ashikmin, Simon Premože & Peter Shirley. *A microfacet-based BRDF generator*. In Proceedings of SIGGRAPH, pages 65–74, New York, NY, USA, 2000. ACM Press.

[Baker 99] S. Baker & S.K. Nayar. *A Theory of Single-Viewpoint Catadioptric Image Formation*. International Journal of Computer Vision, vol. 35, no. 2, pages 175–196, Nov 1999.

[Basri 03] Ronen Basri & David W. Jacobs. *Lambertian Reflectance and Linear Subspaces*. Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 2, pages 218–233, 2003.

[Blinn 77] James F. Blinn. *Models of light reflection for computer synthesized pictures*. In Proceedings of SIGGRAPH, pages 192–198, New York, NY, USA, 1977. ACM.

[Bonfort 06] Thomas Bonfort, Peter Sturm & Pau Gargallo. *General Specular Surface Triangulation*. In Proceedings of Asian Conference on Computer Vision, volume 2, pages 872–881, jan 2006.

[Bouguet 06] Jean-Yves Bouguet. *Camera Calibration Toolbox for MATLAB*, 2006.

[Boyer 87] K. L. Boyer & A. C. Kak. *Color-encoded structured light for rapid active ranging*. Transactions on Pattern Analysis and Machine Intelligence, vol. 9, no. 1, pages 14–28, 1987.

[Chen 06] Tongbo Chen, Michael Goesele & Hans-Peter Seidel. *Mesostructure from Specularity*. In Proceedings of Conference on Computer Vision and Pattern Recognition, volume 2, pages 1825–1832, 2006.

[Chen 07] Tongbo Chen, Hendrik P. A. Lensch, Christian Fuchs & Hans-Peter Seidel. *Polarization and Phase-Shifting for 3D Scanning of Translucent Objects*. In Proceedings of Conference on Computer Vision and Pattern Recognition, 2007.

[Chung 08] H.S. Chung & J. Jia. *Efficient photometric stereo on glossy surfaces with wide specular lobes*. In Proceedings of Conference on Computer Vision and Pattern Recognition. IEEE, 2008.

[Clark 06] James J. Clark. *Photometric Stereo with Nearby Planar Distributed Illuminants*. In Proceedings of Canadian Conference on Computer and Robot Vision, page 16, 2006.

[Cuypers 08a] Tom Cuypers, Yannick Francken & Philippe Bekaert. *Shadow Multiplexing for Single Camera Visual Hull Reconstruction*. In International Workshop on Projector-Camera Systems, 2008.

[Cuypers 08b] Tom Cuypers, Cedric Vanacken, Yannick Francken, Frank Van Reeth & Philippe Bekaert. *A Multi-Camera Framework for Interactive Videogames*. In International Joint Conference on Computer Vision and Computer Graphics Theory and Applications, pages 443–449. INSTICC - Institute for Systems and Technologies of Information, Control and Communication, 2008.

[Cuypers 09a] Tom Cuypers, Tanja Van den Eede, Stijn Ligot, Yannick Francken, Chris Hermans, Frans Arickx & Philippe Bekaert. *StereoWiision: Stereo Vision with Wiimotes*. In Proceedings of International 3D Stereo Film and Technology Festival, 2009.

[Cuypers 09b] Tom Cuypers, Yannick Francken, Johannes Taelman & Philippe Bekaert. *Shadow Multiplexing for Real-Time Silhouette Extraction*. In International Workshop on Projector-Camera Systems, 2009.

[Cuypers 09c] Tom Cuypers, Yannick Francken, Cedric Vanaken, Frank Van Reeth & Philippe Bekaert. *Smartphone Localization on Interactive Surfaces Using the Built-in Camera*. In International Workshop on Projector-Camera Systems, 2009.

[Dana 99] Kristin J. Dana, Bram van Ginneken, Shree K. Nayar & Jan J. Koenderink. *Reflectance and texture of real-world surfaces*. Proceedings of SIGGRAPH, vol. 18, no. 1, pages 1–34, 1999.

[Debevec 97] Paul Debevec & Jitendra Malik. *Recovering high dynamic range radiance maps from photographs*. In Proceedings of SIGGRAPH, pages 369–378, New York, NY, USA, 1997. ACM Press.

[Fischler 81] Martin A. Fischler & Robert C. Bolles. *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*. Communications of the ACM archive, vol. 24, pages 381 – 395, 1981.

[Francken 07a] Yannick Francken, Chris Hermans & Philippe Bekaert. *Screen-Camera Calibration using a Spherical Mirror*. In Proceedings of Canadian Conference on Computer and Robot Vision, pages 11–20, 2007.

[Francken 07b] Yannick Francken, Johan Huysmans & Philippe Bekaert. *Sharing visual information in virtual environments using projective texture mapping*. In International conference on Advances in computer entertainment technology, pages 135–138, New York, NY, USA, 2007. ACM Press.

[Francken 07c] Yannick Francken, Tom Mertens, Jo Gielis & Philippe Bekaert. *Mesostructure from specularity using coded illumination*. In SIGGRAPH '07: ACM SIGGRAPH 2007 sketches, page 73, New York, NY, USA, 2007. ACM Press.

[Francken 08a] Yannick Francken, Tom Cuypers & Philippe Bekaert. *Mesostructure from Specularity Using Gradient Illumination*. In International Workshop on Projector-Camera Systems, pages 1–7, New York, NY, USA, 2008. ACM.

[Francken 08b] Yannick Francken, Tom Cuypers, Tom Mertens & Philippe Bekaert. *Gloss and Normal Map Acquisition Using Gray Codes*. In SIGGRAPH ASIA sketches, New York, NY, USA, 2008. ACM Press.

[Francken 08c] Yannick Francken, Tom Cuypers, Tom Mertens, Jo Gielis & Philippe Bekaert. *High Quality Mesostructure Acquisition Using Specularities*. In Proceedings of Conference on Computer Vision and Pattern Recognition, pages 1–7. IEEE, 2008.

[Francken 08d] Yannick Francken, Chris Hermans, Tom Cuypers & Philippe Bekaert. *Fast Normal Map Acquisition Using an LCD Screen Emitting Gradient Patterns*. In Proceedings of Canadian Conference on Computer and Robot Vision, pages 189–195. IEEE Computer Society, 28-30 May 2008.

[Francken 09a] Yannick Francken, Tom Cuypers, Tom Mertens & Philippe Bekaert. *Gloss and Normal Map Acquisition of Mesostructures Using Gray Codes*. In Proceedings of International Symposium on Visual Computing. Springer, 2009.

[Francken 09b] Yannick Francken, Tom Cuypers, Tom Mertens, Chris Hermans & Philippe Bekaert. *Mesostructures Acquisition Using Planar Illuminants*. In Proceedings of International 3D Stereo Film and Technology Festival, 2009.

[Francken 09c] Yannick Francken, Chris Hermans & Philippe Bekaert. *Screen-Camera Calibration Using Gray Codes*. In Proceedings of Cana-

dian Conference on Computer and Robot Vision. IEEE Computer Society, 25-27 May 2009.

[Frankot 88] Robert T. Frankot & Rama Chellappa. *A Method for Enforcing Integrability in Shape from Shading Algorithms*. Transactions on Pattern Analysis and Machine Intelligence, vol. 10, no. 4, pages 439–451, 1988.

[Funk 07] Nathan Funk & Yee-Hong Yang. *Using a Raster Display for Photometric Stereo*. In Proceedings of Canadian Conference on Computer and Robot Vision, pages 201–207, 2007.

[Gardner 03] Andrew Gardner, Chris Tchou, Tim Hawkins & Paul Debevec. *Linear light source reflectometry*. In Proceedings of SIGGRAPH, pages 749–758. ACM, 2003.

[Geyer 01] Christopher Geyer & Kostas Daniilidis. *Catadioptric Projective Geometry*. International Journal of Computer Vision, vol. 45, no. 3, pages 223–243, 2001.

[Ghosh 09] Abhijeet Ghosh, Tongbo Chen, Pieter Peers, Cyrus A. Wilson & Paul Debevec. *Estimating Specular Roughness and Anisotropy from Second Order Spherical Gradient Illumination*. In Proceedings of Eurographics Symposium on Rendering, jun 2009.

[Goesele 00] M. Goesele, W. Heidrich, H.P.A. Lensch & H.P. Seidel. *Building a photo studio for measurement purposes*. In Proceedings of Vision, Modeling, and Visualization, pages 231–238. IOS Press, November 2000.

[Goldman 05] Dan B. Goldman, Brian Curless, Aaron Hertzmann & Steven M. Seitz. *Shape and Spatially-Varying BRDFs from Photometric Stereo*. In Proceedings of International Conference on Computer Vision, volume 1, pages 341–348, 2005.

[Grossberg 04] M.D. Grossberg & S.K. Nayar. *Modeling the Space of Camera Response Functions*. Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 10, pages 1272–1282, Oct 2004.

[Halstead 96] Mark A. Halstead, Brain A. Barsky, Stanley A. Klein & Robert B. Mandell. *Reconstructing curved surfaces from specular reflection patterns using spline surface fitting of normals*. In Proceedings of SIGGRAPH, pages 335–342, 1996.

[Han 03] Jefferson Y. Han & Ken Perlin. *Measuring bidirectional texture reflectance with a kaleidoscope*. In Proceedings of SIGGRAPH, pages 741–748. ACM, 2003.

[Harris 88] Chris Harris & Mike Stephens. *A Combined Corner and Edge Detector*. In Proceedings of the Fourth Alvey Vision Conference, pages 147–152, Manchester, 1988. The University of Sheffield Printing Unit.

[Hartley 97] Richard I. Hartley. *In Defense of the Eight-Point Algorithm*. Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 6, pages 580–593, 1997.

[Hartley 04] R. I. Hartley & A. Zisserman. Multiple view geometry in computer vision. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[He 91] Xiao D. He, Kenneth E. Torrance, François X. Sillion & Donald P. Greenberg. *A comprehensive physical model for light reflection*. In Proceedings of SIGGRAPH, pages 175–186, New York, NY, USA, 1991. ACM.

[He 92] Xiao D. He, Patrick O. Heynen, Richard L. Phillips, Kenneth E. Torrance, David H. Salesin & Donald P. Greenberg. *A fast and accurate light reflection model*. Proceedings of SIGGRAPH, vol. 26, no. 2, pages 253–254, 1992.

[Healey 88] Glenn Healey & Thomas O. Binford. *Local shape from specularity*. Computer Vision, Graphics, and Image Processing, vol. 42, no. 1, pages 62–86, 1988.

[Healey 94] G. Healey & R. Kondepudy. *Radiometric CCD camera calibration and noise estimation*. Transactions on Pattern Analysis and Machine Intelligence, vol. 16, no. 3, pages 267–276, 1994.

[Hermans 09a] Chris Hermans, Yannick Francken, Tom Cuypers & Philippe Bekaert. *Depth from Encoded Sliding Projections*. In Proceedings of International Symposium on Visual Computing. Springer, 2009.

[Hermans 09b] Chris Hermans, Yannick Francken, Tom Cuypers & Philippe Bekaert. *Depth from Sliding Projections*. In Proceedings of Conference on Computer Vision and Pattern Recognition, 2009.

[Hermans 09c] Chris Hermans, Yannick Francken, Tom Cuypers & Philippe Bekaert. *Mobile Structured Light*. In Proceedings of International 3D Stereo Film and Technology Festival, 2009.

[Hernández 07] Carlos Hernández, George Vogiatzis, Gabriel J. Brostow, Bjoörn Stenger & Roberto Cipolla. *Non-Rigid Photometric Stereo with Colored Lights*. In Proceedings of International Conference on Computer Vision, 2007.

[Hertzmann 05] Aaron Hertzmann. *Example-Based Photometric Stereo: Shape Reconstruction with General, Varying BRDFs*. Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 8, pages 1254–1264, 2005. Member-Steven M. Seitz.

[Holroyd 08] Michael Holroyd, Jason Lawrence, Greg Humphreys & Todd Zickler. *A Photometric Approach for Estimating Normals and Tangents*. Proceedings of SIGGRAPH Asia, vol. 27, no. 5, page 133, 2008.

[Horn 75] B.K.P. Horn. *Obtaining Shape from Shading Information*. In The Psychology of Computer Vision, pages 115–155, McGraw-Hill, New York, USA, 1975.

[Horn 86] B. K. P. Horn. Robot vision. MIT Press, 1986. HOR b2 86:1 1.Ex.

[Iizuka 02] K. Iizuka. Elements of photonics. Wiley-Interscience, 2002.

[Ikeuchi 81] K. Ikeuchi. *Determining Surface Orientation of Specular Surfaces by using the Photometric Stereo Method*. Transactions on Pattern Analysis and Machine Intelligence, vol. 3, nov 1981.

[Johnson 09] Micah K. Johnson & Edward H. Adelson. *Retrographic sensing for the measurement of surface texture and shape*. In Proceedings of Conference on Computer Vision and Pattern Recognition, pages 1070–1077, 2009.

[Koenderink 96] Jan J. Koenderink, Andrea J. van Doorn & Marigo Stavridi. *Bidirectional Reflection Distribution Function Expressed in Terms of Surface Scattering Modes*. In Proceedings of European Conference on Computer Vision, pages 28–39, London, UK, 1996. Springer-Verlag.

[Kovesi 05] Peter Kovesi. *Shapelets Correlated with Surface Normals Produce Surfaces*. In International Conference on Computer Vision, pages 994–1001, Washington, DC, USA, 2005. IEEE Computer Society.

[Kuthirummal 06] S. Kuthirummal & S. K. Nayar. *Multiview Radial Catadioptric Imaging for Scene Capture*. Proceedings of SIGGRAPH, Jul 2006.

[Kutulakos 08] Kiriakos N. Kutulakos & Eron Steger. *A Theory of Refractive and Specular 3D Shape by Light-Path Triangulation*. International Journal of Computer Vision, vol. 76, no. 1, pages 13–29, 2008.

[Lafortune 97] Eric P. F. Lafortune, Sing-Choong Foo, Kenneth E. Torrance & Donald P. Greenberg. *Non-linear approximation of reflectance functions*. In Proceedings of SIGGRAPH, pages 117–126, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[Lalonde 97] Paul Lalonde & Alain Fournier. *A Wavelet Representation of Reflectance Functions*. Transactions on Visualization and Computer Graphics, vol. 3, no. 4, pages 329–336, 1997.

[Lamond 09] Bruce Lamond, Pieter Peers, Abhijeet Ghosh & Paul Debevec. *Image-based Separation of Diffuse and Specular Reflections using Environmental Structured Illumination*. In Proceedings of International Conference on Computational Photography, jun 2009.

[Lanman 06] Douglas Lanman, Megan Wachs, Gabriel Taubin & Fernando Cukierman. *Reconstructing a 3D Line from a Single Catadioptric Image*. In International Symposium on 3D Data Processing, Visualization and Transmission, pages 89–96, Washington, DC, USA, 2006. IEEE Computer Society.

[Lensch 01] Hendrik P. A. Lensch, Michael Goesele, Jan Kautz, Wolfgang Heidrich & Hans-Peter Seidel. *Image-Based Reconstruction of Spatially Varying Materials*. In Proceedings of Eurographics Symposium on Rendering, pages 103–114, London, UK, 2001. Springer-Verlag.

[Ma 07] Wan-Chun Ma, Tim Hawkins, Pieter Peers, Charles-Felix Chabert, Malte Weiss & Paul Debevec. *Rapid Acquisition of Specular and Diffuse Normal Maps from Polarized Spherical Gradient Illumination*. In Proceedings of Eurographics Symposium on Rendering, jun 2007.

[Mallick 05] Satya P. Mallick, Todd E. Zickler, David J. Kriegman & Peter N. Belhumeur. *Beyond Lambert: Reconstructing Specular Surfaces Using Color*. In Proceedings of Conference on Computer Vision and Pattern Recognition, volume 2, pages 619–626, 2005.

[Malzbender 01] Thomas Malzbender, Dan Gelb & Hans J. Wolters. *Polynomial texture maps*. In Proceedings of SIGGRAPH, pages 519–528, 2001.

[Marschner 05] Stephen R. Marschner, Stephen H. Westin, Adam Arbree & Jonathan T. Moon. *Measuring and modeling the appearance of finished wood*. Proceedings of SIGGRAPH, vol. 24, no. 3, pages 727–734, 2005.

[Mitsunaga 99] T. Mitsunaga & S.K. Nayar. *Radiometric Self Calibration*. In Proceedings of Conference on Computer Vision and Pattern Recognition, volume 1, pages 374–380. IEEE Computer Society, Jun 1999.

[Miyazaki 07] Daisuke Miyazaki & Katsushi Ikeuchi. *Shape Estimation of Transparent Objects by Using Inverse Polarization Ray Trac-*

ing. Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 11, pages 2018–2030, 2007.

[Morris 07] Nigel J. W. Morris & Kiriakos N. Kutulakos. *Reconstructing the Surface of Inhomogeneous Transparent Scenes by Scatter-Trace Photography*. In Proceedings of International Conference on Computer Vision, 2007.

[Nayar 90] S. Nayar, Katsushi Ikeuchi & Takeo Kanade. *Determining Shape and Reflectance of Hybrid Surfaces by Photometric Sampling*. Transactions on Robotics and Automation, vol. 6, no. 4, pages 418–431, aug 1990.

[Nayar 97] Shree K. Nayar, Xi-Sheng Fang & Terrance Boult. *Separation of Reflection Components Using Color and Polarization*. International Journal of Computer Vision, vol. 21, no. 3, pages 163–186, 1997.

[Nehab 05] Diego Nehab, Szymon Rusinkiewicz, James Davis & Ravi Ramamoorthi. *Efficiently Combining Positions and Normals for Precise 3D Geometry*. In Proceedings of SIGGRAPH, volume 24, August 2005.

[Nehab 08] Diego Nehab, Tim Weyrich & Szymon Rusinkiewicz. *Dense 3D Reconstruction from Specularity Consistency*. In Proceedings of Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, jun 2008.

[Neubeck 05] A. Neubeck, A. Zalesny & L. Van Gool. *3D Texture Reconstruction from Extensive BTF Data*. In Texture 2005 Workshop in conjunction with International Conference on Computer Vision 2005, pages 13–19, October 2005.

[Ng 03] Ren Ng, Ravi Ramamoorthi & Pat Hanrahan. *All-frequency shadows using non-linear wavelet lighting approximation*. In Proceedings of SIGGRAPH, pages 376–381, New York, NY, USA, 2003. ACM.

[Ngan 05] Addy Ngan, Frédo Durand & Wojciech Matusik. *Experimental Analysis of BRDF Models*. In Proceedings of Eurographics Symposium on Rendering, pages 117–226. Eurographics Association, 2005.

[Nicodemus 77] F.E. Nicodemus, J.C. Richmond, J.J. Hsia, IW Ginsberg & T. Limperis. *Geometrical considerations and nomenclature for reflectance*. NBS monograph, vol. 160, pages 201–231, 1977.

[Nitschke 09] Christian Nitschke, Atsushi Nakazawa & Haruo Takemura. *Display-Camera Calibration from Eye Reflections*. In Proceedings of International Conference on Computer Vision, September 2009.

[Paterson 05] J. A. Paterson, D. Claus & A. W. Fitzgibbon. *BRDF and geometry capture from extended inhomogeneous samples using flash photography*. Computer Graphics Forum (Special Eurographics Issue), vol. 24, no. 3, pages 383–391, September 2005.

[Perlin 85] Ken Perlin. *An image synthesizer*. Proceedings of SIGGRAPH, vol. 19, no. 3, pages 287–296, 1985.

[Pharr 04] Matt Pharr & Greg Humphreys. Physically based rendering: From theory to implementation. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[POV-Ray 09] POV-Ray. `http://www.povray.org/`, 2009.

[Ramamoorthi 01] Ravi Ramamoorthi & Pat Hanrahan. *A signal-processing framework for inverse rendering*. In Proceedings of SIGGRAPH, pages 117–128, New York, NY, USA, 2001. ACM.

[Ramamoorthi 02] Ravi Ramamoorthi & Pat Hanrahan. *Frequency space environment map rendering*. In Proceedings of SIGGRAPH, pages 517–526, New York, NY, USA, 2002. ACM.

[Rushmeier 97] H.E. Rushmeier, G. Taubin & A. Guéziec. Applying shape from lighting variation to bump map capture. IBM TJ Watson Research Center, 1997.

[Salvi 04] Joaquim Salvi, Jordi Pages & Joan Batlle. *Pattern codification strategies in structured light systems*. Pattern Recognition, vol. 37, no. 4, pages 827–849, april 2004.

[Sanderson 88] A.C. Sanderson, L.E. Weiss & S.K. Nayar. *Structured Highlight Inspection of Specular Surfaces*. Transactions on Pattern Analysis and Machine Intelligence, vol. 10, no. 1, pages 44–55, Jan 1988.

[Sato 87] K. Sato & S. Inokuchi. *Range-Imaging System Utilizing Nematic Liquid Crystal Mask*. In Proceedings of International Conference on Computer Vision, pages 657–661, 1987.

[Scharstein 02] Daniel Scharstein & Richard Szeliski. *A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms*. International Journal of Computer Vision, vol. 47, no. 1-3, pages 7–42, 2002.

[Scharstein 03] Daniel Scharstein & Richard Szeliski. *High-Accuracy Stereo Depth Maps Using Structured Light*. In Proceedings of Conference on Computer Vision and Pattern Recognition, volume 1, page 195, 2003.

[Schechner 03] Y.Y. Schechner, S.K. Nayar & P.N. Belhumeur. *A Theory of Multiplexed Illumination*. In Proceedings of International Conference on Computer Vision, volume 2, pages 808–815, oct 2003.

[Schlüns 93] Karsten Schlüns. *Photometric Stereo for Non-Lambertian Surfaces Using Color Information*. In Proceedings of CAIP, pages 444–451, London, UK, 1993. Springer-Verlag.

[Shiu 89] YC. Shiu & S. Ahmad. *3D location of circular and spherical features by monocularmodel-based vision*. In Systems, Man and Cybernetics, 1989. Conference Proceedings., IEEE International Conference on, pages 576–581 vol.2, Cambridge, MA, USA, November 1989.

[Tarini 05] Marco Tarini, Hendrik P. A. Lensch, Michael Goesele & Hans-Peter Seidel. *3D Acquisition of Mirroring Objects*. Graphical Models, vol. 67, no. 4, pages 233–259, July 2005.

[Tong 02] Xin Tong, Jingdan Zhang, Ligang Liu, Xi Wang, Baining Guo & Heung-Yeung Shum. *Synthesis of bidirectional texture functions on arbitrary surfaces*. In Proceedings of SIGGRAPH, pages 665–672, New York, NY, USA, 2002. ACM.

[Torrance 67] K.E. Torrance & E.M. Sparrow. *Theory for off-specular reflection from roughened surfaces*. Journal of the Optical Society of America, vol. 57, no. 9, pages 1105–1114, 1967.

[Tsin 01] Y. Tsin, V. Ramesh & T. Kanade. *Statistical calibration of CCD imaging process*. In Proceedings of International Conference on Computer Vision, volume 1, pages 480–487, 2001.

[Tsumura 03] Norimichi Tsumura, Nobutoshi Ojima, Kayoko Sato, Mitsuhiro Shiraishi, Hideto Shimizu, Hirohide Nabeshima, Syuuichi Akazaki, Kimihiko Hori & Yoichi Miyake. *Image-based skin color and texture analysis/synthesis by extracting hemoglobin and melanin information in the skin*. Proceedings of SIGGRAPH, vol. 22, no. 3, pages 770–779, 2003.

[Umeyama 04] Shinji Umeyama & Guy Godin. *Separation of Diffuse and Specular Components of Surface Reflection by Use of Polarization and Statistical Analysis of Images*. Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 5, pages 639–647, 2004.

[Wang 06] Jing Wang & Kristin J. Dana. *Relief Texture from Specularities*. Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 3, pages 446–457, 2006.

[Ward 92] Gregory J. Ward. *Measuring and modeling anisotropic reflection*. Proceedings of SIGGRAPH, vol. 26, no. 2, pages 265–272, 1992.

[Westin 92] Stephen H. Westin, James R. Arvo & Kenneth E. Torrance. *Predicting reflectance functions from complex surfaces*. In Proceedings of SIGGRAPH, pages 255–264, New York, NY, USA, 1992. ACM.

[Wilburn 08] B. Wilburn, H. Xu & Y. Matsushita. *Radiometric calibration using temporal irradiance mixtures*. In Proceedings of Conference on Computer Vision and Pattern Recognition, pages 1–7, 2008.

[Wolff 89a] Lawrence B. Wolff. *Material classification and separation of reflection components using polarization/radiometric information*. In Proceedings of a workshop on Image understanding workshop, pages 232–244, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.

[Wolff 89b] LB Wolff. *Using polarization to separate reflection components*. In Proceedings of Conference on Computer Vision and Pattern Recognition, pages 363–369, 1989.

[Woodham 80] R. J. Woodham. *Photometric Method for Determining Surface Orientation from Multiple Images*. Optical Engineering, vol. 19, no. 1, pages 139–144, jan/feb 1980.

[Wu 05] Tai-Pang Wu & Chi-Keung Tang. *Dense Photometric Stereo Using a Mirror Sphere and Graph Cut*. In Proceedings of Conference on Computer Vision and Pattern Recognition, volume 2, pages 140–147, 2005.

[Yu 05] Yizhou Yu & Johnny T. Chang. *Shadow Graphs and 3D Texture Reconstruction*. International Journal of Computer Vision, vol. 62, no. 1-2, pages 35–60, 2005.

[Zheng 00] Jiang Yu Zheng & Akio Murata. *Acquiring a Complete 3D Model from Specular Motion under the Illumination of Circular-Shaped Light Sources*. Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pages 913–920, 2000.

[Zickler 02] Todd Zickler, Peter N. Belhumeur & David J. Kriegman. *Helmholtz Stereopsis: Exploiting Reciprocity for Surface Reconstruction*. International Journal of Computer Vision, vol. 49, no. 2-3, pages 215–227, 2002.

[Zongker 99] Douglas E. Zongker, Dawn M. Werner, Brian Curless & David H. Salesin. *Environment matting and compositing*. In Proceedings of SIGGRAPH, pages 205–214, 1999.