# Real-time 2D manipulation of plausible 3D appearance using shading and geometry buffers

Carlos J. Zubiaga

## To cite this version:

Carlos J. Zubiaga. Real-time 2D manipulation of plausible 3D appearance using shading and geometry buffers. Graphics [cs.GR]. Université de Bordeaux, 2016. English. NNT : . tel-01440828

HAL Id: tel-01440828
https://inria.hal.science/tel-01440828
Submitted on 19 Jan 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Public Domain Mark 4.0 International License

# THESIS

PRESENTED AT

# UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE DE MATHMATIQUES ET D'INFORMATIQUE

par **Carlos Jorge Zubiaga Peña**

POUR OBTENIR LE GRADE DE

## DOCTEUR

SPÉCIALITÉ : INFORMATIQUE

---

# Real-time 2D manipulation of plausible 3D appearance using shading and geometry buffers

---

**Date de soutenance :** 7 November 2016

**Devant la commission d'examen compose de :**

| | | |
|---|---|---|
| Diego GUTIERREZ | Professeur, Universidad de Zaragoza | Rapporteur |
| Daniel SÝKORA | Professeur associé, Czech Technical University in Prague | Rapporteur |
| Pascal GUITTON | Professeur, Univeristé Bordeaux | Président |
| David VANDERHAEGHE | Maître de Conferences, Univeristé de Toulouse | Examinateur |
| Xavier GRANIER | Professeur, Institut d'Optique | Examinateur |
| Pascal BARLA | Chargé de recherche, Inria | Directeur |

– 2016 –

**Abstract** Traditional artists paint directly on a canvas and create plausible **appearances** of real-world scenes. In contrast, Computer Graphics artists define objects on a virtual scene (3D meshes, materials and light sources), and use complex algorithms (rendering) to reproduce their **appearance**. On the one hand, painting techniques permit to freely define appearance. On the other hand, rendering techniques permit to modify separately and dynamically the different elements that compose the scene.

In this thesis we present a middle-ground approach to manipulate **appearance**. We offer 3D-like manipulation abilities while working on the 2D space. We first study the impact on shading of materials as band-pass filters of lighting. We present a small set of local statistical relationships between material/lighting and shading. These relationships are used to mimic modifications on material or lighting from an artist-created image of a sphere. Techniques known as LitSpheres/MatCaps use these kinds of images to transfer their appearance to arbitrary-shaped objects. Our technique proves the possibility to mimic 3D-like modifications of light and material from an input artwork in 2D. We present a different technique to modify the third element involved on the visual appearance of an object: its geometry. In this case we use as input rendered images alongside with 3D information of the scene output in so-called auxiliary buffers. We are able to recover geometry-independent shading for each object surface, assuming no spatial variations for each recovered surface. The recovered shading can be used to modify arbitrarily the local shape of the object interactively without the need to re-render the scene.

**Keywords** Appearance, shading, pre-filtered environment map, MatCap, Compositing

**Résumé** Les artistes traditionnels peignent directement sur une toile et créent des apparences plausibles de scènes qui ressemblent au monde réel. A l'opposé, les artistes en informatique graphique définissent des objets dans une scène virtuelle (maillages 3D, matériaux et sources de lumière), et utilisent des algorithmes complexes (rendu) pour reproduire leur apparence. D'un côté, les techniques de peinture permettent de librement définir l'apparence. D'un autre côté, les techniques de rendu permettent de modifier séparément et dynamiquement les différents éléments qui définissent l'apparence.

Dans cette thèse, nous présentons une approche intermédiaire pour manipuler l'apparence, qui permettent certaines manipulations en 3D en travaillant dans l'espace 2D. Mous étudions d'abord l'impact sur l'ombrage des matériaux, tenant en compte des matériaux comme des filtres passe-bande d'éclairage. Nous présentons ensuite un petit ensemble de relations statistiques locales entre les matériaux / l'éclairage et l'ombrage. Ces relations sont utilisées pour imiter les modifications sur le matériaux ou l'éclairage d'une image d'une sphère créée par un artiste. Les techniques connues sons le nom de LitSpheres / MatCaps utilisent ce genre d'images pour transférer leur apparence à des objets de forme quelconque. Notre technique prouve la possibilité d'imiter les modifications 3D de la lumière et de matériaux à partir d'une image en 2D. Nous présentons une technique différente pour modifier le troisième élément impliqué dans l'aspect visuel d'un objet, sa géométrie. Dans ce cas, on utilise des rendus comme images d'entrée avec des images auxiliaires qui contiennent des informations 3D de la scène. Nous récupérons un ombrage indépendant de la géométrie pour chaque surface, ce qui nous demande de supposer qu'il n'y a pas de variations spatiales d'éclairage pour chaque surface. L'ombrage récupéré peut être utilisé pour modifier arbitrairement la forme locale de l'objet de manière interactive sans la nécessité de rendre à nouveau la scène.

**Mots-clés** Apparence, ombrage, cartes d'environnement pré-flitrées, MatCap, Compositing

**Laboratoire d'accueil** LaBRI

# Contents

# Chapter 1

# Introduction

One of the main goals of image creation in Computer Graphics is to obtain a picture which conveys a specific appearance. We first introduce the general two approaches of image creation in the Section 1.1, either by directly painting the image in 2D or by rendering a 3D scene. We also present middle-ground approaches which work on 2D with images containing 3D information. It is important to note that our work will take place using this middle-ground approach. We define our goal in Section 1.2 as 'granting 3D-like control over image appearance in 2D space'. Our goal emerges from the limitations of existing techniques to manipulate 3D appearance in existing images in 2D. Painted images lack any kind of 3D information, while only partial geometric information can be output by rendering. In any case, the available information is not enough to fully control 3D appearance. Finally in Section 1.3 we present the contributions brought by the thesis.

## 1.1 Context

Image creation can be done using different techniques. They can be gathered into two main groups, depending if they work in the 2D image plane or in a 3D scene. On the one hand, traditional painting or the modern digital painting softwares work directly in 2D by assigning colors to a plane. On the other hand, artists create 3D scenes by defining and placing objects and light sources. Then the 3D scene is captured into an image by a rendering engine which simulates the process of taking a picture. There also exist techniques in between that use 3D information into 2D images to create or modify the colors of the image.

### 1.1.1 Painting

Traditional artists create images of observed or imagined real-world scenes by painting. These techniques are based on the deposition of colored paint onto a solid surface. Artists may use different kinds of pigments or paints, as well as different tools to apply them, from brushes to sprays or even body parts. Our perception of the depicted scene depends on intensity and color variations across the planar surface of the canvas. Generated images may be abstract or symbolic, but we are interested in the ones that can be considered as natural or realistic. Artists are capable to depict plausible appearances of the different elements that compose a scene. The complexity of reality is well captured by the design of object's shape and color. Artists achieve good impressions of a variety of materials under different lighting environment. This can be seen in Figure 1.1, where different object are shown ranging from organic nature to hand-crafted.

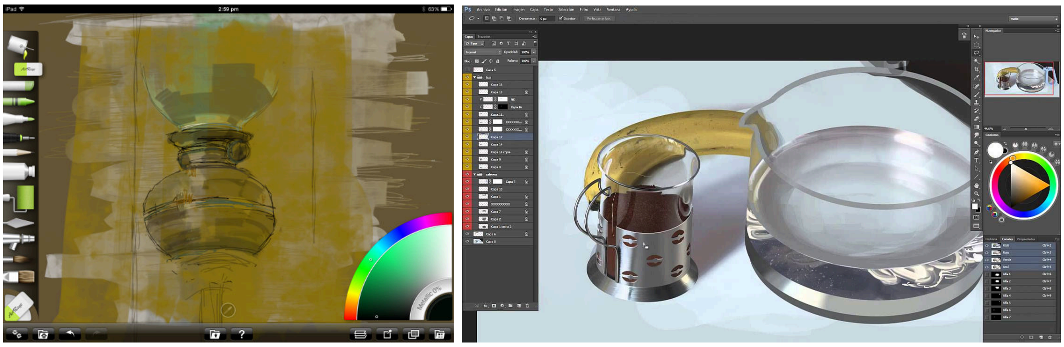
**Bodegon** by Francisco Zurbaran

**Still-life** by Pieter Caesz

**Attributes of Music** by Anne Vallayer-Coster

Figure 1.1: Still-life is a work of art depicting inanimate subjects. Artists are able to achieve a convincing appearance from which we can infer the material of the different objects.

Nowadays painting techniques have been integrated in computer system environments. Classical physical tools, like brushes or pigments, have been translated to digital ones (Figure 1.2). Moreover, digital systems provide a large set of useful techniques like the use of different layers, selections, simple shapes, etc. They also provide a set of image based operators that allow artists to manipulate color in a more complex way, like texturing, embossing or blurring. Despite the differences, both classical painting and modern digital systems share the idea of working directly in image space.



(a) Artrage$^{\text{TM}}$

(b) Photoshop$^{\text{TM}}$

Figure 1.2: Computer systems offer a complete set of tools to create images directly in image space. They provide virtual versions of traditional painting tools, such as different kinds of brushes or pigments, as can be seen in the interface of ArtRage$^{\text{TM}}$ (a). On the right (b) we can see the interface of one of the most well-known image editing softwares, Photoshop$^{\text{TM}}$. They also provide other tools that couldn't exist in traditional painting, like working on layers, different kind of selections or image modifications like scaling or rotations.

Artists are able to depict appearances that look plausible, in a sense that they look real even if they would not be physically correct. Despite our perception of the painted objects as if they were or could be real, artist do not control physical processes. They just manipulate colors either by painting them or performing image based operations. They use variations of colors to represent objects made of different materials and how they would behave under a different illumination. The use of achromatic variations is called *shading*; it is used to convey volume or light source variations (Figure 1.3), as well as material effects. Shading may also correspond to variations of colors, so we can refer to shading in a colored or in a grey scale image.

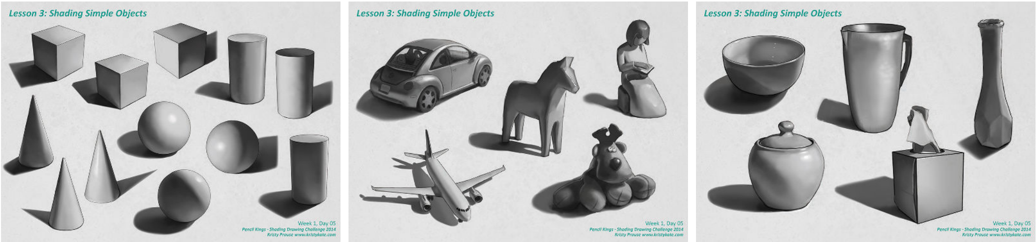
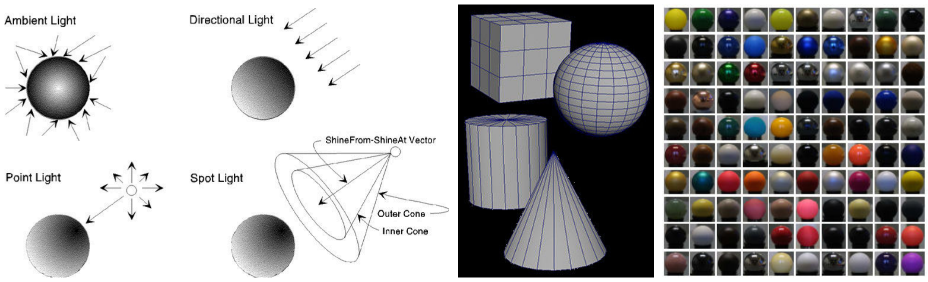


Figure 1.3: Shading refers to depicting depth perception in 3D models or illustrations by varying levels of darkness. It makes possible to perceive volume and infer lighting direction. Image are property of Kristy Kate http://kristykate.com/.

In real life, perceived color variations of an object are the result of the interaction between lighting and object material properties and shape. Despite the difficult understanding of these interactions, artists are able to give good impressions of materials and how they would look like under certain illumination conditions. However, once a digital painting is created it cannot be modified afterwards: shape, material, or lighting cannot be manipulated.

### 1.1.2 Rendering

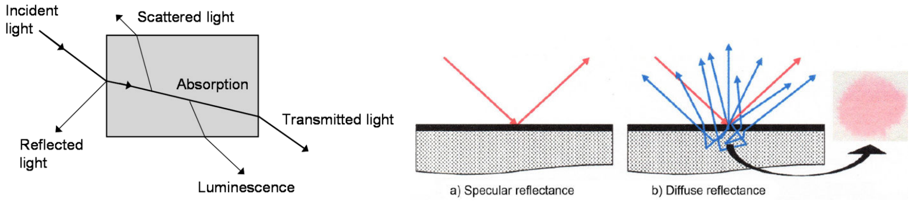
Contrary to 2D techniques, computer graphics provide an environment where artists define a scene based on physical 3D elements and their properties. Artists manipulate objects and light sources, they control object's shape (Fig. 1.4b) and material (Fig. 1.4c) and the type of light sources (Fig. 1.4a), as well as their positions. When an artist is satisfied with the scene definition, he selects a point of view to take a snapshot of the scene and gets an image as a result.



(a) Light sources (b) 3D meshes (c) Materials

Figure 1.4: A 3D scene are composed of lights and objects. where lights may vary in type (a) from ambient, to point, direction, area, etc. Objects are defined by their geometry defined by (b) 3D meshes and (c) materials.

The creation of 2D images from a 3D scene is called *rendering*. Rendering engines are software frameworks that use light transport processes to shade a scene. The theory of light transport defines how light is emitted from the light sources, how it interacts with the different objects of the scene and finally how it is captured in a 2d plane. In practice, light rays are traced from the point of view, per pixel in the image. When the rays reach an object surface, rays are either reflected, transmitted or absorbed, see Figure 1.5a. Rays continue their path until they reach a light source or they disappear by absorption, loss of energy or a limited number of reflections/refractions. At the same time, rays can also be traced from the light sources. Rendering engines usually mix both techniques by tracing rays from both directions, as shown in Figure 1.6.



(a) General material (b) Opaque material

Figure 1.5: In the general case, when a light ray reaches a object surface, it can be reflected, refracted or absorbed. When we focus on opaque objects the reflection can vary from shining (mirror) to matte (diffuse) by varying glossiness.

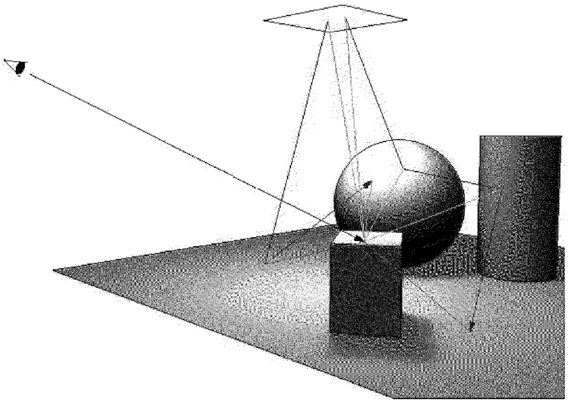


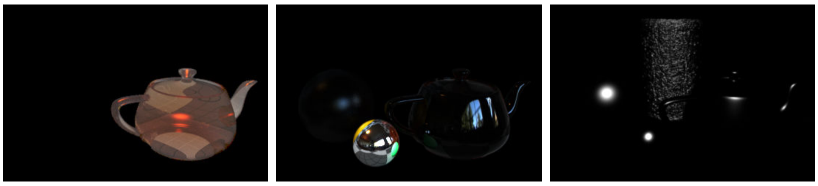
Figure 1.6: Rays may be both traced from the eye or image plane as well as from the light sources. When a ray reaches an object surface it is reflected, transmitted or absorbed.

Object geometry is defined by one or more 3D meshes composed of vertices, which form facets that describe the object surface. Vertices contain information about their 3D position, as well as other properties like their surface normal and tangent. The normal and tangent together describe a reference frame of the geometry at a local scale, which is commonly used in computer graphics to define how materials interact with lighting. This reference frame is used to define the interaction at a macroscopic level. In contrast, real-world interaction of light and a material at a microscopic level may turn out to be extremely complex. When a ray reaches a surface it can be scattered in any possible direction, rather than performing a perfect reflection. The way rays are scattered depends on the surface reflectance for opaque objects or the transmittance in the case of transparent or translucent objects. Materials are usually defined by analytical models with a few parameters; the control of those parameters allows artists to achieve a wide range of object appearances.

Manipulation of all the 3D properties of light, geometry and material allows artists to create images close to real-world appearances. Nevertheless, artists usually tweak images by manipulating shading in 2D until they reach the desired appearance. Those modifications are usually done for artistic reasons that require the avoidance of physically-based restrictions of the rendering engines, which make difficult to obtain a specific result. Artists usually start from the rendering engine output, from which they work to get their imagined desired image.

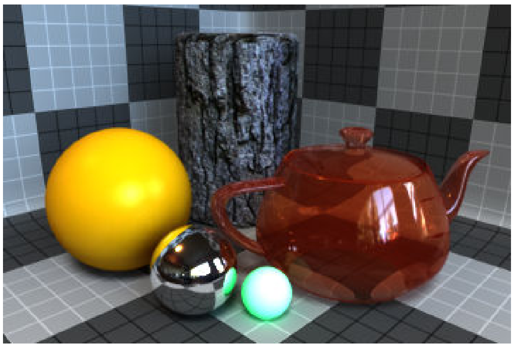
### 1.1.3 Compositing

Shading can be separated into components depending on the effects of the material. Commonly we can treat independently shading coming from the diffuse or the specular reflections (see Figure 1.5b), as well as from the transparent/translucent or the emission effects. Therefore, rendering engines can outputs images of the different components of shading independently. In the post-processing stage, called *compositing*, those images are combined to obtain the final image, as shown in Figure 1.7.



Transparent Reflection Specular

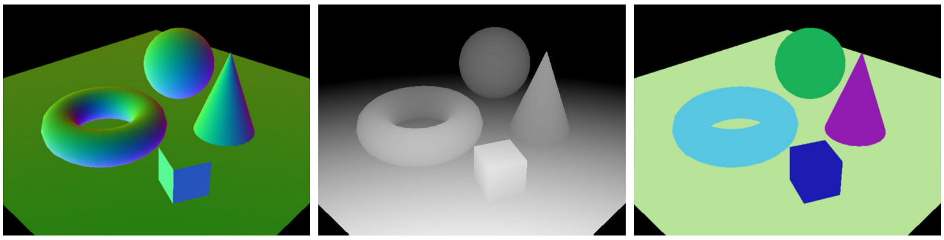
Subsurface Luminous Diffuse



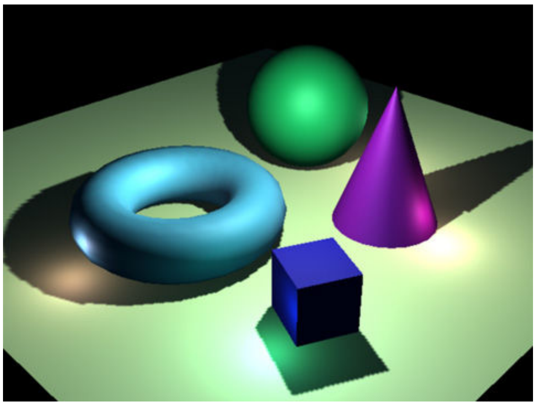
Final shading

Figure 1.7: Rendering engine computes shading per component: diffuse, reflections, transparency, etc. They generate per each component. Those images are used in post-process step called *compositing*. Final image is created as a combination of the different components. This figure shows an example from the software Modo$^{\mathrm{TM}}$.

In parallel with shading images, rendering engines have the capacity to output auxiliary buffers containing 3D information. In general, one can output any kind of 3D information, by assigning them to the projected surface of objects in the image. Usually those buffers are used to test errors for debugging, but they can be used as well to perform shading modifications in post-process. They can be used to guide modifications of shading: for instance, positions or depth information are useful to add fog or create focusing effects like depth of fields. Auxiliary buffers may also be used to add shading locally. Having information about positions, normals and surface reflectance are enough to create new shading by adding local light sources. This is similar to a technique called *Deferred Shading* used in interactive rendering engines. It is based on a multi-pass pipeline, where the first pass produces the necessary auxiliary buffers and the other passes produce shading by adding the contribution of a discrete set of lights, as is shown in Figure 1.8.



Normals Positions Diffuse color

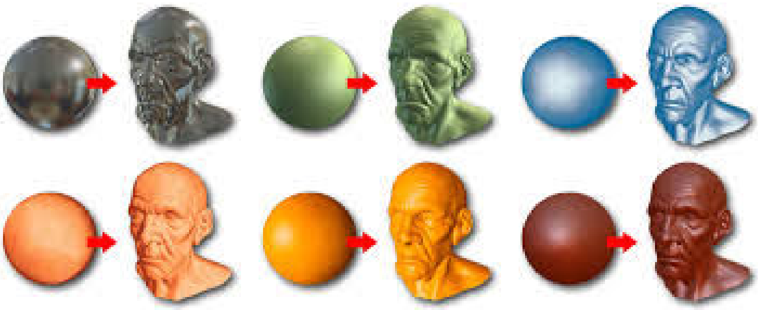


Render

Figure 1.8: Deferred shading computes in a first step a set of auxiliary buffers: positions, normals, diffuse color and surface reflectance. In a second pass those buffers are used to compute shading by adding the contribution of every single light source

Instead of computing shading at each pass we can pre-compute it, if we only consider distant illumination. Distant illumination means that there is no spatial variation on the incoming lighting, therefore it only depends on the surface normal orientation. Thanks to this approximation we only need surface normals to shade an object (usually they are used projected in screen space). Typically, pre-computed shading values per hemisphere direction are given by filtering the values of the environment lighting using the material reflectance properties. These techniques are referred by the name pre-filtered environment maps or PEM (see Chapter 2, Section 2.3). Different material appearances are obtained by using different material filters, as seen in Figure 1.9a. Pre-computed values are stored in spherical structures that can be easily accessed, shading is obtained by fetching using normal buffers. Instead of filtering an environment map, pre-computed shading values may also be created by painting or obtained from images (photographs or artwork). A well known technique, call the *LitSphere*, defines how to fill colors on a sphere from a picture and then use this sphere to shade an object, similarly to pre-filtered environment map techniques. The idea of LitSphere it's been extensively used in sculpting software where it takes the name of *MatCap* (see Figure 1.9b), as shorthand of Material Capture. MatCaps depict plausible materials under an arbitrary environment lighting. In the thesis we decided to use MatCaps instead of LitSpheres to avoid misunderstanding with non photo-realistic shading, like cartoon shading. Despite the limitations of distant lighting (no visibility effects like shadows or inter-reflections), they create convincing shading appearances.

(a) Pre-filtered envirenmont maps



(b) MatCaps

Figure 1.9: Both pre-filtered environment maps (a) and MatCaps (b) can be used to shade arbitrary objects. Shading color per pixel is assigned by fetching the color that corresponds to the same normal in the spherical representation.

### 1.1.4 Summary

On the one hand, painting techniques permit direct manipulation of shading with no restrictions, allowing artists to achieve the specific appearance they desire. In contrast, artists cannot manipulate dynamically the elements represented (object shape and material) and how they are lit. On the other hand, global illumination rendering engines are based on a complete control of the 3D scene and a final costly render. Despite the complete set of tools provided to manipulate a scene before rendering, artists commonly modify the rendering output in post-processing using image-based techniques similar to digital painting. Post-process modifications permit to avoid the physically based restrictions of the light transport algorithms.

As a middle-ground approach between the direct and static painting techniques and the dynamically controlled but physically-based restricted render engines, we find techniques which work in 2D and make use of 3D information in images or buffers. Those techniques may be used in post-process stage called *compositing*. Rendering engines can easily output image buffers with 3d properties like normal, positions or surface reflectance, which are usually called auxiliary buffers. Those buffers permit to generate or modify shading in ways different than digital painting, like the addition of local lighting or a few guided image operations (i.e. fog, re-texturing). Modifications of the original 3D properties (geometry, material or lighting) cannot be performed with a full modification on shading. A different way to employ auxiliary buffers is to use normal buffers alongside with pre-filtered environment maps or MatCaps/LitSpheres to shade objects. The geometry of the objects can be modified arbitrarily, but in contrast once pre-computed shading is defined, their depicted material and lighting cannot be modified.

## 1.2 Problem statement

Dynamic manipulation of appearance requires the control of three components: geometry, material and lighting. When we obtain an image independently of the way it has been created (painted or rendered) we lose the access to all components. **Geometry** is easily accessible, normal buffers may be generated by a rendering engine, but also may be obtained by scanners or estimated from images. **Material** are only accessible when we start from a 3D scene; the reflectance properties of the object can be projected to the image plane. **Lighting** in contrast is not accessible in any case. If we consider rendering engines, lighting is lost in the process of image creation. In the case of artwork, shading is created directly and information of lighting and materials is 'baked-in', therefore we do not have access to lighting or material separately.

Lighting structure is arbitrary complex and the incoming lighting per surface point varies in both the spatial and the angular domain, in other words, it varies per position and normal. The storage of the full lighting configuration is impractical, as we would need to store a complete environment lighting per pixel. Moreover, in the ideal case that we would have access to the whole lighting, the modification of the material, geometry or lighting will require a costly re-rendering process. In that case there will not be an advantage compared to rendering engine frameworks.

Our goal is to *grant 3D-like control of image appearance in 2D space.* We want to incorporate new tools to modify appearance in 2D using buffers containing 3D information. The objective is to be able to modify lighting, material and geometry in the image and obtain a plausible shading color. We develop our technique in 2 steps: first, we focus on the modification of light and material and then on the modification of geometry.

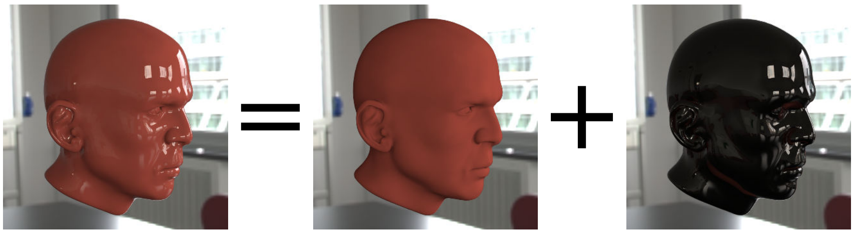
We base our work on the hypothesis that angular variations due to material and lighting can be mimicked by applying modifications directly on shading without having to decouple material and lighting explicitly. For that purpose we use structures similar to pre-filtered environment maps, where shading is stored independently of geometry.

In order to mimic material and lighting variations, we focus MatCaps. They are artist-created images of spheres, which their shading depicts an unknown plausible material under an unknown environment lighting. We want to add dynamic control over lighting, like rotation, and also to the material, like modifications of reflectance color, increasing or decreasing of material roughness or controlling silhouette effects.

In order to mimic geometry modifications, we focus on the compositing stage of the image creation process. Perturbations of normals (e.g. Bump mapping) is a common tool in computer graphics, but it is restricted to the rendering stage. We want to grant similar solutions of the compositing stage. In this stage several shading buffers are output by a global illumination rendering process and at the same time several auxiliary buffers are made available. Our goal in this scenario is to obtain a plausible shading for the modified normals without having to re-render the scene. The avoidance of re-rendering will permit to alter normals interactively.

As described in the previous section, material reflectance, and as a consequence shading, can be considered as the addition of specular and diffuse components. Following this approach we may separate the manipulation of diffuse from specular reflections, which is similar to control differently low-frequency and high-frequency shading content. This approach can be considered in both cases, the MatCap and the compositing stage, see Figure 1.10. Meanwhile rendering engines can output both components separately, MatCaps will require a pre-process step to separate them.

(a) MatCap



(b) Rendering

Figure 1.10: Shading is usually composed as different components. The main components are diffuse and specular reflections. We can see how (a) a MatCap and (b) a rendering are composed as the addition of a diffuse and a specular component.

## 1.3 Contributions

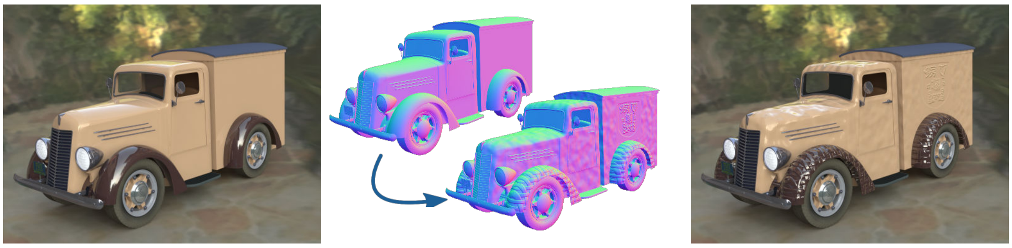
The work is presented in three main chapters that capture the three contributions of the thesis. Chapter 3 present a local statistical analysis of the impact of lighting and material on shading. We introduce a statistical model to represent surface reflectance and we use it to derive statistical relationships between lighting/material and shading. At the end of the chapter we validate our study by analyzing measured materials using statistical measurements.

In Chapter 4 we develop a technique which makes use of the statistical relationships to manipulate material and lighting in a simple scene: an artistic image of a sphere (MatCap). We show how to estimate a few statistical properties of the depicted material on the MatCap, by making assumptions on lighting. Then those properties are used to modify shading by mimicking modifications on lighting or material, see Figure 1.11.

(a) Input MatCap (b) Rotated lighting (c) Color change (d) Rougher look

Figure 1.11: Starting from a stylized image of a sphere (a) our goal is to vary the (b) lighting orientation, (c) the material color and (d) the material roughness.

Chapter 5 introduces a technique to manipulate local geometry (normals) at the compositing stage; we obtain plausible diffuse and specular shading results for the modified normals. To this end, we recover a single-view pre-filtered environment map per surface and per shading component. Then we show how to use these recovered pre-filetered environment maps to obtain plausible shading when modifications on normals are performed, see Figure1.12.



Reference rendering Original → modified normals Our recompositing result

Figure 1.12: Starting from shading and auxiliary buffers, our goal is to obtain a plausible shading color when modifying normals at compositing stage.

# Chapter 2

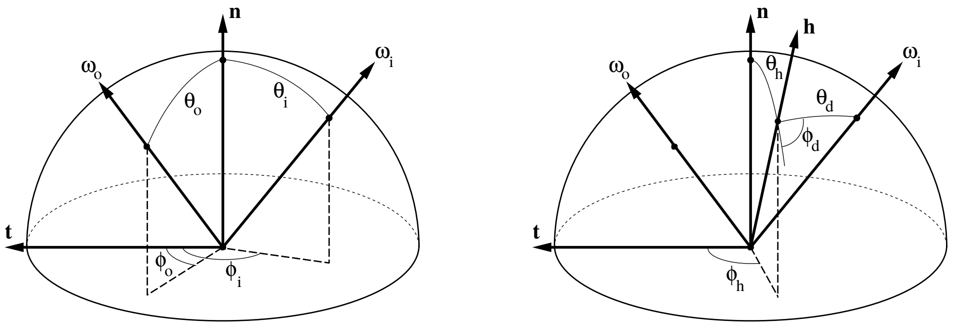
# Related Work

We are interested in the manipulation of shading in existing images. For that purpose we first describe the principles of rendering, in order to understand how virtual images are created as the interaction of geometry, material and lighting (Section 2.1). Given an input image a direct solution to modify its appearance is to recover the depicted geometry, material and lighting. These set of techniques are called inverse rendering (Section 2.2). Recovered components can be modified afterwards and a new rendering can be done. Inverse rendering is limited as it requires assumptions on lighting and materials which forbids its use in general cases. These techniques are restricted to physically-based rendering or photographs and they are not well defined to work with artworks. Moreover, a posterior rendering would limit the interactivity of the modification process. To avoid this tedious process, we found interesting to explore techniques that work with an intermediate representation of shading. Pre-filtered environment maps store the results of the interaction between material and lighting independently to geometry (Section 2.3). These techniques have been proven useful to shade objects in interactive applications, assuming distant lighting. Unfortunately there is no technique which permits to modify lighting or material once PEM are created.

Our work belongs to the domain of appearance manipulation. These techniques are based on the manipulation of shading without the restrictions of physically-based rendering (Section 2.4). However, the goal is to obtain images which appear plausible even if they are not physically correct. Therefore we also explore how the human visual system interprets shading (Section 2.5). We are interested into our capability to infer the former geometry, lighting and material form an image.

## 2.1 Shading and reflectance

We perceive objects by the light they reflect toward our eyes. The way objects reflect light depends on the material they are composed of. In the case of opaque objects it is described by their surface reflectance properties; incident light is considered either absorbed or reflected. Surface reflectance properties define how reflected light is distributed. In contrast, for transparent or translucent objects the light penetrates, scatters inside the object and eventually exists from a different point of the object surface. In computer graphics opaque object materials are defined by the Bidirectional Reflectance Distribution Functions (BRDF or $f_r$), introduced by Nicodemus [Nic65]. They are 4D functions of an incoming $\omega_i$ and an outgoing direction $\omega_o$ (e.g., light and view directions). The BRDF characterizes how much radiance is reflected in all lighting and viewing configurations, and may be considered as a black-box encapsulating light transport at a microscopic scale. Directions are classically parametrized by the spherical coordinates elevation $\theta$ and azimuth $\phi$ angles, according to the reference frame defined by the surface normal $\mathbf{n}$ and the tangent $\mathbf{t}$ as in Figure 2.1a.



(a) Classical parametrization (b) Half-vector parametrization

Figure 2.1: Directions $\omega_o$ and $\omega_i$ can be defined in the classical parametrization of elevation $\theta$ and azimuth $\phi$ angles (a). Or by the the halfvector $(\theta_h, \phi_h)$ and a difference vector $(\theta_d, \phi_d)$ (b). The vectors marked $\mathbf{n}$ and $\mathbf{t}$ are the surface normal and tangent, respectively.

In order to guarantee a physically correct behavior a BRDF must follow the next three properties. It has to be positive $f_r(\omega_i, \omega_o) \geq 0$. It must obey the Helmoth reciprocity: $f_r(\omega_i, \omega_o) = f_r(\omega_o, \omega_i)$ (directions may be swapped without reflectance being changed). It must conserve energy $\forall \omega_o, \int_\Omega f_r(\omega_i, \omega_o) \cos\theta_i \mathrm{d}\omega_i \leq 1$, the reflected radiance must be equal to or less than the input radiance.

Different materials can be represented using different BRDFs as shown in Figure 2.2, which shows renderings of spheres made of five different materials in two different environment illuminations in orthographic view. These images have been obtained by computing the reflected radiance $L_o$ for every visible surface point $\mathbf{x}$ toward a pixel in the image. Traditionally $L_o$ is computed using the reflected radiance equation, as first introduced by Kajiya [Kaj86] :

$$L_o(\mathbf{x}, \omega_o) = \int_\Omega f_r(\mathbf{x}, \omega_o, \omega_i)\ L_i(\mathbf{x}, \omega_i)\ \ \omega_i \cdot \mathbf{n}\ \ \mathrm{d}\omega_i\ , \tag{2.1}$$

with $L_o$ and $L_i$ are the reflected and incoming radiance, $\mathbf{x}$ a surface point of interest, $\omega_o$ and $\omega_i$ the outgoing and ingoing directions, $\mathbf{n}$ the surface normal, $f_r$ the BRDF, and $\Omega$ the upper hemisphere.

Thanks to the use of specialized devices (gonoireflectometers, imaging systems, etc.) we can measure real materials as the ratio of the reflected light from a discrete set of positions on the upper hemisphere. One of the most well-known databases of measured material is the MERL database [Mat03]. This database holds 100 measured BRDFs and displays a wide diversity of material appearances. All BRDFs are isotropic, which means light and view directions may be rotated around the local surface normal with no incurring change in reflectance. When measuring materials we are limited by a certain choice of BRDFs among real-world materials. We are also limited by the resolution of the measurements: we only obtain a discretized number of samples, and the technology involved is subject to measurement errors. Lastly, measured BRDFs are difficult to modify as we do not have parameters to control them. The solution to those limitations has been the modeling of material reflectance properties using analytical functions.

Analytical models have the goal to capture the different effects that a material can produce. The ideal extreme cases are represented by mirror and matte materials. On the one hand, mirror materials reflect radiance only in the reflection direction $\omega_r = 2\ (\omega \cdot \mathbf{n})\ \mathbf{n} - \omega$. On the other hand, matte or lambertian materials reflect light in a uniform way over the whole hemisphere $\Omega$. However, Real-world material are much more complex, they exhibit a composition of different types of reflection. Reflections vary from diffuse to mirror and

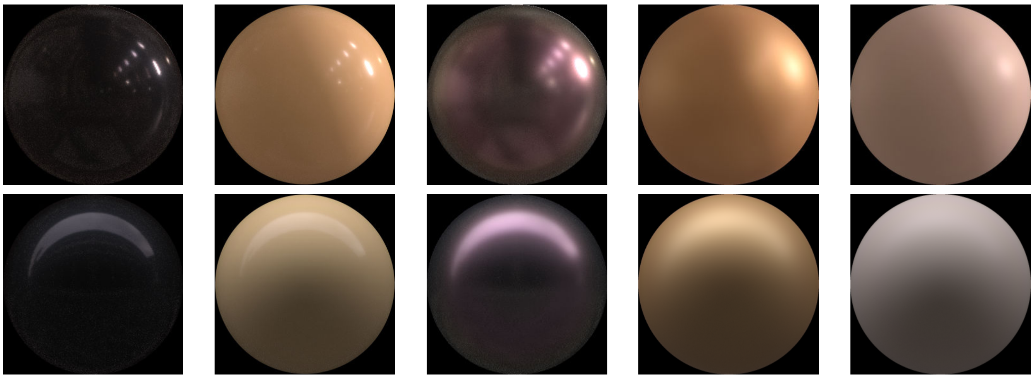


Figure 2.2: Renderings of different BRDF coming from the MERL database (from left to right: `specular-black-phenolic`, `yellow-phenolic`, `color-changing-paint2`, `gold-paint` and `neoprene-rubber`) under two different environment maps (upper row: `galileo`; lower row: `uffizi`). Each BRDF has a different effect on the reflected image of the environment.

therefore materials exhibit different aspects in terms of roughness or glossiness. Materials define the mean direction of the light reflection, it can be aligned with the reflected vector or be shifted like off-specular reflections or even reflect in the same direction (retro-reflections). Materials can also reproduce Fresnel effects which characterize variations on reflectance depending on the viewing elevation angle, making objects look brighter at grazing angles. Variations when varying the view around the surface normals are captured by anisotropic BRDFs. In contrast, isotropic BRDFs imply that reflections are invariant to variations of azimuthal angle of both $\omega_o$ and $\omega_i$. BRDFs may be grouped by empirical models: they mimic reflections using simple formulation; or physically based models: they are based on physical theories. Commonly BRDFs are composed of several terms, we are interested in the main ones: a diffuse and a specular component. The diffuse term is usually characterized with a lambertian term, nevertheless there exist more complex models like Oren-Nayar [ON94].

Regarding specular reflections, the first attempt to characterize them has been defined by Phong [Pho75]. It defines the BRDF as a cosine lobe function of the reflected view vector and the lighting direction, whose spread is controlled by a single parameter. It reproduces radially symmetric specular reflections and does not guarantee energy conservation. An extension of the Phong model has been done in the work of Lafortune et al. [LFTG97] which guarantees reciprocity and energy conservation. Moreover it is able to produce more effects like off-specular reflections, Fresnel effect or retro-reflection. Both models are based on the reflected view vector. Alternatively there is a better representation for BRDFs based on the half vector $h = \frac{(\omega_o+\omega_i)}{||\omega_o+\omega_i||}$, and consequently the ‘difference’ vector, as the ingoing direction in a frame which the halfway vector is at the north pole, see Figure 2.1b. It has been formally described by Rusinkewicz [Rus98]. Specular or retro reflections are better defined in this parametrization as they are aligned to the transformed coordinate angles. Blinn-Phong [Bli77] redefined the Phong model by using the half vector instead of the reflected vector. The use of the half vector produces asymmetric reflections in contrast to the Phong model. Those model, Phong, Lafortune and Blinn-Phong are empirical based on cosine lobes. Another empirical model, which is commonly used, is the one defined by Ward [War92]. This model uses the half vector and is based on Gaussian Lobes. It is designed to reproduce anisotropic reflections and to fit measured reflectance, as it was introduced alongside with a measuring device.

The most common physically-based models are the ones who follow the micro-facet theory. This theory assumes that a BRDF defined for a macroscopic level is composed by a set of micro-facets. The Torrance-Sparrow model [TS67] uses this theory by defining the BRDF as:

$$f_r(\omega_o, \omega_i) = \frac{G(\omega_o, \omega_i, h)D(h)F(\omega_o, h)}{4|\omega_o n||\omega_i n|}, \tag{2.2}$$

where $D$ is the Normals distributions, $G$ is the Geometric attenuation and $F$ is the Fresnel factor. The normal distribution function $D$ defines the orientation distribution of the micro-facets. Normal distributions often use Gaussian-like terms as Beckmann [BS87], or other distributions like GGX [WMLT07]. The geometric attenuation $G$ accounts for shadowing or masking of the micro-facets with respect to the light or the view. It defines the portion of the micro-facets that are blocked by their neighbor micro-facets for both the light and view directions. The Fresnel factor $F$ gives the fraction of light that is reflected by each micro-facet, and is usually approximated by the Shlick approximation [Sch94].

To understand how well real-world materials are simulated by analytical BRDFs we can fit the parameters of the latter to approximate the former. Ngan et al. [NDM05] have conducted such an empirical study, using as input measured materials coming from the MERL database [Mat03]. It shows that a certain number of measured BRDFs can be well fitted, but we still can differentiate them visually when rendered (even on a simple sphere) when comparing to real-world materials.

The use of the reflected radiance equation alongside with the BRDF models tell us how to create virtual images using a forward pipeline. Instead we want to manipulate existing shading. Moreover we want those modifications to behave in a plausible way. The goal is to modify shading in image space as if we were modifying the components of the reflectance radiance equation: material, lighting or geometry. For that purpose we are interested in the impact of those components in shading.

## 2.2 Inverse rendering

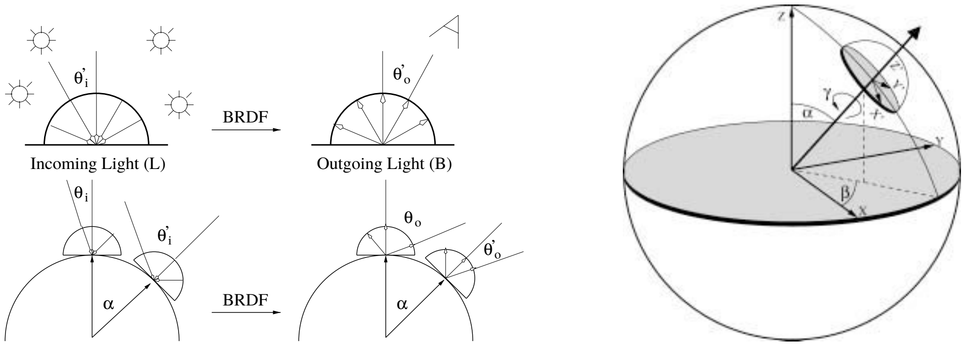
An ideal solution to the manipulation of appearance from shading would be to apply inverse-rendering. It consists in the extraction of the rendering components: geometry, material reflectance and environment lighting, from an image. Once they are obtained they can be modified and then used to re-render the scene, until the desired appearance is reached. In our case we focus on the recovery of lighting and material reflectance assuming known geometry. Inverse rendering has been a long-standing goal in Computer Vision with no easy solution. This is because material reflectance and lighting properties are of high dimensionality, which makes their recovery from shading an under-constrained problem.

Different combinations of lighting and BRDFs may obtain similar shading. The reflection of a sharp light on a rough material would be similar to a blurry light source reflected by a shiny material. At specific cases it is possible to recover material and/or lighting as described in [RH01b]. In the same paper the authors show that interactions between lighting and material can be described as a 2D spherical convolution where the material acts a low-pass filter of the incoming radiance. This approach requires the next assumptions: Convex curved object of uniform isotropic material lit by distant lighting. These assumptions make radiance dependent only on the surface orientation, different points with the same normal sharing the same illumination and BRDF. Therefore the reflectance radiance Equation (2.1) may be rewritten using a change of domain, by obtaining the rotation which transform the surface normal to the $z$ direction. This rotations permits to easily transforms directions in local space to global space, as shown in Figure 2.3 for the 2D and the 3D case. They rewrite

Equation (2.1) as a convolution in the spherical domain:

$$\begin{aligned} L_o(R, \omega'_o) &= \int_{\Omega'} \hat{f}_r(\omega'_i, \omega'_o)\ L_i(R\omega'_i)\mathrm{d}\omega'_i \\ &= \int_{\Omega} \hat{f}_r(R^{-1}\omega_i, \omega'_o)\ L_i(\omega_i)\mathrm{d}\omega_i \\ &= \hat{f}_r * L, \end{aligned}$$

where $R$ is the rotation matrix which transforms the surface normal to the $z$ direction. Directions $\omega_o$, $\omega_i$ and the domain $\Omega$ are primed for the local space and not primed on the global space. $\hat{f}_r$ indicates the BRDF with the cosine term encapsulated. The equation is rewritten as a convolution, denoted by $*$.



2D scheme

Complete 3D scheme

Figure 2.3: Different orientations of the surface correspond to rotations of the upper hemisphere and BRDF, with global directions (not primed) corresponding to local directions (primed).

Ramamoorthi et al. used the convolution approximation to study the reflected radiance equation in the frequency domain. For that purpose they use Fourier basis functions in the spherical domain, which correspond to Spherical Harmonics. They are able to recover lighting and BRDF from an object with these assumptions using spherical harmonics. Nevertheless this approach restricts the BRDF to be radially symmetric like: lambertian, Phong or re-parametrized micro-facets BRDF to the reflected view vector.

Lombardi et al. [LN12] manage to recover both reflectance and lighting, albeit with a degraded quality compared to ground truth for the latter. They assume real-world natural illumination for the input images which permits to use statistics of natural images with a prior on low entropy on the illumination. The low entropy is based on the action of the BRDF as a bandpass filter causing blurring: they show how histogram entropy increase for different BRDFs. They recovered isotropic directional statistics BRDFs [NL11] which are defined by a set of hemispherical exponential power distributions. This kind of BRDF is made to represent the measured materials of the MERL database [Mat03]. The reconstructed lighting environments exhibit artifacts (see Figure 2.4), but these are visible only when re-rendering the object with a shinier material compared to the original one.

In their work Lombardi et al. [LN12] compare to the previous work of Romeiro et al [RZ10]. The latter gets as input a rendered sphere under an unknown illumination and extracts a monochromatic BRDF. They do not extract the lighting environment which restricts its use to re-use the BRDF under a different environment and forbids the manipulation of the input image. Similar to the work of Lombardi they use priors on natural lighting, in this case they study statistics of a set of environment maps projected in the Haar wavelet

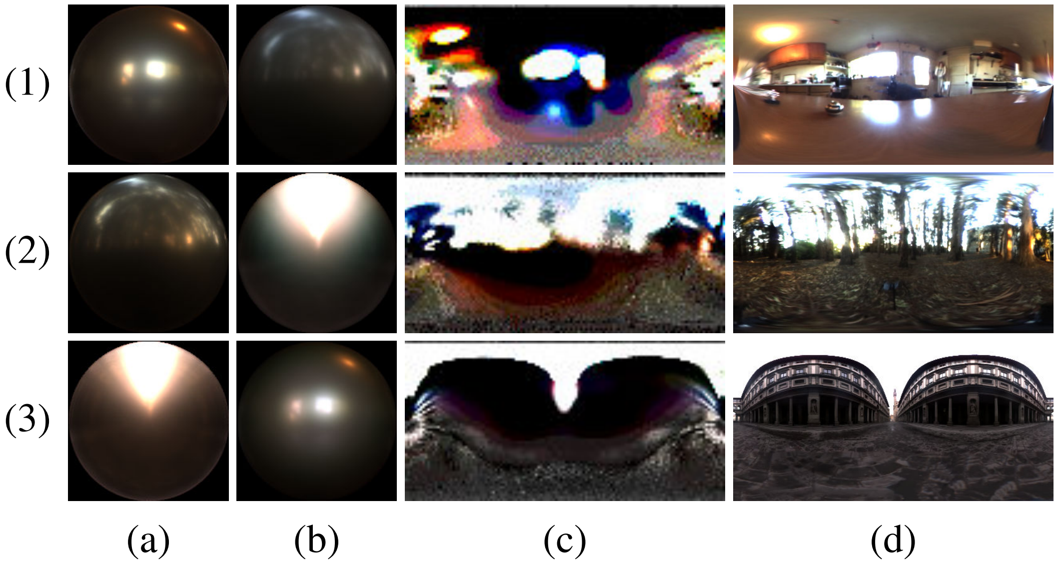


Figure 2.4: Results of the alum-bronze material under three lighting environments using Lombardi et al. [LN12] method. Column (a) shows the ground truth alum-bronze material rendered with one of the three lighting environments, column (b) shows a rendering of the estimated BRDF with the next ground truth lighting environment, column (c) shows the estimated illumination map and column (d) shows the ground truth illumination map. The lighting environments used were `Kitchen` (1), `Eucalyptus Grove` (2), and the `Uffizi Gallery` (3). The recovered illumination is missing high frequency details lost at rendering.

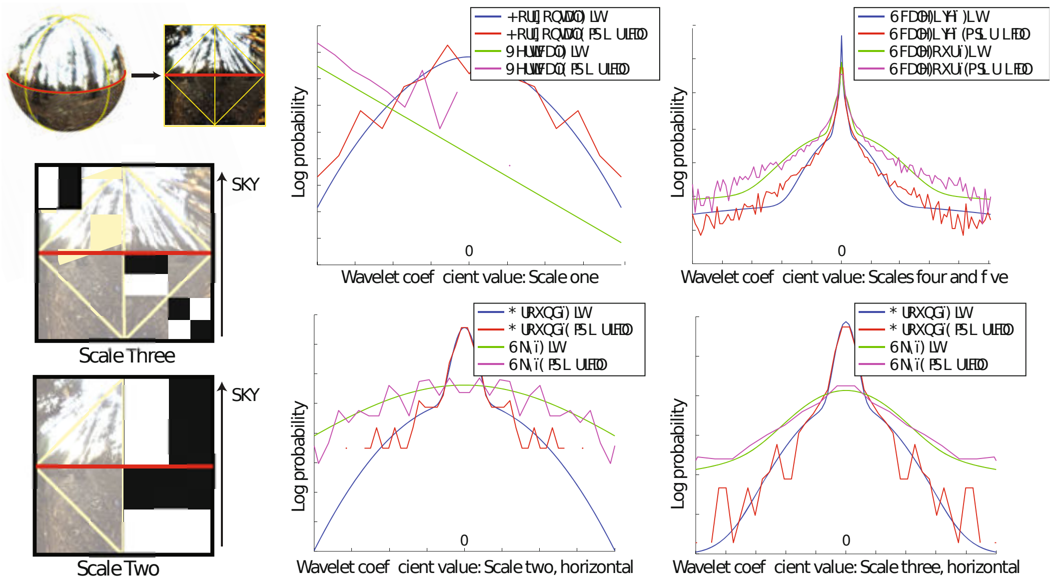


Figure 2.5: Left: lighting is represented using a Haar wavelet basis on the octahedral domain discretized to 32 × 32. Statistics of wavelet coefficients are non-stationary, so distinct distributions for coefficients above and below the horizon are fitted. Right: Empirical distributions and their parametric fits for a variety of wavelet coefficient groups.

basis, see Figure 2.5. Those statistics are used to find the most likely reflectance under the studied distribution of probable illumination environments. The type of recovered BRDF is defined in a previous work of the same authors [RVZ08]. That work recovers BRDFs using rendered spheres under a known environment map. They restrict BRDFs to be isotropic and they add a further symmetry around the incident plane, which permits to rewrite the BRDF as a 2D function instead of the general 4D function.

Other methods that perform BRDF estimation always require a set of assumptions, such as single light sources or controlled lighting. The work from Jaroszkiewicz [Jar03] assumes a single point light. It extracts BRDFs from a digitally painted sphere using homomorphic factorization. Ghosh et al. [GCP+09] uses controlled lighting based on spherical harmonics. This approach reconstructs spatially varying roughness and albedo of real objects. It employs 3D moments (in Cartesian space) up to order 2 to recover basic BRDF parameters from a few views. Aittala et al. [AWL13] employs planar Fourier lighting patterns projected using a consumer-level screen display. They recover Spatially Varying-BRDFs of planar objects.

As far as we know there is no algorithm that works in a general case and extracts a manipulable BRDF alongside with the environment lighting. Moreover, as we are interested in the manipulation of appearance in an interactive manner, re-rendering methods are not suitable. A re-rendering process uses costly global illumination algorithms once material and lighting are recovered. In contrast, we expect that manipulation of shading does not require to decouple the different terms involved in the rendering equation. Therefore, we rather apply approximate but efficient modifications directly to shading, mimicking modifications of the light sources or the material reflectance. Moreover, all these methods work on photographs; in contrast we also want to manipulate artwork images.

## 2.3 Pre-filtered lighting

Pre-filtered environment maps [KVHS00] take an environment lighting map and convolve it with a filter defined by a material reflectance. The resulting values are used to shade arbitrary geometries in an interactive process, giving a good approximation of reflections. Distant lighting is assumed, consequently reflected radiance is independent of position. In the general case a pre-filtered environment would be a 5 dimensional function, depending on the outgoing direction $\omega_o$, and on the reference frame defined by the normal $\mathbf{n}$ and the tangent $\mathbf{t}$. Nevertheless some dependencies can be dropped. Isotropic materials are independent of the tangent space. Radially symmetric BRDFs around either the normal (e.g. lambertian) or the reflected view vector (e.g. Phong) are 2 dimensional functions. When the pre-filtered environment maps are reduced to a 2 dimensional function they can be stored in a spherical maps. A common choice is the dual paraboloid map [HS98], which is composed of a front and a back image with the z value given by $1/2-(x^2+y^2)$. This method is efficient in terms of sampling and the introduced distortion is small, see Figure 2.6.

Unfortunately effects dependent on the view direction, like Fresnel, cannot be captured in a single spherical representation as in the last mentioned technique. Nevertheless it can be added afterwards, and several pre-filtered environment maps can be combined with different Fresnel functions. A solution defined by Cabral et al. [CON99] constructs a spare set of view-dependent pre-filtered environment maps. Then, for a new viewpoint they dynamically create a view-dependent pre-filtered environment map by warping and interpolating pre-computed environment maps.

A single view-dependent pre-filtered environment map is useful when we want to have a non expensive rendering for a fixed view direction. Sloan et al. [SMGG01] introduce a technique which creates shaded images of spheres from paintings, which can be used as

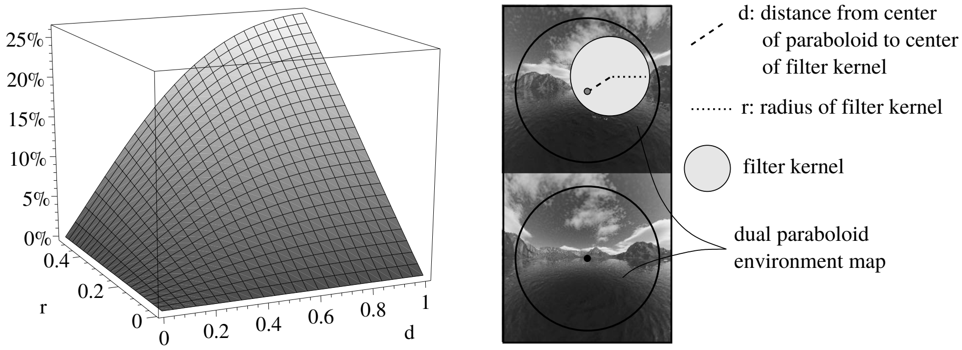
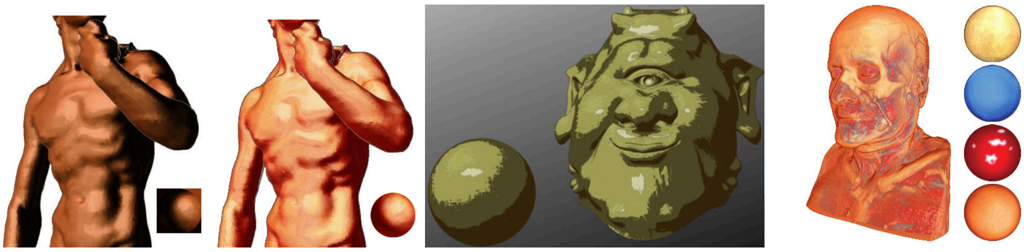


Figure 2.6: Distortion of a circle when projected from a paraboloid map back to the sphere.



(a) Sloan et al. [SMGG01]

(b) Todo et al. [TAY13]

(c) Bruckner [BG07]

Figure 2.7: Renderings using LitSphere: (a) the original LitSphere technique, (b) a non-photorealistic approach and (c) a technique focused on scientific illustration

view-dependent pre-filtered environment maps (Fig. 2.7a). This technique differs from the previous as it generates shading directly without specifying an environment map and a BRDF. It has been proven useful for non-photorealistic approaches like [TAY13] as well as for technique which convey plausible materials (Fig. 2.7b). The latter is extensively used in sculpting software auch as ZBrush or MudBox, where it is usually called MatCaps (a shorthand for 'Material Capture"). It has been as well shown useful for scientific illustration [BG07] (Fig. 2.7c).

Rammamorthi et al. have shown how diffuse pre-filetered environment maps are well approximated by 9 Spherical Harmonics coefficients [RH01a] corresponding to the lowest-frequency modes of the illumination. It is proven that the resulting values differ on average 1% of the ground truth. For that purpose they project the environment lighting in the first 2 orders of spherical harmonics, which is faster than applying a convolution with a diffuse-like filter. The diffuse color is obtained by evaluating a quadratic polynomial in Cartesian space using the surface normal.

Pre-filtered lighting maps store appearance independently of geometry for distant lighting. This permits to easily give the same material appearance to any geometry, for a fixed material and environment lighting. As a drawback, when we want to modify the material or the environment lighting they need to be reconstructed, which forbids interactive appearance manipulation. In the case of the artwork techniques, LitSpheres/MatCaps are created for a single view, which forbids rotations as shading is tied to the camera.

## 2.4 Appearance manipulation

The rendering process is designed to be physically realistic. Nevertheless, sometimes we want to create images with a plausible appearance without caring about their physically correctness. There exist some techniques which permit different manipulations of appearance using different kinds of input, ranging form 3D scenes to 2D images. Those techniques reproduce visual cues of the physically-based image creation techniques but without being restricted by them. At the same time they take advantage of the inaccuracy of human visual system to distinguish plausible from accurate.

Image-based material editing of Khan et al. [KRFB06] takes as input a single image in HDR of a single object and is able to change its material. They estimate both, the geometry of the object, and the environment lighting. Then estimated geometry and environment lighting are used alongside with a new BRDF to re-render the object. Geometry is recovered following the heuristic of darker is deeper. Environment lighting is reconstructed from the background. First the hole left by the object is filled with other pixels from the image, to preserve the image statistics. Then the image is extruded to form a hemisphere. The possible results range from modifications of glossiness, texturing of the object, replacement of the BRDF or even simulation of transparent or translucent objects, see Figure 2.8.

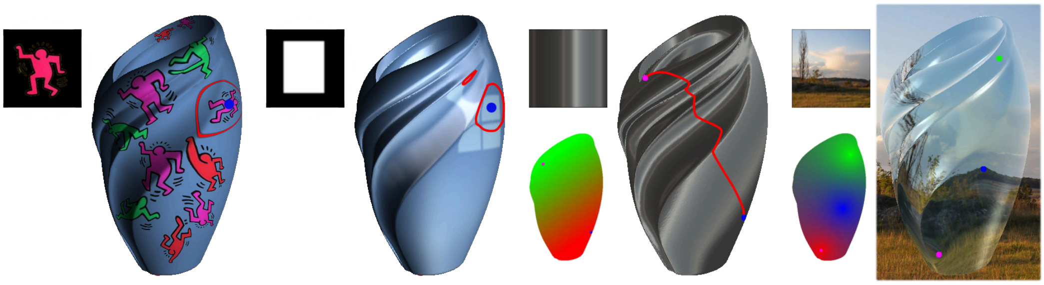
Figure 2.8: Given a high dynamic range image such as shown on the left, the Image Based Material Editing technique makes objects transparent and translucent (left vases in middle and right images), as well as apply arbitrary surface materials such as aluminium-bronze (middle) and nickel (right).

The interactive reflection editing system of Ritschel et al. [ROTS09] makes use of a full 3D scene to directly displace reflections on top of object surfaces, see Figure 2.9. The method takes inspiration on paintings where it is common to see refections that would not be possible in real life, but we perceive them as plausible. To define the modified reflections the user define constraints consisting on the area where he wants the reflections and another area which defines the origin of the reflections. This technique allows users to move reflections, adapt their shape or modify refractions.

The Surface Flows method [VBFG12] warps images using depth and normal buffers to create 3D shape impressions like reflections or texture effects. In this work they performed a differential analysis of the reflectance radiance Equation (2.1) in image space. From that differentiation of the equation they identify two kind of variations: a first order term related to texturing (variations on material) and a second order variation related to reflection (variations on lighting). Furthermore they use those variations to define empirical equations to deform pixels of an image following the derivatives of a depth buffer in the first case and the derivatives of a normal buffer in the second case. As a result they introduce a set of tools: addition of decal textures or reflections and shading using gradients or images (Fig. 2.10).

The EnvyLight system [Pel10] permits to make modifications on separable features of the environment lighting by selecting them from a 3D scene. Users make scribbles on rendered image of the scene to differentiate the parts that belong to a lighting feature from the ones that do not. The features can be diffuse reflections, highlights or shadows. The geometry

Figure 2.9: Users can modify reflections using constraints. (a) The mirror reflection is changed to reflect the dragon's head, instead of the tail. (b) Multiple constraints can be used in the same scene to make the sink reflected on the pot. (c) The ring reflects clearer the face by using two constraints. (d) The reflection of the tree in the hood is modified and at the same time the highlight on the door is elongated.



(a) Brush tool textures (b) Brush tool shading (c) Gradient tool (d) Image tool

Figure 2.10: Tools using Surface Flows. Deformed textures (a) or shading patterns (b) are applied at arbitrary sizes (red contours) and locations (blue dots). (c) Smooth shading patterns are created by deforming a linear gradient (red curve). Two anchor points (blue and red dots) control its extremities. (d) A refraction pattern is manipulated using anchor points. Color insets visualize weight functions attached to each anchor point.

of the zones containing the feature permits to divide the environment map on the features that affect those effect from the rest. The separation of the environment lighting permits to edit them separately as well as to make other modifications like: contrast, translation, blurring or sharpening, see Figure 2.11.

Appearance manipulation techniques are designed to help artists achieve a desired appearance. To this end they might need to evade from physical constraints in which computer graphics is based. Nevertheless, obtained appearance might still remain plausible for the human eye. As artists know intuitively that the human visual system is not aware of how light physically interacts with objects.

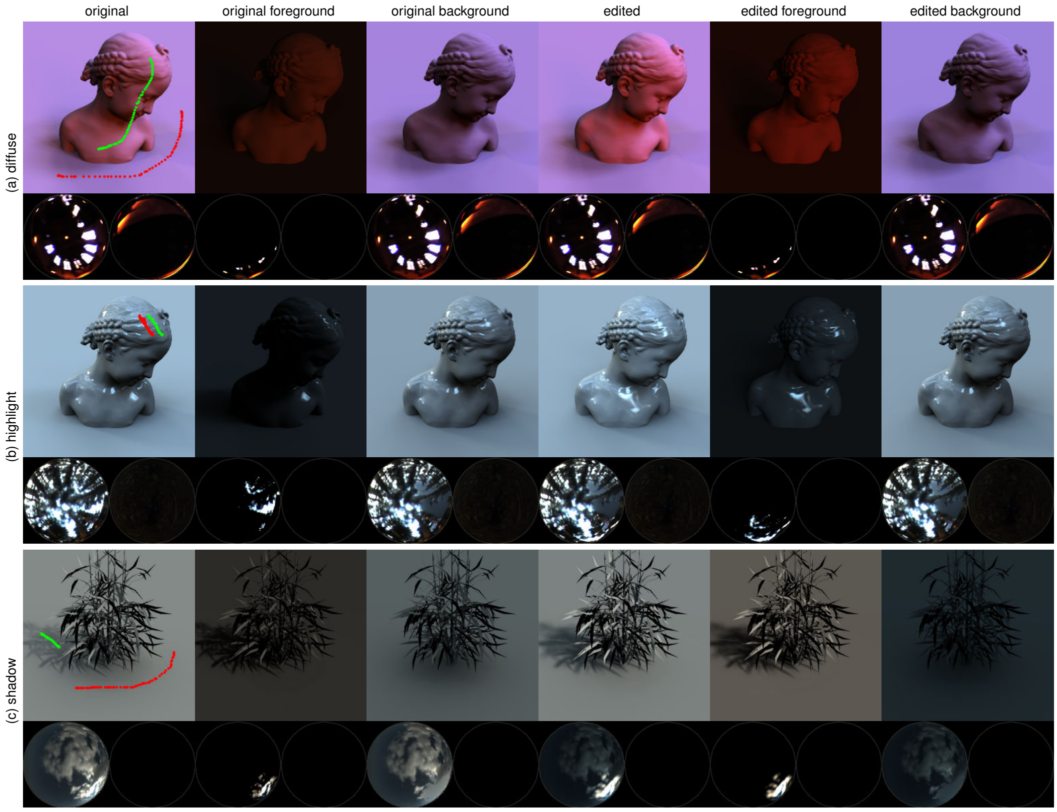
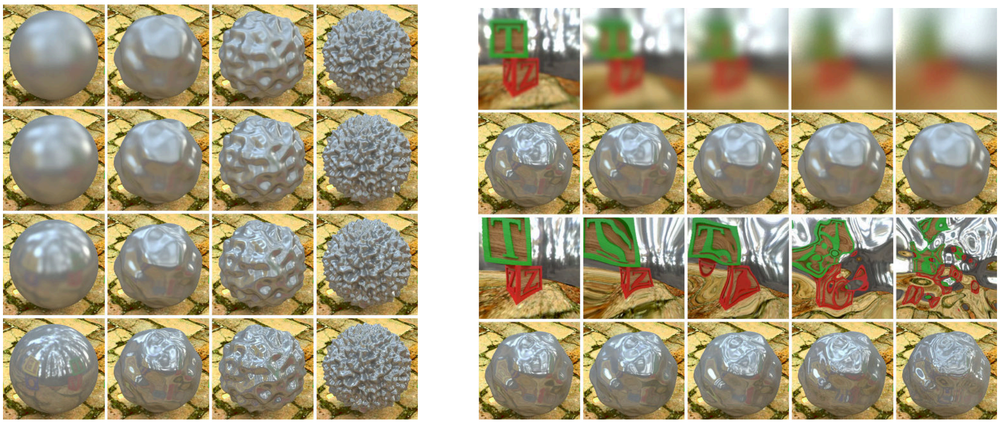


Figure 2.11: Example edits performed with envyLight. For each row, we can see the environment map at the bottom, as top and bottom hemispheres, and the corresponding rendered image at the top. Designers mark lighting features (e.g. diffuse gradients, highlights, shadows) with two strokes: a stroke to indicate parts of the image that belong to the feature (shown in green), and another stroke to indicate parts of the image that do not (shown in red). envyLight splits the environment map into a foreground and background layer, such that edits to the foreground directly affect the marked feature and such that the sum of the two is the original map. Editing operations can be applied to the layers to alter the marked feature. (a) Increased contrast and saturation of a diffuse gradient. (b) Translation of a highlight. (c) Increased contrast and blur of a shadow.

## 2.5 Visual perception

Created or manipulated images are evaluated with respect to a ‘reference’ image (e.g. photograph, ground truth simulation). Measurements of *Visual Quality* consist in computing the perceived fidelity and similarity or the perceived difference between an image and the ‘reference’. Traditionally numerical techniques like MAE (mean absolute error), MSE (mean square error), or similar have been used to measure signal fidelity in images. They are used because of their simplicity and because of their clear physical meaning. However, those metrics are not good descriptors of human visual perception. In the vast majority of cases human beings are the final consumer of images and we judge them based on our perception. Visual perception is an open domain of research which presents many challenging problems. In computer graphics perception is very useful when a certain appearance is desired, without relying completely on physical control. A survey of image quality metrics from traditional numeric to visual perception approaches is provided in [LK11].

Ramanarayanan et al. [RFWB07] have shown how the human visual system is not able to perceive certain image differences. They develop a new metric for measuring how we judge images as visually equivalent in terms of appearance. They prove that we are not mostly able to detect variations on environment lighting. Users judge the equivalence of two objects that can vary in terms of bumpiness or shininess, see Figure 2.12. Objects are rendered under transformations (blurring or warping) of the same environment lighting. The results prove that we judge images as equivalent, despite their visual difference. This limitation of the human visual system is used in computer graphics to design techniques of appearance manipulation, like shown in the previous section.


(a) Objects varying in bumpiness and roughness

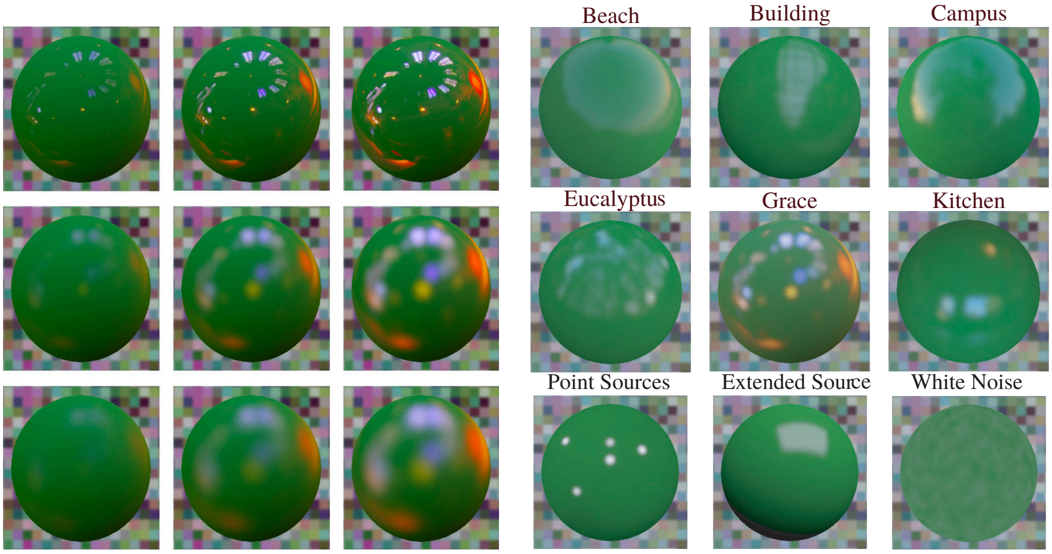
(b) Env. Lighting modified by warping or blurring

Figure 2.12: (a) Objects varying in bumpiness from left to right and the roughness of the material increase vertically. (b) Environment lighting map is blurred in the upper image and warped in the bottom row.

Despite the tolerance of the human visual system to visual differences we are able to differentiate image properties of objects. To distinguish the material of an object we use visual cues like color, texture or glossiness. The latter is often defined as the achromatic component of the surface reflectance. In a BRDF, gloss is responsible for changes in the magnitude and spread of the specular highlight as well as the change in reflectance that occurs as light moves away from the normal toward grazing angles. Hunter [Hun75] introduced six visual properties of gloss: specular gloss, sheen, luster, absence-of-bloom, distinctness-of-image and surface-uniformity. He suggests that, except for surface-uniformity, all of these visual properties may be connected to reflectance (i.e., BRDF) properties. There exists standard test methods for measuring some of these properties (such as ASTM D523, D430 or D4039).

The measurements of Hunter as well as the standard methods are based on optical measurements of reflections. However, perceived glossiness does not have a linear relationships with physical measurements. The work of Pellacini [PFG00] re-parametrized the Ward model [War92] to vary linearly in relation to human perceived glossiness. Wills et al. [WAKB09] performs an study from the isotropic BRDF database of MERL. From them they create a 2D perceptual space of gloss.

Moreover, perceived reflectance depends on the environment lighting around an object. The work of Doerschner et al. [DBM10] tries to quantify the effect of the environment lighting on the perception of reflectance. They look for a transfer function of glossiness between pairs of BRDF glossiness and environment lighting. Fleming et al. [FDA03] perform a series of experiments about the perception of surface reflectance under natural illumination. Their experiments evaluate how we perceive variation on specular reflectance and roughness of a material under natural and synthetic illuminations, see Figure 2.13. Their results show that we estimate better material properties under natural environments. Moreover they have tried to identify natural lighting characteristics that help us to identify material properties. Nevertheless, they show that our judgment of reflectance seems to more related to certain lighting features than to global statistics of the natural light.



(a) Materials vary on roughness and specularity (b) Natural or analytical environment lighting

Figure 2.13: (a) Rendered spheres are shown by increasing roughness from top to bottom, and by increasing specular reflectance from left to right. The scale of these parameters are re-scaled to fit visual perception as proposed by Pellacini et al. [PFG00] All spheres are rendered under the same environment lighting `Grace`. (b) Rendered spheres with the same intermediate values of roughness and specular reflectance are rendered under different environment maps. The first two columns use natural environment lighting, whether the last column use artificial analytical environment lighting.

As we have seen the perception of gloss has been largely studied [CK15]. However, we believe that explicit connections between physical and visual properties of materials (independently of any standard or observer) remain to be established.

## 2.6 Summary

Work on visual perception shows how humans are tolerant to inaccuracies in images. The human visual system may perceive as plausible images with certain deviations from physically correctness. Nevertheless we are able to distinguish material appearance under different illuminations, despite the fact that we are not able to judge physical variations linearly. Manipulation appearance techniques take advantage of these limitations to alter images by overcoming physical restrictions on rendering while keeping results plausible. We pursue a similar approach when using techniques like pre-filtered environment maps, where shading is pre-computed as the interaction of lighting and material. We aim to manipulate dynamically geometry-independent stored shading (similar to pre-filtered environment maps) and be able to mimic variations on lighting and material within it. The use of these structures seems a good intermediate alternative to perform appearance modification in comparison to the generation of images using a classical rendering pipeline.

# Chapter 3

# Statistical Analysis

The lightness and color (shading) of an object are the main characteristics of its appearance. Shading is the result of the interaction between the lighting and the surface reflectance properties of the object. In computer graphics lighting-material interaction is guided by the reflected radiance equation [Kaj86], explained in Section 2.1:

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

Models used in computer graphics that define reflectance properties of objects are not easily connected to their final appearance in the image. To get a better understanding we perform an analysis to identify and relate properties between shading on one side, and material reflectance and lighting on the other side.

The analysis only considers opaque materials which are are well defined by BRDFs [Nic65], leaving outside of our work transparent or translucent materials. We consider uniform materials, thus we only study variations induced by the viewing direction. When a viewing direction is selected the BRDF is evaluated as 2D function, that we call a **BRDF slice**. In that situation the material acts a filter of the incoming lighting. Our goal is to characterize the visible effect of BRDFs, and how their filtering behavior impacts shading. For that purpose we perform an analysis based on statistical properties of the local light-material interaction. Specifically, we use `moments` as quantitative measures of a BRDF slice shape.

Moments up to order can be used to obtain the classical mean and variance, and the energy as the zeroth moment. We use those statistical properties: energy, mean and variance to describe a BRDF slice model. In addition we make a few hypothesis on the BRDF slice shape to keep the model simple. Then, this model is used to develop a Fourier analysis where we find relationships on the energy, mean and variance between material/lighting and shading.

Finally we use our moment-based approach to analyze measured BRDFs. We show in plots how statistical properties evolve as functions of the view direction. We can extract common tendencies of different characteristics across all materials. The results verifies our previous hypothesis and show correlations among mean and variance.

This work have been published in the SPIE Electronic Imaging conference with the collaboration of Laurent Belcour, Carles Bosch and Adolfo Muñoz [ZBB+15]. Specifically, Laurent Belcour has helped with the Fourier Analysis, meanwhile Carles Bosch has made fittings on the measured BRDF analysis.

## 3.1 BRDF slices

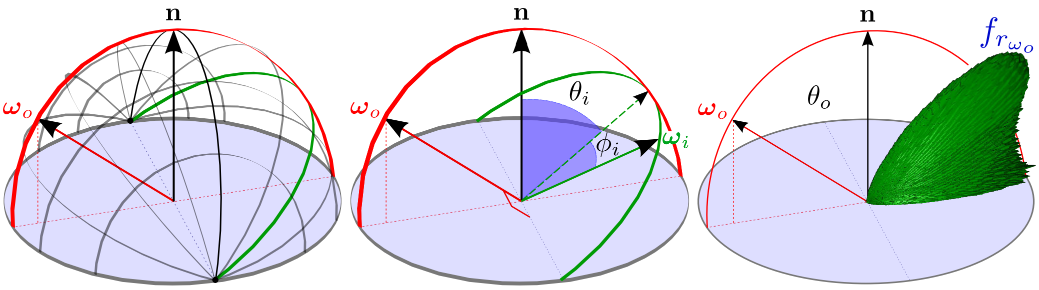
When we fix the view direction $\omega_o$ at a surface point $\mathbf{p}$ a 4D BRDF $f_r$ is restricted to a 2D **BRDF slice**. We define it as scalar functions on a 2D hemispherical domain, which we write $f_{r_{\omega_o}}(\omega_i) : \Omega \to \mathbb{R}$, where the underscore view direction $\omega_o$ indicates that it is fixed, and $\mathbb{R}$ denotes reflectance. Intuitively, such BRDF slices may be seen as a filter applied to the environment illumination. We suggest that some statistical properties of this filter may be directly observable in images, and thus may constitute building blocks for material appearance.

### 3.1.1 View-centered parametrization

Instead of using a classical parametrization in terms of elevation and azimuth angles for $\Omega$, we introduce a new view-centered parametrization with poles orthogonal to the view direction $\omega_o$, see Fig. 3.1. This parametrization is inspired by the fact that most of the energy of a BRDF slice is concentrated around the scattering plane spanning the view direction $\omega_o$ and the normal $\mathbf{n}$, then it minimize distortions around this plane. It is also convenient to our analysis. First, it permits to define a separable BRDF slice model, which is useful to perform the Fourier analysis separately per coordinate, see Section 3.2. Second, it enables the computation of statistical properties by avoiding periodical domains, see Section 3.3. Formally, we specify it by a mapping $m : [\text{-}\frac{\pi}{2}, \frac{\pi}{2}]^2 \to \Omega$, given by:

$$m(\theta, \phi) = (\sin\theta\cos\phi, \sin\phi, \cos\theta\cos\phi), \tag{3.1}$$

where $\phi$ is the angle made with the scattering plane, and $\theta$ the angle made with the normal *in* the scattering plane.



(a) Parametrization isolines (b) Parametrization angles (c) 2D BRDF slice

Figure 3.1: (a) Our parametrization of the hemisphere has poles orthogonal to the view direction $\omega_o$, which minimizes distortions in the scattering plane (in red). (b) It maps a pair of angles $(\theta_i, \phi_i) \in [-\frac{\pi}{2}, \frac{\pi}{2}]^2$ to a direction $\omega_i \in \Omega$. (c) A 2D BRDF slice $f_{r_{\omega_o}}$ is directly defined in our parametrization through this angular mapping.

The projection of a BRDF slice into our parametrization is then defined by:

$$f_{r_{\omega_o}}(\theta_i, \phi_i) := f_r(m(\theta_o, \phi_o), m(\theta_i, \phi_i)), \tag{3.2}$$

where $\theta_o$, $\phi_o$ and $\theta_i$, $\phi_i$ are the coordinates of $\omega_\mathbf{o}$ and $\omega_\mathbf{i}$ respectively in our parametrization. In the following we consider only isotropic BRDF which are invariant to azimuthal view angle. This choice is coherent with the analysis of measured BRDFs as the MERL database only contains isotropic BRDFs. Then, BRDF slices of isotropic materials are only dependent on the viewing elevation angle $\theta_o$; we denote them as $f_{r_{\theta_o}}$.

### 3.1.2 Statistical reflectance radiance model

We define a BRDF slice model in our parametrization using statistical properties. This model will be useful to perform a statistical analysis to study the impact of material on shading. Specifically it allows us to derive a Fourier analysis in 1D that yields statistical relationships between shading and material/lighting.

Our BRDF slice model is based on Gaussian lobes, which is a common choice to work with BRDFs. Gaussian functions are described by their mean $\mu$ and variance $\sigma^2$. The mean $\mu$ describes the expected or the central value of the Gaussian distribution. BRDF Gaussian lobes centered on the reflected view vector or on the normal are commonly used in Computer Graphics. The variance $\sigma^2$ describes the spread of the function. In the case of BRDF Slices, variance may be seen as a representation of material roughness. Wider lobes are representative of rough or diffuse material, meanwhile narrow lobes represent shiny or specular materials.

To define our BRDF slice model we have made a few assumptions by observing the measured materials from the MERL database. We have observed that BRDF slices of measured materials exhibits close to symmetric behavior around the scattering plane. Moreover, while BRDF slice lobes stay centered on this plane, the mean direction can vary, from the normal direction to the tangent plane direction, passing through the reflected view vector direction. These observations lead us to assume a perfect symmetry around the scattering plane. This assumption alongside with our view-centered parametrization allows us to define our model using a pair of 1D Gaussians. Therefore our BRDF slice model is defined as:

$$f_{r_{\theta_o}}(\theta_i, \phi_i) = \alpha(\theta_o) \;\; g_{\sigma_\theta(\theta_o)}(\theta_i - \mu_\theta(\theta_o)) \;\; g_{\sigma_\phi(\theta_o)}(\phi_i) \;, \tag{3.3}$$

where $g_{\sigma_\theta}$ and $g_{\sigma_\phi}$ are normalized 1D Gaussians [1] of variance $\sigma_\theta^2$ for the $\theta$ axis and variance $\sigma_\phi^2$ for the $\phi$ axis of our parametrization. The Gaussian $g_{\sigma_\theta}$ is centered at $\mu_\theta$, meanwhile the Gaussian $g_{\sigma_\phi}$ is centered at 0. The energy $\alpha$ is similar to the directional albedo (the ratio of the radiance which is reflected). But it differs in two ways: it does not take into account the cosine term of Equation (2.1), and is defined in our parametrization. As $\alpha$ is a ratio, it is bounded between 0 and 1, which guarantees energy conservation. A representation of the model is shown in Figure 3.2.

Energy, mean and variance can be defined using statistical quantities called `moments`, which characterize the shape of a distribution $f$:

$$\mu_k[f] = \int_{-\infty}^{\infty} x^k \; f(x) \; dx, \tag{3.5}$$

where $k$ is the moment order. In our case we use moments up to order 2 to define the meaningful characteristics: energy, mean and and variance. The energy $\alpha$ is the 0th order moment, which describes the integral value of the function. The energy is required to be 1 to guarantee $f$ to be a distribution, as moments are only defined for distribution functions. The mean $\mu$ is the 1st moment and the variance $\sigma^2$ is the 2nd central moments, where $\mu$ is used to center the distribution.

We emphasize out that this model does not ensure reciprocity, and shall thus not be used outside of this statistical analysis.

---

[1] Both Gaussians correspond to normal distributions that have been rescaled to guarantee energy conservation on our parametrization domain. The scaling term is given by:

$$A = \int_{-\frac{\pi}{2}-t_0}^{\frac{\pi}{2}-t_0} e^{-\frac{t^2}{2\sigma^2}} \,\mathrm{d}t = \sqrt{\frac{\pi\sigma^2}{2}} \Big[\mathrm{erf}\Big(\frac{\pi/2 - t_0}{\sqrt{2\sigma^2}}\Big) - \mathrm{erf}\Big(-\frac{\pi/2 - t_0}{\sqrt{2\sigma^2}}\Big)\Big], \tag{3.4}$$

where we have restricted the domain to $[-\frac{\pi}{2}, \frac{\pi}{2}]$ and centered it on $t_0$. This accommodates both Gaussians with $t_0 = \mu_\theta(\theta_o)$ in one case, and $t_0 = 0$ in the other.

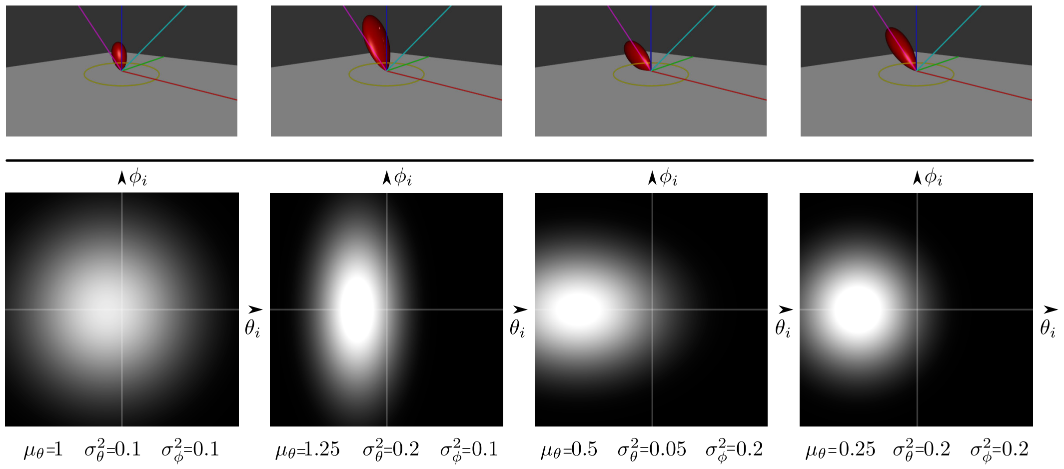


Figure 3.2: Different BRDF slices for the same viewing elevation angle at $45^o$ are shown in our view-centered parametrization. Different values of $\alpha$, $\mu_\theta$, $\sigma_\theta^2$ and $\sigma_\phi^2$ will define different BRDF slice and therefore different material appearances. These values can vary independently in terms of elevation angle $\theta_o$.

## 3.2 Fourier analysis

We conduct a local Fourier analysis that yields direct relationships between reflected radiance and BRDF/lighting of energy, mean and variance around a *fixed* view elevation.

### 3.2.1 Local Fourier analysis

Our analysis begins with a change of variable in Equation (2.1) using our parametrization. Analysis is performed in a local tangent frame for simplicity, with the domain of integration being the input space of $m$:

$$L_o(\theta_o,\phi_o) = \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} f_{r\theta_o}(\theta_i,\phi_i)\ L_i\big(m(\theta_i,\phi_i)\big)\cos\theta_i\cos^2\phi_i\,\mathrm{d}\theta_i\,\mathrm{d}\phi_i, \tag{3.6}$$

where the 3rd coordinate of $\omega_i = m(\theta_i,\phi_i)$ (given by $\cos\theta_i\cos\phi_i$ according to Equation. (3.1)) stands for the cosine term in tangent space.

Replacing $f_{r\theta_o}$ with our BRDF slice model (Equation (3.3)) yields:

$$L_o(\theta_o,\phi_o) = \alpha(\theta_o)\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} g_{\sigma_\theta(\theta_o)}(\theta_i-\mu_\theta(\theta_o))\left(\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} g_{\sigma_\phi(\theta_o)}(\phi_i)L_i\big(m(\theta_i,\phi_i)\big)\cos\theta_i\cos^2\phi_i\mathrm{d}\phi_i\right)\mathrm{d}\theta_i. \tag{3.7}$$

Now, since our BRDF slice model is separable in $\theta_i$ and $\phi_i$, we may pursue our study in either dimension independently. Let us focus on $\theta_i$. If we fold in the integral of terms over $\phi_i$ and cosines and write:

$$L_{i_\phi}(\theta_i) = \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} g_{\sigma_\phi(\theta_o)}(\phi_i)L_i\big(m(\theta_i,\phi_i)\big)\cos\theta_i\cos^2\phi_i\,\mathrm{d}\phi_i,$$

then Equation 3.7 turns into a 1D integral of the form:

$$L_o(\theta_o,\phi_o) = \alpha(\theta_o)\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} g_{\sigma_\theta(\theta_o)}(\theta_i-\mu_\theta(\theta_o))L_{i_\phi}(\theta_i)\,\mathrm{d}\theta_i. \tag{3.8}$$

Our next step is to approximate this 1D integral with a convolution. To this end, we make local approximations of our BRDF slice model in a 1D angular window around $\theta_o$. We assume the energy and variance to be locally constant: $\alpha(\theta_o+t) \approx \alpha(\theta_o)$ and $\sigma_\theta^2(\theta_o+t) \approx \sigma_\theta^2(\theta_o)$. For the mean, we rather make use of a first-order approximation: $\mu_\theta(\theta_o+t) \approx \mu_\theta(\theta_o) + \frac{d\mu_\theta}{dt}|_{\theta_o} t$. As a result, $L_o$ may be approximated by a 1D convolution of the form:

$$L_o(\theta_o+t, \phi_o) \approx \alpha\big(g_{\sigma_\theta} * L_{i_\phi}\big)(\theta_o+t), \text{ with } t \in [-\epsilon, +\epsilon], \tag{3.9}$$

where we have dropped the dependencies of both $\alpha$ and $\sigma_\theta$ on $\theta_o$ since they are assumed locally constant.

In Fourier space, this convolution turns into the following product:

$$\mathcal{F}[L_o](\xi) \approx \alpha \; \mathcal{F}[g_{\sigma_\theta}](\xi) \; \mathcal{F}[L_{i_\phi}](\xi), \tag{3.10}$$

where $\xi$ is the Fourier variable corresponding to $t$. Note that Fourier shifts $e^{i\theta_o}$ due to the centering on $\theta_o$ cancel out since they appear on both sides of the equation. Equation (3.9) bears similarities with previous work [DHS<sup>+</sup>05, RMB07], with the difference that our approach provides direct connections with moments thanks to our BRDF slice model.

### 3.2.2 Relationships between moments

An important property of moments is that they are directly related to the Fourier transform of a function $f$ by [Bul65]:

$$\mathcal{F}[f](\xi) = \sum_k \frac{(i\xi)^k}{k!} \mu_k[f] \tag{3.11}$$

where $\mu_k[f]$ is the $k$-th moment of $f$. We thus re-write Equation (3.10) as a product of moment expansions:

$$\mathcal{F}[L_o](\xi) = \alpha \left( \sum_k \frac{(i\xi)^k}{k!} \mu_k[g_{\sigma_\theta}] \right) \left( \sum_l \frac{(i\xi)^l}{l!} \mu_l[L_{i_\phi}] \right). \tag{3.12}$$

To establish relationships between moments, we extract the moments from $\mathcal{F}[L_o]$ using its *own* moment expansion. This is done by differentiating $\mathcal{F}[L_o]$ at $\xi = 0$ [Bul65]:

$$\mu_0[L_o] = \mathcal{F}[L_o](0) \tag{3.13}$$

$$\mu_1[L_o] = Im\left\{ \frac{\mathrm{d}\mathcal{F}[L_o]}{\mathrm{d}\xi}(0) \right\} \tag{3.14}$$

$$\mu_2[L_o] = -Re\left\{ \frac{\mathrm{d}^2\mathcal{F}[L_o]}{\mathrm{d}\xi^2}(0) \right\}. \tag{3.15}$$

Next, we expand Equation 3.12 and its derivatives at $\xi = 0$ and plug them inside Equations (3.13) through (3.15):

$$\mu_0[L_o] = \alpha\mu_0[g_{\sigma_\theta}]\mu_0[L_{i_\phi}], \tag{3.16}$$

$$\mu_1[L_o] = \mu_1[g_{\sigma_\theta}] + \mu_1[L_{i_\phi}], \tag{3.17}$$

$$\mu_2[L_o] = \mu_2[g_{\sigma_\theta}] + \mu_2[L_{i_\phi}] + 2\mu_1[g_{\sigma_\theta}]\mu_1[L_{i_\phi}]. \tag{3.18}$$

Since $g_{\sigma_\theta}$ is normalized, $\mu_0[g_{\sigma_\theta}] = 1$. However, $\mu_0[L_o] \neq 1$ in the general case, and we must normalize moments of order 1 and 2 before going further. We write $\bar{L}_o = L_o/\mu_0[L_o]$, which yields $\mu_k[\bar{L}_o] = \frac{\mu_k[L_o]}{\mu_0[L_o]}$. It can then be easily shown that Equations (3.17) and (3.18) remain valid after normalization.

Lastly, we write the variance of $\bar{L}_o$ in terms of moments: $\mathrm{Var}[\bar{L}_o] = \mu_2[\bar{L}_o] - \mu_1^2[\bar{L}_o]$. After carrying out simplifications, we get: $\mathrm{Var}[\bar{L}_o] = \mathrm{Var}[\bar{g}_{\sigma_\theta}] + \mathrm{Var}[\bar{L}_{i_\phi}]$. Putting it all together, we obtain the following moment relationships for a given viewing elevation $\theta_o$:

$$\begin{aligned} \mu_0[L_o](\theta_o) &= \alpha(\theta_o)\ \mu_0[L_{i_\phi}](\theta_o), & (3.19)\\ \mathrm{E}[\bar{L}_o](\theta_o) &= \mu_\theta(\theta_o) + \mathrm{E}[\bar{L}_{i_\phi}](\theta_o), & (3.20)\\ \mathrm{Var}[\bar{L}_o](\theta_o) &= \sigma_\theta^2(\theta_o) + \mathrm{Var}[\bar{L}_{i_\phi}](\theta_o), & (3.21) \end{aligned}$$

where we have used $\mathrm{E}[\bar{g}_{\sigma_\theta}](\theta_o) = \mu_\theta(\theta_o)$ and $\mathrm{Var}[\bar{g}_{\sigma_\theta}](\theta_o) = \sigma_\theta^2(\theta_o)$. The reasoning is similar when studying the integral along $\phi_i$, in which case we must define a $L_{i_\theta}$ term analogous to $L_{i_\phi}$. We then obtain similar moment relationships, except in this case $\mathrm{E}[\bar{g}_{\sigma_\phi}] = 0$, $\mathrm{Var}[\bar{g}_{\sigma_\phi}] = \sigma_\phi^2$, and $L_{i_\phi}$ is replaced by $L_{i_\theta}$.

## 3.3 Measured material analysis

We compute statistical moments of BRDF slices up to order 2 (energy, mean and variance) on a set of measured materials of the MERL database. Moments are computed as functions of viewing angle which we call moment profiles. We experimentally show that such moment profiles are well approximated by parametric forms: a Hermite spline for the energy, a linear function for the mean, and a constant for the variance. Parametric forms for these functions are obtained through fitting, and additionally we show that mean and variance statistics are correlated.

On the implementation side, we have made use of BRDF Explorer [Bur12], which we have extended to incorporate the computation of moments. Carles Bosch have performed fitting using Mathematica.

### 3.3.1 Moments of scalar functions

We analyze moments on BRDF slices of measured materials without making any hypothesis. Therefore we use general tensors to capture moments of a scalar distribution $f$:

$$\mu_k[f] = \int_X \underbrace{\mathbf{x} \otimes \cdots \otimes \mathbf{x}}_{k \text{ factors}}\ f(\mathbf{x})\,\mathrm{d}\mathbf{x}, \tag{3.22}$$

where $X$ is the domain of definition of $f$ and $\otimes$ denotes a tensor product. As a result, a moment of order $k$ is a tensor of dimension $k+1$: a scalar at order 0, a vector at order 1, a matrix at order 2, etc.

Similarly to our BRDF Slice model we analyze moments up to order 2 to study the energy, mean and variance of the BRDF slices. Despites that, the analysis is easily extensible to higher order moments: the 3rd and 4th order moments (skewness and kurtosis) are given in the AppendixA.

Now, for a scalar distribution defined over a 2D domain, we write $\mathbf{x} = (x, y)$ and define:

$$\mu_{n,m}[f] := \mathrm{E}_f[x^n y^m] = \iint_X x^n y^m\ f(x,y)\,\mathrm{d}x\mathrm{d}y, \tag{3.23}$$

which corresponds to a scalar coefficient of the moment tensor of order $k = n + m$. In the following, we mostly make use of Equation (3.23) instead of Equation (3.22) as it makes our computations more explicit.

### 3.3.2 Choice of domain

The classical parametrization in terms of elevation and azimuth angles is not adapted to the computation of moments Indeed, the *periodicity* of the azimuthal dimension is problematic because domains are anti-symmetric when the power involved in the computation of moments is odd, see Equation (3.23). This incompatibility is avoided when using our parametrization. The projected result of the BRDF slices using our view-centered parametrization is shown in Figure 3.3.

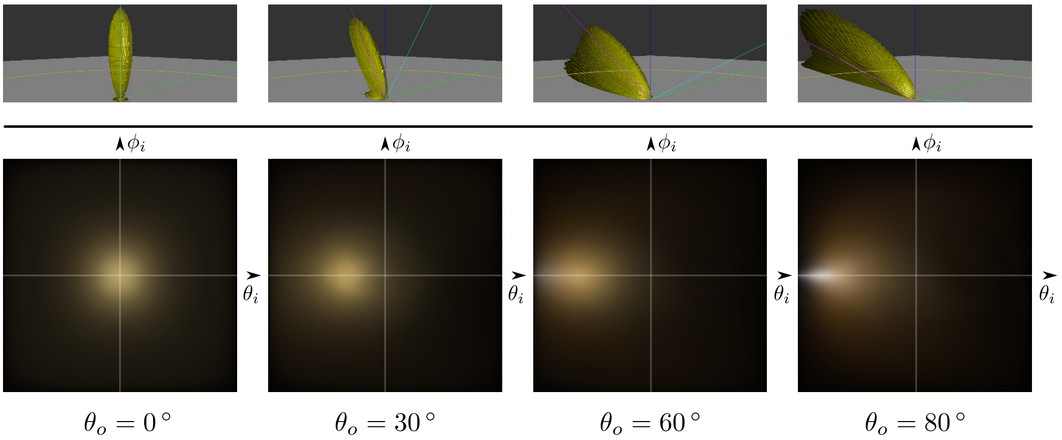


Figure 3.3: Top row: 3D visualization of four slices of the `gold-paint` BRDF at increasing viewing angles. Bottom row: the same BRDF slices in our view-dependent parametrization.

A different solution to deal with the periodicity of the hemispherical domain would be to compute 3D moments using Cartesian coordinates as done by Ghosh et al. [GCP$^+$09]. However, this would not only make analysis harder (relying on 3D tensors), but it would also unnecessarily introduce distortions at grazing angles, where hemispherical and Euclidean distances differ markedly. An alternative would be to rely on statistics based on lighting elevation, as done by Havran et al [HS14] for the purpose of material characterization. Unfortunately, this approach is not adapted to our purpose since it reduces a priori the complexity of BRDFs by using a 1D analysis. Instead, we compute moments using a planar 2D parametrization that introduces as little distortion as possible for isotropic BRDFs.

### 3.3.3 BRDF slice components

Moments are not good descriptors for *multimodal* functions. They are only well-defined for *unimodal* functions; computed statistics are not meaningful otherwise.

In contrast to our BRDF slice model, we have observed that many BRDFs from the MERL database display multi-modal components with a near constant (i.e. close to Lambertian) diffuse component. We rely on a simple heuristic method to separate that diffuse component, leaving the rest of the data as a specular component. Such a perfect diffuse component can be extracted using a simple thresholding scheme: we sample the BRDF at a viewing elevation of 45 degrees and retrieve the minimum reflectance value. We then remove this constant from the data in order to obtain its specular component. We will analyze only the remaining specular component, on which we ground our study.

However, even after removing a Lambertian component from BRDF data, some materials still show multi-modal behaviors. We simply remove these BRDFs from our set manually, leaving a total of 40 unimodal BRDFs [2]. They still span a wide range of appearances, from rough to shiny materials.

### 3.3.4 Moment profiles

We compute $2D$ moments of the specular component of the projected BRDF slices (see Fig. 3.3) using a discretized version of the Equation (3.23). Using moments up to order 2 we have as a result a scalar value for the energy, a $2D$ vector for the mean and a $2 \times 2$ matrix for the variance. We show how those statistical properties vary as functions of the viewing elevation $\theta_o$, which we call *moment profiles*. In practice, we sample the $\theta_o$ dimension uniformly in angles, each sample yielding a projected BRDF slice.We then use a Monte Carlo estimator to evaluate the 2D moments for each of these slices:

$$\mu_{n,m}[f_r](\theta_o) \approx \frac{\pi^2}{N} \sum_{i=1}^{N} \theta_i^n \phi_i^m \; f_{r_{\theta_o}}(\theta_i, \phi_i), \tag{3.24}$$

where $\mathbf{x}_i = (\theta_i, \phi_i)$ is the $i$th randomly generated sample in the slice, and $N$ is the number of samples.

In the following, we present moment profiles computed at increasing orders, as shown in Figs. 3.4 and 3.5. For the sake of clarity, we will omit the dependence on $\theta_o$ both for BRDF slices and 2D moments.

#### Energy

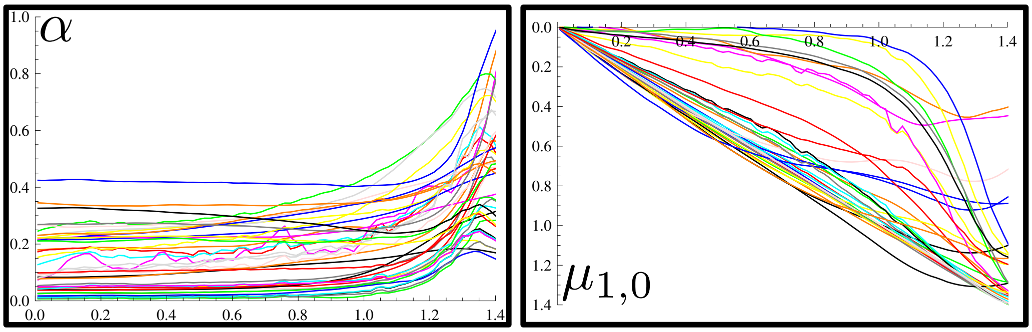
As seen in these plots, the energy $\alpha$ stays below 1, which indicates that a portion of the light is reflected. They look mostly constant except near grazing angles where they tend to increase. We show $\alpha$ profiles for the red channel only of all our selected BRDFs.

#### Mean

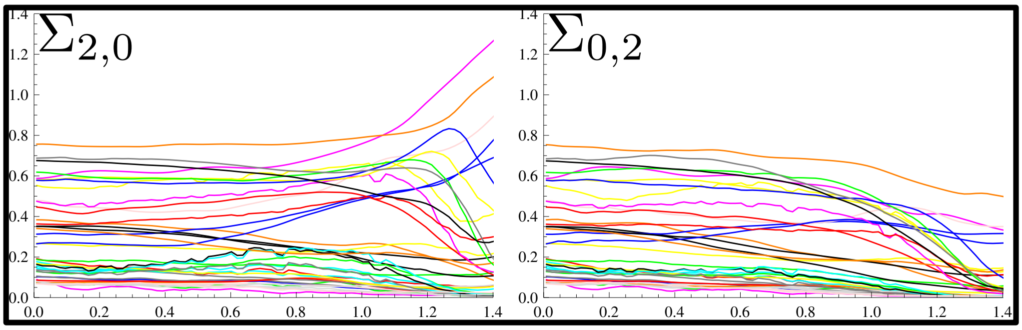
For moments of order 1 and higher, we must normalize by the 0th order moment in order to guarantee that $f_r$ is a distribution, see Sec. 3.1.2. We thus write $\bar{f}_r = \frac{f_r}{\alpha}$. The coefficients are now given by $\mu_{n,m}[\bar{f}_r] = E_{\bar{f}_r}\left[\theta_i^n \phi_i^m\right]$ for $n + m = 1$. The profile for $\mu_\theta := \mu_{1,0}$ is shown in Fig. 3.4b: our selected BRDFs exhibit profiles that have different slopes, with deviations from a line occurring toward grazing angles. In contrast the profile of $\mu_\phi := \mu_{0,1}$, as shown in Fig. 3.5a, remains close to zero for all values of $\theta_o$. This is coherent with the near-symmetry of the BRDF slice around the scattering plane.

---

[2]
- yellow-phenolic
- yellow-matte-plastic
- white-paint
- white-marble
- white-acrylic
- violet-acrylic
- two-layer-gold
- tungsten-carbide
- ss440
- specular-violet-phenolic
- specular-green-phenolic
- specular-blue-phenolic
- specular-black-phenolic
- silver-paint
- silicon-nitrade
- red-metallic-paint
- pvc
- pure-rubber
- pearl-paint
- nickel
- neoprene-rubber
- hematite
- green-metallic-paint2
- green-metallic-paint
- gold-paint
- gold-metallic-paint3
- gold-metallic-paint2
- color-changing-paint3
- color-changing-paint2
- color-changing-paint1
- chrome
- chrome-steel
- brass
- blue-metallic-paint2
- blue-metallic-paint
- black-phenolic
- black-obsidian
- aventurnine
- aluminium
- alum-bronze

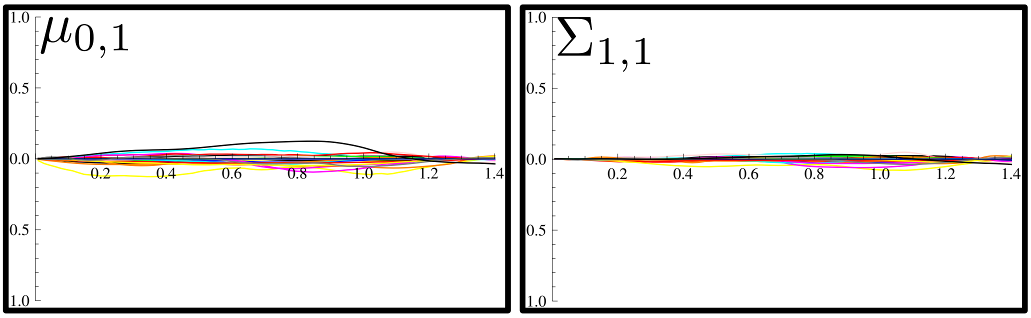


(a) Energy profiles (b) Mean profiles



(c) Co-Variance profiles

Figure 3.4: Moment profiles computed from our selected BRDFs are shown at increasing moment orders. (a) Energy (b) Mean at $\theta$ (c) Co-Variance at $\theta\theta$ and $\phi\phi$.



(a) Mean profiles (b) Co-Variance profiles

Figure 3.5: Near-zero moment profiles occur for (a) mean at $\phi$ and (b) co-variance $\theta\phi$ likely due to BRDF slice symmetry.

### Co-variance

It is defined as the *centralized* moment matrix $\Sigma$ of order 2, which consists of moments of $\bar{f}_r$ centered on its mean. In our case, since $\mu_{0,1} \approx 0$, the coefficients of the co-variance matrix may be written using a slightly simpler formula: $\Sigma_{n,m}[\bar{f}_r] = E_{\bar{f}_r}\left[(\theta_i - \mu_\theta)^n \phi_i^m\right]$ for $n + m = 2$. This matrix characterizes how the BRDF slice is spread around its mean, with larger values in either dimension implying larger spread. The profiles for the diagonal coefficients $\sigma_\theta^2 := \Sigma_{2,0}$ and $\sigma_\phi^2 := \Sigma_{0,2}$ are shown in Fig. 3.4c: our selected BRDFs exhibit profiles of different variances, with slight deviations from the average occurring toward grazing viewing angles. The off-diagonal coefficient $\Sigma_{1,1}$ remains close to zero as shown in Fig. 3.5b, again due to the near-symmetry of the BRDF slice.

### Interim discussion

Plots exhibit common behaviors along all the selected BRDFs where we can study their causes. First, both moments which correspond to anti-symmetric functions in the $\phi_i$ dimension $m = 1$, $\mu_{0,1}$ and $\Sigma_{1,1}$, exhibits close to null profiles. This strongly suggests that they are due to the *near-symmetry* of most isotropic BRDFs about the scattering plane (i.e. along $\phi_i$), as seen in Fig. 3.3. This affirms the symmetry hypothesis made for the definition of our BRDF slice model. Second, values at incident view $\theta_o = 0$ start at 0 for the mean and share the same value for $\sigma^2_\theta$ and $\sigma^2_\phi$. The reason for this is that slices of isotropic BRDFs are near radially symmetric around the normal (the origin in our parametrization) at incidence.

Lastly, all materials tend to exhibit deviations with respect to a simple profile toward grazing viewing angles. This might be due to specific modes of reflectance such as asperity scattering [KP03] coming in to reshape the BRDF slice.

Those three common behaviors are coherent with the further results obtained with the study of the skewness and kurtosis as presented in the Appendix A.

### 3.3.5 Fitting & correlation

In order to have a better understanding of material behavior we fit analytical functions to moment profiles. We look for similarities for a same moment order, as well as correlations among different orders. Naturally, we will less focus on fitting accuracy than on concision: a minimal set of parameters is necessary if we wish to compare profiles across many measured materials.

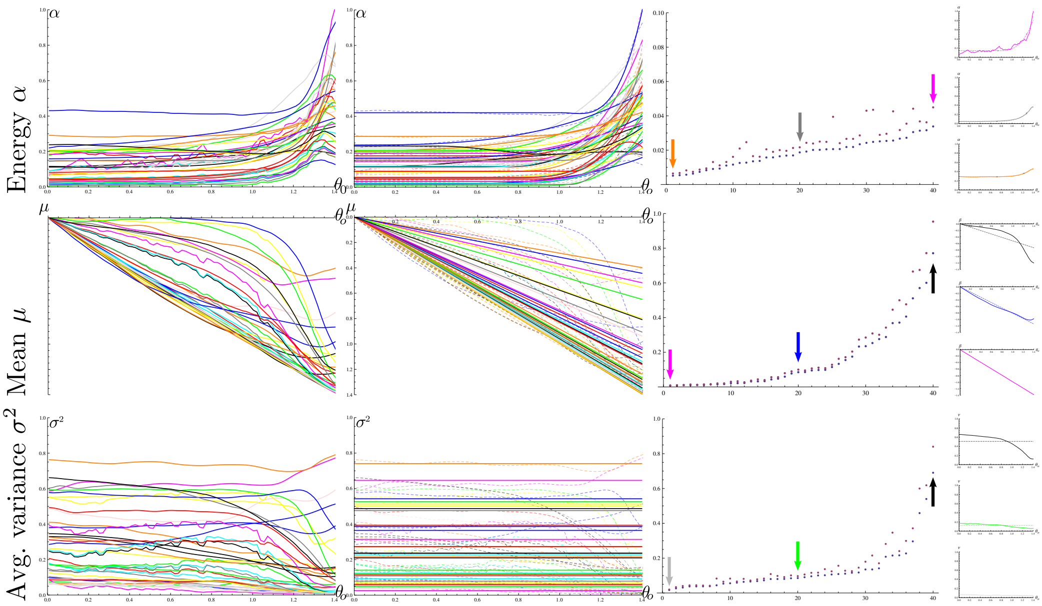
Regarding color channels, we have tried fitting them separately, or fitting their average for each slice directly. We only report fits based on averages since for our selected materials we found differences across fits for different color channels to be negligible. It must be noted that for the energy, profiles for each color channel will obviously be different; however they are merely offseted with respect to each other. It is thus also reasonable to fit their average since we are mostly interested in the shape of profile functions.

Figure 3.6 shows the fitting results for the energy, mean and variance profiles, as detailed below. It shows computed profiles, fitted profiles, fitting errors and representative ‘best’, ‘intermediate’ and ‘worse’ fits. We introduce our choices of analytical function for each moment order in turn.

### Energy

As seen in Fig. 3.6, the energy profiles $\alpha(\theta_o)$ for our selection of BRDFs exhibit a constant behavior up to grazing angles, at which point they tend to increase progressively. We model this profile with $\alpha(\theta_o) \approx \tilde{\alpha}(\theta_o) = \tilde{\alpha_b} + \tilde{\alpha_s}(\theta_o)$ where $\tilde{\alpha_b}$ represents the constant base energy and $\tilde{\alpha_s}$ is an Hermite spline that deals with the increase in energy. The Hermite spline itself is modeled with two knots defined by their angular position $\theta_0$, $\theta_1$, energy value $\alpha(\theta_0)$, $\alpha(\theta_1)$ and slope $m_0$, $m_1$ (see Fig. 3.7a).

We use a non-linear optimization to fit these parameters to each energy profile, using $\theta_0 = 45$ and $\theta_1 = 75$ degrees as initial values, and constraining $m_0 = 0$ to reproduce the constant profile far from grazing angles. We first fit knot positions independently for each material, which yields an average $\theta_0$ of 35.3 degrees with a standard deviation of 14.8, and an average $\theta_1$ of 71.5 degrees with a standard deviation of 5.2. Combined with the observation that $\alpha_1 > \alpha_0$ in all our materials, this suggests that all our materials exhibit an energy boost confined to grazing viewing angles. We then fit the same knot positions for all our materials; this yields $\theta_0 = 38.7$ degrees and $\theta_1 = 69.9$ degrees, which confirms the grazing energy boost tendency.



(a) computed profiles (b) fitted profiles (c) fitting errors

Figure 3.6: We fitted the moment profiles (from top to bottom: energy, mean and average variance) from the selected materials list. We provide in each column the computed moment profiles (a), the fitted profiles (b), and the corresponding fitting errors (c). The error of our fits is computed using both the Mean Absolute Error (MAE in blue) and the Root Mean Square Error (RMSE in purple). The small inset profiles correspond to worse, intermediate and best fits.

### Mean

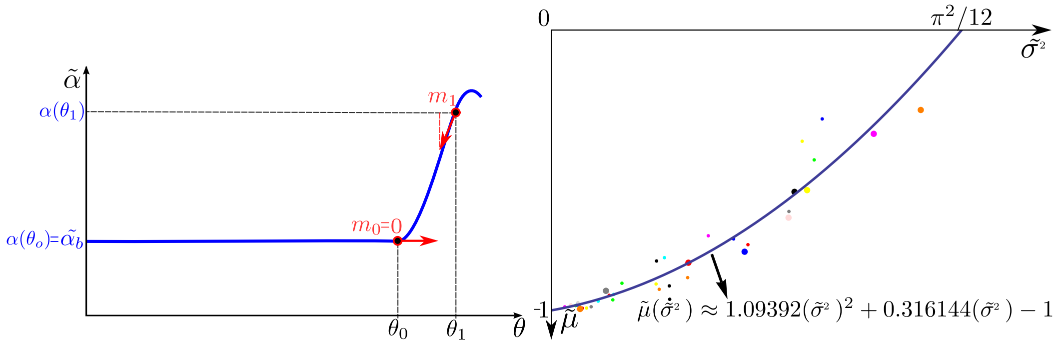
Concerning the mean profile $\mu_\theta(\theta_o)$, the vast majority of cases show a linear tendency, with slopes proportional to the specularity of the material. Moreover, all profile functions go through the origin, as previously observed in Sec. 3.3.4. This suggests that a linear fit $\mu_\theta(\theta_o) \approx \tilde{\mu}\theta_o$ is appropriate for representing this behavior. We fit the single slope parameter $\tilde{\mu}$ using a least-squares optimization, which always leads to a negative value due to our choice of parametrization (e.g., the mirror direction is given by $\tilde{\mu} = -1$). It is interesting to observe that materials exhibit mean slopes nearly spanning the entire range from $-1$ to 0.

### Variance

We have observed in Sec. 3.3.4 that $\sigma^2_\theta(0) \approx \sigma^2_\phi(0)$, which is due to radial symmetry at viewing incidence. Our data also reveals that the deviations from a constant behavior observed around grazing angles tend to increase for $\sigma^2_\theta$ and decrease for $\sigma^2_\phi$. We thus choose to study the average variance using a constant profile, hence $\tilde{\sigma}^2 \approx \frac{(\sigma^2_\theta(\theta_o)+\sigma^2_\phi(\theta_o))}{2}$. The constant parameter is obtained using a least-squares fitting as before, with values ranging from 0 for a mirror to $\frac{\pi^2}{12}$ for a Lambertian [3]. Once again, our materials exhibit a large range of average variances.

### Correlation

Looking at Fig. 3.6b, one may observe a seeming correlation between the fitted mean slope $\tilde{\mu}$ and average variance $\tilde{\sigma}^2$: *the lower the variance, the steeper the mean.* To investigate this potential correlation, we plot one parameter against the other in Fig. 3.7b, which indeed exhibits a correlation. We thus perform a least-squares fitting with a quadratic function, using parameters of a mirror material ($\tilde{\mu} = -1$ and $\tilde{\sigma}^2 = 0$) and of a Lambertian one ($\tilde{\mu} = 0$ and $\tilde{\sigma}^2 = \pi^2/12$) as end point constraints. We conjecture that this correlation is due to hemispherical clamping. Because of clamping, the distribution mean is slightly offset toward incidence compared to the distribution peak, and this effect is all the more pronounced for distributions of high variance: wider distributions are clamped early on relative to narrow ones.



(a) Energy profile function $\tilde{\alpha}$

(b) Correlation between $\tilde{\mu}$ and $\tilde{\sigma}^2$

Figure 3.7: (a) The energy profile $\tilde{\alpha}$ may exhibit a silhouette effect, which we model by a Hermite spline starting at $\theta_0$ with $m_0 = 0$, and ending at $\theta_1$ with a fitted $m_1$. (b) We fit the correlation between the mean slope $\tilde{\mu}$ and the average variance $\tilde{\sigma}^2$ using a quadratic function.

## 3.4 Discussion

Relationships defined in Sec. 3.2.2 provide insights on how material influence shading locally, which is observable in pictures. Moreover the conclusions of the analysis of measured BRDFs in Sec. 3.3.5 describe similar behaviors across different BRDFs.

As an example, let's consider a distant environment illumination reflected off a sphere: in this case, each $(\theta_o, \phi_o)$ pair corresponds to a point on the sphere surface. First, the energy

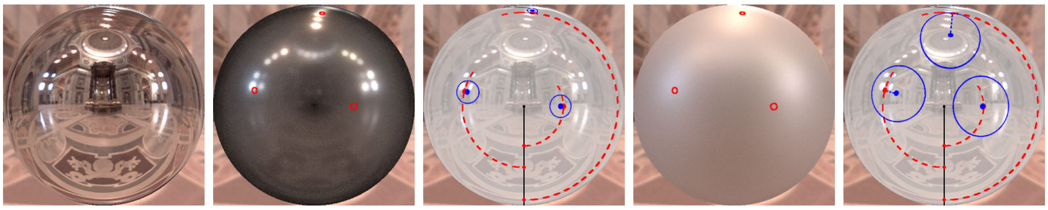
---

[3] A Lambertian material corresponds to $f_{r_L} = \frac{1}{\pi^2}$ in our angular parametrization, irrespective of the view direction.

Since we are working on a closed space $[-\frac{\pi}{2}..\frac{\pi}{2}]^2$, the formulation of moments for a constant function is not the same than in an infinite domain: moments of a Lambertian BRDF are thus finite. We can simplify the expression of the average variance as the mean of a Lambertian is zero, $\mu_1[f_{r_L}] = 0$, hence $\mathrm{Cov}[f_{r_L}] = \mu_2[f_{r_L}]$. Furthermore, the formulation of the variance along $\theta_i$ and along $\phi_i$ provide the same result due to the symmetry of the integration space and integrands. Thus the average variance $\nu[f_{r_L}]$ is equal to the variance along $\theta_i$, leading to the formula:

$$\nu[f_{r_L}] = \frac{1}{\pi^2} \int\int_{\theta_i,\phi_i \in [-\frac{\pi}{2}..\frac{\pi}{2}]^2} {\theta_i}^2 d\theta_i d\phi_i = \frac{1}{\pi} \left[\frac{{\theta_i}^3}{3}\right]_{-\frac{\pi}{2}}^{\frac{\pi}{2}} = \frac{\pi^2}{12}.$$

relationship Equation (3.19) tells us that the material color has a *multiplicative* effect on shading. In addition the energy fitting describe this multiplication as a constant effect ($\tilde{\alpha}_b$) with an additional boost ($\tilde{\alpha}_s$) toward the silhouette. Second, the Equation (3.20) acts as a warping that constantly increases toward the silhouette. Third, a constant blur is applied locally as defined by Equation (3.21). It has a constant average behavior for both dimension $\tilde{\sigma}^2$. Both mean and variance are linked to material roughness and their effect is correlated (Fig. 3.7b). It is significant that the only effect which is applied depending on the $\phi$ direction is the blurring which is consistent with the symmetry of BRDF slices around the scattering plane



(a) Mirror sphere (b) Spec. material (c) Spec. filtering (d) Diff. material (e) Diffuse filtering

Figure 3.8: (a) A sphere made of an ideal mirror material rendered using the `StPeter` environment map. The reflected environment is extremely sharp and warped toward silhouettes. (b) A rendering using the `Specular-black-phenolic` BRDF. The reflected environment is slightly blurred and highly warped. This is explained in (c) by the filtering characteristics (in blue) of the BRDF at 3 different locations: the filter is narrow and remains close to the evaluation point (in red). Note that at similar viewing elevations (dashed red arcs), the filters are rotated copies of each other. (d) A rendering using the `Pearl-paint` BRDF. The reflected environment is this time much more blurred and exhibits less warping. This is explained in (e) by the filtering characteristics of the BRDF: the filter is wide and offset toward the center of the sphere for locations closer to the silhouette. This confirms the mean/variance correlation that we have observed in our study.

Figure 3.8 provides an illustration of the effect described above. We start from an ideal mirror BRDF in Fig. 3.8a, which exhibits extreme sharpness and warping of the environment toward the silhouette as expected. We then study the effect of two BRDFs: `specular-black-phenolic` and `pearl-paint`, with renderings shown in Figure 3.8b and Figure 3.8d. In both case, we focus on three points increasingly closer to the silhouette (i.e., at increasingly grazing viewing elevation). Our analysis reveals that a BRDF directly acts as an *image filter* whose properties are governed by statistical moments. In Figure 3.8c and Figure 3.8e, we show the means and co-variance ellipses (in blue) for both BRDFs at the three picked locations. The filters corresponding to the `specular-black-phenolic` BRDF remain close to the evaluation position (in red), and their spread is narrow, resulting in a small blur. In contrast, the filters corresponding to the `pearl-paint` BRDF exhibit a stronger blur, and are offseted toward the center of the sphere for increasing viewing angles. As a result, the warping due to the BRDF is less pronounced in this case, a subtle effect of this BRDF which illustrates the impact of correlation between mean and variance.

Our statistical analysis has shown that using a simple BRDF slice model based on energy, mean and variance we can derive relationships between lighting/material and shading. Those relationships are observable in images induced by coloring due to $\alpha$, warping by $\mu$ and blurring by $\sigma^2$. Our BRDF slice model is coherent with isotropic BRDFs as shown for the selected list of materials from the MERL database. As output of this study we obtain similar behaviors of the BRDF slices as functions depending on the viewing elevation angle, as well as a correlation between mean and variance.

# Chapter 4

# Dynamic Appearance Manipulation of MatCaps

Object appearance is the result of complex interactions between shape, lighting and material. Instead of defining those components and performing a rendering process afterwards, we rather intend to manipulate existing appearance by directly modifying the resulting shading colors. We have shown how two of these components: lighting and materials are related to the resulting shading. In this chapter we are interested in using the studied relationships between material/lighting and shading to mimic modification of both material and lighting from existing shading.

We focus our work on artwork inputs, as we find interesting the artist perspective to create plausible appearances by direct painting, instead of the tedious rendering process of trial and error. We use artistic images of sphere which are created without having to specify material or lighting properties. Appearance described in these images is easily transferable to an arbitrary-shaped object with a simple lookup based on *screen-space* normals. This approach was first introduced by the LitSphere technique introduced by Sloan et al. [SMGG01]. It is also known under the name of 'MatCaps' for rendering individual objects, typical applications include scientific illustration (e.g., in MeshLab and volumetric rendering [BG07]) and 3D sculpting (e.g., in ZBrush, MudBox or Modo). In this work we use the term 'MatCap' to refer to LitSphere images that convey plausible material properties, and we leave out of our work non-photorealistic LitSpheres approaches (e.g., [TAY13]).

The main limitation of a MatCap is that it describes a static appearance: lighting and material are 'baked in' the image. For instance, lighting remains tied to the camera and cannot rotate independently; and material properties cannot be easily modified. A full separation into physical material and lighting representations would not only be difficult, but also unnecessary since a MatCap is unlikely to be physically-realistic. Instead, our approach is to keep the simplicity of MatCaps while permitting dynamic appearance manipulations in real time. Hence we do *not* fully separate material and lighting, but rather decompose an input MatCap (Figure 4.1a) into a pair of spherical image-based representations (Figure 4.1b). Thanks to this decomposition, common appearance manipulations such as rotating lighting, or changing material color and roughness are performed through simple image operators (Figures 4.1c, 4.1d and 4.1e).

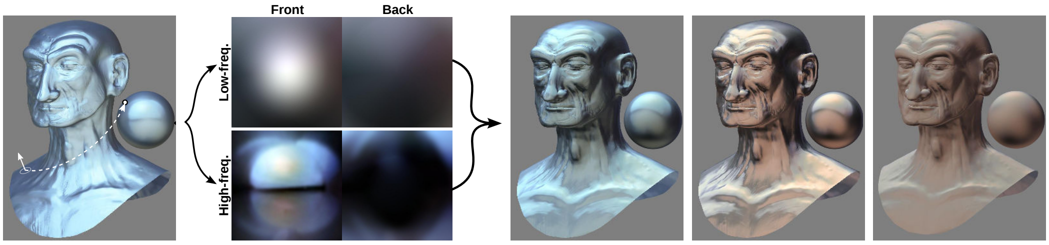
Our approach makes the following contributions:

- We assume that the material acts as an image *filter* in a MatCap and we introduce a simple algorithm to estimate the parameters of this filter (Section 4.1);
- We next decompose a MatCap into high- and low-frequency components akin to diffuse and specular terms. Thanks to estimated filter parameters, each component is then

unwarped into a spherical representation analogous to *pre-filtered* environment maps (Section 4.2);

- We perform appearance manipulation in real-time from our representation by means of image operations, which in effect *re-filter* the input MatCap (Section 4.3).

As shown in Section 4.4, our approach permits to convey a plausible, spatially-varying appearance from one or more input MatCaps, without ever having to recover physically-based material or lighting representations.



(a) Input (b) Decomposed MatCap (c) Rotation of lighting (d) Change in color (e) Appearance turn rougher

Figure 4.1: Our approach decomposes a MatCap into a representation that permits dynamic appearance manipulation via image filters and transforms. (a) An input MatCap applied to a sculpted head model (with a lookup based on screen-space normals). (b) The low- & high-frequency (akin to diffuse & specular) components of our representation stored in dual paraboloid maps. (c) A rotation of our representation orients lighting toward the top-left direction. (d) Color changes applied to each component. (e) A rougher-looking material obtained by bluring, warping and decreasing the intensity of the high-frequency component.

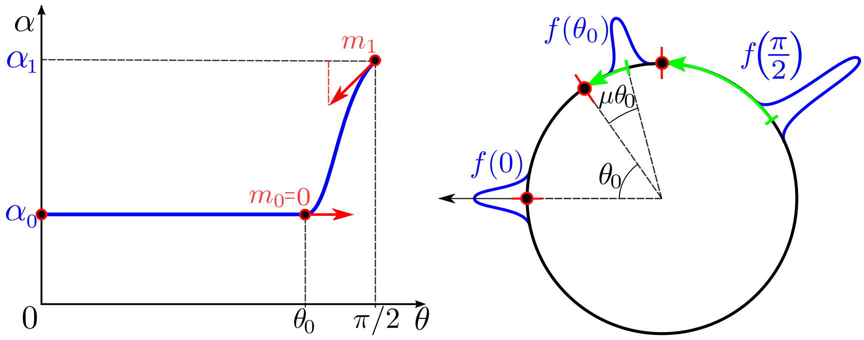
This work have been published in the Eurographics Symposium of Rendering at the 'Experimental Ideas & Implementations' with the collaboration of Adolfo Muñoz, Laurent Belcour and Carles Bosch [ZMB+15].

## 4.1 Appearance model

We hypothesize that the material depicted in a MatCap image acts as a filter of constant size in the spherical domain (see Figure 4.2b). Our goal is then to estimate the parameters of this filter from image properties *alone*. We first consider that such a filter has a pair of diffuse and specular terms. The corresponding diffuse and specular MatCap components may either be given as input, or approximated (see Section 4.2.1). The remaining of this section applies to either component considered independently.

### 4.1.1 Definitions

We consider a MatCap component $L_o$ to be the image of a Sphere in orthographic projection. Each pixel is uniquely identified by its *screen-space* normal using a pair $(\theta, \phi)$ of angular coordinates. The color at a point in $L_o$ is assumed to be the result of filtering an *unknown* lighting environment $L_i$ by a material filter $F$. Therefore we can apply the approach described by Ramamoorthi et al. [RH04], of considering rendering as a 2D spherical convolution where the material acts a low-pass filter of the incoming radiance; we may write $L_o = F * L_i$. We make the same assumptions that we have introduced in Section 2.2: 'Convex curved object of uniform isotropic material lit by distant lighting'.



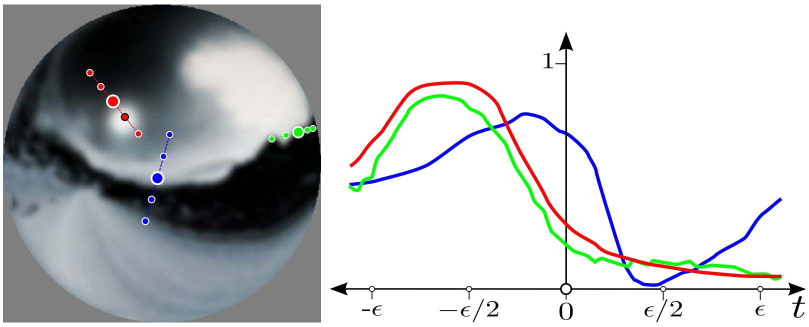
(a) Energy $\alpha(\theta)$ (b) Material filter

Figure 4.2: (a) the filter energy $\alpha(\theta)$ is the sum of a base color $\alpha_0$ and a Hermite funtion for silhouette effects (with control parameters $\theta_0$, $m_0 = 0$, $m_1$ and $\alpha_1$). (b) three slices of our material filter for $\theta = \{0, \theta_0, \frac{\pi}{2}\}$ (red points). Observe how the filter (in blue) is shifted in angles by $\mu\,\theta$ (green arrows), with its energy increasing toward $\theta = \frac{\pi}{2}$.

Even though MatCaps are artist-created images that are *not* directly related to radiance, they still convey material properties. We further restrict $F$ to be radially-symmetric on the sphere, which simplifies the estimation of these properties as it allows us to study $L_o$ in a single dimension. A natural choice of dimension is $\theta$ (see Figure 4.3a), since it also corresponds to viewing elevation in tangent space along which most material variations occur. We thus re-write the 2D spherical convolution as a 1D angular convolution of the form:

$$L_o(\theta + t, \phi) = (f * L_{i_\phi})(\theta + t),\ t \in [-\epsilon, +\epsilon], \tag{4.1}$$

where $f$ is a 1D slice of $F$ along the $\theta$ dimension, and $L_{i_\phi}$ corresponds to $L_i$ integrated along the $\phi$ dimension.



(a) Sampling (b) Intensity plots

Figure 4.3: (a) A MatCap is sampled uniformly in the $\theta$ dimension, around three different locations (in red, green and blue). (b) Intensity plots for each 1D window.

We have used the same approach in our Fourier analysis on Section 3.2. Where we have shown that starting from the Equation (3.9), similar to Equation (4.1), we obtain simple formula relating 1D image statistics to statistics of lighting and material. Moreover, we use the simplified relationship equations defined after studying the measured materials of the MERL database (Section 3.3.5). These formulas are trivially adapted to the angular parametrization based on screen-space normals (a simple change of sign in Equation (4.3)).

For a point $(\theta, \phi)$, we have:

$$\begin{aligned} K[L_o] &= K[L_{i_\phi}]\,\alpha(\theta), && (4.2)\\ \mathrm{E}[\bar{L}_o] &= \mathrm{E}[\bar{L}_{i_\phi}] - \mu\,\theta, && (4.3)\\ \mathrm{Var}[\bar{L}_o] &= \mathrm{Var}[\bar{L}_{i_\phi}] + \nu, && (4.4) \end{aligned}$$

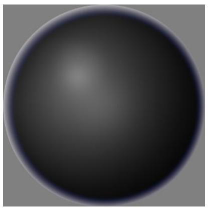
where $K$ denotes the energy of a function, hat functions are normalized by energy (e.g., $\bar{L}_o = \frac{L_o}{K[L_o]}$), and E and Var stand for statistical mean and variance respectively.

The filter parameters associated to each statistic are $\alpha$, $\mu$ and $\nu$. We make use of our fitting study (Section 3.3.5) to make a number of simplifying assumptions to ease their estimation. Equation (4.2) shows that the filter energy $\alpha(\theta)$ acts as a multiplicative term. We define it as the sum of a constant $\alpha_0$ and an optional Hermite function that accounts for silhouette effects (see Figure 4.2a). We assume only $\alpha_0$ varies per color, hence we call it the *base color* parameter. Equation (4.3) shows that the angular location of the filter is additive. The assumption here is that it is a linear function of viewing elevation (i.e., the material warps the lighting environment linearly in $\theta$); hence it is controlled by a *slope* parameter $\mu \in [0, 1]$. Lastly, Equation (4.4) shows that the *size* of the filter $\nu$ acts as a simple additive term in variance. We make the assumption that this size parameter is constant (i.e., the material blurs the lighting environment irrespective of viewing elevation). One may simply use $\mu = 0$ for the diffuse component and $\mu = 1$ for the specular component. However, we have shown evidence of correlation between $\mu$ and $\nu$, which are likely due to grazing angle effects. We use the correlation function $\mu(\nu) = 1 - 0.3\nu - 1.1\nu^2$, in effect defining slope as a function of filter size.

Putting it all together, we define our filter $F$ as a 2D spherical Gaussian: its energy varies according to $\alpha(\theta)$, it is shifted by $\mu\,\theta$ and has constant variance $\nu$. This is illustrated in Figure 4.2b, where we draw filter slices $f$ for three different viewing elevations. In the following, we first show how to evaluate the filter energy $\alpha$ (Section 4.1.2), then its size $\nu$ (Section 4.1.3), from which we obtain its slope $\mu$.

### 4.1.2 Energy estimation

The filter energy is modeled as the sum of a constant base color and an optional silhouette effect function. However, silhouette effects are scarce in MatCaps, as they require the artist to consistently apply the same intensity boost along the silhouette. In our experience, the few MatCaps that exhibit such an effect (see inset) clearly show an additive combination, suggesting a rim lighting configuration rather than a multiplicative material boost. We thus only consider the base color for estimation in artist-created MatCaps. Nevertheless, we show in Section 4.3.2 how to incorporate silhouette effects in a proper multiplicative way.



The base color $\alpha_0$ is a multiplicative factor that affects an entire MatCap component. If we assume that the brightest light source is pure white, then the corresponding point on the image is the one with maximum luminance. All MatCaps consist of low-dynamic range (LDR) images since they are captured from LDR images or painted in LDR. Hence at a point of maximum luminance, $\alpha_0$ is directly read off the image since $K[L_{i_\phi}] = 1$ in Equation (4.2). This corresponds to *white balancing* using a grey-world assumption (see Figure 4.6c). This assumption may not always be correct, but it is important to understand that we do not seek an absolute color estimation. Indeed, user manipulations presented in Section 4.3 are only made *relative* to the input MatCap.

### 4.1.3 Variance estimation

The filter size corresponds to material variance. It is related to image variance (see Eq. (4.4)).

#### Image variance

We begin by explaining how we compute image variance, the left hand side in Equation (4.4). To this end we must define a 1D window with compact support around a point $(\theta, \phi)$, and sample the MatCap along the $\theta$ dimension as shown in Figure 4.3b. In practice, we weight $L_o$ by a function $W_\epsilon : [-\epsilon, +\epsilon] \to [0, 1]$, yielding:

$$L_{o_\epsilon}(\theta + t, \phi) = L_o(\theta + t, \phi) W_\epsilon(t), \tag{4.5}$$

where $W_\epsilon$ is a truncated Gaussian of standard deviation $\epsilon/3$. Assuming $L_o$ to be close to a Gaussian as well on $[-\epsilon, +\epsilon]$, the variance of $L_o$ is related to that of $L_{o_\epsilon}$ by [Bro03]:

$$\mathrm{Var}[\bar{L}_o] \overset{\epsilon}{\simeq} \frac{\mathrm{Var}[\bar{L}_{o_\epsilon}] \cdot \mathrm{Var}[W_\epsilon]}{\mathrm{Var}[\bar{L}_{o_\epsilon}] - \mathrm{Var}[W_\epsilon]}. \tag{4.6}$$

The image variance computed at a point $(\theta, \phi)$ depends on the choice of window size.

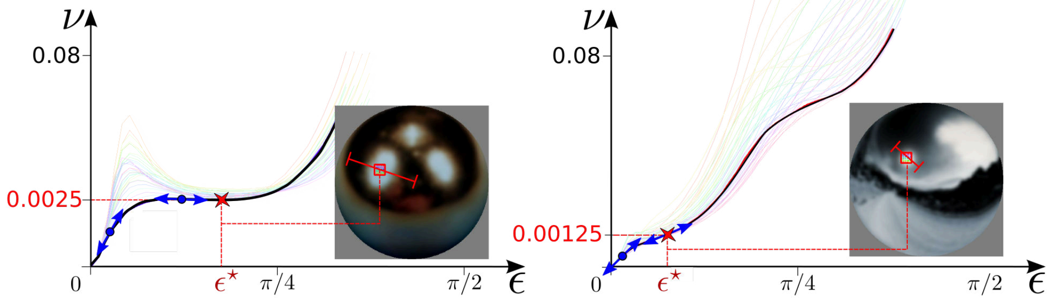
We find the most relevant window size (and the corresponding variance value) using a simple differential analysis in scale space, as shown in Figure 4.4.

Variance exhibits a typical signature: after an initial increase that we attribute to variations of $W_\epsilon$, it settles down (possibly reaching a local minimum), then raises again as $W_\epsilon$ encompasses neighboring image features. We seek the window size $\epsilon^\star$ at which the window captures the variance best, which is where the signature settles. We first locate the second inflection point which marks the end of the initial increase. Now $\epsilon^\star$ either corresponds the location of the next minimum (Figure 4.4a) or the location of the second inflection if no minimum is found (Figure 4.4b). If no second inflection occurs, we simply pick the variance at the largest window size $\epsilon^\star = \frac{\pi}{2}$ (Figure 4.4c). The computation may become degenerated, yielding negative variances (Figure 4.4d). Such cases occur in regions of very low intensity that compromise the approximation of Equation (4.6); we discard the corresponding signatures.

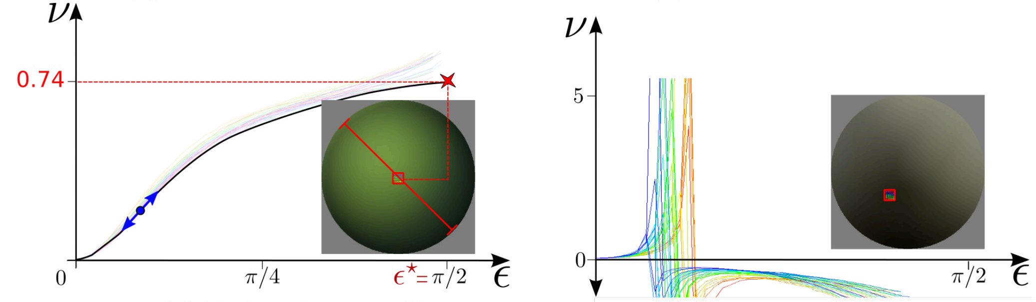
#### Material variance

The estimation of $\nu$ from Equation (4.4) requires to make assumptions on the variance of the integrated lighting $L_{i_\phi}$. If we assume that the lighting environment contains sharp point or line light sources running across the $\theta$ direction, then at those points we have $\mathrm{Var}[\bar{L}_{i_\phi}] \approx 0$ and thus $\nu \approx \mathrm{Var}[\bar{L}_o]$. Moreover, observe that Equation (4.1) remains valid when replacing $L_o$ and $L_{i_\phi}$ by their derivatives $L'_o$ and $L'_{i_\phi}$ in the $\theta$ dimension. Consequently Equation (4.4) may also be used to recover $\nu$ by relying on the $\theta$-derivative of a MatCap component. In particular, if we assume that the lighting environment contains sharp edge light sources, then at those points we have $\mathrm{Var}[\bar{L}'_{i_\phi}] \approx 0$ and thus $\nu \approx \mathrm{Var}[\bar{L}'_o]$.

In practice, we let users directly provide regions of interest (ROI) around sharpest features by selecting a few pixel regions in the image. We run our algorithm on each pixel inside a ROI, and pick the minimum variance over all pixels to estimate the material variance. The process is fast enough to provide interactive feedback, and it does not require accurate user inputs since variance is a centered statistic. An automatic procedure for finding ROIs would be interesting for batch conversion purposes, but is left to future work. Our approach is similar in spirit to that of Hu and de Hann [HDH06], but is tailored to the signatures of Figure 4.4. Note that since MatCap images are LDR, regions where intensity is clamped to 1 produce large estimated material variances. This seems to be in accordance with the way material perception is altered in LDR images [PFL09].



(a) Estimation at minimum (b) Estimation at inflexion



(c) Estimation at $\pi/2$ (d) Degenerated signatures

Figure 4.4: Our algorithm automatically finds the relevant window size $\epsilon^\star$ around a ROI (red square on MatCaps). We analyze image variances for all samples in the ROI (colored curves) as a function of window size $\epsilon$, which we call a signature. The variance estimate (red cross) is obtained by following signature inflexions (blue tangents), according to four cases: (a) Variance is taken at the first minimum after the second inflexion; (b) There is no minimum within reach, hence variance is taken *at* the second inflection; (c) There is no second inflexion, hence the variance at the widest window size is selected; (d) The signatures are degenerated and the ROI is discarded. The signature with minimum variance (black curve) is selected for material variance.

### Validation

We validate our estimation algorithm using analytical primitives of known *image* variance, as shown in Figure 4.5. To make the figure compact, we have put three primitives of different sizes and variance in the first two MatCaps. We compare ground truth image variances to estimates given by $\mathrm{Var}[\bar{L}'_o]$ (for ROIs `A` and `D`) or $\mathrm{Var}[\bar{L}_o]$ (all other ROIs), at three image resolutions. Our method provides accurate variance values compared to the ground truth, independently of image resolution, using $L_o$ or $L'_o$. The slight errors observed in `D`, `E` and `F` are due to primitives lying close to each other, which affects the quality of our estimation. The small under-estimation in the case of `I` might happen because the primitives is so large that a part is hidden from view.

To compute *material* variance, our algorithm considers the location that exhibits minimum image variance. For instance, if we assume the first two MatCaps of Figure 4.5 to be made of homogeneous materials, then their material variances will be those of `A` and `D` respectively. This implicitly assumes that the larger variances of other ROIs are due to blurred lighting features, which is again in accordance with findings in material perception [FDA03].

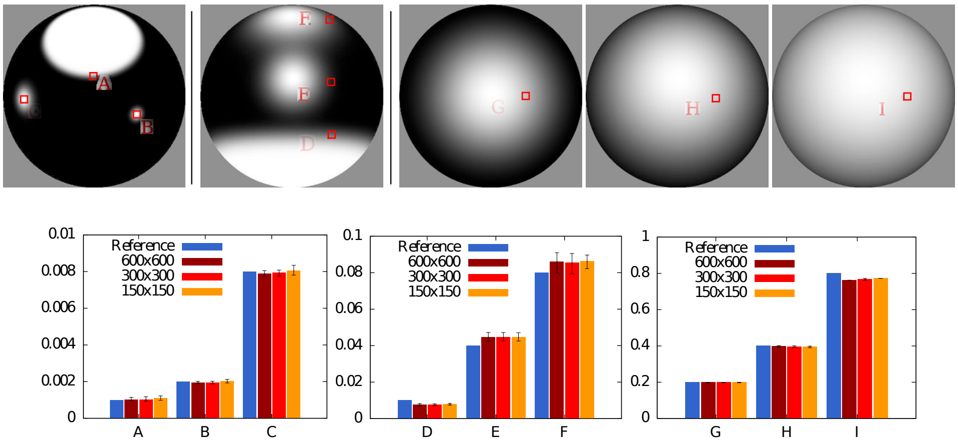


Figure 4.5: We validate our estimation algorithm on analytic primitives of known *image* variance in MatCaps. This is done at three resolutions for nine ROI marked `A` to `I`. Comparisons between known variances (in blue) and our estimates (with black intervals showing min/max variances in ROI) reveal that our algorithm is both accurate and robust.

## 4.2 MatCap decomposition

We now make use of estimated filter parameters to turn a MatCap into a representation amenable to dynamic manipulation. Figure 4.6 shows a few example decompositions. Please note that all our MatCaps are artist-created, except for the comparisons in Figs. 4.8 and 4.12.

### 4.2.1 Low-/High-frequency separation

Up until now, we have assumed that a MatCap was readily separated into a pair of components akin to diffuse and specular effects. Such components may be provided directly by the artist during the capture or painting process, simply using a pair of layers. However, most MatCaps are given as a single image where both components are blended together.

Separating an image into diffuse and specular components without additional knowledge is inherently ambiguous. Existing solutions (e.g., [NVY⁺14]) focus specifically on specular highlights, while we need a full separation. Instead of relying on complex solutions, we provide a simple heuristic separation into low-frequency and high-frequency components, which we find sufficient for our purpose. Our solution is based on a gray-scale morphological opening directly inspired by the work of Sternberg [Ste86]. It has the advantage of outputting *positive* components without requiring any parameter tuning, which we found in no other technique.

We use morphological opening to extract the low-frequency component of a MatCap. An opening is the composition of an erosion operator followed by a dilation operator. Each operator is applied *once* to all pixels in parallel, per color channel. For a given pixel $\mathbf{p}$:

$$\begin{aligned} \texttt{erode}(\mathbf{p}) &= \min_{\mathbf{q}\in\mathcal{P}} \left(\frac{v_{\mathbf{q}}}{(\mathbf{n_p}\cdot\mathbf{n_q})}\right); && (4.7)\\ \texttt{dilate}(\mathbf{p}) &= \max_{\mathbf{q}\in\mathcal{P}} \Big(v_{\mathbf{q}}\ (\mathbf{n_p}\cdot\mathbf{n_q})\Big), && (4.8) \end{aligned}$$

where $\mathcal{P} = \{\mathbf{q} \mid (\mathbf{n_p}\cdot\mathbf{n_q}) > 0\}$ is the set of valid neighbor pixels around $\mathbf{p}$, and $v_{\mathbf{q}}$ and $\mathbf{n_q}$ are the color value and *screen-space* normal at a neighbor pixel $\mathbf{q}$ respectively.

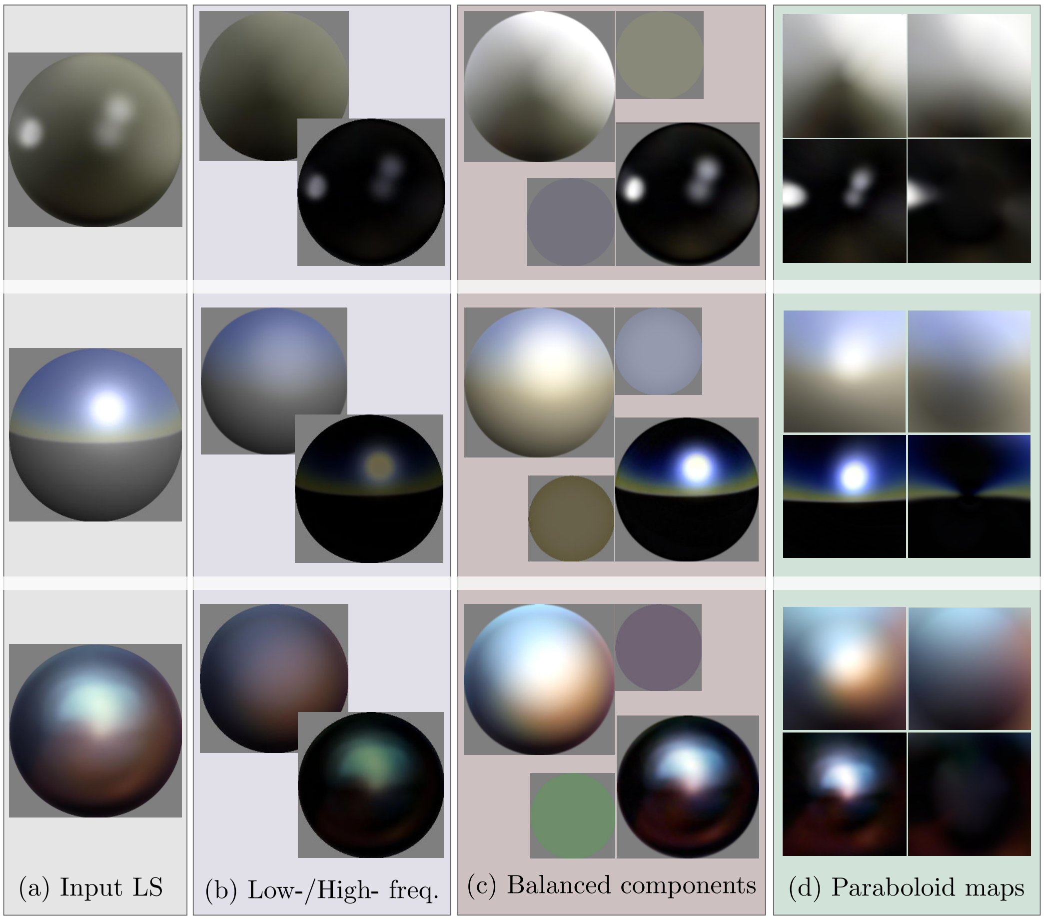
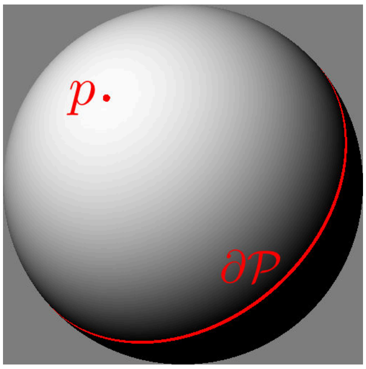


Figure 4.6: Each row illustrates the entire decomposition process: (a) An input MatCap is decomposed into (b) low- and high-frequency components; (c) white balancing separates shading from material colors; (d) components are unwarped to dual paraboloid maps using slope and size parameters.

The dot product between normals reproduces cosine weighting, which dominates in diffuse reflections. It is shown in the inset figure along with the boundary $\partial\mathcal{P}$ of neighbor pixels.



The morphological opening process is illustrated in Figure 4.7. The resulting low-frequency component is subtracted from the input to yield the high-frequency component. Figure 4.8 shows separation results on a *rendered* sphere compared to veridical diffuse and specular components. Differences are mostly due to the fact that some low-frequency details (due to smooth lighting regions) occur in the veridical specular component. As a result the specular component looks brighter compared to our high-frequency component, while the diffuse component looks dimmer than our low-frequency component. Nevertheless, we found that this approach provides a sufficiently plausible separation when no veridical diffuse and specular components exist, as with artist-created MatCaps (see Figure 4.6b for more examples).

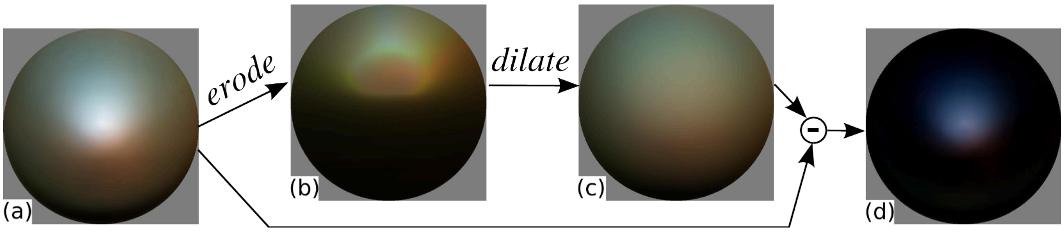
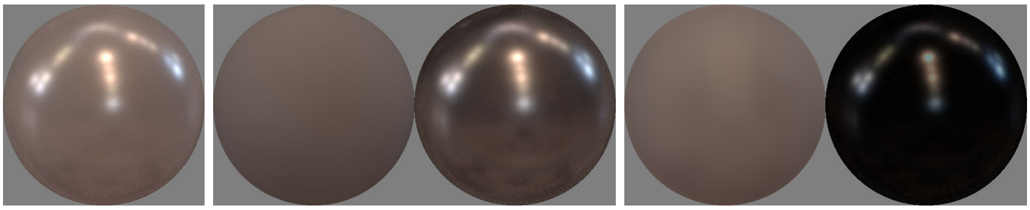


Figure 4.7: (a) An input MatCap is (b) eroded then (c) dilated to extract its low-frequency component. The high-frequency component is obtained by (d) subtracting the low-frequency component from the input MatCap.



(a) Input (b) Diffuse (c) Specular (d) Low-frequency (e) High-frequency

Figure 4.8: A rendered Matcap (a) is separated into veridical diffuse & specular components (b,c). Our low-/high-frequency separation (d,e) provides a reasonable approximation. Intensity differences are due to low-frequency details in the specular component (c) that are falsely attributed to the low-frequency component (d) in our approach. Note that $(a) = (b) + (c) = (d) + (e)$ by construction.

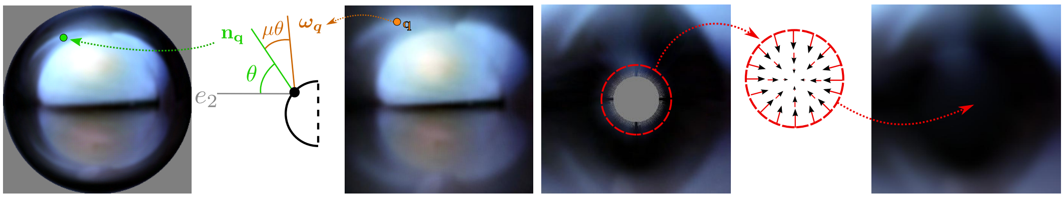
### 4.2.2 Spherical mapping & reconstruction

Given a pair of low- and high-frequency components along with estimated filter parameters, we next convert each component into a spherical representation. We denote a MatCap component by $L_o$, the process being identical in either case.

We first divide $L_o$ by its base color parameter $\alpha_0$. This yields a white-balanced image $L_o^\star$, as shown in Figure 4.6c. We then use the filter slope parameter $\mu$ to unwarp $L_o^\star$ to a spherical representation, and we use a dual paraboloid map [HS98] for storage purpose. In practice, we apply the inverse mapping to fill in the dual paraboloid map, as visualized in Figure 4.9. Each texel $\mathbf{q}$ in the paraboloid map corresponds to a direction $\omega_\mathbf{q}$. We rotate it back to obtain its corresponding normal $\mathbf{n_q} = rot_{\mathbf{u_q},-\mu\theta}(\omega_\mathbf{q})$ where $\mathbf{u_q} = \frac{\mathbf{e_2} \times \omega_\mathbf{q}}{\|\mathbf{e_2} \times \omega_\mathbf{q}\|}$, $\theta = acos(\mathbf{e_2} \cdot \omega_\mathbf{q})/(1+\mu)$ and $\mathbf{e}_2 = (0,0,1)$ stands for the (fixed) view vector in *screen space*. Since for each texel $\mathbf{q}$ we end up with a different rotation angle, the resulting transformation is indeed an image warping. The color for $\mathbf{q}$ is finally looked up in $L_o^\star$ using $\mathbf{n_q}$.

Inevitably, a disc-shaped region on the back-side of the dual paraboloid map will receive no color values. We call it the blind spot and its size depends on $\mu$: the smaller the slope parameter, the wider the blind spot. Since in our approach the slope is an increasing function $\mu(\nu)$ of filter size, a wide blind spot will correspond to a large filter, and hence a low-frequency content. It is thus reasonable to apply inpainting techniques without having to introduce new details in the back paraboloid map. In practice, we apply Poisson image editing [PGB03] with a radial guiding gradient that propagates boundary colors of the blind spot toward its center (Fig. 4.9b).

This decomposition process results in a pair of white-balanced dual paraboloid maps, one for each component, as illustrated in Figure 4.6d. They well suited to real-time rendering as they are analogous to pre-filtered environment maps (e.g., [KVHS00, RH02]).



(a) Unwraping

(b) Inpainting

Figure 4.9: We illustrate the reconstruction process, starting from a white-balanced MatCap component. (a) A dual paraboloid map is filled by warping each texel $\mathbf{q}$ to a normal $\mathbf{n_q}$; the color is then obtained by a MatCap lookup. (b) This leaves an empty region in the back paraboloid map (the 'blind spot') that is filled with a radial inpainting technique.

## 4.3 Appearance manipulation

Rendering using our decomposition is the inverse process of Section 4.2.2. The color at a point $\mathbf{p}$ on an arbitrary object is given as a function of its screen-space normal $\mathbf{n_p}$. For each component, we first map $\mathbf{n_p}$ to a direction $\omega_{\mathbf{p}}$ in the sphere: we apply a rotation $\omega_{\mathbf{p}} = rot_{\mathbf{u_p},\mu\theta}(\mathbf{n_p})$, with $\mathbf{u_p} = \frac{\mathbf{e_2}\times\mathbf{n_p}}{\|\mathbf{e_2}\times\mathbf{n_p}\|}$ and $\theta = acos(\mathbf{e_2}\cdot\mathbf{n_p})$. A shading color is then obtained by a lookup in the dual paraboloid map based on $\omega_{\mathbf{p}}$, which is then multiplied by the base color parameter $\alpha_0$. The low- and high-frequency components are finally added together.

### 4.3.1 Lighting manipulation

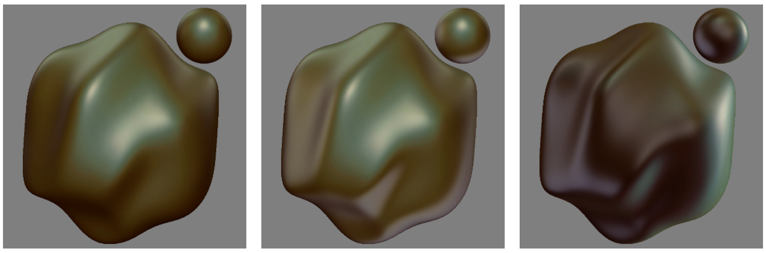
Lighting may be edited by modifying our representation given as a pair of dual paraboloid maps. We provide a painting tool to this end, as illustrated in Figure 4.10b. The user selects one of the components and paints *on the object* at a point $\mathbf{p}$. The screen-space normal $\mathbf{n_p}$ and slope parameter $\mu$ are used to accumulate a brush footprint in the dual paraboloid map. To account for material roughness, the footprint is blurred according to $\nu$. We use Gaussian- and Erf-based footprints to this end, since they enable to perform such a blurring analytically. We also provide a light source tool, which is similar to the painting tool, and is shown in Figure 4.10e. It takes as input a bitmap image that is blurred based on $\nu$. However, instead of accumulating it as in painting, it is simply moved around.

A major advantage of our decomposition is that it permits to *rotate* the whole lighting environment around. This is applied to both low- and high-frequency components in synchronization. In practice, it simply consists in applying the *inverse* rotation to $\mathbf{n_p}$ prior to warping. As shown in Figure 4.10c,f, this produces convincing results that remain coherent even with additional reflections.

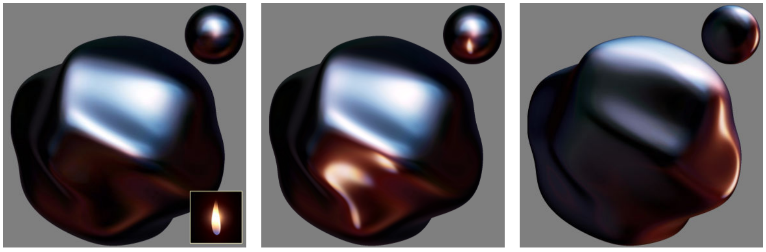
### 4.3.2 Material manipulation

Manipulating apparent material roughness requires the modification of $\nu$, but also $\mu$ since it depends on $\nu$. This is trivial for light sources that have been added or painted, as one simply has to re-render them. However, low- and high-frequency components obtained through separation of the input MatCap require additional filtering. For a rougher material look (Figure 4.11b), we decrease the magnitude of $\mu$ and blur the dual paraboloid map to increase $\nu$. For a shinier material look (Figure 4.11c), we increase the magnitude of $\mu$ and manually add reflections with a lower $\nu$ to the dual paraboloid map. We have tried using simple sharpening operators, but avoided that solution as it tends to raise noise in images.

For the manipulation of apparent material color, we take inspiration from color variation

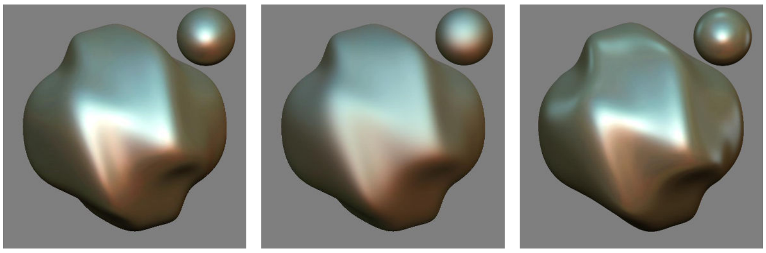


(a) Initial MatCap (b) Painted reflections (c) Rotated lighting

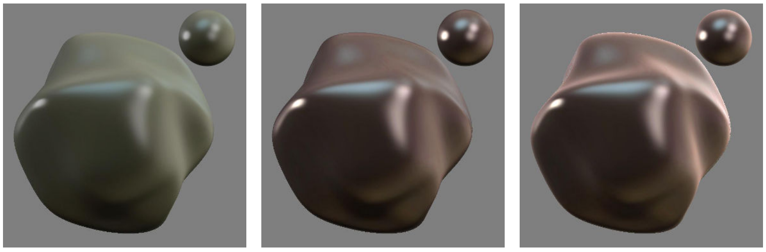


(d) Initial MatCap (e) Added reflection (f) Rotated lighting

Figure 4.10: Lighting manipulation. Top row: (a) Starting from a single reflection, (b) we modify the lighting by painting two additional reflections (at left and bottom right); (c) we then apply a rotation to orient the main light to the right. Bottom row: (d) We add a flame reflection to a dark glossy environment by (e) blurring and positioning the texture; (f) we then rotate the environment.



(a) Initial MatCap (b) Rougher material (c) Shinier material



(d) Initial MatCap (e) Modified color (f) Silhouette effect

Figure 4.11: Material manipulation. Top row: (a) Starting from a glossy appearance, (b) we increase filter size to get a rougher appearance, or (c) decrease it and add a few reflections to get a shinier appearance. Warping is altered in both cases since it is a function of filter size. Bottom row: (d) The greenish color appearance is turned into (e) a darker reddish color with increased contrast in both components; (f) a silhouette effect is added to the low-frequency component.

tools in image processing software. We let users modify the base color parameter $\alpha_0$ in HSV space, as well as the relative intensities of low- and high-frequency components as shown in Figure 4.11e. Even though silhouette effects are uncommon in input MatCaps, we provide means to incorporate them at the rendering stage as illustrated in Figure 4.11f. Each color channel is increased by the same silhouette function (see Figure 4.2), with users controlling the $\theta_0$, $\alpha_1$ and $m_1$ parameters.

## 4.4 Results and comparisons

Our material estimation algorithm (Section 4.1) is implemented on the CPU and runs in real-time on a single core of an Intel i7-2600K 3.4GHz, allowing users to quickly select appropriate ROIs. The decomposition process (Section 4.2) is implemented in Gratin (a GPU-tailored nodal software available at `http://gratin.gforge.inria.fr/`), using an Nvidia GeForce GTX 555. Performance is largely dominated by the low-/high-frequency separation algorithm, which takes from 2 seconds for a $400 \times 400$ MatCap, to 6 seconds for a $800 \times 800$ one. Rendering (Section 4.3) is implemented in Gratin as well and runs in real-time on the GPU, with a negligible overhead compared to rendering with a simple MatCap. We provide GLSL shaders for rendering with our representation in supplemental material.

A benefit of our approach is the possibility to rotate lighting independently of the view. One may try to achieve a similar behavior with a mirrored MatCap to form an entire sphere. However, this is equivalent to a spherical mapping, in which case highlights do not move, stretch or compress in a plausible way.

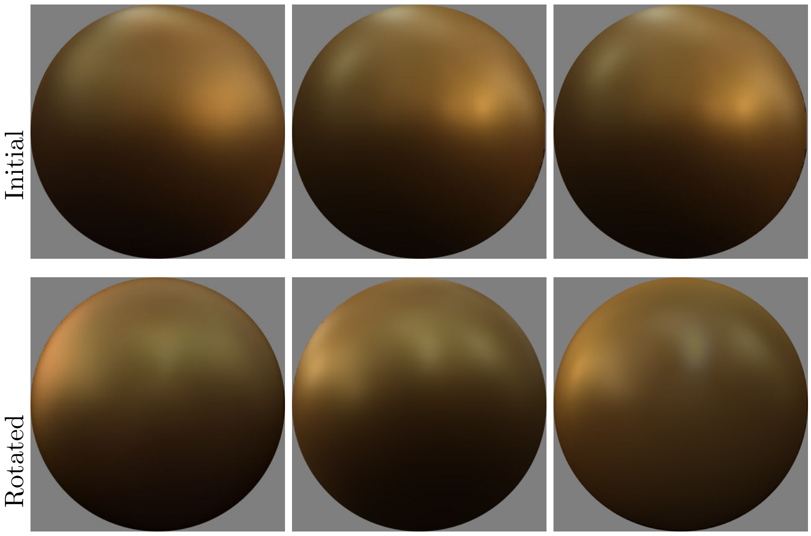
In this paper, we have focused on artist-created MatCaps for which there is hardly any ground truth to compare to. Nevertheless, we believe MatCaps should behave similarly to rendered spheres when lighting is rotated. Figure 4.12 shows a lighting rotation applied to the rendering of a sphere, for which a ground truth exists. We also compare to a rotation obtained by the method of Lombardi et al. [LN12]. For the *specific case* of lighting rotation, our approach appears superior; in particular, it reproduces the original appearance exactly. However, the method of Lombardi et al. has an altogether different purpose, since it explicitly separates material *and* lighting. For instance, they can re-render a sphere with the same lighting but a different material, or with the same material but a different lighting.

Up to this point, we have only exploited a single MatCap in all our renderings. However, we may use low- and high-frequency components coming from different MatCaps, as shown in Figure 4.13. Different MatCaps may of course be used on different object parts, as seen in Figure 4.14. Our approach offers several benefits here: the input Matcaps may be aligned, their color changed per components, and they remain aligned when rotated.

Our representation also brings interesting spatial interpolation abilities, since it provides material *parameters* to vary. Figure 4.15 shows how bitmap textures are used to vary high- and low-frequency components separately. Figure 4.16 successively makes use of an ambient occlusion map, a diffuse color map, then silhouette effects to convey object shape. Our approach thus permits to obtain spatial variations of appearance, which are preserved when changing input MatCaps.

## 4.5 Discussion

We have shown how to decompose a MatCap into a representation more amenable to dynamic appearance manipulation. In particular, our approach enables common shading operations such as lighting rotation and spatially-varying materials, while preserving the appeal of artist-created MatCaps. We are convinced that our work will quickly prove useful in software that already make use of MatCaps (firstly 3D sculpting, but also CAD and scientific



(a) Lombardi et al. (b) Ground truth (c) Our approach

Figure 4.12: Comparison on lighting rotation. The top and bottom rows show initial and rotated results respectively. (b) Ground truth images are rendered with the `gold paint` material in the Eucalyptus Grove environment lighting. (a) The method of Lombardi et al. [LN12] makes the material appear rougher both before and after rotation. (c) Our approach reproduces exactly the input, and better preserves material properties after rotation.

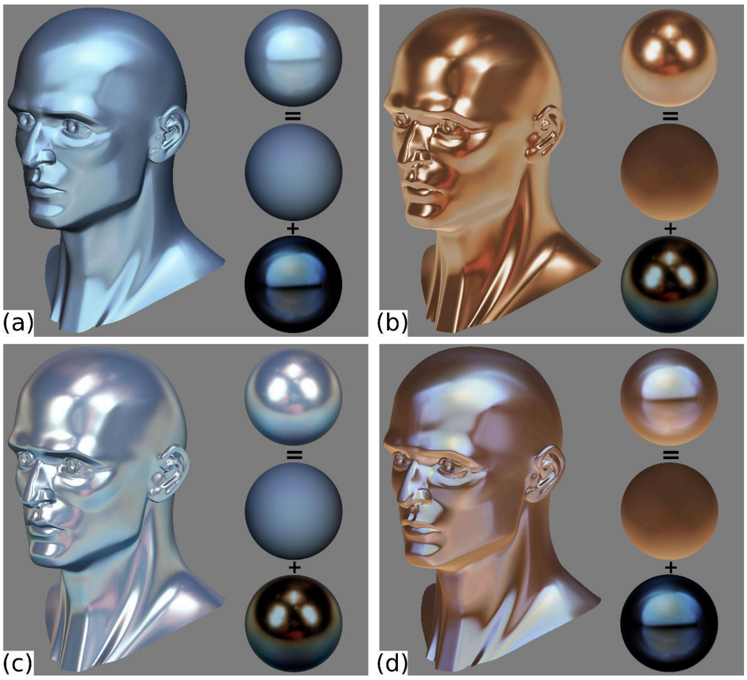


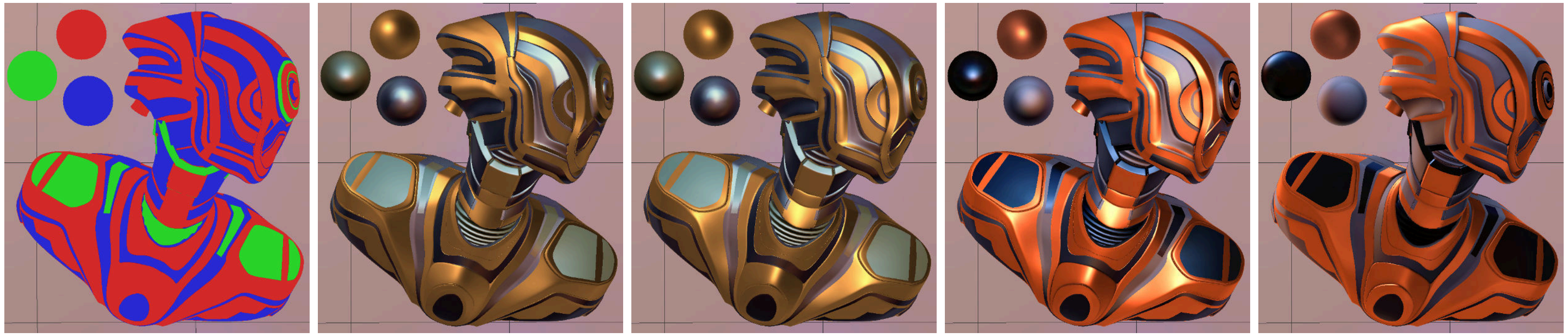
Figure 4.13: Mixing components. (a,b) Two different MatCaps applied to the same head model. Thanks to our decomposition, components may be mixed together: (c) shows the low-frequency component of (a) added to the high-frequency component of (b); (d) shows the reverse combination.

visualization), with a negligible overhead in terms of performance but a greater flexibility in terms of appearance manipulation.

We believe that this work restricted to MatCaps is easily transfered to other kind of inputs, once an spherical representation of shading is obtained. We presume that the estimation of materials will be valid independently if the input comes from an artwork, a photograph or a rendering. However, sharp lighting assumption might not always be met, in which case material parameters will be over- or under-estimated. This will not prevent our approach from working, since it will be equivalent to having a slightly sharper or blurrier lighting. Interestingly, recent psycho-physical studies (e.g., [DBM10]) show that different material percepts may be elicited only by changing lighting content. This suggests that our approach could be in accordance with visual perception, an exciting topic for future investigations.

Our decomposition approach makes a number of assumptions that may not always be satisfied. We assume an additive blending of components, whereas artists may have painted a MatCap using other blending modes.

For further explanation of our technique limitations like the restriction of radially symmetric BRDFs or the needed improvements of the filling-in technique we refer to the Chapter 6.



(a) Material Ids

(b) Initial MatCaps

(c) Aligned MatCaps

(d) Material changes

(e) Lighting rotation

Figure 4.14: Using material IDs (a), three MatCaps are assigned to a robot object (b). Our method permits to align their main highlight via individual rotations (c) and change their material properties (d). All three MatCaps are rotated together in (e).

(a) Initial appearance

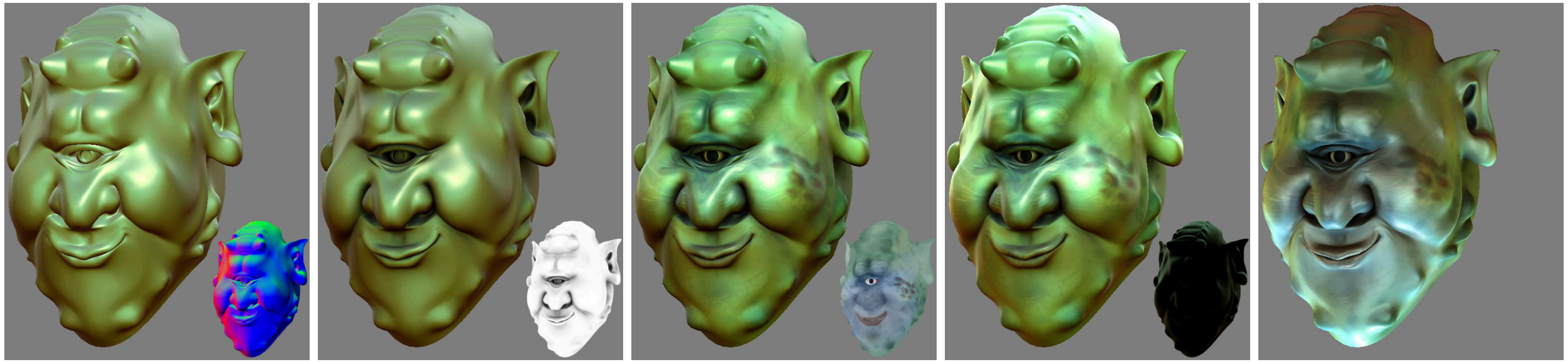
(b) Color tex. low-frequency

(c) Color tex. high-frequency

(d) Roughness variations

(e) Lighting rotation

Figure 4.15: Spatially-varying colors. (a) The MatCap of Figure 4.6 (2nd row) is applied to a cow toy model. A color texture is used to modulate (b) the low-frequency component, then (c) the high-frequency component. (d) A binary version of the texture is used to increase roughness outside of dark patches (e.g., on the cheek). In (e) we rotate lighting to orient it from behind.



(a) Initial appearance

(b) Occlusion map

(c) Diffuse map

(d) Silhouette effects

(e) Lighting rotations

Figure 4.16: Shape-enhancing variations. (a) A variant of the MatCap of Figure 4.10 (1st row) is applied to an ogre model. (b) An occlusion map is used to multiply the low- and high-frequency components. (c) A color texture is applied to the low-frequency component. (d) Different silhouette effects are added to each component. (e) Lighting is rotated so that it comes from below.

# Chapter 5

# Local Shape Editing at the Compositing Stage

Images created by rendering engines are often modified in post-process, by making use of independent, additive shading components such as diffuse or reflection shading or transparency (Section 1.1.3). Most modern off-line rendering engines output these shading components in separate image buffers without impeding rendering performance. In the same way, it is possible to output auxiliary components such as position or normal buffers. These auxiliary buffer permit additional modifications: adding lights in post-process (using normals) or depth of field (using positions) for instance. Nevertheless, these modifications are limited and if we would like to modify auxiliary buffers holding 3D normals or positions, this would have no effect on shading buffers.

Following our goal to modify existing appearance we want to grant geometry modifications. We specifically focus on ways to obtain a plausible shading color when modifying local shape (normals). If one wants to make these kinds of modifications, the straightforward solution will be to completely re-render the scene in 3D. This is a time-consuming process that we want to avoid to be able to explore modifications interactively. Such post-processing techniques are routinely used in product design applications (e.g., Colorway) or in movie production (e.g., Nuke or Flame) to quickly test alternative compositions. They are most often preferred to a full re-rendering of the 3D scene that would require considerably larger assets, longer times and different artistic skills.

The main issue when one wants to modify shape at the compositing stage is that lighting information is not available any more as is is lost in the rendering process. The recovery of the environment lighting would not be possible as we lack much of the necessary 3D data. We instead strive for a plausible result, ensuring that the input diffuse and reflection shading buffers are recovered when reverting to the original normals

As in Chapter 4 we will be working with pre-filtered environments maps. While with MatCaps we use pre-filtered environments maps to mimic modifications of material or lighting, now we use them to allow geometry modifications. The key idea of our method is to *reconstruct* a pair of pre-filtered environments per object/material: one for the diffuse term, the other for the reflection term. Obtaining new shading colors from arbitrarily modified normals then amounts to perform a pair of lookups in the respective prefiltered environment maps. Modifying local shape in real-time then becomes a matter of *recompositing* the reconstructed shading buffers.

Alternatively we could export an environment lighting map, pre-filtered or not, per object/material in the rendering process. Afterwards they will be used to perform desired modifications of local shape. This solution requires to obtain the environment maps using light-transport techniques to capture the effects of the interaction between different objects (i.e. shadows or reflections). That results in a tedious and costly approach because each environment map would need a complete re-render of the whole scene. Moreover retro-reflections could not be obtained as each object needs to be replaced by a sphere to get its environment map.

Our approach is a first step toward the editing of surface shape (and more generally object appearance) at the compositing stage, which we believe is an important and challenging problem. The rendering and compositing stages are clearly *separate* in practice, involving different artistic skills and types of assets (i.e., 3D scenes vs render buffers). Providing compositing artists with more control over appearance will thus require specific solutions. Our paper makes the following contributions toward this goal (see Figure 5.1):

- Diffuse and reflection shading environments are *automatically* reconstructed in pre-process for each object/material combinations occurring in the reference rendering (Section 5.1);
- The reconstructed environments are used to create new shading buffers from *arbitrarily* modified normals, which are recomposited in real time with the reference shading buffers (Section 5.2).

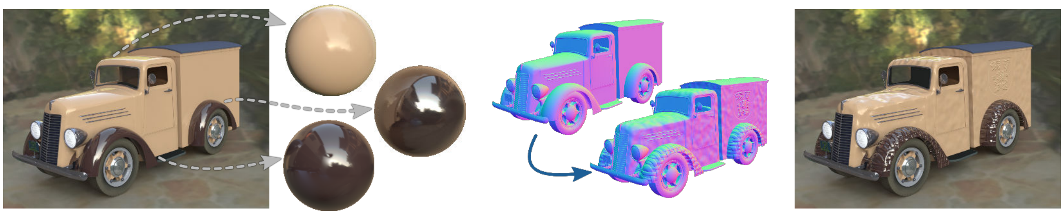

Reference rendering    Reconstructed env.    Original → modified normals    Our recompositing result

Figure 5.1: Our method permits to modify surface shape by making use of the shading and auxiliary buffers output by modern renderers. We first reconstruct shading environments for each object/material combination of the `Truck` scene, relying on normal and shading buffers. When normals are then modified by the compositing artist, the color image is recomposited in real-time, enabling interactive exploration. Our method reproduces inter-reflections between objects, as seen when comparing the reconstructed environments for rear and front mudguards.

This work have been published in the Eurographics Symposium of Rendering at the 'Experimental Ideas & Implementations' with the collaboration of Gael Guennebaud and Romain Vergne [ZPGVB16]. Specifically, Gael Guennebaud has developed the reconstruction of the diffuse component and the hole-filling and regularization has been developed by Gael Guennebaud and Romain Vergne.

## 5.1 Reconstruction

In this work, we focus on modifying the apparent shape of *opaque* objects at the compositing stage. We thus only consider the diffuse and reflection buffers, the latter resulting from reflections off either specular or glossy objects. These shading buffers exhibit different frequency content: diffuse shading is low-frequency, while reflection shading might contain arbitrarily high frequencies. As a result we use separate reconstruction techniques for each of them.

Both techniques take their reference shading buffer $D_0$ or $R_0$ as input, along with an auxiliary normal buffer $\mathbf{n}$. Reconstruction is performed separately for each surface patch $\mathcal{P}$ belonging to a same object with a same material, and identified thanks to a surface ID buffer. We also take as input ambient and reflection occlusion buffers $\alpha_D$ and $\alpha_R$, which identify diffuse and specular visibility respectively (see Appendix). The latter is used in the reconstruction of the reflection environment. We use Modo to export all necessary shading and auxiliary buffers, as well as camera data. The normals are transformed from world to screen-space prior to reconstruction; hence the diffuse and reflection environments output by our reconstruction techniques are also expressed in screen space.

### 5.1.1 Diffuse component

In this section our goal is to reconstruct a complete prefiltered diffuse environment $D : S^2 \to \mathbb{R}^3$, parametrized by surface normals $\mathbf{n}$. The map $D$ should match the input diffuse buffer inside $\mathcal{P}$: for each selected pixel $\mathbf{x} \in \mathcal{P}$, $D(\mathbf{n}(\mathbf{x}))$ should be as close as possible to $D_0(\mathbf{x})$. However, as illustrated by the Gauss map in Figure 5.2, this problem is highly ill-posed without additional prior. First, it is under-constrained in regions of the unit sphere *not covered* by the Gauss map of the imaged surface. This is due to sharp edges, occluded regions and highly curved features producing a very sparse sampling. Second, the problem might also be over-constrained in areas of the Gauss map covered by *multiple sheets* of the imaged surface, as they might exhibit different diffuse shading values due to global illumination (mainly occlusions).

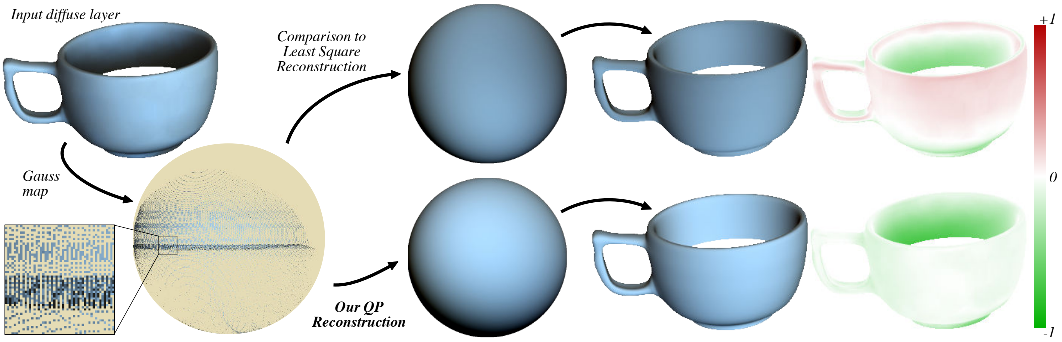


Figure 5.2: Reconstruction of the prefiltered diffuse environment map. From left to right: the pixels of the input diffuse layer of the selected object are scattered inside the Gauss map. This shading information is then approximated by low-order spherical harmonics using either a Least Square fit (top) or our Quadratic Programming formulation (bottom) leading to the respective reconstructed environment maps. To evaluate the reconstruction quality, those environment maps are then applied to the original objects, and the signed residual is shown using a color code (shown at right). Observe how our QP reconstruction guarantees a negative residual.

Since diffuse shading exhibits very low frequencies, we address the coverage issue by representing $D$ with low-order Spherical Harmonics (SH) basis functions, which have the double advantage of being globally supported and of exhibiting very good extrapolation capabilities. We classically use order-2 SH, only requiring 9 coefficients per color channel [RH01a]. The reconstructed prefiltered diffuse environment is thus expressed by:

$$D(\mathbf{n}) \quad = \quad \sum_{l=0}^{2} \sum_{m=-l}^{l} c_{l,m} Y_{l,m}(\mathbf{n}). \tag{5.1}$$

The multiple-sheets issue is addressed by reconstructing a prefiltered environment as if no occlusion were present. This choice will facilitate shading edition as the local residual $D_0(\mathbf{x}) - D(\mathbf{n}(\mathbf{x}))$ is then directly correlated to the amount of local occlusion. Formally, it amounts to the following constrained quadradic minimization problem:

$$\begin{aligned} c^\star_{l,m} &= \arg\min_{c_{l,m}} \sum_{\mathbf{x}\in\mathcal{P}} \|D_0(\mathbf{x}) - D(\mathbf{n}(\mathbf{x}))\|^2, \\ s.t. \quad & D(\mathbf{n}(\mathbf{x})) \geq D_0(\mathbf{x}), \end{aligned}$$

which essentially says that the reconstruction should be as close as possible to the input while enforcing negative residuals. This is a standard Quadratic Programming (QP) problem that we efficiently solve using a dual iterative method [GI83].

Figure 5.2 compares our approach to a standard least squares (LS) reconstruction. As made clear in the two right-most columns, our QP method produces shading results more plausible than the LS method: residuals are negative by construction and essentially correspond to darkening by occlusion.

#### Validation

The left column of Figure 5.6 shows reconstructed diffuse shading for a pair of environment illuminations. We project the light probes onto the SH basis and use them to render a 3D teapot model, from which we reconstruct the diffuse environments. In this case, there is no occlusion, only direct lighting: our reconstruction exhibits very good results as shown by the difference images with the ground truth environments.

### 5.1.2 Reflection component

At first glance, the problem of reconstructing a prefiltered reflection environment map is similar to the diffuse case. Our goal is to reconstruct a complete prefiltered reflection environment $R : S^2 \to \mathbb{R}^3$ parametrized by the reflected view vector $\mathbf{r} = \text{reflect}(\mathbf{v}, \mathbf{n})$, where $\mathbf{v}$ and $\mathbf{n}$ are the local view and normal vectors respectively. As before, $R$ should match the input reflection buffer: for each selected pixel $\mathbf{x} \in \mathcal{P}$, $R(\mathbf{r}(\mathbf{x}))$ should be as close as possible to $R_0(\mathbf{x})$.

On the other hand, the reflection buffer contains arbitrarily high-frequencies, which prohibits the use of a SH representation. We thus propose to represent and store $R$ in a high resolution dual-paraboloid map [HS98] that we fill in three steps:

1. Mapping from $R_0$ to $R$ while dealing with overlapping sheets;
2. Hole-filling of $R$ using a spherical harmonic interpolation;
3. Smoothing of the remaining discontinuities.

#### Partial reconstruction

For the reconstruction of the diffuse map, interpreting each input pixel as a simple point sample proved to be sufficient. However, in order to reconstruct a sharp reflection map, it becomes crucial to fill the gaps between the samples with as much accuracy as possible. To this end, we partition the input set of pixels into smooth patches according to normal and depth discontinuities. Each smooth patch has thus a continuous (i.e., hole-free) image once mapped on $S^2$ through the reflected directions $\mathbf{r}$. Due to the regular structure of the input pixel grid, the image of the patch is composed of adjacent spherical quads and triangles (at patch boundaries). This is depicted in Figure 5.3 for a block of $3 \times 3$ pixels.

Depending on object shape and camera settings, each patch may self-overlap, and the different patches can also overlap each other. In other words, a given reflection direction $\mathbf{r}$ might coincide with several polygons. We combine shading information coming from these different image locations using two types of weights. First, we take as input an auxiliary

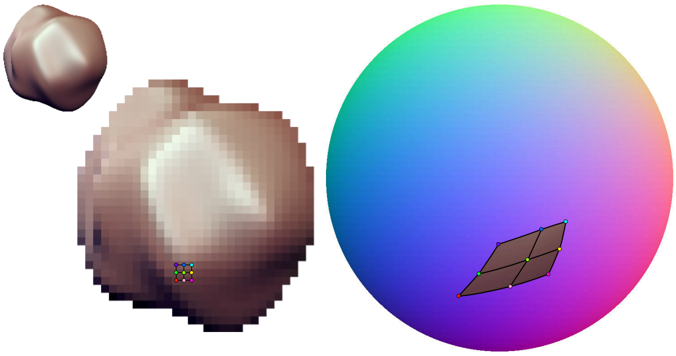


Figure 5.3: A 3×3 pixel neighborhood (shown at left on a low-resolution image for clarity) is mapped to four contiguous spherical quads on the Gauss sphere of rectified normals (right). The color inside each spherical quad is computed by bilinear interpolation inside the input image using spherical barycentric coordinates.

buffer $\alpha_R$ storing *reflection occlusions*. Examples are shown in the right of Figure 5.7.This information tells us whether the majority of reflected rays used to compute the shading value come from the environment (i.e., other objects). This buffer can be treated as a binary mask to discard polygons exhibiting self-reflections as those polygons are unreliable to reconstruct the environment. Second, among the remaining polygons, we favor the ones that have a small extent on the unit sphere; they correspond to image locations where the object is the least curved, and hence provide the most accurate shading information. Formally, we compute a weight $w_k = (1 - \cos^{-1}(\ell_k)/\pi)^\eta$ for each polygon $k$, with $\ell_k$ the longest arc defined by any pair of its vertices. We use $\eta = 200$, which has the effect of selecting the smallest polygons with only a slight blend between nearly equal-sized ones. A partial reflection environment is thus obtained by combining recovered shading values for any query direction $\mathbf{r}$:

$$R(\mathbf{r}) = \frac{\sum_{k=1}^{N(\mathbf{r})} w_k \sum_{j \in Q_k} \lambda_j^k(\mathbf{r}) R_0(\mathbf{x}_j)}{\sum_{k=1}^{N(\mathbf{r})} w_k}, \tag{5.2}$$

where $N(\mathbf{r})$ is the number of *unoccluded* spherical polygons containing $\mathbf{r}$, $Q_k$ is the set of corner indices of the $k$-th polygon, and $\lambda_j^k$ are barycentric coordinates enabling the interpolation of the shading colors inside the $k$-th polygon. For polygons embedded on a sphere, barycentric coordinates can be computed as described in Equation 8 of [LBS06]. We use normalized spherical coordinates, which amounts to favor the partition of unity property over the linear precision property on the sphere (i.e., Equation 2 instead of 3 in [LBS06]).

In order to quickly recover the set of spherical polygons containing $\mathbf{r}$, we propose to first warp the set $S^2$ of reflected directions to a single hemisphere so that the search space can be more easily indexed. To this end, we compute a *rectified normal* buffer $\mathbf{n}'$ such that $\mathbf{r} = \text{reflect}(\mathbf{z}, \mathbf{n}')$, where $\mathbf{z} = (0, 0, 1)^T$ as shown in Figure 5.4. This is obtained by the bijection $\mathbf{n}'(\mathbf{r}) = \frac{\mathbf{r}+\mathbf{z}}{\|\mathbf{r}+\mathbf{z}\|}$. In a preprocess, we build a 2D grid upon an orthographic projection of the Gauss map of these rectified screen-space normals. For each polygon corresponding to four or three connected pixels, we add its index in the cells it intersects. The intersection is carried out conservatively by computing the 2D convex hull of the projected spherical polygon. Then, for each query reflection vector $\mathbf{r}$, we compute the spherical barycentric coordinates $\lambda_j^k$ of each polygon of index $k$ in the cell containing $\mathbf{n}'(\mathbf{r})$, and pickup the polygons having all $\lambda_j^k$ positive. In our implementation, for the sake of simplicity and consistency with our 2D grid construction, we compute spherical barycentric coordinates with respect to the rectified normals $\mathbf{n}'$, for both intersections and shading interpolation (Equation (5.2)).

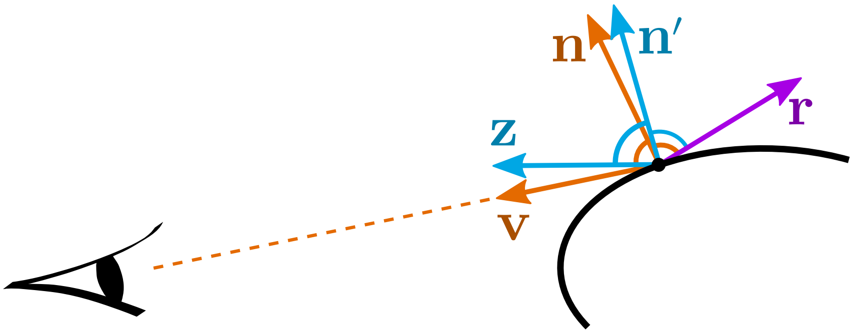


Figure 5.4: A normal **n** is 'rectified' to **n**′ prior to reconstruction. The reflection of the view vector **v** is then given by $\mathbf{r} = \text{reflect}(\mathbf{z}, \mathbf{n}')$.

### Hole-filling and regularization

Evaluating Equation (5.2) for each direction **r** in a dual-paraboloid map yields a partial reconstruction, with holes in regions not covered by a single polygon and discontinuities at the transition between different smooth parts.

For instance, the bright red region in the top row of Figure 5.5 correspond to reflection directions where no shading information is available (i.e., $N(\mathbf{r}) = 0$). It is necessary to fill these empty regions to guarantee that shading information is available for all possible surface orientations. In practice, we perform a harmonic interpolation directly on a densely tessellated 3D sphere, with vertices indexed by **r** matching exactly the pixels of the dual-paraboloid map. The tessellation is thus perfectly regular except at the junction between the front and back hemispheres. We use a standard Finite Element Discretization with linear basis functions over a triangular mesh to solve the Laplacian differential equation, while setting shading values recovered by Equation (5.2) as Dirichlet boundary constraints [IGG+14].

A result is shown in the middle row of Figure 5.5: holes are properly filled in, but some shading discontinuities remain. Those are caused by spatial discontinuities in Equation (5.2) occurring when spatially disconnected polygons are used for neighboring directions in the environment. We thus apply a last post-processing step where we slightly blur the environment along those discontinuities. We identify them by computing a second dual-paraboloid map storing the 2D image position of the polygon that contributed the respective shading color. This map is simply obtained by replacing the shading values $R_0(\mathbf{x}_j)$ in Equation (5.2) by the 2D coordinates $\mathbf{x}_j$. We then compute the gradient of these maps and use its magnitude to drive a spatially-varying Gaussian blur. The effect is to smooth the radiance on discontinuities caused by two or more remote polygons projected next to one another. An example of regularized environment is shown in the bottom row of Figure 5.5.

### Visualization

Dual paraboloid maps are only used for practical storage purposes in our approach. It should be noted that once applied to a 3D object, most of the shading information in the back part of the map gets confined to the silhouette, as shown in the right column of Figure 5.5. In the following, we thus prefer to use shaded 3D spheres seen in orthographic projection (i.e., Lit Spheres [SMGG01]) to visualize the reconstructed shading environments (both diffuse and

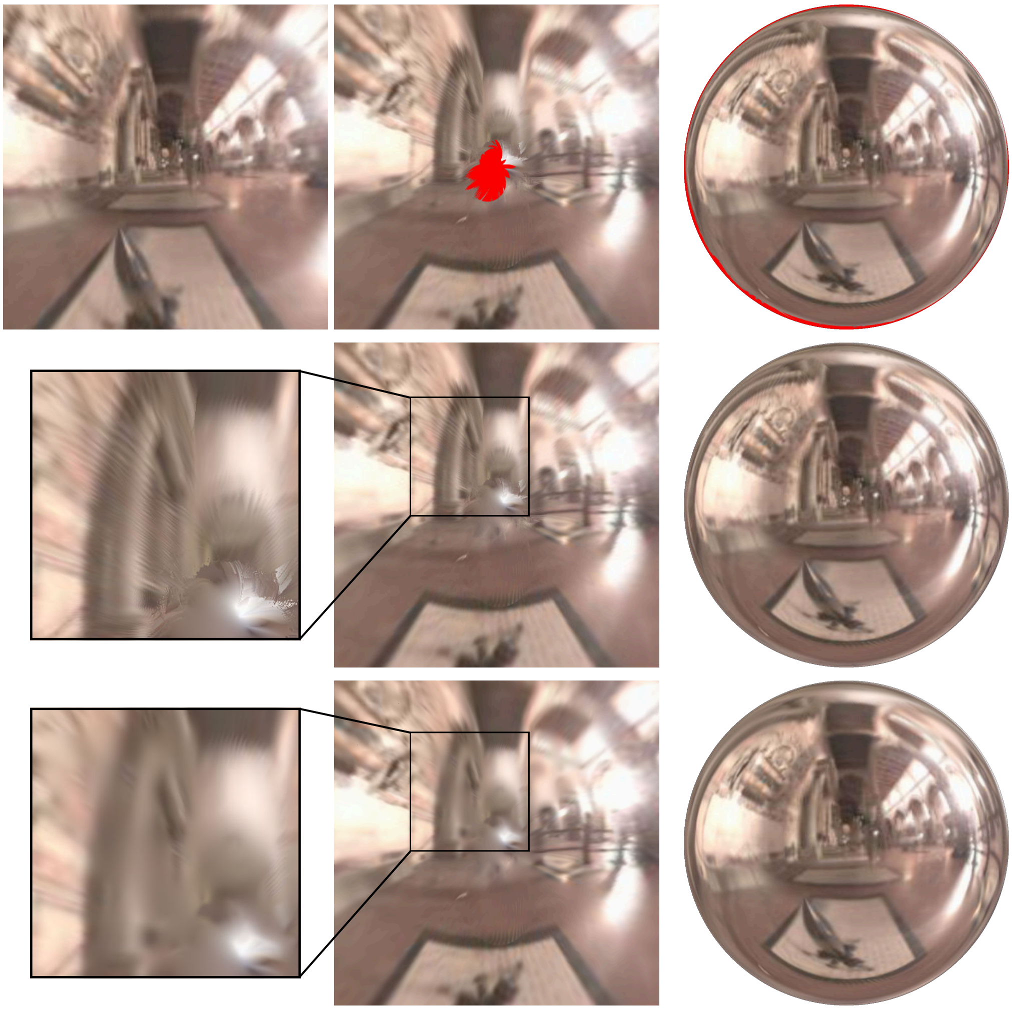


Figure 5.5: A dual paraboloid map reconstructed with our approach is shown in the top row: it only partially covers the space of reflected view vectors. The missing information, shown in red, appears confined to the silhouette when visualized with a LitSphere at right. We focus on the back part of the paraboloid map in the middle row to show the result of the hole-filling procedure. Missing regions are completed, but some discontinuities remain; they are corrected by our regularization pass as shown in the bottom row.

reflections). In practice, perspective projection only lets appear a subset of filled-in shading values close to object contours.

### Validation

The right column of Figure 5.6 shows reconstruction results for a pair of known environment lighting. As before, a 3D teapot is rendered using the light probe, and then used for reconstructing reflection environment maps. The difference between our reconstruction and the ground truth is small enough to use it for shape editing.

Figure 5.6: Reconstruction results for diffuse (left column) and reflection (right column) shading, using `Galileo` (top row) and `rnl` (bottom row) light probes. Each quadrant shows the reference prefiltered environment, along with the rendering of a 3D teapot assuming distant lighting and no shadowing or inter-reflections. The teapot image is then used to reconstruct the prefiltered environment using either the method of Section 5.1.1 or 5.1.2, and a boosted absolute color difference with the reference environment is shown.

## 5.2 Recompositing

The outcome of the reconstruction process is a set of SH coefficients and dual paraboloid maps for all object/material combinations appearing in the image. Obtaining reconstructed shading buffers simply amounts to evaluate shading in the appropriate SH basis or environment map, using arbitrary screen-space normals. A benefit of this approach is that any normal manipulation may then be used in post-process; we give some practical examples in Section 5.3.

However, we must also ensure that the independently reconstructed shading buffers are seamlessly recombined in the final image. In particular, when a normal is left untouched by the compositing artist, we must guarantee that we reproduce the reference diffuse and reflection shading colors exactly. This is the goal of the recompositing process: taking as input an arbitrarily modified normal buffer, it combines the reconstructed prefiltered environments with rendered input buffers to produce a final color image where the apparent shape of objects has been altered. It works in parallel on all pixels; hence we drop the dependence on $\mathbf{x}$ for brevity.

### Combined diffuse term

Given a modified normal $\tilde{\mathbf{n}}$, we define the combined diffuse term $\tilde{D}$ by:

$$\tilde{D} \quad = \quad \alpha_D \left\lfloor \underbrace{D_0 - D(\mathbf{n})}_{\text{residual}} + D(\tilde{\mathbf{n}}) \right\rfloor + (1 - \alpha_D)\, D_0, \tag{5.3}$$

where the ambient occlusion term $\alpha_D$ is used to linearly interpolate between the reference and reconstructed diffuse colors. The rationale is that highly occluded areas should be preserved to prevent the introduction of unnatural shading variations. The $D_0 - D(\mathbf{n})$ term is used to re-introduce residual differences between the reference and reconstructed buffers. It corresponds to local darkening of diffuse shading that could not be captured by our global reconstruction. The $\lfloor \cdot \rfloor$ symbol denotes clamping to 0, which is necessary to avoid negative diffuse shading values. This is still preferable to a multiplicative residual term $D_0/D(\mathbf{n})$ as it would raise numerical issues when $D(\mathbf{n}) \approx 0$. Observe that if $\mathbf{n} = \tilde{\mathbf{n}}$ then $\tilde{D} = D_0$: the reference diffuse shading is exactly recovered when the normal is not modified.

### Combined reflection term

Contrary to the diffuse case, we cannot apply the residual approach between the reference and reconstructed reflection buffers as it would create ghosting artifacts. This is because reflections are not bound to low-frequencies as in the diffuse case. Instead, given a modified normal $\tilde{\mathbf{n}}$ and a corresponding modified reflection vector $\tilde{\mathbf{r}} = \mathrm{reflect}(\mathbf{v}, \tilde{\mathbf{n}})$, we define the combined reflection term $\tilde{R}$ by:

$$\tilde{R} \quad = \quad \nu_{\mathbf{r},\tilde{\mathbf{r}}}\, \alpha_R\; R(\tilde{\mathbf{r}}) + (1 - \nu_{\mathbf{r},\tilde{\mathbf{r}}}\, \alpha_R)\; R_0, \tag{5.4}$$

where $\nu_{\mathbf{r},\tilde{\mathbf{r}}} = \min(1, \cos^{-1}(\mathbf{r} \cdot \tilde{\mathbf{r}})/\epsilon)$ computes the misalignment between original and modified reflection vectors (we use $\epsilon = 0.005\pi$), and $\alpha_R$ is the reflection occlusion term. The latter serves the same purpose as the ambient occlusion term for the diffuse case. Equation (5.4) performs a linear interpolation between reference and reconstructed reflection colors based on $\nu_{\mathbf{r},\tilde{\mathbf{r}}}\, \alpha_R$. As a result, if $\mathbf{n} = \tilde{\mathbf{n}}$, then $\nu_{\mathbf{r},\tilde{\mathbf{r}}} = 0$ and $\tilde{R} = R_0$: the reference reflection shading is exactly recovered when the normal is left unmodified.

### Final composition

The final image intensity $\tilde{I}$ is given by:

$$\tilde{I} \quad = \quad \alpha \left(k_D\, \tilde{D} + k_R\, \tilde{R}\right)^{\frac{1}{\gamma}} + (1 - \alpha)\, I, \tag{5.5}$$

where the diffuse and reflection coefficients $k_D$ and $k_R$ are used to edit the corresponding shading term contributions ($k_D = k_R = 1$ is the default), $\gamma$ is used for gamma correction (we use $\gamma = 2.2$ in all our results), and $\alpha$ identifies the pixels pertaining to the background (e.g., showing an environment map), which are already gamma corrected in our input color image $I$. Equation (5.5) is carried out on all color channels separately. Figure 5.7 shows an example of the recompositing process on a simple scene holding a single object, where input normals have been corrupted by a 2D Perlin noise restricted to a square region. The final colors using original and modified normals are shown in the leftmost column; the remaining columns show the different gamma-corrected shading terms. The top row focuses on the diffuse term ($k_R = 0$), while the bottom row focuses on the reflection term ($k_D = 0$). The importance of recombining reconstructed and reference diffuse shading done in Equation (5.3) becomes apparent when comparing $D(\tilde{\mathbf{n}})$ and $\tilde{D}$. In particular, it permits to seamlessly reproduce $D_0$ outside of the square region (e.g., inside the ear). Similarly, using Equation (5.4) permits to remove implausible bright reflections in reflection shading (e.g., inside the ear or below the eyebrow).

## 5.3 Experimental results

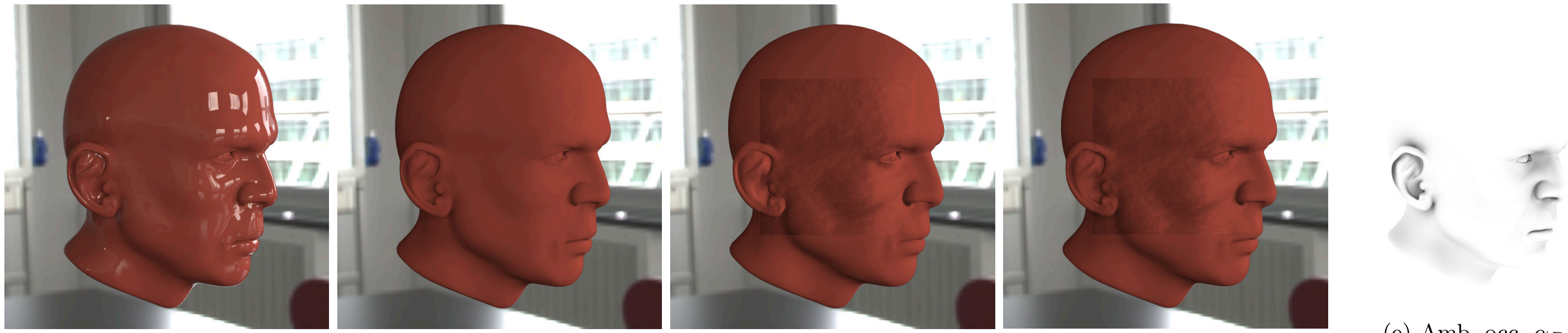
We have implemented the recompositing process of Section 5.2 in Gratin [VB15], a programmable node-based system working on the GPU. It permits to test various normal modification algorithms in 2D by programming them directly in GLSL, while observing results in real-time as demonstrated in the supplemental video. Alternatively, normal variations can be mapped onto 3D objects and rendered as additional auxiliary buffers at a negligible fraction of the total rendering time. It then grants compositing artists the ability to test and combine different variants of local shape details in post-process. We demonstrate both the 2D and 3D normal editing techniques in a set of test scenes rendered in global illumination.

We start with three simple 3D scenes, each showing a different object with a same material in a same environment illumination. The normal buffer and global illumination rendering for each of these scenes are shown in the first two columns of Figure 5.8. Diffuse and reflection environments are reconstructed from these input images and shown in the last two columns, using Lit Spheres. The reconstructed diffuse environments are nearly identical for all three objects. However, the quality of reconstruction for the reflection environment depends on object shape. The sphere object serves as a reference and only differs from the LitSphere view due to perspective projection. With increasing shape complexity, in particular when highly curved object features are available, the reflection environment becomes less sharp. However, this is usually not an issue when we apply the reconstructed environment to the same object, as shown in Figures 5.9 and 5.10.

We evaluate the visual quality of our approach on the head and vase objects in Figure 5.9. The alternative normal buffer is obtained by applying a Voronoi-based bump map on the object in 3D. We use the reconstructed environments of Figure 5.8 and our recompositing pipeline to modify shading buffers in the middle column. The result is visually similar to a re-rendering of the scene using the additional bump map, shown in the right column. A clear benefit of our approach is that it runs in real-time independently of the rendered scene complexity. In contrast, re-rendering takes from several seconds to several minutes depending on the scene complexity.

Figure 5.10 demonstrates three interactive local shape editing tools that act on a normal $\mathbf{n} = (n_x, n_y, n_z)$. Normal embossing is inspired from the LUMO system [Joh02]: it replaces the $n_z$ coordinate with $\beta\, n_z$ where $\beta \in (0, 1]$ and renormalize the result to make the surface appear to “bulge”. Bump mapping perturbs the normal buffer with an arbitrary height map, here a fading 2D ripple pattern (the same manipulation is applied in Figure 5.7 with a 2D Perlin noise). Bilateral smoothing works on projected normals $\bar{\mathbf{n}} = (n_x, n_y)$ using an auxiliary depth buffer to preserve object contours.

In more complex scenes, images of objects may appear in the reflections of each other. This is what occurs in Figure 5.11, which shows a `Table top` scene with various objects: a cup made of porcelain with a metal spoon, a reddish coated kettle with an aluminum handle, and a vase made of a glossy material exhibiting blurry reflections. Despite the increased



(a) Ref. image $I$ (b) Ref. shading $D_0$ (c) Reconst. $D(\tilde{\mathbf{n}})$ (d) Combined $\tilde{D}$ (e) Amb. occ. $\alpha_D$

(f) Mod. image $\tilde{I}$ (g) Ref. shading $R_0$ (h) Reconst. $R(\tilde{\mathbf{n}})$ (i) Combined $\tilde{R}$ (j) Amb. occ. $\alpha_S$

Figure 5.7: Illustration of the different terms involved in recompositing (Equations (5.3)-(5.5)). The left column shows reference and modified images, where we have perturbed the normal buffer with a 2D perlin noise in a square region. This permits to show the behavior of our approach on both modified & unmodified portions of the image (i.e., inside & outside the square); in the latter case, $\tilde{I} = I$. The remaining columns present the diffuse (top) and reflection (bottom) terms. Starting with the reference shading at left, we then show reconstructed shading using prefiltered environments. Reference and reconstructed shading terms slightly differ in some unmodified regions (e.g., inside the ear). The combined shading terms correct these effects with the aid of ambient and reflection occlusion buffers, shown in the rightmost column.

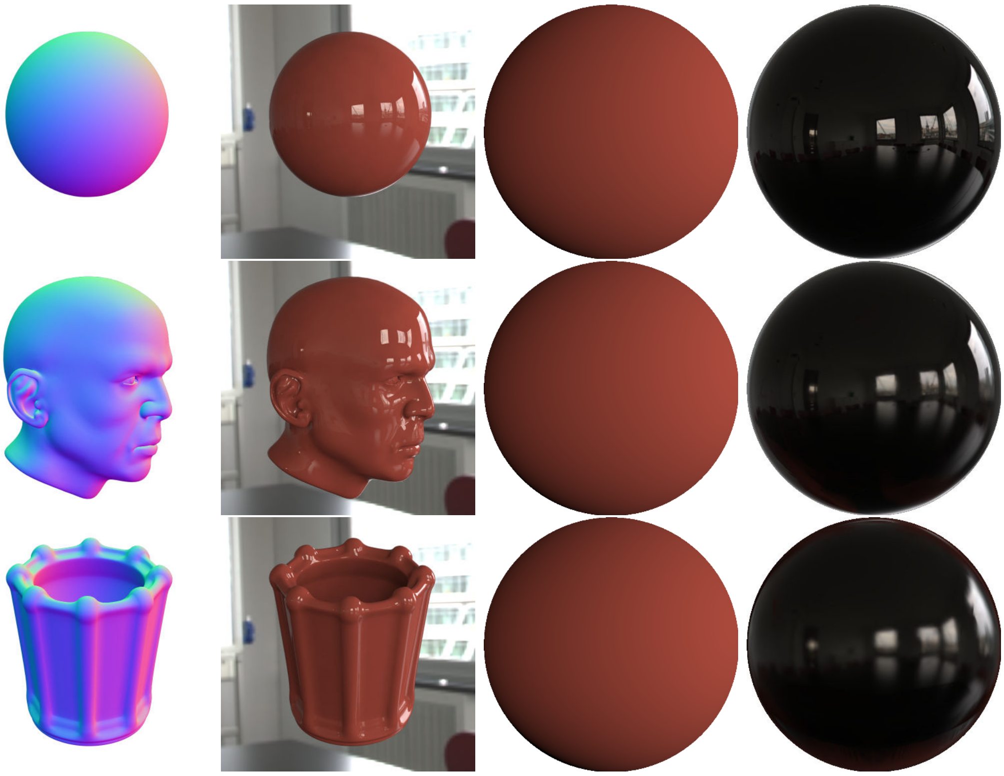


Figure 5.8: *Reconstructed* diffuse and reflection prefiltered environment maps for three objects: the red sphere (top), head (middle) and vase (bottom) models. The diffuse environment reconstructions are nearly identical in all three cases. The sharpness of the reflection environment reconstructions vary with object shape.

complexity, our method still produces a plausible result when normals are modified with a noise texture on the cup, an embossed symbol on the kettle body and a carved pattern on the vase. The results remain plausible even when the material properties are edited, as shown in the right column where we decrease the diffuse intensity and increase the specular intensity. The reconstructed diffuse and reflection environments are shown combined in Figure 5.12, before and after material editing has been performed. Observe in particular how the reflections of nearby objects have been properly reconstructed. The reflected window appears stretched in the cup environment. This is because it maps to the highly curved rim of the cup. However, when reapplied to the same object, stretching goes unnoticed.

The `Truck` scene of Figure 5.1 is more challenging: not only object parts are in contact, but each cover a relatively small subset of surface orientations. Nevertheless, as shown in the reconstructed shading environments, our method manages to capture a convincing appearance that reproduces inter-reflections between different parts. This permits to generate a plausible result when normals are modified to apply a noise texture and an embossed emblem to the truck body, and corrugations to the front and rear mudguards.

### Performance

Reconstruction timings for all edited objects in the paper are given in Table 5.1, using a single CPU core on an i7-4790k@4.00GHz. The reconstruction of the diffuse environment is negligible compared to that of the reflection environment. Our partial reconstruction could be easily optimized with a parallel implementation. The performance of the hole-filling

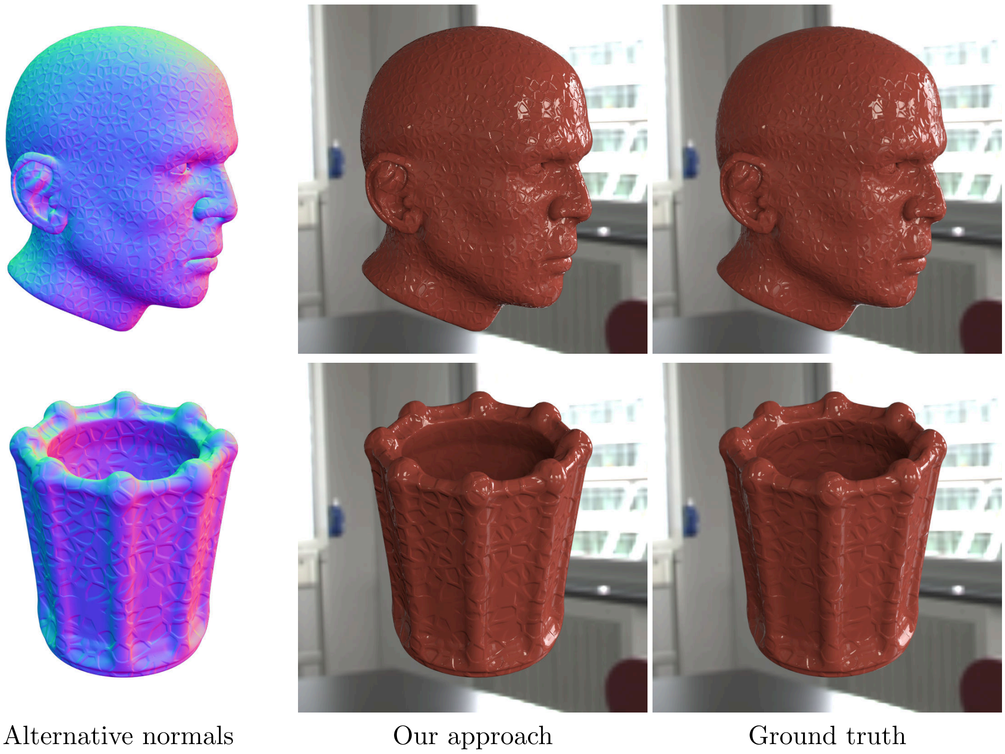

Alternative normals Our approach Ground truth

Figure 5.9: A Voronoi-based bump texture is mapped onto the red head and vase models in 3D, yielding an alternative normal buffer. Our approach is used to modify shading at the compositing stage in real-time, with a resulting appearance similar to re-rendering the object using the bump map (ground truth).

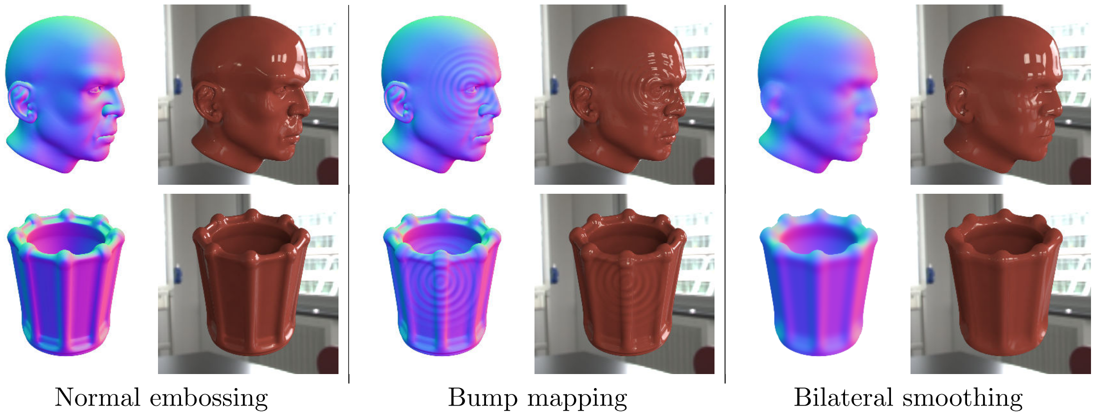

Normal embossing Bump mapping Bilateral smoothing

Figure 5.10: Our technique permits to apply arbitrary modifications to the normal buffer, while still yielding plausible shading results. Normal embossing is applied to the eyebrow of the head, and the whole vase, resulting in an apparent bulging of local shape. Any bump texture may be applied as a decal to modify normals: we use a 2D fading ripple pattern, affecting both diffuse and reflection shading. Local shape details may also be removed: we apply a cross bilateral filter on normals, using an auxiliary depth buffer to preserve occluding contours.

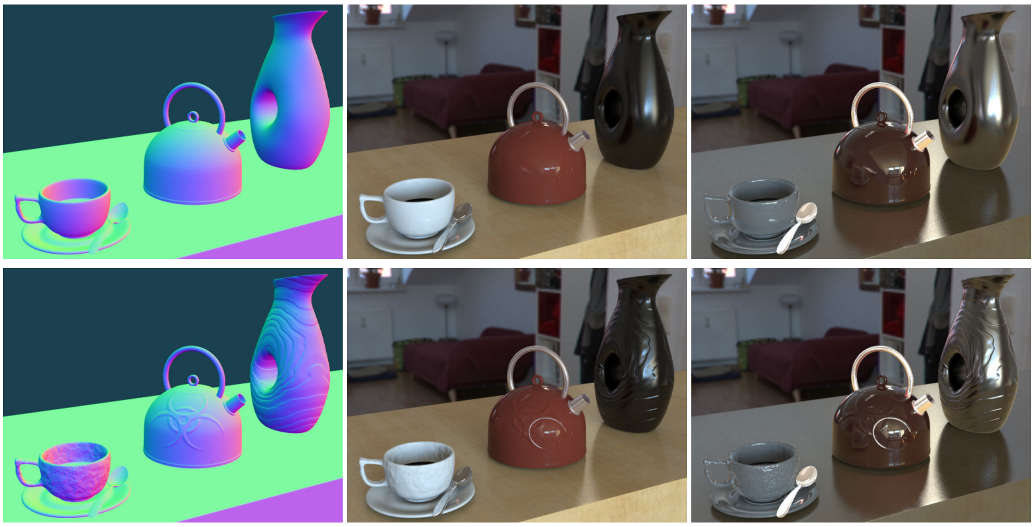


Figure 5.11: The left column shows the original normal buffer of the `Table top` scene (top), as well as a modified version (bottom) where a noise texture, an embossed symbol and a carved pattern have been respectively applied to the cup, kettle body and vase. In the middle column, we show the result of our approach (bottom) on the corresponding color image (top), using the reconstructed diffuse and reflection environments shown in Figure 5.12 (1st row). In the right column, we have edited the intensities of the diffuse and reflection components in both the original and modified scene. Reflections of nearby objects become more clearly apparent, as is also seen in Figure 5.12 (2nd row).

process highly depends on the size of the hole; it could be greatly improved by using an adaptive sphere tesselation strategy. Nevertheless, reconstruction is not too performance-demanding as it is done only once in pre-process. Most importantly, recompositing works in real-time in all our examples.

| Model | Diffuse rec. | Reflection Partial rec. | Reflection Hole filling |
|---|---|---|---|
| Red sphere | 110 | 315 | 215 |
| Red head | 184 | 470 | 280 |
| Red vase | 147 | 385 | 192 |
| Cup | 29 | 45 | 325 |
| Kettle | 65 | 97 | 650 |
| Black vase | 64 | 133 | 4000 |
| Truck body | 119 | 120 | 6000 |
| Front mudguard | 25 | 45 | 2500 |
| Rear mudguard | 14 | 26 | 3700 |

Table 5.1: Timings in ms for each reconstruction stage.

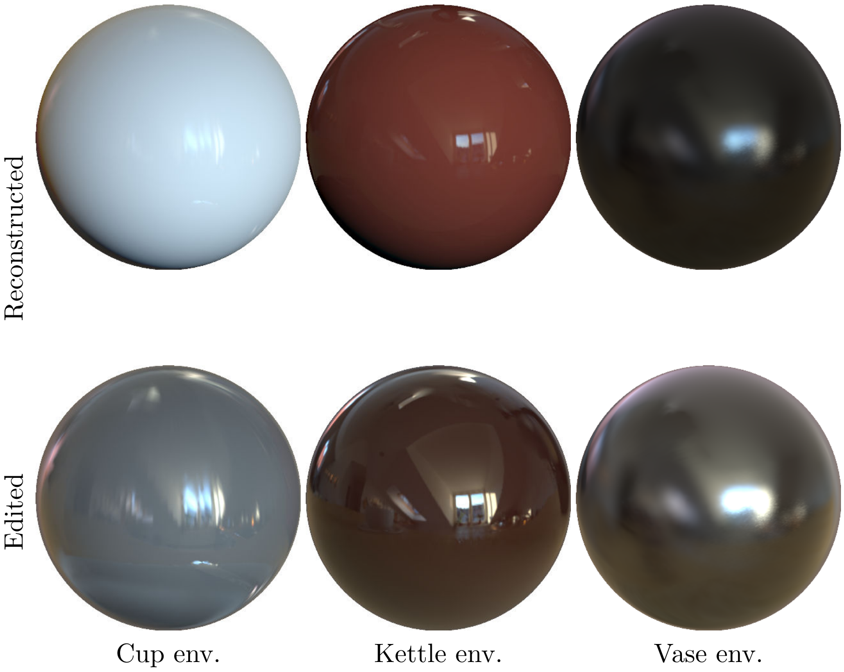


Figure 5.12: Combined diffuse and reflection environments reconstructed from the cup, kettle and vase objects of Figure 5.11. The bottom row shows edited materials where the diffuse term is divided by 4 and the reflection term multiplied by 4: the supporting table and other nearby objects appear more clearly in the reflections.

## 5.4 Discussion and future work

We have demonstrated a first solution for the edition of surface shape at the compositing stage, based on environment reconstruction and real-time re-compositing.

Our technique is limited in terms of the kinds of materials that we can work with and is quite dependent on the geometry of the input. We are restricted to homogeneous isotropic opaque materials. A first easy extension would be to treat spatially varying materials, meanwhile more complex materials would require more involved improvements. Geometry restricts the quality and the viability of our reconstruction. If object shape is too simple, it will not provide enough shading information, which will require to fill-in wider regions. If object shape is too detailed with respect to image resolution, it will tend to reduce the accuracy of the reconstructed shading, as seen in the right column of Figure 5.8 when object complexity is increased. We only modify normals, which mimics geometry modifications without the displacement of vertices that would be needed in an ideal case. Similarly, we are not able to reproduce other effects related to visibility, such as inter-reflections. For further explanation of these limitations and their possible solutions we refer to the Chapter 6.

Besides the described limitations, as demonstrated in Section 5.2, our approach gives satisfying results in practical cases of interest, granting interactive exploration of local shape variations with real-time feedback. We believe it could already be useful to greatly shorten trial and error decisions in product design and movie production.

# Chapter 6

# Conclusions

We have introduced a middle-ground approach for the control of appearance; it works in-between 2D image creation and 3D scene rendering. The use of auxiliary buffers (mostly normal buffers) permits to situate our techniques between painting in 2D and the control of a 3D scene for rendering. Our technique is developed to manipulate appearance for arbitrary-shaded images. We can work with artwork (MatCaps) and rendering (Compositing), and we expect it can be easily extended to photographs. In Chapter 4 I have shown how to modify shading from an input MatCap to mimic modifications on lighting and material without having to retrieve them separately. In Chapter 5 I have shown how to recover single-view pre-filtered environment maps at the compositing stage, and how these pre-filtered environment maps are used to obtain plausible shading when modifying local geometry.

In the following I will discuss the main limitations of our approach, as well as possible solutions. I will conclude by presenting a series of future work directions to extend our approach.

## 6.1 Discussion

In this section I enumerate the basic restrictions of our techniques. Some restrictions are inherited from the use of structures similar to pre-filtered environment maps. They limit the kind of materials that we can represent and restrict lighting to be distant. Nevertheless, we propose possible solutions to work with an extended set of materials (Section 6.1.1), as well as to reproduce effects related to local lighting (Section 6.1.4). Alongside with these limitations, we present other problems related to the separation of shading and material components (Section 6.1.2) and the filling of missing shading information (Section 6.1.3).

### 6.1.1 Non-radially symmetric and anisotropic materials

We store geometry-independent shading values into a spherical representation (dual paraboloid maps) that we use to manipulate appearance. Our representation can be seen as a pre-filtered environment map. Similarly to pre-filtered environment maps we are restricted to work with opaque objects. These kind of structures are not adapted to work with transparent or translucent objects, as they depend on complex light paths inside objects.

Moreover, we have restricted the manipulations of MatCaps to radially symmetric BRDFs. Input MatCaps define shading tied to the camera. In order to enable modifications on lighting from an input MatCap, specifically rotation, we have turned MatCaps into a spherical representation that behaves as a pre-filtered environment lighting of a radially symmetric BRDF. Radially symmetric BRDFs permit to create 2D view-independent pre-filtered environment maps, and therefore enable the rotation of lighting independently of the view. The limitation of BRDFs to be radially symmetric also eases the estimation of material properties from a MatCap. Symmetry is also used to compute the correlation function between mean and variance, as variance in measured materials is computed as the average along the $\theta_i$ and $\phi_i$ dimensions. The radial symmetry is finally used in the definition of filters that mimic rougher materials by blurring using radial/circular filters.

To incorporate arbitrary isotropic materials in our work, we should start with a deeper study of variances in both $\theta_i$ and $\phi_i$ directions. We expect that a better understating of variance will help us to define non-radial filters, and to estimate material properties for $\theta_i$ and $\phi_i$ directions independently. In contrast, if we assume that MatCaps depict arbitrary isotropic BRDFs instead of radially symmetric ones, rotations of lighting will not be straightforward. A solution would be to ask the artist to depict the same MatCap for different view directions, but it will turn the appealing simplicity of MatCaps into a tedious process.

In the Compositing stage technique we reconstruct pre-filtered environment maps for a fixed view-direction. In this case, arbitrary isotropic BRDFs are possible since we are tied to the camera used for rendering. Nevertheless, this leaves out anisotropic materials since anisotropy is linked to tangent fields. Therefore we will need to make use of tangent buffers alongside with normal buffers. Moreover, we would need to have a 3D structure, instead of a 2D spherical representation. This increment on complexity will impact the reconstruction of geometry-independent shading. First, it increments the storage size while the input shading data stays similar, which makes the reconstructions more sparse. Second, partial reconstruction and filling-in techniques are not straightforward to extend to the required 3D structure.

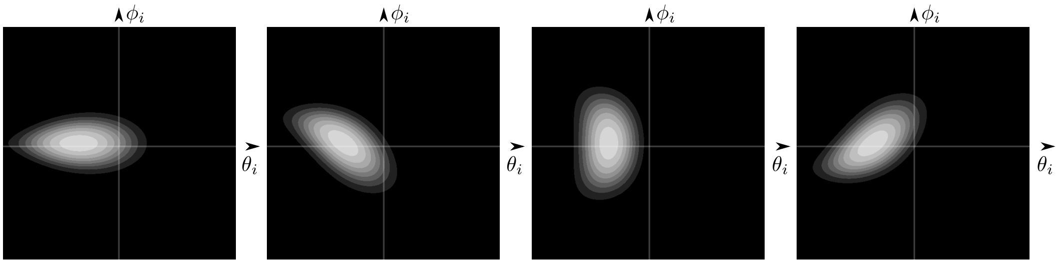
If we want to manipulate anisotropic BRDFs for arbitrary views it will require 5 dimensional structures in the naive case, since all possible views can be parametrized with 2 additional dimensions: the shading at a surface point would then depend on the 2D view direction and the 3D reference frame (normal and tangent). This increases the complexity even further. Working with 5D structure would not be straightforward, as opposed to the simplicity of a spherical representation.

Anisotropic materials are not considered in our statistical approach. Their inclusion would imply to study BRDF slice statistics as variations of viewing direction, instead of simply the viewing elevation angle. Moreover, it would require to use a different database of measured materials because the MERL database only contains isotropic BRDFs. However, at the time of writing existing databases of anisotropic BRDFs, like the UTIA BTF Database [FV14], are not sampled densely enough or do not contain enough materials.

A different approach to increase the dimensionality of our approach would be to use a similar 2D spherical structure with a set of filters that adapt it to local viewing and anisotropy configurations. In the case of non-radially symmetric materials we would apply non-radial filters and for anisotropic materials we would need to introduce local rotations as well. Non-radial symmetry can be see in the left of Figure 6.1 where the spread of the BRDF differs between dimensions. Anisotropic materials (Figure 6.1, three remaining images) have the effect of rotating the filter kernel in our parametrization. This approach seems promising, as our goal is to mimic modifications from a basic shading, while producing plausible results.

### 6.1.2 Shading components

We treat both shading and BRDF as the addition of diffuse and specular components. At the compositing stage we obtain a perfect separation of diffuse and specular components thanks to the capabilities of the rendering engines. In contrast, we need to separate those components for measured materials as well as in MatCaps. Because of our simple heuristic on measured BRDF decomposition, we were forced to consider a subset of the MERL



$\phi_o = [0^o, 190^o, 360^o]$ $\phi_o = [45^o, 225^o]$ $\phi_o = [90^o, 270^o]$ $\phi_o = [135^o, 315^o]$

Figure 6.1: BRDF slices on our parametrization for the anisotropic Ward BRDF. Slices vary in azimuthal angle, while remaining at the same elevation angle $\theta$ of $30^o$. From left to right images correspond to azimuthal angles of [$0^o$,$180^o$,$180^o$], [$45^o$,$225^o$], [$90^o$,$270^o$] and [$135^o$,$315^o$]. To reproduce the effect of anisotropy, filters will have to be rotated.

database. It would thus be interesting to devise clever decomposition schemes so that each component could be studied using a separate moment analysis. We have shown that our MatCap separation into low and high frequency components is a good approximation of diffuse and specular components. Despite that, we attribute all low-frequency content to the diffuse component, which is not the case in real world materials.For instance, hazy gloss (see Section 6.2.1) is a low-frequency effect that belongs to the specular component. Moreover, material reflectance is composed of more than just diffuse and specular components. It would be interesting to treat separately those different components. We can relate them to different effects like grazing angle effects (e.g. asperity scattering), retro reflection or off-specular reflection. We would need to be able to separate them in our statistical analysis to understand better their effects. In the case of manipulation of shading from rendering engine outputs, we could again take advantage of their capability to render them separately.

As we have already discussed, techniques related to pre-filtered environment maps are not ready to work with translucency or transparency materials. Nevertheless we are interested in trying to recover translucency shading. Depending on the material, it can look similar to a diffuse material. Moreover, it has been already shown by Khan et al. [KRFB06] that the human visual system would accept inaccurate variations of translucent materials as plausible.

### 6.1.3 Filling-in of missing shading

The construction of our geometry-independent shading structure, for both the MatCap and the compositing approach, requires the filling of some parts. The missing parts in MatCaps depend on the estimated roughness of the depicted material, which describes a circle in the back paraboloid map, that we called the 'blind spot'. When retrieving shading information from renderings at the compositing stage we are restricted by input geometry. The corresponding missing parts may be bigger than the ones defined by the 'blind spot' and their shape are arbitrary, which makes the filling-in more complex. This is illustrated in Figure 6.2, which shows the reconstructed reflection environment for the spoon in the `Table top` scene of Figure 5.11 shown in Chapter 5.

One way to improve the filling-in would be to take into account structured shading, as shown in Figure 6.3. It is important to note that in the case of MatCaps this is not a blocking issue since users may correct in-painting results by hand, which is consistent with the technique as it is an artistic approach.

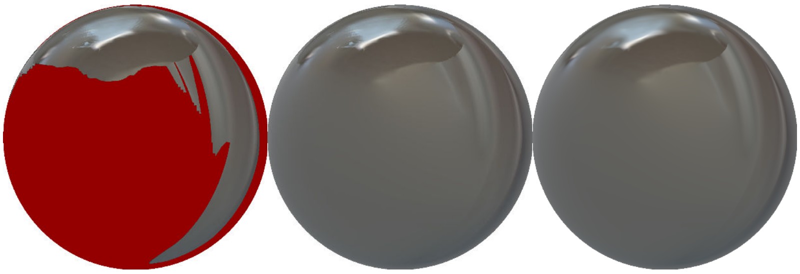
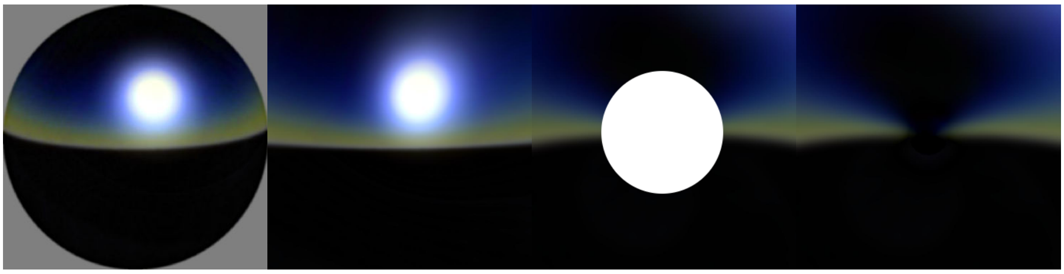


Figure 6.2: The reconstructed reflection environment for the spoon object in Figure 5.11 does not contain enough shading information to grant editing, even after hole-filling and regularization.



(a) Original MatCap (b) Front parabol. map (c) Back parabol. map (d) Blind-Spot filling

Figure 6.3: Filling-in techniques could be improved by approaches that preserve structured shading. The original MatCap (a) is turned from the image of a sphere into a spherical representation. This representation is stored in a dual paraboloid map (b, c). The back paraboloid map (c) cannot be filled entirely, and an empty circle (blind-spot) need to be in-painted. Our current technique diffuses colors (d) independently of the structure in shading. The structure of horizontal line should be taken into account in order to prevent it from fading out.

### 6.1.4 Visibility and inter-reflections

Our approach does not offer any solution to control or mimic local light transport.We do not take into account visibility effects like shadows or inter-reflections. Working with pre-filtered environment maps requires to assume distant illumination. As a solution, when working with MatCaps we plan to use different MatCaps to shade illuminated and shadowed parts of an object. Similarly, for the case of compositing we plan to create separated PEM for shadowed and un-shadowed parts thanks to the information obtained in auxiliary buffers.

Geometry modifications would required to displace vertex in addition to modifying normals. A plausible solution would be to apply displacement of vertices to update occlusion buffers. Another limitation is the control of inter-reflections when recovering shading at the compositing stage. It could be interesting to recover shading for those specific zones. Available information is sparse and would not be enough to recover a good quality PEM. A solution would be to get more involved into the rendering process by recognizing these parts and output more shading samples that characterize the inter-reflections zones.

## 6.2 Future work

As long-term goals we would like to extend our analysis of BRDFs and their impact on shading (Section 6.2.1). A second major future goal is to be able to manipulate more complex 3D scenes. The most important challenge would be to deal with the spatial shading variations due to both material and lighting, not just angular variations (Section 6.2.2). Finally we explain how our technique could be useful to other applications in graphics or even perception (Section 6.2.3).

### 6.2.1 Extended statistical analysis

We have performed our analysis by considering simple shapes (spheres). Therefore we have studied the implication of material and lighting into shading without taking into account geometry. When considering more complex shapes, our observations may still be similar by considering a surface patch on the object. However, surface curvatures will impose restrictions on the window sizes we use for establishing relationships between material/lighting and shading on this patch. In particular, high curvatures will lead to rapid changes of the view direction in surface tangent space. In such situations, our local approximation will be valid only in small 1D windows. This suggests that the effect of a BRDF will tend to be less noticeable on bumpy surfaces, which is consistent with existing perceptual experiments [VLD07].

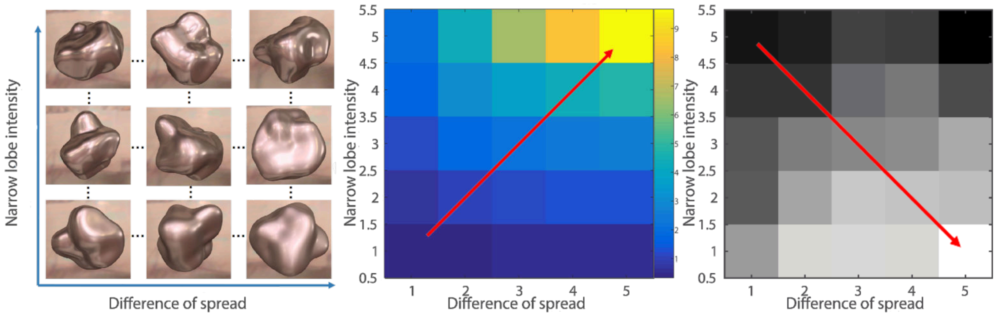
We have considered orthographic or orthographic-corrected projections along the whole thesis. To get complete relationships between shading and their components, we should consider the effect of perspective projection on reflected radiance. This will of course depend on the type of virtual sensor used. We may anticipate that foreshortening will tend to compress radiance patterns at grazing angles. This suggests that some grazing angle effects will get ‘squeezed’ in a thin image region around the silhouette.

We have focused on moments up to order 2, but as we shown in Appendix A it can be extended to higher-order moments to study skewness and kurtosis. Skewness quantifies the asymmetry of a distribution, meanwhile kurtosis measures its peakedness. We have shown how energy, mean and variance of a BRDF slice are perceived in the image, as coloring, warping and blurring. One question is whether similar perceptible effects could be related to skewness and kurtosis. To study these effects would require to introduce both skewness and kurtosis into the statistical model and consequently into the Fourier Analysis. This would increase the complexity of the statistical analysis; hence we have performed a perceptual study to first identify whether they have a perceptible effect [VBZPF16].

We have focused on kurtosis with the idea to identify it as a cue to hazy gloss. We performed a series of experiment with a BRDF model made of a pair of Gaussian lobes.The difference between the lobe intensities and their spread produces different kurtosis and different haze effects, as shown in Figure 6.4a. Using these stimuli we have studied how human subjects perceive haziness of gloss. Our conclusion is that perceived haziness does *not* vary according to kurtosis, as shown in Figure 6.4b and Figure 6.4c. We suggest that haziness depends on the sepparation of the specular component into two sub-components, which are not directly the two Gaussians used to define the BRDF. Instead, haziness effects would be characterized by a central peak plus a wide component characteristic of the halo effect of haziness. If this hypothesis is correct, then maybe other sub-decompositions can be performed for other BRDF components.

### 6.2.2 Spatially-varying shading

In this thesis we only have considered shading as variations in the angular domain. This approximation has lead us to satisfactory results in simple scenes. However, for a good



(a) Estimuli (b) BRDF Kurtosis (c) Perceived haziness

Figure 6.4: In our perceptual experiments we use a two lobe Gaussian model BRDF controlled separately by its intensity and spread to produce a hazy gloss appearance. The sum of of intensities and spread of the wider lobe are keep constant. (a) A set of stimuli is presented by increasing the intensity of the narrow lobe from bottom to top and by increasing the difference in spread from left to right. (b) We measured kurtosis for the BRDF of our stimuli, and we can see how variations differs from (c) how much subjects rated ‘haziness’ for each material.

representation and manipulation of shading in complex scenes we should consider spatial variations as well. As we have shown, variations of shading depend of variations of material and of lighting.

In the case of variations of materials our compositing approach would be easily extended to objects with spatially-varying reflectance, provided that diffuse (resp. specular) reflectance buffers are output *separately* by the renderer. Our method would then be used to reconstruct multiple diffuse (resp. specular) shading buffers, and the final shading would be obtained through multiplication by reflectance at the re-compositing stage.

When considering variations of lighting, we may separate shading depending on the origin of the incoming radiance. We can distinguish and manipulate differently the shading due to local light sources or the reflection of close objects. This would help to deal with a problem that arises with extended objects: their incoming lighting may vary spatially and come from completely different sources.

Ideally we should store shading in a 4D representation for the variations in spatial and angular domains, which can be seen as a light field. This is equivalent to reconstructing a pre-filtered environment map per pixel instead of per surface. To deal with this issue, we would like to explore the reconstruction of multiple pre-filtered environment maps for extended surfaces and recombining them through warping depending on pixel locations.

We have considered out-of-the-box render buffers and have focused on algorithms working at the compositing stage. We would also like to work at the rendering stage to export more detailed shading information while retaining a negligible impact on rendering performance. For example we would like to export information about light paths, in a sense that we could have more information about where the incoming radiance came from. Another useful solution would be to have a fast pre-analysis of render output to know which parts will not provide enough information to recover shading and output more information for these parts.

Rendering engines are usually made to generate a set of images that will form an animation. We plan to extend our technique to animations, which will require a temporally consistent behavior.

### 6.2.3 New applications

We believe our approach could prove useful in a number of applications in Computer Graphics and Visual Perception.

#### Dynamic creation of MatCaps

We have shown how to use existing MatCaps and later on apply our technique to enable modification of lighting and material. Instead we could consider the creation of MatCaps directly on a spherical representation (i.e. dual parabolid maps) using our tools. The rotation of the MatCap would avoid problems in the blind-spot by permitting to fill it during the creation process. When using paint brushes or light sources to create shading, the material roughness could be taken into account and blur the created shading accordingly.

#### Material estimation on photographs

We estimate a few material properties from MatCaps making assumptions on lighting. We believe that this technique could be extended to other kind of inputs, like rendering or, more interestingly, photographs. This will require some knowledge over lighting moments, either explicit or hypothesized. This technique should be complemented with a geometry estimation from images. In any case, for a correct behavior it will require the study of the impact of geometry in shading.

#### Editing of measured materials

Moments have proved to be a good method to analyze the BRDF effect on shading. We believe that they could be used as a way to edit measured BRDFs, by finding operators in BRDF space that preserve some moments while modifying others. Ideally, users could be granted control over a measured BRDF directly through its moment profiles. A sufficient accuracy of these edits would require a better decomposition of BRDFs.

#### Perceptual studies

Finally, we believe that BRDF moments may also be well adapted to the study of material perception. The end goal of such an approach is to explicitly connect perceptually-relevant gloss properties to measurable BRDF properties. Experiments should be conducted to evaluate the BRDF moments humans are most sensitive to, and check whether the statistical analysis can predict perceived material appearance.

# Appendix A

# Skewness and Kurtosis Analysis of Measured BRDFs

We have shown how to compute 2D moments of arbitrary order. By now we have used moments up to order 2 to define the statistical properties of energy, mean and variance. Here we extend our analysis on measured materials to moments of order 3 and 4, which allow us to define skewness and kurtosis. They seem to be important properties of material appearance, related to asymmetry and peakedness of a BRDF slice respectively.

Co-skewness and co-kurtosis are defined as the *standardized* moment tensors of order 3 and 4 respectively. Standardized moments are computed by both centering $\bar{f}_r$ on its mean and scaling it by respective variances. Since in our case $\Sigma_{1,1} \approx 0$, we may write standardized moments using $\gamma_{n,m}[\bar{f}_r] = E_{\bar{\rho}}[(\frac{\theta_i - \mu_\theta}{\sigma_\theta})^n (\frac{\phi_i}{\sigma_\phi})^m]$, where $\sigma_\theta = \sqrt{\Sigma_{2,0}}$ and $\sigma_\phi = \sqrt{\Sigma_{0,2}}$. The coefficients of the co-skewness and co-kurtosis tensors are then given by $\gamma_{n,m}[\bar{f}_r]$ for $n+m = 3$ and $n + m = 4$ respectively. It is common to use the modern definition of kurtosis, also called excess kurtosis, which is is equal to 0 for a Normal distribution. In our case (i.e., with $\mu_{0,1} = 0$ and $\Sigma_{1,1} = 0$), it can be shown that excess kurtosis coefficients are given by $\gamma_{4,0} - 3$, $\gamma_{3,1}$, $\gamma_{2,2} - 1$, $\gamma_{1,3}$ and $\gamma_{0,4} - 3$. For simplicity, we will make an abuse of notation and refer to excess kurtosis coefficients as $\gamma_{n,m}[\bar{f}_r]$ for $n + m = 4$.

The co-skewness tensor characterizes asymmetries of the BRDF slices in different dimensions. The profile for two of its coefficients, $\gamma_{3,0}$ and $\gamma_{1,2}$, are shown in Fig. A.1d: our selected BRDFs exhibit increasing skewness toward grazing angles, which are more pronounced for $\gamma_{3,0}$ than $\gamma_{1,2}$. The other two profiles are close to zero and shown in Fig. A.2c. The co-kurtosis tensor characterizes the peakedness of the BRDF slice in different dimensions. The profile for three of its coefficients, $\gamma_{4,0}$, $\gamma_{2,2}$ and $\gamma_{0,4}$, are shown in Fig. A.1e: our selected BRDFs exhibit roughly constant profiles, with deviations occurring at grazing angles, which are most pronounced in $\gamma_{4,0}$. The other two profiles are close to zero and shown in Fig. A.2d.

### Conclusions

Plots of co-skewness and co-kurtossis follow the same insights described in Section 3.3.4. First, moments where $m$ is odd are close to null, similar to $\mu_{0,1}$ and $\Sigma_{1,1}$. This enforces the symmetry about the scattering plane. The second symmetry at incident view is as well confirmed by the fact that co-skewness starts at 0, meanwhile co-kurtosis starts at the same value, similarly to mean and variance respectively. In general all deviations from a single profile occur toward grazing angles. This effect appears stronger for moments related to $\theta$ than for those along $\phi$. We thus conjecture that such grazing-angle deviations are due in part to the clamping of directions by hemispherical boundaries. Indeed, such a clamping will have more influence at grazing angles in directions parallel to $\theta_i$ (see Fig. 3.3).

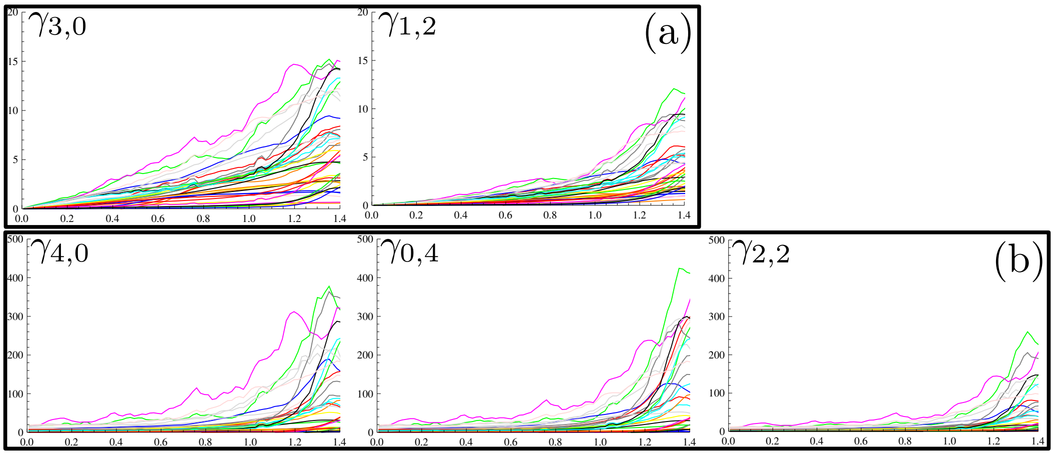


Figure A.1: Skewness (a) and Kurtosis (b) profiles computed from our selected BRDFs

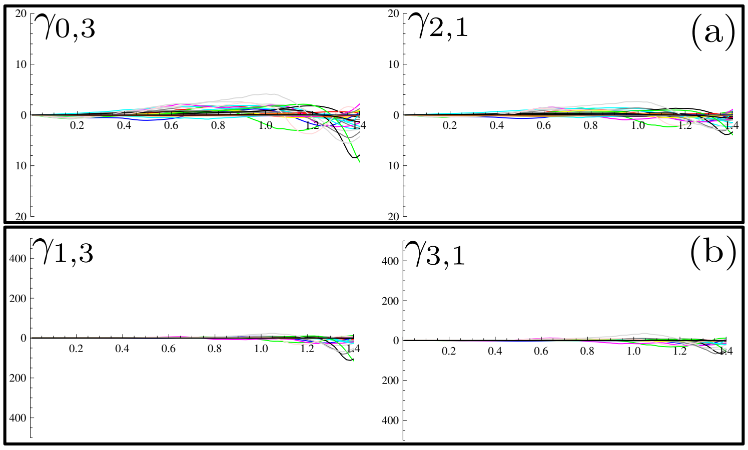


Figure A.2: Near-zero moment profiles occur for Skewness (a) and Kurtosis (b) are likely due to BRDF slice symmetry.

### Discussion

It would be interested to study potential correlations between moments of different orders, as we have done for the mean and the variance. We have already observed interesting deviations from simple behaviors at grazing angles in skewness and kurtosis profiles. They may be related to known properties such as off-specular peaks, but could as well be due to hemispherical clamping once again. Moreover, we would like to extend our local Fourier analysis to include co-skewness and co-kurtosis tensors, in order to characterize their effects in the image.

# Bibliography

[AWL13] Miika Aittala, Tim Weyrich, and Jaakko Lehtinen. Practical svbrdf capture in the frequency domain. *ACM Trans. Graph.*, 32(4):110–1, 2013.

[BG07] Stefan Bruckner and M Eduard Gröller. Style transfer functions for illustrative volume rendering. In *Computer Graphics Forum*, volume 26, pages 715–724. Wiley Online Library, 2007.

[Bir13] Jeremy Birn. *Digital lighting and rendering*. Pearson Education, 2013.

[Bli77] James F Blinn. Models of light reflection for computer synthesized pictures. In *ACM SIGGRAPH Computer Graphics*, volume 11, pages 192–198. ACM, 1977.

[Bli78] James F Blinn. Simulation of wrinkled surfaces. In *ACM SIGGRAPH computer graphics*, volume 12, pages 286–292. ACM, 1978.

[Bro03] P Bromiley. Products and convolutions of gaussian distributions. *Medical School, Univ. Manchester, Manchester, UK, Tech. Rep*, 3:2003, 2003.

[BS87] Petr Beckmann and Andre Spizzichino. The scattering of electromagnetic waves from rough surfaces. *Norwood, MA, Artech House, Inc., 1987, 511 p.*, 1987.

[BSS$^+$13] Laurent Belcour, Cyril Soler, Kartic Subr, Nicolas Holzschuch, and Fredo Durand. 5d covariance tracing for efficient defocus and motion blur. *ACM Transactions on Graphics (TOG)*, 32(3):31, 2013.

[Bul65] Michael George Bulmer. Principles of statistics. 1965.

[Bur12] Brent Burley. BRDF Explorer. http://www.disneyanimation.com/technology/brdf.html, 2012.

[CK15] AC Chadwick and RW Kentridge. The perception of gloss: a review. *Vision research*, 109:221–235, 2015.

[CON99] Brian Cabral, Marc Olano, and Philip Nemec. Reflection space image based rendering. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 165–170. ACM Press/Addison-Wesley Publishing Co., 1999.

[DBM10] Katja Doerschner, Huseyin Boyaci, and Laurence T Maloney. Estimating the glossiness transfer function induced by illumination change and testing its transitivity. *Journal of Vision*, 10(4):8–8, 2010.

[DHS$^+$05] Frédo Durand, Nicolas Holzschuch, Cyril Soler, Eric Chan, and François X Sillion. A frequency analysis of light transport. *ACM Transactions on Graphics (TOG)*, 24(3):1115–1126, 2005.

[FDA03] Roland W Fleming, Ron O Dror, and Edward H Adelson. Real-world illumination and the perception of surface reflectance properties. *Journal of vision*, 3(5):3–3, 2003.

[FV14] J. Filip and R. Vávra. Template-based sampling of anisotropic brdfs. *Computer Graphics Forum (Proceedings of Pacific Graphics 2014)*, 2014.

[GCP$^+$09] Abhijeet Ghosh, Tongbo Chen, Pieter Peers, Cyrus A Wilson, and Paul Debevec. Estimating specular roughness and anisotropy from second order spherical gradient illumination. In *Computer Graphics Forum*, volume 28, pages 1161–1170. Wiley Online Library, 2009.

[GI83] Donald Goldfarb and Ashok Idnani. A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical programming*, 27(1):1–33, 1983.

[HDH06] Haibo Hu and G De Hann. Low cost robust blur estimator. In *Image Processing, 2006 IEEE International Conference on*, pages 617–620. IEEE, 2006.

[HS98] Wolfgang Heidrich and Hans-Peter Seidel. View-independent environment maps. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware*, pages 39–ff. ACM, 1998.

[HS14] Vlastimil Havran and Mateu Sbert. Statistical characterization of surface reflectance. In *Proceedings of the Eurographics 2014 Workshop on Material Appearance Modeling: Issues and Acquisition*, pages 15–18. Eurographics Association, 2014.

[Hun75] Richard Sewall Hunter. *The measurement of appearance*. John wiley, 1975.

[IGG$^+$14] Ivo Ihrke, Xavier Granier, Gaël Guennebaud, Laurent Jacques, and Bastian Goldluecke. An introduction to optimization techniques in computer graphics. In *Eurographics 2014-Tutorials*, 2014.

[Jar03] Rafal Jaroszkiewicz. Fast extraction of brdfs and material maps from images. 2003.

[Joh02] Scott F Johnston. Lumo: illumination for cel animation. In *Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, pages 45–ff. ACM, 2002.

[Kaj86] James T Kajiya. The rendering equation. In *ACM Siggraph Computer Graphics*, volume 20, pages 143–150. ACM, 1986.

[KP03] Jan Koenderink and Sylvia Pont. The secret of velvety skin. *Machine vision and applications*, 14(4):260–268, 2003.

[KRFB06] Erum Arif Khan, Erik Reinhard, Roland W Fleming, and Heinrich H Bülthoff. Image-based material editing. *ACM Transactions on Graphics (TOG)*, 25(3):654–663, 2006.

[KVHS00] Jan Kautz, Pere-Pau Vázquez, Wolfgang Heidrich, and Hans-Peter Seidel. *A unified approach to prefiltered environment maps*. Springer, 2000.

[LBS06] Torsten Langer, Alexander Belyaev, and Hans-Peter Seidel. Spherical barycentric coordinates. In *Symposium on Geometry Processing*, pages 81–88, 2006.

[LFTG97] Eric PF Lafortune, Sing-Choong Foo, Kenneth E Torrance, and Donald P Greenberg. Non-linear approximation of reflectance functions. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 117–126. ACM Press/Addison-Wesley Publishing Co., 1997.

[LK11] Weisi Lin and C-C Jay Kuo. Perceptual visual quality metrics: A survey. *Journal of Visual Communication and Image Representation*, 22(4):297–312, 2011.

[LN12] Stephen Lombardi and Ko Nishino. Reflectance and natural illumination from a single image. In *Computer Vision–ECCV 2012*, pages 582–595. Springer, 2012.

[LRR$^+$14] Gerrit Lochmann, Bernhard Reinert, Tobias Ritschel, Stefan Müller, and Hans-Peter Seidel. Real-time reflective and refractive novel-view synthesis. In *VMV*, pages 9–16, 2014.

[Mat03] Wojciech Matusik. *A data-driven reflectance model*. PhD thesis, Citeseer, 2003.

[NDM05] Addy Ngan, Frédo Durand, and Wojciech Matusik. Experimental analysis of brdf models. *Rendering Techniques*, 2005(16th):2, 2005.

[Nic65] Fred E Nicodemus. Directional reflectance and emissivity of an opaque surface. *Applied optics*, 4(7):767–775, 1965.

[NL11] Ko Nishino and Stephen Lombardi. Directional statistics-based reflectance model for isotropic bidirectional reflectance distribution functions. *JOSA A*, 28(1):8–18, 2011.

[NVY$^+$14] Tam Nguyen, Quang Nhat Vo, Hyung-Jeong Yang, Soo-Hyung Kim, and Guee-Sang Lee. Separation of specular and diffuse components using tensor voting in color images. *Applied optics*, 53(33):7924–7936, 2014.

[ON94] Michael Oren and Shree K Nayar. Generalization of lambert's reflectance model. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 239–246. ACM, 1994.

[Pel10] Fabio Pellacini. envylight: an interface for editing natural illumination. *ACM Transactions on Graphics (TOG)*, 29(4):34, 2010.

[PFG00] Fabio Pellacini, James A Ferwerda, and Donald P Greenberg. Toward a psychophysically-based light reflection model for image synthesis. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 55–64. ACM Press/Addison-Wesley Publishing Co., 2000.

[PFL09] Jonathan B Phillips, James A Ferwerda, and Stefan Luka. Effects of image dynamic range on apparent surface gloss. In *Color and Imaging Conference*, volume 2009, pages 193–197. Society for Imaging Science and Technology, 2009.

[PGB03] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 313–318. ACM, 2003.

[Pho75] Bui Tuong Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, 1975.

[RFWB07] Ganesh Ramanarayanan, James Ferwerda, Bruce Walter, and Kavita Bala. Visual equivalence: towards a new standard for image fidelity. *ACM Transactions on Graphics (TOG)*, 26(3):76, 2007.

[RH01a] Ravi Ramamoorthi and Pat Hanrahan. An efficient representation for irradiance environment maps. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 497–500. ACM, 2001.

[RH01b] Ravi Ramamoorthi and Pat Hanrahan. A signal-processing framework for inverse rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 117–128. ACM, 2001.

[RH02] Ravi Ramamoorthi and Pat Hanrahan. Frequency space environment map rendering. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 517–526. ACM, 2002.

[RH04] Ravi Ramamoorthi and Pat Hanrahan. A signal-processing framework for reflection. *ACM Transactions on Graphics (TOG)*, 23(4):1004–1042, 2004.

[RMB07] Ravi Ramamoorthi, Dhruv Mahajan, and Peter Belhumeur. A first-order analysis of lighting, shading, and shadows. *ACM Transactions on Graphics (TOG)*, 26(1):2, 2007.

[RMZ13] Fabrice Rousselle, Marco Manzi, and Matthias Zwicker. Robust denoising using feature and color information. In *Computer Graphics Forum*, volume 32, pages 121–130. Wiley Online Library, 2013.

[ROTS09] Tobias Ritschel, Makoto Okabe, Thorsten Thormählen, and Hans-Peter Seidel. Interactive reflection editing. In *ACM Transactions on Graphics (TOG)*, volume 28, page 129. ACM, 2009.

[Rus98] Szymon M Rusinkiewicz. A new change of variables for efficient brdf representation. In *Rendering techniques 98*, pages 11–22. Springer, 1998.

[RVZ08] Fabiano Romeiro, Yuriy Vasilyev, and Todd Zickler. Passive reflectometry. In *Computer Vision–ECCV 2008*, pages 859–872. Springer, 2008.

[RZ10] Fabiano Romeiro and Todd Zickler. Blind reflectometry. In *Computer Vision–ECCV 2010*, pages 45–58. Springer, 2010.

[Sch94] Christophe Schlick. An inexpensive brdf model for physically-based rendering. In *Computer graphics forum*, volume 13, pages 233–246. Wiley Online Library, 1994.

[SMGG01] Peter-Pike J Sloan, William Martin, Amy Gooch, and Bruce Gooch. The lit sphere: A model for capturing npr shading from art. In *Graphics interface*, volume 2001, pages 143–150. Citeseer, 2001.

[Ste86] Stanley R Sternberg. Grayscale morphology. *Computer vision, graphics, and image processing*, 35(3):333–355, 1986.

[TAY13] Hideki Todo, Ken Anjyo, and Shunichi Yokoyama. Lit-sphere extension for artistic rendering. *The Visual Computer*, 29(6-8):473–480, 2013.

[Tok15] Y Tokuyoshi. Specular lobe-aware filtering and upsampling for interactive indirect illumination. In *Computer Graphics Forum*, volume 34, pages 135–147. Wiley Online Library, 2015.

[TS67] Kenneth E Torrance and Ephraim M Sparrow. Theory for off-specular reflection from roughened surfaces. *JOSA*, 57(9):1105–1112, 1967.

[VB15] Romain Vergne and Pascal Barla. Designing gratin, a gpu-tailored node-based system. *Journal of Computer Graphics Techniques*, 4(4):17, 2015.

[VBFG12] Romain Vergne, Pascal Barla, Roland Fleming, and Xavier Granier. Surface flows for image-based shading design. *ACM Transactions on Graphics*, 31(3):94–1, 2012.

[VBZPF16] Peter Vangorp, Pascal Barla, Carlos Jorge Zubiaga Peña, and Roland Fleming. Specular kurtosis and the perception of hazy gloss. In *Vision Science Society* , St Petersburg, United States, May 2016.

[VLD07] Peter Vangorp, Jurgen Laurijssen, and Philip Dutré. The influence of shape on the perception of material reflectance. *ACM Transactions on Graphics (TOG)*, 26(3):77, 2007.

[WAKB09] Josh Wills, Sameer Agarwal, David Kriegman, and Serge Belongie. Toward a perceptual space for gloss. *ACM Transactions on Graphics (TOG)*, 28(4):103, 2009.

[War92] Gregory J Ward. Measuring and modeling anisotropic reflection. *ACM SIGGRAPH Computer Graphics*, 26(2):265–272, 1992.

[WMLT07] Bruce Walter, Stephen R Marschner, Hongsong Li, and Kenneth E Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pages 195–206. Eurographics Association, 2007.

[ZBB+15] Carlos J Zubiaga, Laurent Belcour, Carles Bosch, Adolfo Munoz, and Pascal Barla. Statistical analysis of bidirectional reflectance distribution functions. In *IS&T/SPIE Electronic Imaging*, pages 939808–939808. International Society for Optics and Photonics, 2015.

[ZMB+15] Carlos J Zubiaga, Adolfo Munoz, Laurent Belcour, Carles Bosch, and Pascal Barla. Matcap decomposition for dynamic appearance manipulation. In *Eurographics Symposium on Rendering 2015*, 2015.

[ZPGVB16] Carlos Jorge Zubiaga Peña, Gael Guennebaud, Romain Vergne, and Pascal Barla. Local Shape Editing at the Compositing Stage. In *EGSR*, Dublin, Ireland, June 2016.

[ZRJ+15] Henning Zimmer, Fabrice Rousselle, Wenzel Jakob, Oliver Wang, David Adler, Wojciech Jarosz, Olga Sorkine-Hornung, and Alexander Sorkine-Hornung. Path-space motion estimation and decomposition for robust animation filtering. In *Computer Graphics Forum*, volume 34, pages 131–142. Wiley Online Library, 2015.