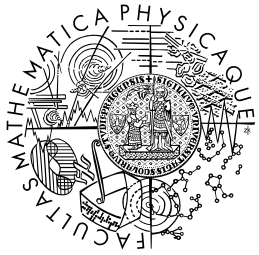


FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

DOCTORAL THESIS

Asen Atanasov

Efficient and Expressive Microfacet Models

Department of Software and Computer Science Education

Supervisor of the doctoral thesis:	doc. Dr. Alexander Wilkie
Study programme:	Informatics
Study branch:	Computer Graphics and Image Analysis

Prague 2023

I declare that I carried out this doctoral thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date ....

Author's signature

I could not have started or completed this endeavor without the help and support of these three individuals to whom I am deeply grateful.

To Vladimir Koylazov, my manager at Chaos Software Bulgaria, who adopted my idea to follow doctoral studies at Charles University while continuing to be a full-time researcher at Chaos. I thank him for believing in me and being patient with me throughout the many years we worked together. He has always been an inspiring example and the imprint he left on my career can hardly be exaggerated. He also contributed to all our publications listed in this thesis.

To my former advisor Jaroslav Křivánek who agreed to distantly supervise my doctoral studies even though both of us were working for competitors at the time. He guided me through the initial phase of my studies when I had the least idea of what to do. I was granted the privilege to learn from him, but I also regret that I did not have the chance to learn more. We started to write the first article together, but he never saw its publication. This first article is dedicated to him.

To my advisor Alexander Wilkie who extended his hand and offered to supervise my studies at the time when I had no idea if and how I would continue with this work. I thank him for being very positive and encouraging, and for identifying the potential in some ideas. He sparked my interest in color science and spectral rendering and it was always a pleasure to learn from his expertise in the field.

Also, I appreciate the effort that my fellow students Martin Mirbauer and Thomas Nindel put into reviewing the manuscript of this work and the suggestions they made.

I am glad to close this chapter of my life, which was rather difficult on multiple fronts. I cannot imagine it without the love, care and strength that I received from my wonderful wife Ivana. I am forever thankful to my loving parents Eliza and Petar.

Above all, I praise my Lord Jesus Christ without whom my life would be a scene with no light.

to Ivana

Title: Efficient and Expressive Microfacet Models

Author: Asen Atanasov

Department: Department of Software and Computer Science Education

Supervisor: doc. Dr. Alexander Wilkie, Department of Software and Computer Science Education

Abstract: In realistic appearance modeling, rough surfaces that have microscopic details are described using so-called *microfacet models*. These include analytical models that statistically define a physically-based microsurface. Such models are extensively used in practice because they are inexpensive to compute and offer considerable flexibility in terms of appearance control. Also, small but visible surface features can easily be added to them through the use of a normal map. However, there are still areas in which this general type of model can be improved: important features like anisotropy control sometimes lack analytic solutions, and the efficient rendering of normal maps requires accurate and general filtering algorithms.

We advance the state of the art with regard to such models in these areas: we derive analytic anisotropic models, reformulate the filtering problem and propose an efficient filtering algorithm based on a novel filtering data structure.

Specifically, we derive a general result in microfacet theory: given an arbitrary microsurface defined via standard microfacet statistics, we show how to construct the statistics of its linearly transformed counterparts. This leads to a simple closed-form expression for anisotropic variations of a given surface that generalizes previous work by supporting all microfacet distributions and all invertible tangential linear transformations. As a consequence, our approach allows transferring macrosurface deformations to the microsurface, so as to render its corresponding complex anisotropic appearance.

Furthermore, we analyze the filtering of the combined effect of a microfacet BRDF and a normal map. We show that the filtering problem can be expressed as an *Integral Histogram (IH)* evaluation. Due to the high memory usage of IHs, we develop the *Inverse Bin Map (IBM)*: a form of an IH that is very compact and fast to build. Based on the IBM, we present a highly memory-efficient technique for filtering normal maps that is targeted at the accurate rendering of glints, but in contrast with previous approaches also offers roughness control.

Keywords: computer graphics, realistic rendering, appearance modeling, microfacet theory, anisotropy, glints, normal map filtering

Contents

Chapter 1

Introduction

In the mid-18th century, Lambert [1760] published his book *Photometria* in which he measured the optical properties of materials and computed illumination based on a complete system of principles and quantities. Among others, this included the fundamental *Lambert's cosine law* [Wikipedia, 2023b] – stating that the intensity observed from an ideally diffuse surface is proportional to the cosine of the angle between the observer and the surface normal. This principle defines a perfectly matte surface known as *Lambertian* surface, and also serves as the basis for other shading models – these are mathematical models that describe the interaction of light and matter [Pharr et al., 2016].

In the same year, Bouguer's *Optical Treatise on the Gradation of Light* [1760] was published. In this work, Bouguer reasoned that the glossy reflection off a rough surface can be conceptualized as a reflection from a continuous surface, composed of microscopic flat facets (so-called *microfacets*) that have random orientations, and behave like mirrors [Trowbridge and Reitz, 1975].

Today, these same ideas are extensively used in practice and form the foundation of *microfacet theory*, which continues to be an active research area. Therefore, we can regard the year 1760 as the birth of this theory. See the front covers of these two foundational books in Figure 1.1.

Although the first concepts from microfacet theory were introduced in the 18th century, the main theory took shape in the optics community during the 20th century [Berry, 1923, Pokrowski, 1924, 1926, Barkas, 1939, Middleton and Mungall, 1952, Christie, 1953, Beckmann and Spizzichino, 1963, Torrance and Sparrow, 1967]. During the 19th century, however, wave optics theory was developed. Fresnel's model [1823] of light as a transverse elastic wave was backed by experiments and convinced the scientific community to adopt it [Wikipedia, 2023c]. Microfacet theory, like many light transport algorithms [Veach, 1998, Pharr et al., 2016], is grounded in geometric optics, and therefore wave optical phenomena like diffraction, interference and dispersion are excluded. This reduces both the conceptual and computational complexity of the models while at the same time being able to explain the vast majority of light interaction phenomena that we encounter daily. Yet, the specular reflection which is governed by *Fresnel equations* is an essential part of microfacet models [Walter et al., 2007, Pharr et al., 2016].

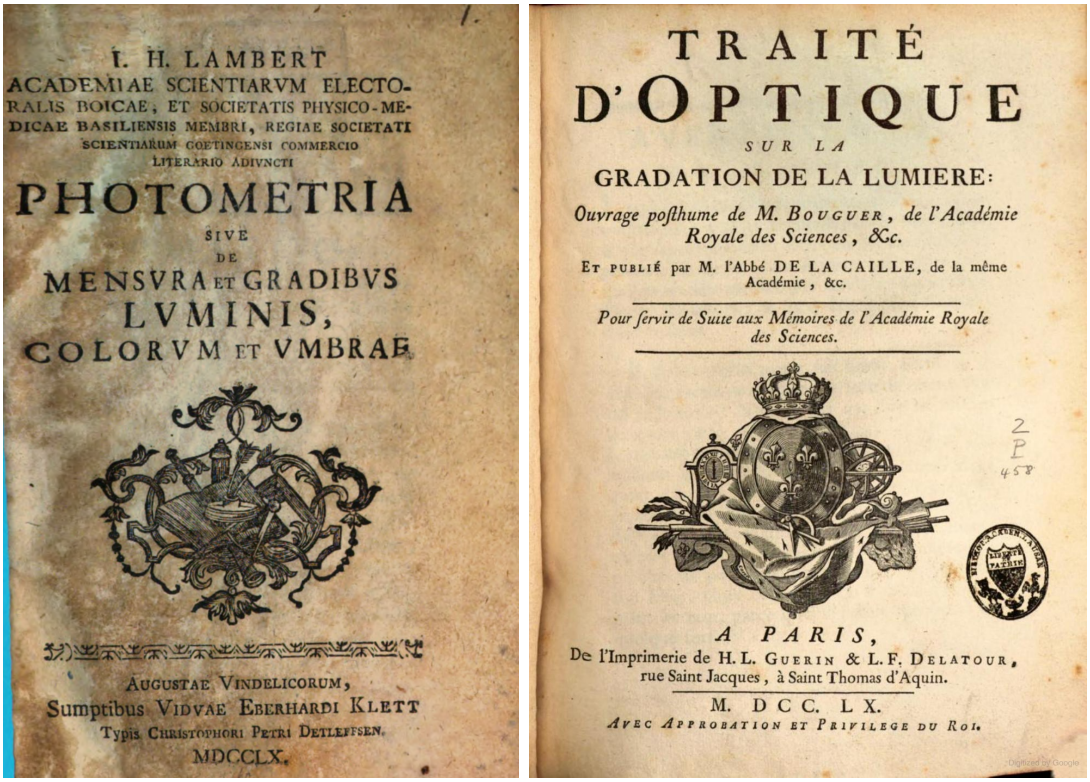


Figure 1.1: **Foundations of microfacet theory.** Front covers of J. H. Lambert's *Photometria* [1760] and P. Bouguer's *Optical Treatise on the Gradation of Light* [1760], both published in the same year.

1.1 Microfacet shading models

Since the advent of computers, the quest for photorealistic computer-generated images has had researchers focus on physically-based models of reflectance. Early on, microfacet theory was introduced to computer graphics [Blinn, 1977, Cook and Torrance, 1982]. Gradually, it was adopted as a standard instrument, to the point where virtually all rendering systems use it to describe rough surfaces, including educational [Pharr et al., 2016], research [Jakob, 2010, Bitterli, 2014, Nimier-David et al., 2019] and production [Burley, 2012, Hill et al., 2017].

At an abstract level, microfacet models are represented by a *microsurface* that is aligned with a *macrosurface*. The microsurface is composed of infinitesimal flat reflective facets that approximate the rough surface. The macrosurface is the shaded geometric object to which the microfacet model is applied. Microsurface features are assumed to be much smaller in comparison with the object's curvature so that the macrosurface is assumed locally flat, see Figure 1.2. On the other hand, it is assumed that the microfacets are sufficiently larger than the wavelength of light so that the model's geometric-optical approximation is reasonably accurate.

The resulting purely homogeneous surface appearance only matches our visual experience when viewing objects from a sufficient distance, Figure 1.3 **a)-c)**. For the close-up and mid-range views, which are much more common in our everyday experience, the effect of light interacting with larger details of the surface structure can often be resolved by the naked eye, Figure 1.3 **d)-e)**.

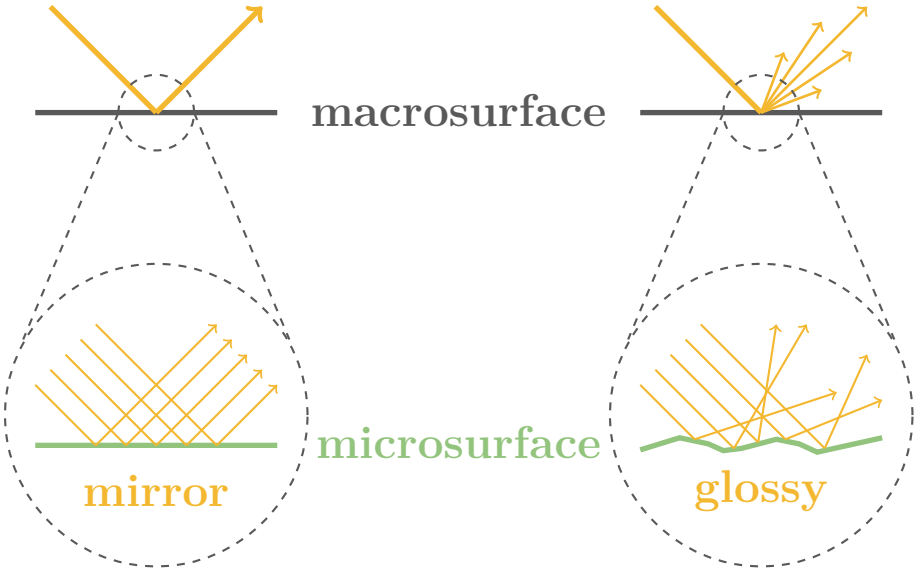


Figure 1.2: **Microfacet model.** At a macroscopic level (top) the surface on the left reflects light like a mirror while the reflection from the surface on the right is glossy. At a microscopic level (bottom) both surfaces reflect light specularly, however, the surface on the left is smooth while the surface on the right is rough.

Independently of microfacet approaches, Blinn [1978] presented bump mapping, a technique that adds visible surface detail by perturbing the surface normal according to a heightfield that is provided via a texture. When the tangent space normal is directly stored in the texture, this technique is referred to as normal mapping.

Rendering the combined appearance of a smooth microfacet model modulated by a normal map was investigated from the perspective of microfacet theory by Schüssler et al. [2017]. This is a problem of high practical relevance — virtually all rendering systems support normal mapping and the user can additionally control the roughness of the underlying surface. For example, metal surfaces with different roughness can be represented by a microfacet model while scratches can be added through the use of a normal map, see Figure 1.3 **d)**.

A large number of material appearances can be accomplished using this general type of microfacet model: rough metals and plastics, leather [Cook and Torrance, 1982, Yan et al., 2014, 2016], velvet and cloth's sheen [Ashikmin et al., 2000, Estevez and Kulla, 2017], along with compound, layered materials [Weidlich and Wilkie, 2007, Belcour, 2018]. Furthermore, microfacet models were extended successfully to rough refractive interfaces, e.g. frosted glass [Walter et al., 2007], hair scattering [Huang et al., 2022] and diffraction effects [Holzschuch and Pacanowski, 2017, Yan et al., 2018]. The problem of multiple scattering in microfacet surfaces has been addressed using stochastic techniques [Heitz et al., 2016b, Wang et al., 2022, Bitterli and d'Eon, 2022], however, due to their computational overhead, we restrict our models to the first scattering event.

In Appendix A, we provide microscope photographs of several materials that have a rough microstructure, separated into two categories — *conductors*, i.e., metals and *dielectrics*. It is common to classify materials by their ability to conduct electricity, because this property affects their appearance, as determined by Fresnel equations. *Semiconductors* are the third type of materials that are

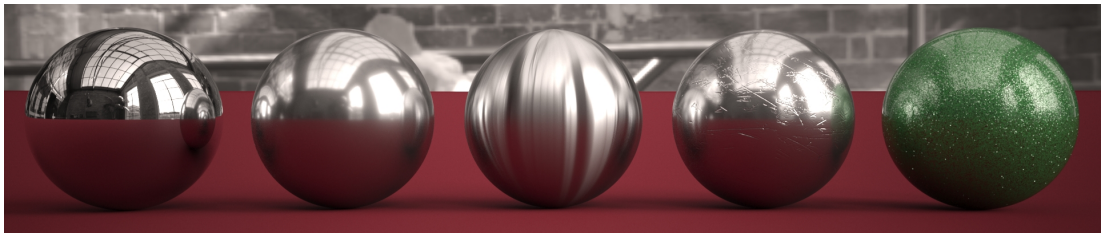


Figure 1.3: **Rendered aluminum microsurfaces.** From left to right: **a) Mirror:** The microsurface coincides with the macrosurface resulting in a perfectly flat interface (see Figure 1.2, left); **b) Blurry:** The microsurface facets follow a smooth statistical distribution resulting in a blurred, so-called *glossy reflection*; **c) Anisotropic:** The microsurface is smoother horizontally and rougher vertically, creating strong anisotropic reflection; **d) Scratched:** The rough surface from **b)**, modulated by a scratch normal map; **e) Metallic paint:** Aluminum flakes suspended in a green medium. The flakes are modeled via a normal map. (The scene is lit by an environment map provided by Rhys Dippie.)

not usually introduced in rendering systems, due to being absent from common macroscopic objects.

1.2 Motivation

In this thesis, we focus on two reflective phenomena.

1.2.1 Anisotropic appearance

Surfaces with anisotropic properties that are smoother in one direction and rougher in another direction exhibit a distinguishable reflective appearance. Brushed metals [Wikipedia, 2023a] are probably the most common example, see Figure 1.4. They are extensively used in modern architecture: stainless steel finishes are often used in interiors and exteriors alike. From facades and monuments to interior panels, and elevator doors and cabins. Brushed metals are also used in the automotive and product industries, for example in electrical appliances, also in kitchen and bathroom accessories. Components manufactured using a lathe machine also have a distinctive anisotropic look.

While objects with a strong anisotropic appearance are usually metals, there are examples of anisotropic dielectric objects. For example, gramophone records are inscribed with a spiral groove that creates a strong anisotropic reflection, similar to the bottom of the pot in Figure 1.4. Also, wood has a complex texture, and its top layer or coating can show anisotropy, like the weak anisotropy of the bamboo wireless charger in Figure A.2. The top layer of deforming human skin also reflects light anisotropically, and this effect is crucial to the natural look of rendered skin [Nagano et al., 2015]. Finally, many elastic objects manifest dynamic anisotropic reflection, for example, the rubber balloon in Figure 1.5 and Figure A.2.

We develop a framework to render the appearance of anisotropic surfaces, including such driven by dynamic deformations.

Figure 1.4: **Photographs of highly anisotropic brushed metals.** From left to right: a door handle with a similar appearance to the rendered ball in Figure 1.3 **c)**; an interior column covered with linearly brushed metal; a side panel of a truck with anisotropic circles; anisotropic pots and lids.

1.2.2 Glinty appearance

Rough surfaces with microscopic features that reflect light (nearly) specularly, especially under high-frequency illumination, produce spatially varying, illumination- and view-dependent micro-highlights referred to as *glints*, *glitter*, *sparkles*, *coherent scratches*, etc. These include natural phenomena like sand, snow and the ocean surface, but also manufactured materials like matte plastics, rough and scratched metals and metallic paints, see Figure 1.6.

The perception of color and texture of metallic car paint glitter has been investigated in the color science literature due to its importance to the automotive industry [McCamy, 1996, 1998, Dekker et al., 2011, Kirchner et al., 2007].

In computer graphics, these effects are often modeled using specular normal maps, however, they prove very difficult to converge using the standard stochastic sampling of Monte Carlo ray tracers, and to tackle this problem filtering algorithms need to be employed [Yan et al., 2014, 2016]. This is because a large number of glints could be contained in a single pixel, but only a fraction of them reflects light toward the observer. For example, metallic car paint can be modeled using a flake normal map [Günther et al., 2005]. While the flakes are often modeled as specular, empirical measurements support that the roughness of the individual flakes has an important contribution to the overall appearance [Sung et al., 2002].

We design a filtering algorithm for a glinty appearance that provides roughness control and demonstrates that a small change in the roughness affects the appearance significantly.

1.3 List of original contributions

In this thesis, we describe two novel microfacet models and list their contributions.

1.3.1 Linearly Transformed Microsurface

The first model [Atanasov et al., 2022] is described in Chapter 4. It constitutes an addition to microfacet theory that facilitates anisotropic modifications to surfaces. Given an arbitrary microsurface described by standard microfacet

$C = 56cm$ $C = 67cm$ $C = 77cm$

Figure 1.5: **Photographs of a rubber balloon in different stages of inflation.** From left to right the balloon has circumferences $C = (56cm, 67cm, 77cm)$. The rubber is a rough surface that is getting smoother with inflation. This causes the highlight to become sharper (the photographs are scaled to have highlights of the same size). Note that the surface is stretched out unevenly, thus the resulting reflection is anisotropic.

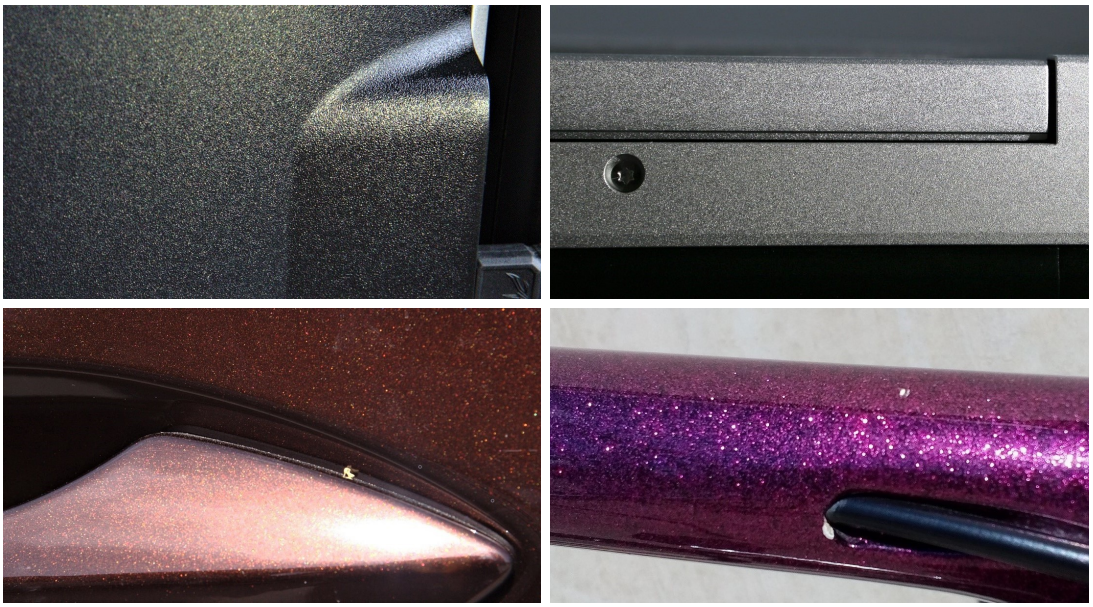


Figure 1.6: **Photographs of glinty appearance.** The back of a computer monitor, made from rough plastic (top left); a laptop frame, made from rough aluminum (top right); a door handle of a car, painted with metallic paint (bottom left); a bike frame, painted with metallic paint with large flakes (bottom right).

statistics, it defines the statistics of its linearly transformed counterparts, creating an infinite family of new appearances. We list its main contributions and practical implications.

Valid microsurface transformations. Of all the possible 3×3 linear transformation matrices we show that the 2D invertible tangential matrices (the upper xy part) lead to new valid microsurfaces, while both rotations outside the xy plane and shearing parallel to the xy plane break the microsurface. Furthermore, we show that the scaling of the microsurface's height results in a valid microsurface, but a microsurface with the same statistics can also be obtained by applying inverse scaling to the xy plane. Thus, we show that all valid microsurfaces obtainable as a result of linearly transformed microsurfaces can be expressed by a 2D tangential transformation matrix.

Closed-form statistics. We derive a full set of simple closed-form equations that provide physically-based microsurface configurations and efficient sampling:

- We introduce the family of *Linearly Transformed Microfacet Distributions (LTMD)*, which is an infinite family of normalized microfacet distributions, parametrized by all valid microsurface transformations. To this end, we compute the Jacobian determinant of the nonlinear transformation that multiplies a vector with a linear transformation and normalizes the result on the sphere.

- We generalize the *masking probability invariance* property [Heitz, 2014] to all valid microsurface transformations, beyond just stretching. This property states that when a microsurface-direction configuration is stretched, the proportion of the shadowed-unshadowed area with respect to the stretched direction is preserved. Consequently, we can re-parametrize the 1-dimensional shadowing-masking function to apply accurate shadowing to all transformed surfaces.

- We show how the sampling equations for (visible) microfacet normals of the original non-transformed microsurface can be employed to sample (visible) microfacet normals on the transformed microsurface.

Stretch anisotropy. We show that our model for linearly transformed microsurfaces can be used to create anisotropic variations of microfacet-based surfaces, regardless of the underlying microfacet distribution, by stretching/compressing them along the axes. We also show that our method is a generalization of the elliptical anisotropy — the standard method based on specifying different roughness values along the two axes. It has been shown that elliptical anisotropy is guaranteed to have a closed-form solution for the special class of *shape-invariant* microfacet distributions [Heitz, 2014], while it is often the case that all other *(shape-variant)* distributions cannot be integrated analytically (e.g. the GTR microfacet distribution [Burley, 2012]). For all shape-invariant distributions, we prove that our method produces the same result as the elliptical anisotropy, while our closed-form solution is valid in the entire space of microfacet distributions.

Nonlinear transformations. Another practical application of our model is simulating the appearance of light reflected from a deformed surface. To this end, we track the tangential deformations of the rendered mesh using per-face

linear transformations. Due to our model being parametrized in the space of the microsurface normals, we can directly feed these linear transformations into our model. Alternatively, we demonstrate that for an analytic nonlinear transformation we can directly provide its Jacobian matrix to our model, and render the corresponding appearance.

1.3.2 Multiscale Microfacet Model

The second model [Atanasov et al., 2021] is presented in Chapter 5. It is a normal map filtering approach that can resolve an accurate glinty appearance as well as a glossy appearance by offering roughness control on top of the normal map. Follows the list of contributions.

Integral histogram normal map filtering. We formulate the general problem of filtering the combined effect of a microfacet BRDF and a normal map, and by discretizing the normal distribution function into directional bins, we reduce the problem to an evaluation of an *integral histogram (IH)* [Porikli, 2005]. The integral histogram is a data structure, which has a variety of filtering applications [Ballester-Ripoll and Pajarola, 2019]. Our formulation can serve as the basis for developing normal map filtering algorithms.

Inverse Bin Maps. While classical integral histograms are attractive due to their constant-time queries in axis-aligned regions, they require a full data-size *Summed-area table (SAT)* [Crow, 1984] for each sparse data bin. This redundancy is tolerable for problems that require a moderate number of data bins, but when a large number of data bins is needed, memory usage becomes impractically large. To tackle this issue, we have designed the *Inverse Bin Map (IBM)* data structure which is a compact integral histogram. It is fast to build and extremely memory efficient. This is achieved at the cost of reducing the query speed from constant to logarithmic. Another advantage of IBMs is that they support arbitrary-shaped query regions.

Practical normal map filtering. We propose an accurate normal map filtering technique based on our integral histogram formulation and the IBM data structure. In contrast to previous work, the technique features a brief precomputation, modest memory requirements and importantly filters the accurate appearance at all scales. While concurrent accurate approaches are specialized at filtering specular normal maps, our method exposes Beckmann roughness control and filters both specular and glossy normal maps. A key to the efficiency of this algorithm is the variable bin resolution which is a function of the *Beckmann flake roughness*: we assign a higher number of bins to normal maps combined with lower-roughness materials (approaching specular normal maps), while we give a lower number of bins to those combined with higher-roughness materials. We derive such a relation that for each evaluation we query a fixed number of IBM bins regardless of the Beckmann roughness. This promotes an efficient implementation and increases the accuracy of the more challenging low-roughness normal maps.

1.4 List of publications

The author was the primary investigator of two journal articles published in *Computer Graphics Forum* and three short papers published in *ACM SIGGRAPH* conference proceedings:

- *Microsurface Transformations*
  Asen Atanasov, Vladimir Koylazov, Rossen Dimov, Alexander Wilkie
  Eurographics Symposium on Rendering EGSR 2022

- *A Multiscale Microfacet Model Based on Inverse Bin Mapping*
  Asen Atanasov, Alexander Wilkie, Vladimir Koylazov, Jaroslav Křivánek
  Eurographics 2021

- *Efficient Multiscale Rendering of Specular Microstructure*
  Asen Atanasov, Jaroslav Křivánek, Vladimir Koylazov, Alexander Wilkie
  ACM SIGGRAPH 2020 Talks

- *Adaptive Environment Sampling on CPU and GPU*
  Asen Atanasov, Vladimir Koylazov, Blagovest Taskov, Alexander Soklev, Vassillen Chizhov, Jaroslav Křivánek
  ACM SIGGRAPH 2018 Talks

- *A Practical Stochastic Algorithm for Rendering Mirror-Like Flakes*
  Asen Atanasov, Vladimir Koylazov
  ACM SIGGRAPH 2016 Talks

This dissertation is based on the two journal articles at the top of the list.

Chapter 2

Measuring Reflection

In this chapter, we introduce central concepts from computer graphics that are related to shading models. We start with the main radiometric quantities that allow us to take measurements of reflected light. *Radiometry* is the branch of physics that deals with the measurement of electromagnetic radiation. Of primary interest is the definition of the BRDF (Bidirectional Reflectance Distribution Function) — the main tool that assists the integration of reflective materials in rendering systems. The BTDF (Bidirectional Transmission Distribution Function) [Walter et al., 2007] is beyond the scope of this work. Then we formulate two basic BRDFs: the Lambert BRDF, which represents perfect diffuse reflection, and the specular BRDF, which represents a perfect mirror reflection, modulated by Fresnel equations.

The main notation for this chapter is summarized in Table 2.1. In the following chapters, we will append more specific notation. Notice that all directions on the unit hemisphere are in the tangent space of the surface normal $\mathbf{n}$, meaning that $\mathbf{n}$ is always pointing up towards the zenith. For a brief exposition on spherical coordinates and integration on the hemisphere refer to Appendix B.

2.1 Radiometry

In order to perform computations with light, we need to take a rigorous quantitative approach. Light is electromagnetic radiation which consists of tiny bits of energy, i.e., *photons*. Each photon is characterized by its wavelength λ, and has energy

$$Q = \frac{hc}{\lambda}, \tag{2.1}$$

where h is Planck's constant and c is the speed of light. The energy is measured in joules $[J]$. The highest energy photons in the visible spectrum compound the violet light, which reaches energy of up to $5.32 \times 10^{-19} J$ for a single photon.

Theoretically, we can trace and count individual photons, but this is computationally infeasible[1] — even a single candle (assuming $5W$ of visible light) emits more than 10^{19} photons per second in the visible spectrum. Consisting of

[1] This statement should not be confused with particle tracing techniques like *photon mapping* [Jensen, 2001], where the so-called *photons* carry much larger portions of energy than physical photons.

$\mathcal{H}^2$	Unit hemisphere
$\mathcal{D}$	Unit disk
x	Point on the shaded surface
$\mathbf{n}$	Surface normal $(0, 0, 1)^T$
$\mathbf{v}$	Direction on the unit hemisphere with Cartesian coordinates (v_x, v_y, v_z) and spherical coordinates $(\theta_\mathbf{v}, \phi_\mathbf{v})$
$\mathbf{i}$	Incoming light direction
$\mathbf{o}$	Outgoing light direction
$\mathbf{h}$	Half vector $(\mathbf{i} + \mathbf{o})/\|\mathbf{i} + \mathbf{o}\|$
$\mathbf{r}$	Direction of perfect specular reflection $2(\mathbf{i} \cdot \mathbf{n})\mathbf{n} - \mathbf{i}$
$L(x, \mathbf{i})$	Incident radiance at the point x along $\mathbf{i}$
$L(x, \mathbf{o})$	Reflected radiance from the point x along $\mathbf{o}$
ρ	Diffuse albedo in $[0, 1]$
η, k	Index of refraction and absorption coefficient
F	Fresnel term for conductors or dielectrics
$\delta_\mathbf{o}, \delta_\mathbf{h}$	Hemispherical Dirac delta distributions associated with differential solid angles d$\mathbf{o}$ and d$\mathbf{h}$, respectively
f_x	The BRDF at the point x
f^0	Specular BRDF

Table 2.1: Basic notation related to shading models.

such a large number of photons in common everyday scenarios, despite its discrete nature, the radiant energy Q is modeled as a continuous quantity. Such an approach to describe a very large number of very small objects using continuous quantities is common in mathematical modeling, and in this thesis, we shall encounter another instance of it — although real-world microfacets are a finite number and are of finite area, analytic microfacet models assume an infinite number of infinitesimal microfacets. While continuous radiant energy makes light transport simulation viable [Veach, 1998], continuous distribution of microfacets makes microfacet models efficient.

2.1.1 Radiometric quantities

Here we list the related radiometric quantities:

- **Flux.** The *flux*, also known as *radiant power*, measures energy per unit time, or watts $[W = J \cdot s^{-1}]$, and it is defined as

$$\Phi(t) = \frac{\mathrm{d}Q(t)}{\mathrm{d}t}. \tag{2.2}$$

  For example, the flux of a light bulb is the energy, hence the spectrum of photons that is emitted per second.

- **Irradiance.** The *irradiance* is measured as the flux per unit area $[W \cdot m^{-2}]$. At a point x on a surface, the irradiance is defined as

$$E(x) = \frac{\mathrm{d}\Phi(x)}{\mathrm{d}A(x)}, \tag{2.3}$$

where $\mathrm{d}A(x)$ is a differential surface area around x.

- **Radiance.** The *radiance* is the flux per unit projected area per unit solid angle $[W \cdot m^{-2} \cdot sr^{-1}]$

$$L(x, \mathbf{v}) = \frac{\mathrm{d}^2\Phi(x, \mathbf{v})}{(\mathbf{v} \cdot \mathbf{n})\mathrm{d}A(x)\mathrm{d}\mathbf{v}}, \tag{2.4}$$

  where $\mathbf{n}$ is the surface normal at the point x. Therefore, the radiance measures the flux that arrives from a solid angle around the direction $\mathbf{v}$ at the differential area around the point x that is projected along $\mathbf{v}$.

For an extended discussion on radiometric quantities, we refer the reader to the book by Pharr et al. [2016] or the dissertation of Veach [1998].

2.1.2 BRDF

We introduce the BRDF (Bidirectional Reflectance Distribution Function) — a mathematical description of surface reflection at a point. It was proposed as a representation of surface reflective properties, which is capable of emulating a wide variety of materials [Nicodemus et al., 1977].

Intuitively, the BRDF answers the question "What portion of the light that arrives at the point x along the incoming light direction $\mathbf{i}$ is reflected towards the outgoing light direction $\mathbf{o}$?", see Figure 2.1. Formally, it is defined as the differ-

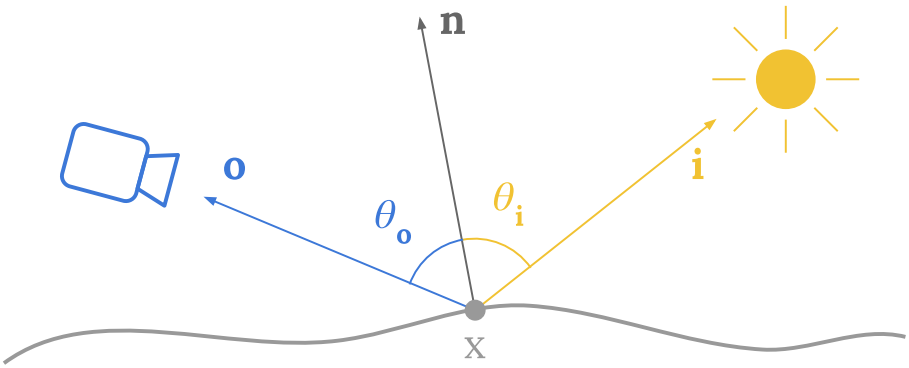


Figure 2.1: **Bidirectional Reflectance Distribution Function (BRDF).** The BRDF defines the surface reflection properties at a point x with a surface normal $\mathbf{n}$. The BRDF value is the portion of the light coming from the light source along the incoming light direction $\mathbf{i}$ that is reflected towards the sensor (e.g. eye or camera) along the outgoing light direction $\mathbf{o}$. The angles $\theta_\mathbf{o}$ and $\theta_\mathbf{i}$ between the surface normal and the directions $\mathbf{o}$ and $\mathbf{i}$ are known as *angles of incidence*.

ential radiance reflected towards the outgoing light direction $\mathbf{o}$, due to differential irradiance arriving from direction $\mathbf{i}$

$$f_x(\mathbf{i}, \mathbf{o}, \mathbf{n}) = \frac{\mathrm{d}L(x, \mathbf{o})}{\mathrm{d}E(x, \mathbf{i})} = \frac{\mathrm{d}L(x, \mathbf{o})}{L(x, \mathbf{i})(\mathbf{i} \cdot \mathbf{n})\mathrm{d}\mathbf{i}}. \tag{2.5}$$

At the point x with normal $\mathbf{n}$, the BRDF is a 4-dimensional function, parametrized with the spherical coordinates of the incoming and outgoing

light directions $(\theta_\mathbf{i}, \phi_\mathbf{i}, \theta_\mathbf{o}, \phi_\mathbf{o})$. In cases when the BRDF does not depend on the surface position x, we drop the subscript x. From Equation (2.5), we can see that the BRDF is measured in inverse steradians $[sr^{-1}]$.

Physical properties impose the following constraints on BRDFs:

- **Non-negativity.** Reflecting a negative amount of light has no physical meaning, and therefore the BRDF must be greater or equal to zero for all input directions
$$f_x(\mathbf{i}, \mathbf{o}, \mathbf{n}) \geq 0. \tag{2.6}$$

- **Helmholtz reciprocity.** If the directions $\mathbf{i}$ and $\mathbf{o}$ are exchanged, i.e, the direction of light flow is reversed, then the reflected amount must not change
$$f_x(\mathbf{i}, \mathbf{o}, \mathbf{n}) = f_x(\mathbf{o}, \mathbf{i}, \mathbf{n}). \tag{2.7}$$

- **Energy conservation.** The total reflected light from the point x must never exceed the incident light at x
$$\int_{\mathcal{H}^2} f_x(\mathbf{i}, \mathbf{o}, \mathbf{n})(\mathbf{o} \cdot \mathbf{n}) \mathrm{d}\mathbf{o} \leq 1, \ \forall \mathbf{i} \in \mathcal{H}^2. \tag{2.8}$$

These three properties are necessary but not sufficient to describe a physically-based BRDF. In microfacet theory, physically-based BRDFs are those that correspond to a correct geometrical configuration of the microsurface [Heitz, 2014].

2.1.3 Reflection equation

The BRDF at the point x on a surface of interest is used to compute the reflected light from this point. We want to compute the reflected radiance $L(x, \mathbf{o})$ along a given outgoing direction $\mathbf{o}$. Incident radiance $L(x, \mathbf{i})$ coming from all directions $\mathbf{i} \in \mathcal{H}^2$, for which the BRDF is positive $f_x(\mathbf{i}, \mathbf{o}, \mathbf{n}) > 0$, contribute to the reflected radiance in direction $\mathbf{o}$, and we need to sum up all contributions $L(x, \mathbf{i}) f_x(\mathbf{i}, \mathbf{o}, \mathbf{n})$.

By definition, the incident radiance $L(x, \mathbf{i})$ from an incoming light direction $\mathbf{i}$ is the flux per unit solid angle around $\mathbf{i}$, per unit projected area along $\mathbf{i}$. In order to convert the projected areas along directions $\mathbf{i}$ into the same surface area at x, we need to multiply by $\cos\theta_i = \mathbf{i} \cdot \mathbf{n}$, see Equation (2.4). This is needed, because the radiance that is coming along larger incident angles θ_i is spread over larger surface areas, and therefore it is weaker per unit area, compared to the same radiance coming along small incident angles, i.e., Lambert's law. Consequently, the reflected radiance is computed using the *reflection equation*

$$L(x, \mathbf{o}) = \int_{\mathcal{H}^2} L(x, \mathbf{i}) f_x(\mathbf{i}, \mathbf{o}, \mathbf{n})(\mathbf{i} \cdot \mathbf{n}) \mathrm{d}\mathbf{i}, \tag{2.9}$$

which is illustrated in Figure 2.2. Notice that for a specified outgoing direction $\mathbf{o}$, the 4-dimensional BRDF is reduced to a 2-dimensional function in $(\theta_\mathbf{i}, \phi_\mathbf{i})$, which is integrated on the hemisphere.

The reflection equation has analytic solutions only in very simple scenarios, thus it is mainly solved numerically. In light transport algorithms, it is commonly solved using Monte Carlo methods which introduce high variance, manifested in the form of noise. Therefore, rendering systems employ variance reduction

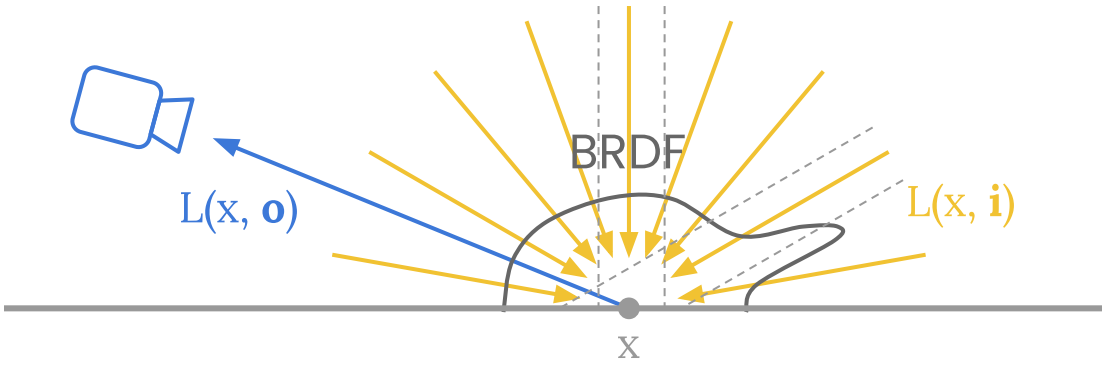


Figure 2.2: **The reflection equation.** To compute the outgoing radiance $L(x, \mathbf{o})$, the reflection equation integrates the product of the BRDF and the cosine-weighted incoming radiance $L(x, \mathbf{i})(\mathbf{i} \cdot \mathbf{n})$. The light that is reflected along $\mathbf{o}$ could have arrived at the point x from any direction on the upper hemisphere, depending on the surface properties. For a fixed direction $\mathbf{o}$, the BRDF is a function on the hemisphere (dark gray), centered at x. Two light beams of light with the same cross-section (gray dashed lines) illuminate different areas around x, depending on their inclinations. The light that is coming from directions with larger incident angles spread over larger areas, and therefore its power per unit area is reduced. This is compensated by the cosine factor $(\mathbf{i} \cdot \mathbf{n}) = \cos\theta_{\mathbf{i}}$, see Equation (2.9).

techniques, with *importance sampling* being the most widely used due to its efficiency [Pharr et al., 2016].

Importance sampling is aiming at generating more samples in the important regions, where the integrand value is high. Ideally, the density of the samples must be proportional to the integrand. Due to this, successful utilization of importance sampling requires well-suited sampling routines. In particular, general importance sampling strategies for the reflection equation do not exist due to the complex geometric nature of the problem — the BRDF, potentially being fairly complex by itself, is a property of the surface, while the incident radiance is coming from various light sources, which could be (partially) occluded.

Therefore, a very practical and robust technique that is used in rendering systems is to combine sampling strategies that are targeting different factors of the integrand using *multiple importance sampling (MIS)* [Veach and Guibas, 1995]. To this end, specialized sampling techniques are developed to sample the light sources and the BRDF, respectively. In Figure 2.3, we sketch light sampling and BRDF sampling.

Hence, practical implementations of BRDFs must have the following features:

- **Evaluation.** Efficient formula or algorithm to compute the BRDF value $f_x(\mathbf{i}, \mathbf{o}, \mathbf{n})$.
- **Sampling.** Efficient BRDF sampling strategy: this is sampling directions with a density that is similar to the BRDF shape. Ideally, this is a sampling procedure with density proportional to the product $f_x(\mathbf{i}, \mathbf{o}, \mathbf{n})(\mathbf{i} \cdot \mathbf{n})$, although such sampling equations are rarely available.

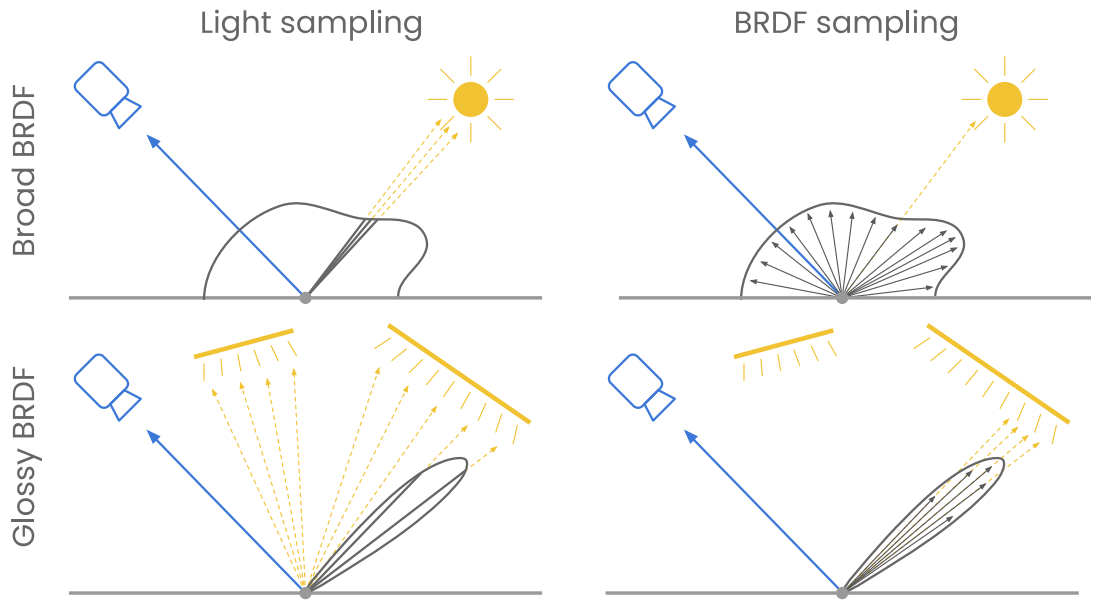


Figure 2.3: **Illustration of sampling strategies for the reflection equation.** For a fixed outgoing light direction (blue) the BRDF is visualized as a function on the hemisphere (dark gray).
Left column: Sampling the light source: this is usually done by choosing random points on the light source or sampling random directions in the solid angle that the light source subtends on the unit hemisphere at the point x. **Right column:** Sampling the BRDF: sampling directions that are at best proportional to the product of the BRDF lobe and the cosine term, or at least are generated with higher probability in the regions where the value of this product is high. **Top row:** A diffuse-like BRDF with a broad lobe that reflects light in all directions. The light sampling strategy usually performs well in such cases since all generated light directions that connect the light source with the point x, contribute to the reflected light. On the other hand, the BRDF sampling could be very inefficient, especially for small light sources — samples are distributed in all directions, with a low probability of hitting the light source. **Bottom row:** A glossy BRDF with an elongated directional lobe that reflects light around a single direction. For such BRDFs, the light sampling strategy could be inefficient, especially for large light sources, because from all generated directions few reflect light towards the outgoing direction. On the other hand, sampling the BRDF in this case could hit the large light source. In practical scenarios with many lights of different intensities and sizes, and different BRDF shapes, it is not easy to pick the right strategy. Therefore, both strategies are computed, and their contributions are weighed depending on their probabilities using MIS [Veach and Guibas, 1995].

- **Probability.** Efficient routine to compute the probability density function (PDF) of the BRDF sampling.

2.2 Diffuse and specular reflection

As illustrated in Figure 2.3, the BRDF shapes vary from broad diffuse-like lobes to directional glossy lobes. The two extremes in terms of directionality are the perfect diffuse BRDF, known as Lambertian BRDF, and the perfect specular BRDF, which reflects all incident light in a single direction. These are important BRDFs that can be used directly or serve as building blocks in more complex BRDFs, e.g. microfacet models.

2.2.1 Lambert BRDF

A surface that scatters light uniformly in all directions on $\mathcal{H}^2$ is referred to as a *Lambertian surface*. This is an idealized model for perfect matte reflection which could be a reasonable approximation for unfinished wood, paper or wall paint. *Spectralon* is a manufactured material that is designed to exhibit nearly perfect Lambertian reflection [Georgiev and Butler, 2007].

The only parameter of Lambertian materials is their albedo $\rho \in [0, 1]$ which defines what portion of the incoming light is reflected, while the rest $1 - \rho$ is absorbed. Therefore, the Lambert BRDF is proportional to its albedo $\kappa\rho$, where κ is the proportionality constant. In order to find κ we use the energy conservation property, Equation (2.8)

$$\int_{\mathcal{H}^2} \kappa\rho(\mathbf{o} \cdot \mathbf{n}) \mathrm{d}\mathbf{o} = \kappa\rho \int_0^{2\pi} \int_0^{\frac{\pi}{2}} \cos\theta_\mathbf{o} \sin\theta_\mathbf{o} \mathrm{d}\theta_\mathbf{o} \mathrm{d}\phi_\mathbf{o} = \tag{2.10}$$

$$\kappa\rho \int_0^{2\pi} \int_0^{\frac{\pi}{2}} \sin\theta_\mathbf{o} \mathrm{d}\sin\theta_\mathbf{o} \mathrm{d}\phi_\mathbf{o} = \kappa\rho \int_0^{2\pi} \left(\frac{\sin^2\theta_\mathbf{o}}{2} \Big|_0^{\frac{\pi}{2}} \right) \mathrm{d}\phi_\mathbf{o} = \frac{\kappa\rho}{2} \int_0^{2\pi} \mathrm{d}\phi_\mathbf{o} = \tag{2.11}$$

$$\kappa\rho\pi \leq 1. \tag{2.12}$$

Since the albedo could be at most $\rho = 1$ in the case when all the energy is reflected, then the constant κ must be equal to $\frac{1}{\pi}$, thus the Lambert BRDF is

$$f(\mathbf{i}, \mathbf{o}, \mathbf{n}) = \frac{\rho}{\pi}. \tag{2.13}$$

2.2.2 Fresnel reflectance

A common assumption in microfacet models for reflection from rough surfaces is that the microfacets are perfect mirrors. In order to formulate these models one must understand the perfect specular reflection — this is the reflection from an ideally smooth surface. When light reaches such a surface, a portion of the light is reflected while the rest of the light is transmitted and/or absorbed, depending on the physical material of the surface. The amount of reflected light depends on the angle of incidence $\theta_\mathbf{i}$, and it is described by Fresnel equations — these are a solution to Maxwell's equations for the reflection of an electromagnetic wave at the flat boundary between two different homogeneous media.

Each medium is characterized by its index of refraction (IOR) (denoted η), which indicates how much the light slows down in the medium, compared to the speed of light in a vacuum (with $\eta = 1$). These include homogeneous dielectric media like air ($\eta \approx 1.00029$), water ($\eta \approx 1.33$), glass ($\eta \approx 1.5$) and diamond ($\eta = 2.42$). Another important class of materials is conductors, which include all metals. Metals are opaque, thus they do not transmit light but absorb it. In addition to the IOR η, they are characterized by an *absorption coefficient* k, which by definition is zero for dielectric materials.

For the purposes of this work, we formulate the Fresnel equations for the vacuum-material interface, since the vacuum has a similar IOR with air, and the material is given by the pair (η, k). The plane that contains the incoming and the reflected ray is named the *plane of incidence*. Since light is a transversal wave, each photon wave has an orientation along its propagation direction, i.e., *polarization*. Fresnel equations express the reflectivities for two orthogonal configurations — R_s for waves that are perpendicular to the plane of incidence, and R_p for waves that are parallel to the plane of incidence:

$$R_s = \frac{a^2 + b^2 - 2a\cos\theta_\mathbf{i} + \cos^2\theta_\mathbf{i}}{a^2 + b^2 + 2a\cos\theta_\mathbf{i} + \cos^2\theta_\mathbf{i}} \tag{2.14}$$

and

$$R_p = R_s \frac{(a^2 + b^2)\cos^2\theta_\mathbf{i} - 2a\cos\theta_\mathbf{i}(1 - \cos^2\theta_\mathbf{i}) + (1 - \cos^2\theta_\mathbf{i})^2}{(a^2 + b^2)\cos^2\theta_\mathbf{i} + 2a\cos\theta_\mathbf{i}(1 - \cos^2\theta_\mathbf{i}) + (1 - \cos^2\theta_\mathbf{i})^2}, \tag{2.15}$$

where

$$a^2 = \frac{1}{2}\left(\sqrt{c^2 + 4\eta^2k^2} + c\right), \tag{2.16}$$

$$b^2 = \frac{1}{2}\left(\sqrt{c^2 + 4\eta^2k^2} - c\right) \tag{2.17}$$

and

$$c = \eta^2 - k^2 + \cos^2\theta_\mathbf{i} - 1. \tag{2.18}$$

It is noteworthy that these equations can be simplified for dielectrics because they do not have an absorption coefficient ($k = 0$), and therefore it is a common practice in rendering systems to keep separate implementations for conductors and dielectrics [Pharr et al., 2016].

Furthermore, the IOR and the absorption coefficient are both wavelength-dependent properties. Due to this, light with different wavelengths is refracted along different angles, leading to the well-known decomposition of light into its spectrum — a phenomenon known as *dispersion*. In the case of metals, this wavelength dependence is the cause of their colors.

A common assumption in graphics is that the light is *unpolarized* — meaning that photons have random polarizations. In this case, the Fresnel reflectance is

$$F(\mathbf{i}\cdot\mathbf{n}) = F(\cos\theta_\mathbf{i}) = \frac{1}{2}\left(R_s^2 + R_p^2\right). \tag{2.19}$$

2.2.3 Specular BRDF

Specular reflection occurs at perfectly smooth interfaces. The incoming light along the direction $\mathbf{i} = (\theta_\mathbf{i}, \phi_\mathbf{i})$ is reflected in direction $\mathbf{r} = (\theta_\mathbf{i}, \phi_\mathbf{i} + \pi)$. The

directions $\mathbf{i}$ and $\mathbf{r}$, together with the surface normal $\mathbf{n}$ are contained in the plane of incidence. The reflection direction is computed as

$$\mathbf{r} = 2(\mathbf{i} \cdot \mathbf{n})\mathbf{n} - \mathbf{i}. \tag{2.20}$$

To derive the specular BRDF we need a tool to express that all the reflected energy is concentrated in a single direction. To this end, we employ a hemispherical Dirac delta distribution, defined for a differential solid angle d$\mathbf{o}$ containing the outgoing direction:

$$\delta_{\mathbf{o}}(\mathbf{v}) = \begin{cases} \infty, & \mathbf{v} = \mathbf{o} \\ 0, & \text{otherwise} \end{cases}, \ \mathbf{v} \in \mathcal{H}^2, \tag{2.21}$$

which has the property

$$\int_{\mathcal{H}^2} \delta_{\mathbf{o}}(\mathbf{v}) \mathrm{d}\mathbf{v} = 1. \tag{2.22}$$

As a consequence, this property allows the evaluation of integrands containing the delta function:

$$\int_{\mathcal{H}^2} \delta_{\mathbf{o}}(\mathbf{v}) f(\mathbf{v}) \mathrm{d}\mathbf{v} = f(\mathbf{o}). \tag{2.23}$$

The specular BRDF takes the form

$$f^0(\mathbf{i}, \mathbf{o}, \mathbf{n}) = \kappa \delta_{\mathbf{o}}(\mathbf{r}) F(\mathbf{i} \cdot \mathbf{n}), \tag{2.24}$$

where κ is a normalization constant. This BRDF is non-zero (and proportional to the Fresnel term) only when the outgoing light direction $\mathbf{o}$ coincides with the direction of perfect reflection $\mathbf{r}$. To calculate the constant κ we use that the total reflected light must be equal to the Fresnel term

$$\int_{\mathcal{H}^2} f^0(\mathbf{i}, \mathbf{o}, \mathbf{n})(\mathbf{o} \cdot \mathbf{n}) \mathrm{d}\mathbf{o} = F(\mathbf{i} \cdot \mathbf{n}) \tag{2.25}$$

$$\int_{\mathcal{H}^2} \kappa \delta_{\mathbf{o}}(\mathbf{r}) F(\mathbf{i} \cdot \mathbf{n})(\mathbf{o} \cdot \mathbf{n}) \mathrm{d}\mathbf{o} = F(\mathbf{i} \cdot \mathbf{n}) \tag{2.26}$$

$$\kappa F(\mathbf{i} \cdot \mathbf{n})(\mathbf{r} \cdot \mathbf{n}) = F(\mathbf{i} \cdot \mathbf{n}) \tag{2.27}$$

$$\kappa = \frac{1}{\mathbf{r} \cdot \mathbf{n}}. \tag{2.28}$$

Notice that by definition $\theta_{\mathbf{r}} = \theta_{\mathbf{i}}$, thus

$$\mathbf{r} \cdot \mathbf{n} = \mathbf{i} \cdot \mathbf{n}. \tag{2.29}$$

Therefore, the normalized specular BRDF is

$$f^0(\mathbf{i}, \mathbf{o}, \mathbf{n}) = \frac{\delta_{\mathbf{o}}(\mathbf{r}) F(\mathbf{i} \cdot \mathbf{n})}{\mathbf{i} \cdot \mathbf{n}}. \tag{2.30}$$

Another formulation of the specular BRDF is useful when the integration changes from light directions to half vectors

$$\mathbf{h} = \frac{\mathbf{i} + \mathbf{o}}{\|\mathbf{i} + \mathbf{o}\|}. \tag{2.31}$$

This formulation is convenient when working with microfacet BRDFs, as we will see in the next chapter. A similar identity to Equation (2.29) holds for the half vector:

$$\mathbf{i} \cdot \mathbf{h} = \mathbf{o} \cdot \mathbf{h}. \tag{2.32}$$

A specular reflection occurs when the light directions $\mathbf{i}$ and $\mathbf{o}$ are such that the half vector $\mathbf{h} = \mathbf{n}$. Therefore, we can use another delta function $\delta_{\mathbf{h}}$, that is associated with a differential solid angle around the half vector $\mathrm{d}\mathbf{h}$. To account for the change of measure we must calculate the Jacobian of the transformation from directions (vectors) to half vectors

$$\delta_{\mathbf{o}} \mathrm{d}\mathbf{o} = \delta_{\mathbf{h}} \mathrm{d}\mathbf{h} \tag{2.33}$$

$$\delta_{\mathbf{o}} = \frac{\mathrm{d}\mathbf{h}}{\mathrm{d}\mathbf{o}} \delta_{\mathbf{h}} \tag{2.34}$$

$$\delta_{\mathbf{o}} = J_{\mathbf{h}} \delta_{\mathbf{h}} \tag{2.35}$$

This Jacobian was derived algebraically by Stam [2001] and geometrically by Walter et al. [2007], and it is

$$J_{\mathbf{h}} = \frac{1}{4(\mathbf{i} \cdot \mathbf{n})}. \tag{2.36}$$

Finally, the specular BRDF associated with the half vector measure is

$$f^0(\mathbf{i}, \mathbf{o}, \mathbf{n}) = \frac{\delta_{\mathbf{h}}(\mathbf{n}) J_{\mathbf{h}} F(\mathbf{i} \cdot \mathbf{n})}{\mathbf{i} \cdot \mathbf{n}} = \frac{\delta_{\mathbf{h}}(\mathbf{n}) F(\mathbf{i} \cdot \mathbf{n})}{4(\mathbf{i} \cdot \mathbf{n})^2}. \tag{2.37}$$

Chapter 3

Microfacet Theory

In this chapter, we introduce the relevant concepts from microfacet theory. We begin with the standard microfacet statistics: the pair of microfacet distribution and shadowing-masking function that defines a geometrically valid microsurface. Follow the notion of *shape invariance* and its relation to the standard approach to extend microfacet distributions to anisotropy. Then we give the main definitions upon which we build our microfacet models in the following chapters — the microfacet BRDF and the combined BRDF, which is a microfacet BRDF modulated by a normal map.

The main notation related to microfacet theory is collected in Table 3.1. In Appendix C, we provide three analytic models for microsurfaces that are used in our work.

3.1 Microsurface statistics

In microfacet theory, the microsurface is defined statistically. This approach offers both computational efficiency and flexibility in terms of intuitive control parameters.

As mentioned earlier, from the standpoint of the microsurface, the macrosurface is assumed flat. In our exposition, all vectors are in the local frame of the macrosurface, and we assume its local normal as $\mathbf{n} = (0, 0, 1)^T$. A differential area $\mathrm{d}A$ on the macrosurface is projected along a direction $\mathbf{v} \in \mathcal{H}^2$ to differential area $(\mathbf{v} \cdot \mathbf{n})\mathrm{d}A$, Figure 3.1.

Each microsurface is associated with a profile that describes how its microstructure is organized. In this thesis, we adopt the state-of-the-art Smith [1967] microsurface profile, which assumes that the heights and the orientations of neighboring microfacets are not correlated since this profile is more realistic than the alternatives [Heitz, 2014].

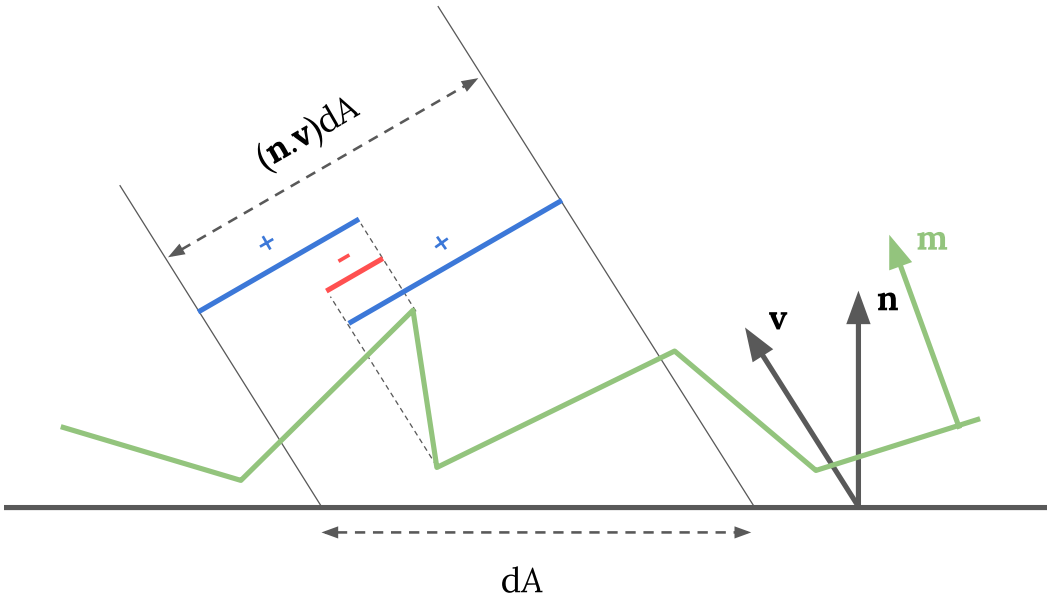
3.1.1 Microfacet distribution

The microsurface is statistically determined by its profile and *microfacet distribution function* [Walter et al., 2007, Heitz, 2014]

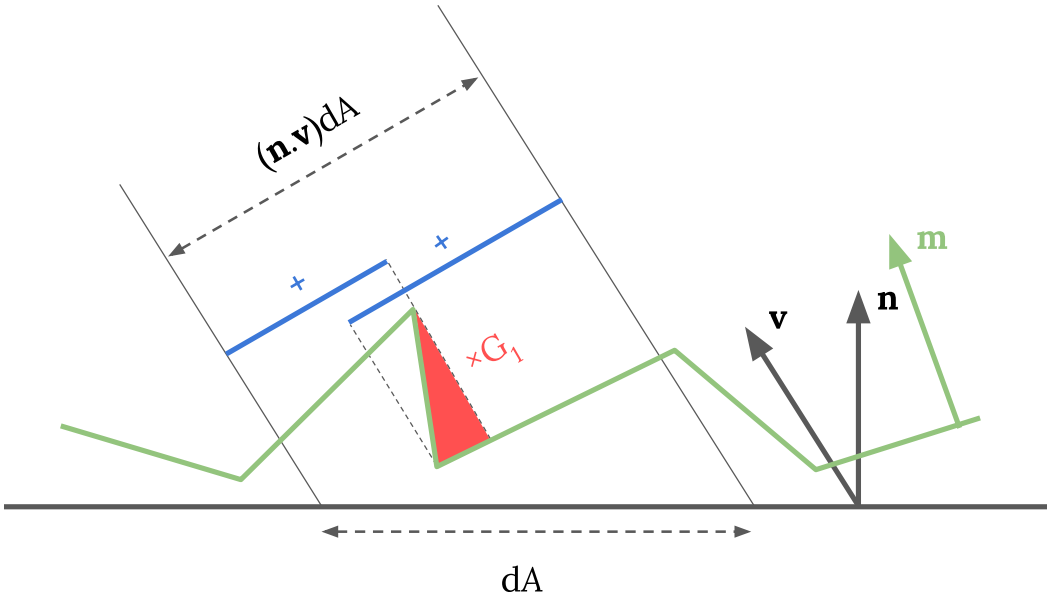
$$0 \leq D(\mathbf{m}) \leq \infty. \tag{3.1}$$

Microsurface-related symbols	
$\mathbf{m}$	Microsurface normal (micro-normal)
D	Microfacet distribution function
$D_{\mathbf{v}}$	Distribution of visible normals from direction $\mathbf{v}$
P_{22}	Slope distribution of the microsurface
S	Average visibility function
G_1	Monodirectional shadowing function
G	Shadowing-masking function
α	Roughness of the microfacet distribution
α_x, α_y	Roughness values along x and y for anisotropy
f_m	General Microfacet BRDF
f_μ	Micro-BRDF aligned with each microfacet
f^a	Microfacet BRDF with specular microfacets
Normal map-related symbols	
$\mathbf{t}_k$	Normal of k-th texel
T_k	Texture space region corresponding to k-th texel
N	Total number of normal map texels
$\|X\|$	Surface area of the region X
I_X	Indicator function of the region X
D_x	Normal distribution function (NDF) of the normal map
G_x	Shadowing-masking function of the normal map
C	Texel contribution function
f_x^a	Combined BRDF: microfacet BRDF modulated by a normal map

Table 3.1: Notation for microfacet BRDFs.



a) Constraint on D, Equation (3.4)



b) Constraint on G_1, Equation (3.9)

Figure 3.1: **The microfacet distribution and its shadowing function.** A differential area dA on the macrosurface with normal $\mathbf{n}$. The projection of this area along direction $\mathbf{v} \in \mathcal{H}^2$ is $(\mathbf{n} \cdot \mathbf{v})\mathrm{d}A$. The microsurface (green) is continuous and it is composed of microfacets with micro-normals $\mathbf{m}$. Microsurface statistics are subject to constraints. **a) Microfacet distribution** D: The projected areas along $\mathbf{v}$ of all forward-facing microfacets (blue) minus the projected areas of all backward-facing microfacets must be equal to the projected area of the macrosurface $(\mathbf{n} \cdot \mathbf{v})\mathrm{d}A$. **b) Shadowing function** G_1: When the total projected area along $\mathbf{v}$ of forward-facing microfacets (blue) is greater than the projected area of the macrosurface $(\mathbf{n} \cdot \mathbf{v})\mathrm{d}A$, the shadowing factor G_1 must diminish it to compensate for the shadowed regions (red).

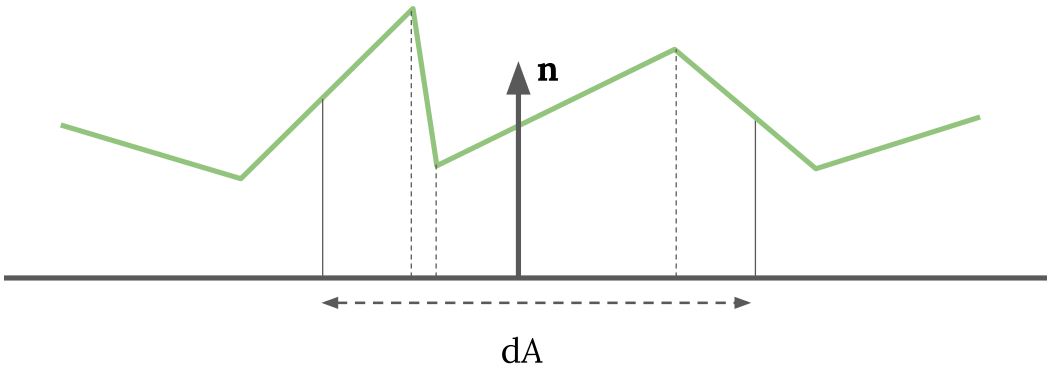


Figure 3.2: **Normalized microfacet distributions.** The projections of all microfacets onto the macrosurface fully cover the macrosurface and do not overlap with each other, see Equation (3.5).

Note that this definition includes delta distributions. For a given micro-normal $\mathbf{m} \in \mathcal{H}^2$, $D(\mathbf{m})$ evaluates to the differential area of the microsurface oriented with $\mathbf{m}$. That is why in some older writings the microfacets are referred to as *microareas*, and the microfacet distribution — *microarea distribution* [Trowbridge and Reitz, 1975]. Rigorously, for a macrosurface differential area $\mathrm{d}A$ and a differential solid angle around the micro-normal $\mathrm{d}\mathbf{m}$, the total differential area of the microsurface whose micro-normals are contained in $\mathrm{d}\mathbf{m}$ is $D(\mathbf{m})\mathrm{d}\mathbf{m}\mathrm{d}A$ [Walter et al., 2007].

In a valid configuration, the differential area of a non-flat microsurface is never less than the differential macrosurface area

$$\int_{\mathcal{H}^2} D(\mathbf{m})\mathrm{d}\mathbf{m}\mathrm{d}A \geq \mathrm{d}A \tag{3.2}$$

and the equality is reached when the microsurface is flat, thus coinciding with the macrosurface. This means that the total microsurface area is never less than the total macrosurface area, which leads to a weak constraint on the distribution

$$\int_{\mathcal{H}^2} D(\mathbf{m})\mathrm{d}\mathbf{m} \geq 1. \tag{3.3}$$

Notice that the integration is over differential solid angles, and therefore the microfacet distribution is measured in inverse steradians $[sr^{-1}]$.

Furthermore, the signed projected area of the microsurface must be equal to the projected area of the macrosurface along an arbitrary direction $\mathbf{v} \in \mathcal{H}^2$ [Walter et al., 2007], Figure 3.1 a). This property enforces a normalization constraint on the microfacet distribution:

$$\int_{\mathcal{H}^2} D(\mathbf{m})(\mathbf{v} \cdot \mathbf{m})\mathrm{d}\mathbf{m} = (\mathbf{v} \cdot \mathbf{n}). \tag{3.4}$$

It is sufficient to ensure that the projected microfacets along the macrosurface normal do not overlap and perfectly cover the macrosurface, see Figure 3.2. Hence, Equation (3.4) is substituted with $\mathbf{v} := \mathbf{n}$:

$$\int_{\mathcal{H}^2} D(\mathbf{m})(\mathbf{n} \cdot \mathbf{m})\mathrm{d}\mathbf{m} = 1. \tag{3.5}$$

This simplified constraint can be used to find normalization factors for microfacet distributions. In Chapter 4, we use it to validate our linearly transformed microfacet distributions.

Finally, any microfacet distribution can be converted to a distribution of slopes

$$P_{22}\left(-\frac{m_x}{m_z}, -\frac{m_y}{m_z}\right) = D(\mathbf{m})\cos^4\theta_{\mathbf{m}}, \tag{3.6}$$

where $-\frac{m_x}{m_z}$ and $-\frac{m_y}{m_z}$ are the slopes of a microfacet with micro-normal $\mathbf{m} = (m_x, m_y, m_z)^T$. The factor $\cos^3\theta_{\mathbf{m}}$ is for the change of measure from solid angle to slopes and the factor $\cos\theta_{\mathbf{m}}$ is for the projection of the microfacets onto the macrosurface [Walter et al., 2007, Heitz, 2014].

3.1.2 Microsurface self-shadowing

Not all microfacets are visible from all directions. In fact, all microfacets are guaranteed to be visible only along the macrosurface normal $\mathbf{n}$, see Figure 3.2. To conduct reflectance measurements, the self-shadowing of the microsurface needs to be taken into account. This is facilitated by a geometric factor derived from $D(\mathbf{m})$ — the monodirectional shadowing

$$0 \leq G_1(\mathbf{v}, \mathbf{m}) \leq 1 \tag{3.7}$$

yields the fraction of the microfacets with normals $\mathbf{m} \in \mathcal{H}^2$ that are visible from a specified direction $\mathbf{v} \in \mathcal{H}^2$, Figure 3.1 b). For microsurfaces with Smith profile the shadowing function G_1 is represented by the average visibility function $S(\mu)$:

$$G_1(\mathbf{v}, \mathbf{m}) = \begin{cases} S(\mu), & \mathbf{v}\cdot\mathbf{m} > 0 \\ 0, & \text{otherwise} \end{cases}, \tag{3.8}$$

where $\mu = \cot\theta_{\mathbf{v}}$ and S is the fraction of the microsurface that is visible from direction $\mathbf{v}$ [Walter et al., 2007]. G_1 is subject to a normalization constraint [Smith, 1967, Ashikmin et al., 2000, Heitz, 2014]:

$$\int_{\mathcal{H}^2} D(\mathbf{m})G_1(\mathbf{v}, \mathbf{m})(\mathbf{v}\cdot\mathbf{m})\mathrm{d}\mathbf{m} = (\mathbf{v}\cdot\mathbf{n}). \tag{3.9}$$

Notice the similarity between the two constraints: the negative projected area from the back-facing microfacets in Equation (3.4) $(\mathbf{v}\cdot\mathbf{m}) \leq 0$ is clamped in Equation (3.9) because $G_1 = 0$, and instead the shadowing for the forward-facing microfacets is compensated by the S term, see Equation (3.8). This resemblance is illustrated with the two diagrams in Figure 3.1. In Chapter 4, we use Equation (3.9) to validate the shadowing functions of our linearly transformed microsurfaces.

The monodirectional shadowing G_1 is also used in the definition of the *distribution of visible normals* [Heitz, 2014, Heitz and D'Eon, 2014]

$$D_{\mathbf{v}}(\mathbf{m}) = \frac{D(\mathbf{m})G_1(\mathbf{v}, \mathbf{m})(\mathbf{v}\cdot\mathbf{m})}{\mathbf{v}\cdot\mathbf{n}}. \tag{3.10}$$

This is a view-dependent distribution: for a given direction $\mathbf{v}$ it is the distribution of all microfacets that are visible from $\mathbf{v}$, i.e. $\mathbf{v} \cdot \mathbf{m} > 0$. $D_{\mathbf{v}}$ is based on Equation (3.9) and its normalization is a direct consequence of this equation

$$\int_{\mathcal{H}^2} D_{\mathbf{v}}(\mathbf{m}) \mathrm{d}\mathbf{m} = \frac{1}{\mathbf{v} \cdot \mathbf{n}} \int_{\mathcal{H}^2} D(\mathbf{m}) G_1(\mathbf{v}, \mathbf{m})(\mathbf{v} \cdot \mathbf{m}) \mathrm{d}\mathbf{m} = \frac{\mathbf{v} \cdot \mathbf{n}}{\mathbf{v} \cdot \mathbf{n}} = 1. \tag{3.11}$$

At the end of this chapter, we will see that $D_{\mathbf{v}}$ is important for the efficient sampling of microfacet BRDFs.

Sometimes, the microfacet distribution is abbreviated *NDF (Normal Distribution Function)* and the distribution of visible normals — *VNDF (Visible Normal Distribution Function)*.

Microfacet BRDFs require a tool to evaluate what portion of microfacets with normal $\mathbf{m}$ are seen from both the incoming and outgoing light directions $\mathbf{i}$ and $\mathbf{o}$. This is achieved by the shadowing-masking function

$$0 \leq G(\mathbf{o}, \mathbf{i}, \mathbf{m}) \leq 1 \tag{3.12}$$

which can be approximated using the shadowing G_1. The simplest variant is the separable shadowing-masking function [Walter et al., 2007]

$$G(\mathbf{o}, \mathbf{i}, \mathbf{m}) = G_1(\mathbf{o}, \mathbf{m}) G_1(\mathbf{i}, \mathbf{m}). \tag{3.13}$$

A more accurate and nearly as efficient variant is the height-correlated shadowing-masking function [Ross et al., 2005, Heitz, 2014]

$$G(\mathbf{o}, \mathbf{i}, \mathbf{m}) = \frac{G_1(\mathbf{o}, \mathbf{m}) G_1(\mathbf{i}, \mathbf{m})}{G_1(\mathbf{o}, \mathbf{m}) + G_1(\mathbf{i}, \mathbf{m}) - G_1(\mathbf{o}, \mathbf{m}) G_1(\mathbf{i}, \mathbf{m})}. \tag{3.14}$$

For a detailed derivation of shadowing-masking functions for microfacet surfaces we refer the reader to the excellent material by Heitz [2014].

3.1.3 Elliptical anisotropy

The classical anisotropy approach and the shape invariance property presented here are based on Heitz's article [2014].

Commonly, microfacet distributions $D(\mathbf{m})$ have a roughness parameter to control the spread of the distribution, which is denoted by α. Traditionally, isotropic distributions are extended to anisotropy by varying the roughness in the xy tangential plane. This is done by specifying roughness α_x along x axis and α_y along y axis and replacing the isotropic argument $\frac{1}{\alpha^2}$ with the ellipse

$$\frac{\cos^2 \phi_{\mathbf{m}}}{\alpha_x^2} + \frac{\sin^2 \phi_{\mathbf{m}}}{\alpha_y^2}. \tag{3.15}$$

This leads to a distribution that is not properly normalized. A normalization factor can be computed if Equation (3.5) can be integrated into a closed-form expression. This is not always possible — for example, the GTR distribution does not have a closed-form normalization except for the special case of the GGX distribution [Burley, 2012].

3.1.4 Shape invariance

Shape-invariant isotropic distributions have the property that their shape does not change when the roughness changes. They are an important sub-class of distributions: for all roughness values, the distribution curves are scaled copies of each other. Such distributions take the form:

$$D(\mathbf{m}) = \frac{1}{(\mathbf{m}\cdot\mathbf{n})^4}\frac{1}{\alpha^2} f\left(\frac{\tan\theta_{\mathbf{m}}}{\alpha}\right), \tag{3.16}$$

where the function f is a 1D distribution. The anisotropic version based on Equation (3.15) is

$$D(\mathbf{m}) = \frac{1}{(\mathbf{m}\cdot\mathbf{n})^4}\frac{1}{\alpha_x\alpha_y} f\left(\tan\theta_{\mathbf{m}}\sqrt{\frac{\cos^2\phi_{\mathbf{m}}}{\alpha_x^2}+\frac{\sin^2\phi_{\mathbf{m}}}{\alpha_y^2}}\right). \tag{3.17}$$

To complete the anisotropic microsurface configuration, the shadowing function $G_1(\mathbf{v},\mathbf{m})$ must be evaluated for the projected roughness onto the direction $\mathbf{v}$:

$$\alpha_{\mathbf{v}} = \sqrt{\cos^2\phi_{\mathbf{v}}\alpha_x^2 + \sin^2\phi_{\mathbf{v}}\alpha_y^2}. \tag{3.18}$$

3.2 Microfacet BRDFs

We define the BRDF that corresponds to the microsurface, which is based on the statistics from the previous section. In its general form, the microfacet BRDF is

$$f_m(\mathbf{i},\mathbf{o},\mathbf{n}) = \int_{\mathcal{H}^2} \frac{(\mathbf{i}\cdot\mathbf{m})(\mathbf{o}\cdot\mathbf{m})}{(\mathbf{i}\cdot\mathbf{n})(\mathbf{o}\cdot\mathbf{n})} f_\mu(\mathbf{i},\mathbf{o},\mathbf{m}) D(\mathbf{m}) G(\mathbf{i},\mathbf{o},\mathbf{m}) \mathrm{d}\mathbf{m}, \tag{3.19}$$

where f_μ is the micro-BRDF — a BRDF associated with each microfacet, aligned with its micro-normal $\mathbf{m}$. A derivation is provided by Heitz [2014], however, in contrast to his formulation, our formulation in Equation (3.19) assumes the upper hemisphere of the normal $\mathbf{n}$. Additionally, the cases in which one or the two light directions are in the opposite hemisphere of the microfacet normal $\mathbf{m}$, i.e. $(\mathbf{i}\cdot\mathbf{m}) < 0$ and/or $(\mathbf{o}\cdot\mathbf{m}) < 0$, are implicitly zeroed out by the shadowing term G, see Equations (3.14) and (3.8).

When all microfacets are aligned with the macrosurface normal $\mathbf{n}$, then the microfacet distribution $D(\mathbf{m})$ is the delta distribution $\delta_{\mathbf{n}}(\mathbf{m})$, and all microfacets are visible from all directions, i.e. $G(\mathbf{i},\mathbf{o},\mathbf{m}) = 1$, thus

$$f_m(\mathbf{i},\mathbf{o},\mathbf{n}) = \int_{\mathcal{H}^2} \frac{(\mathbf{i}\cdot\mathbf{m})(\mathbf{o}\cdot\mathbf{m})}{(\mathbf{i}\cdot\mathbf{n})(\mathbf{o}\cdot\mathbf{n})} f_\mu(\mathbf{i},\mathbf{o},\mathbf{m}) \delta_{\mathbf{n}}(\mathbf{m}) \mathrm{d}\mathbf{m} = f_\mu(\mathbf{i},\mathbf{o},\mathbf{n}). \tag{3.20}$$

In this special case, the microfacet BRDF is reduced to the micro-BRDF.

The two most common choices for the micro-BRDF f_μ are the Lambert BRDF and the specular BRDF from Equations (2.13) and (2.37), respectively. In the case of Lambertian microfacets, Equation (3.19) cannot be integrated into a closed-form expression. Oren and Nayar [1994] demonstrated that the Lambertian BRDF is a poor representation for materials like clay, plaster, sand and cloth because these materials tend to reflect more light as the outgoing light direction $\mathbf{o}$

approaches the incoming light direction $\mathbf{i}$. They proposed a fit for Equation (3.19) with Lambertian microfacets and Gaussian microfacet distribution that is more adequate at approximating such materials. Heitz and Dupuy [2015] proposed a stochastic evaluation technique for Equation (3.19) with Lambert micro-BRDF and demonstrated it for anisotropic microsurfaces.

3.2.1 Specular microfacets

The most widely used microfacet model is the one with specular microfacets. It is capable of representing a large range of appearances, and it does have a closed-form formula. It is derived by substituting the specular BRDF from Equation (2.37) in Equation (3.19):

$$f_m(\mathbf{i},\mathbf{o},\mathbf{n}) = \int_{\mathcal{H}^2} \frac{(\mathbf{i}\cdot\mathbf{m})(\mathbf{o}\cdot\mathbf{m})}{(\mathbf{i}\cdot\mathbf{n})(\mathbf{o}\cdot\mathbf{n})} f^0(\mathbf{i},\mathbf{o},\mathbf{m}) D(\mathbf{m}) G(\mathbf{i},\mathbf{o},\mathbf{m}) \mathrm{d}\mathbf{m} \tag{3.21}$$

$$= \int_{\mathcal{H}^2} \frac{(\mathbf{i}\cdot\mathbf{m})(\mathbf{o}\cdot\mathbf{m})}{(\mathbf{i}\cdot\mathbf{n})(\mathbf{o}\cdot\mathbf{n})} \frac{\delta_{\mathbf{h}}(\mathbf{m}) F(\mathbf{i}\cdot\mathbf{m})}{4(\mathbf{i}\cdot\mathbf{m})^2} D(\mathbf{m}) G(\mathbf{i},\mathbf{o},\mathbf{m}) \mathrm{d}\mathbf{m} \tag{3.22}$$

$$= \frac{(\mathbf{i}\cdot\mathbf{h})(\mathbf{o}\cdot\mathbf{h}) F(\mathbf{i}\cdot\mathbf{h}) D(\mathbf{h}) G(\mathbf{i},\mathbf{o},\mathbf{h})}{(\mathbf{i}\cdot\mathbf{n})(\mathbf{o}\cdot\mathbf{n}) 4(\mathbf{i}\cdot\mathbf{h})^2}. \tag{3.23}$$

This result can further be simplified since $(\mathbf{i}\cdot\mathbf{h}) = (\mathbf{o}\cdot\mathbf{h})$, thus we reach the classical formulation of the microfacet BRDF with specular microfacets

$$f^\alpha(\mathbf{i},\mathbf{o},\mathbf{n}) = \frac{F(\mathbf{i}\cdot\mathbf{h}) D^\alpha(\mathbf{h},\mathbf{n}) G^\alpha(\mathbf{i},\mathbf{o},\mathbf{h},\mathbf{n})}{4(\mathbf{i}\cdot\mathbf{n})(\mathbf{o}\cdot\mathbf{n})}. \tag{3.24}$$

Commonly, the microfacet distribution D and the shadowing-masking function G depend implicitly on the macrosurface normal $\mathbf{n}$ and the roughness parameter α — note that all angles are defined with respect to the normal $\mathbf{n}$. For clarity of our derivation, we make both dependencies explicit.

Importantly, the microfacet BRDF from Equation (3.24) approaches the specular BRDF as the roughness diminishes

$$\lim_{\alpha\to 0} f^\alpha = \frac{F(\mathbf{i}\cdot\mathbf{h}) D^\alpha(\mathbf{h},\mathbf{n}) G^\alpha(\mathbf{i},\mathbf{o},\mathbf{h},\mathbf{n})}{4(\mathbf{i}\cdot\mathbf{n})(\mathbf{o}\cdot\mathbf{n})} = \frac{F(\mathbf{i},\mathbf{h})\delta_{\mathbf{h}}(\mathbf{n})}{4(\mathbf{i}\cdot\mathbf{h})^2} = f^0. \tag{3.25}$$

This is a direct consequence from Equation (3.20) — as the roughness is approaching zero, the microfacet distribution becomes a Dirac delta distribution, and the expression is non-zero only when $\mathbf{h} = \mathbf{n}$. This observation motivates the notation for f^α and f^0.

3.2.2 Normal mapping

The normal map is a collection of N texels, each occupying an equal rectangular region T_k of unit texture space, and it is associated with a normal in tangent space $\mathbf{t}_k \in \mathcal{H}^2$. Alternatively, the normals can be defined as points on the unit disk $\mathcal{D}$, see Appendix B.

The normal distribution function (NDF) based on the normal map is

$$D_x(\mathbf{m}) = \sum_{k=1}^{N} \delta_{\mathbf{t}_k}(\mathbf{m}) I_{T_k}(x), \tag{3.26}$$

which is dependent on the texture space position x and I_{T_k} is the indicator function of the texel region T_k. Our definition is equivalent to the one in Han et al. [Han et al., 2007], however, we do not divide explicitly by N, because in our definition the texel area is $|T_k| = \frac{1}{N}$.

The NDF D_x is a well-defined microfacet distribution, and its corresponding microsurface has a shadowing-masking function G_x. In theory, G_x depends on the normal map, however, precise expressions are not available. In related work, commonly a Smith shadowing-masking function corresponding to a smooth microfacet distribution is used [Yan et al., 2014, Jakob et al., 2014, Yan et al., 2016].

3.2.3 Combined BRDF

To use microfacet BRDFs and normal maps together in a way that is in agreement with assumptions of microfacet theory we follow a derivation similar to Schüssler et al. [2017]. In essence, the normal map provides the visible features of the surface, where its texels are the larger microfacets. On the other hand, each of these facets could be a rough surface in its own right, which is described by a microfacet BRDF with specular microfacets. Formally, in Equation (3.19) we substitute the normal map (D_x, G_x) for the microsurface and f^α for the micro-BRDF f_μ

$$\begin{aligned} f_x^\alpha(\mathbf{i},\mathbf{o},\mathbf{n}) &= \int_{\mathcal{H}^2} \frac{(\mathbf{i}\cdot\mathbf{m})(\mathbf{o}\cdot\mathbf{m})}{(\mathbf{i}\cdot\mathbf{n})(\mathbf{o}\cdot\mathbf{n})} f^\alpha(\mathbf{i},\mathbf{o},\mathbf{m}) D_x(\mathbf{m}) G_x(\mathbf{i},\mathbf{o},\mathbf{m}) \mathrm{d}\mathbf{m} \\ &= \sum_{k=1}^{N} \frac{(\mathbf{i}\cdot\mathbf{t}_k)(\mathbf{o}\cdot\mathbf{t}_k)}{(\mathbf{i}\cdot\mathbf{n})(\mathbf{o}\cdot\mathbf{n})} f^\alpha(\mathbf{i},\mathbf{o},\mathbf{t}_k) G_x(\mathbf{i},\mathbf{o},\mathbf{t}_k) I_{T_k}(x), \end{aligned} \tag{3.27}$$

where $\mathbf{n}$ is the surface normal at position x. The delta function in the definition of D_x breaks the integral into a sum over all texel normals $\mathbf{t}_k$. Furthermore, there is only one term for which $I_{T_k}(x)$ is non-zero — this is the term for which $x \in T_k$. When we expand f^α in this term we reach the desired combined BRDF of a normal map and a microfacet BRDF

$$\begin{aligned} f_x^\alpha(\mathbf{i},\mathbf{o},\mathbf{n}) &= \frac{(\mathbf{i}\cdot\mathbf{t}_k)(\mathbf{o}\cdot\mathbf{t}_k)}{(\mathbf{i}\cdot\mathbf{n})(\mathbf{o}\cdot\mathbf{n})} \frac{F(\mathbf{i},\mathbf{h}) D^\alpha(\mathbf{h},\mathbf{t}_k) G^\alpha(\mathbf{i},\mathbf{o},\mathbf{h},\mathbf{t}_k)}{4(\mathbf{i}\cdot\mathbf{t}_k)(\mathbf{o}\cdot\mathbf{t}_k)} G_x(\mathbf{i},\mathbf{o},\mathbf{t}_k) \\ &= \frac{F(\mathbf{i},\mathbf{h}) D^\alpha(\mathbf{h},\mathbf{t}_k) G^\alpha(\mathbf{i},\mathbf{o},\mathbf{h},\mathbf{t}_k)}{4(\mathbf{i}\cdot\mathbf{n})(\mathbf{o}\cdot\mathbf{n})} G_x(\mathbf{i},\mathbf{o},\mathbf{t}_k) \\ &= F(\mathbf{i},\mathbf{h}) C(\mathbf{i},\mathbf{o},\mathbf{t}_k,\mathbf{n}), \end{aligned} \tag{3.28}$$

where

$$C(\mathbf{i},\mathbf{o},\mathbf{t}_k,\mathbf{n}) = \frac{D^\alpha(\mathbf{h},\mathbf{t}_k) G^\alpha(\mathbf{i},\mathbf{o},\mathbf{h},\mathbf{t}_k) G_x(\mathbf{i},\mathbf{o},\mathbf{t}_k)}{4(\mathbf{i}\cdot\mathbf{n})(\mathbf{o}\cdot\mathbf{n})} \tag{3.29}$$

is the texel contribution.

An illustration of the microsurfaces defined by Equation (3.28) is given in Figure 3.3. When the combined BRDF has roughness zero, i.e. f_x^0, this BRDF is referred to as a *specular normal map*.

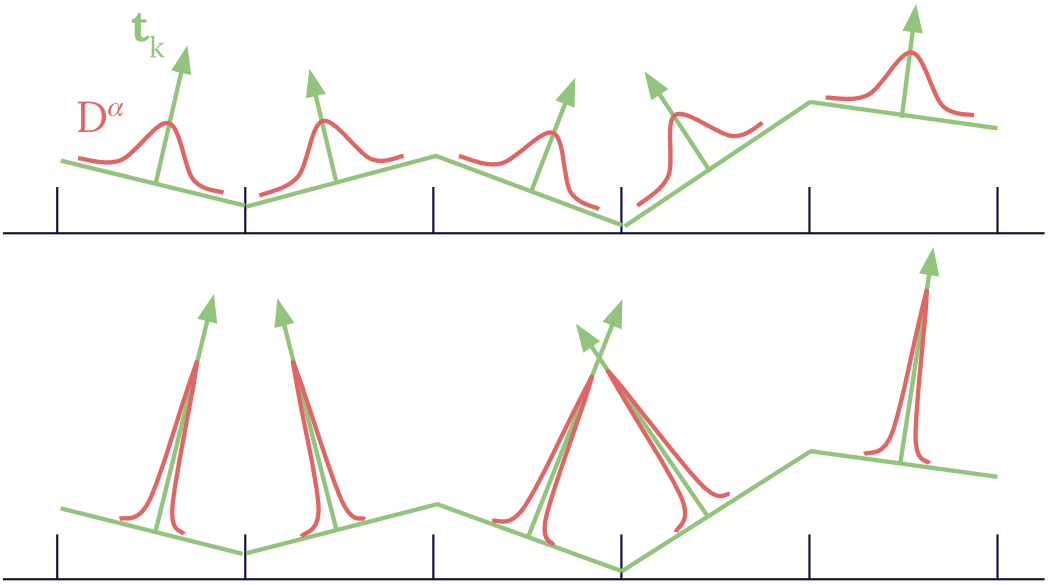


Figure 3.3: **Microsurfaces corresponding to the combined BRDF f_x^α.** The normal map texels (green) define the coarse visible detail of the microsurface while the microfacet distribution associated with each texel (red) represents the roughness of each texel. **Top row:** The normal map texels have higher roughness α, thus the microfacet distribution D^α is wider. **Bottom row:** The microsurface is defined using the same normal map as the one in the top row, however, the roughness of individual texels is lower.

3.2.4 Sampling microfacet BRDFs

The perfect sampling strategy for the microfacet BRDF f_m (Equation (3.19)) would sample directions that are proportional to the product $f_m(\mathbf{i}, \mathbf{o}, \mathbf{n})(\mathbf{i} \cdot \mathbf{n})$. For more details refer to the discussion about the reflection equation in Section 2.1.3.

Furthermore, analytic sampling equations are preferred due to them being compact, accurate and fast. Usually, they are derived using the *inversion method* [Pharr et al., 2016]: first, the *CDF (cumulative density function)* is computed from the *PDF (probability density function)*, and then it is inverted. Uniform random variables that are transformed using the inverse CDF are distributed according to the PDF. Note that this derivation is possible when the inverse CDF can be expressed in closed form. One alternative to analytic sampling is building CDF tables. This is cumbersome because their dimension depends on the number of parameters that parametrize the microfacet distribution, and also requires binary searching for inverting the CDF. It is noteworthy that analytic sampling techniques are not restricted to the inversion method [Heitz, 2020], although it is the most straightforward and common approach.

Unfortunately, analytic sampling equations are available neither for the cosine weighted general microfacet BRDF $f_m(\mathbf{i}, \mathbf{o}, \mathbf{n})(\mathbf{i} \cdot \mathbf{n})$ nor just for the special case of $f^\alpha(\mathbf{i}, \mathbf{o}, \mathbf{n})$ from Equation (3.24), even for the simplest microfacet distributions. Therefore, the standard approach is to sample a microsurface normal $\mathbf{m}'$ and then to sample the micro-BRDF f_μ that is centered at $\mathbf{m}'$. In the most common case of specular microfacets (Equation (3.24)), the sampled direction $\mathbf{r}'$ is computed by reflecting the outgoing light direction $\mathbf{o}$ from the sampled microfacet $\mathbf{m}'$, see

Equation (2.20).

Since all analytic microfacet distributions require that the projected microsurface area $D(\mathbf{m})(\mathbf{m} \cdot \mathbf{n})$ is normalized, see Equation (3.5), their CDF is available. Additionally, for all common distributions, the CDF can be inverted and thus equations for the sampling of micro-normals, proportional to the projected microsurface area

$$\mathbf{m}' \propto D(\mathbf{m})(\mathbf{m} \cdot \mathbf{n}) \tag{3.30}$$

are available [Walter et al., 2007, Burley, 2012, Ribardière et al., 2017].

The sampling strategy that employs Equation (3.30) is far from ideal. It has the following issues:

- Low input directions ($\mathbf{o}$) could often be below the sampled micro-normal $\mathbf{m}'$ ($\mathbf{o} \cdot \mathbf{m}' < 0$) and therefore the sample must be rejected.
- When high input directions ($\mathbf{o}$) are reflected from tail microfacets ($\mathbf{m}' \cdot \mathbf{n} \ll 1$), they often end up in the negative hemisphere. The sample is rejected.
- Reflected directions $\mathbf{r}'$ with large incident angles ($\mathbf{r}' \cdot \mathbf{n} \ll 1$) have much smaller contribution due to the shadowing $G_1(\mathbf{r}', \mathbf{m}')$, but they are not sampled with the corresponding lower probability.
- The Fresnel term is not taken into account.

Sampling from the distribution of normals that are visible from direction $\mathbf{o}$ — $D_{\mathbf{o}}$ (Equation (3.10)) solves only the first issue, but it is already a significant improvement [Heitz and D'Eon, 2014]. Unfortunately, equations for sampling $D_{\mathbf{o}}$ are available only for Beckmann and GGX distributions [Jakob, 2014, Heitz, 2018].

Chapter 4

Linearly Transformed Microsurface

4.1 Introduction

Microfacet models that can be extended to anisotropy are valuable in practical applications, as this functionality adds substantial additional control over the appearance of the shaded object, besides the basic parameters of the microfacet distribution function. Many real-world objects exhibit anisotropic reflections [Kajiya, 1985], so models that are capable of describing their appearance are needed.

The current state of the art for creating analytic anisotropic distributions from existing analytic isotropic ones is based on varying the roughness of the isotropic distribution in an ellipse. This idea can be traced back at least to the elliptical Gaussian model of Ward [1992]. It was later adapted to microfacet models [Ashikmin et al., 2000, Kurt et al., 2010] and has been used since then in modern rendering systems [Nimier-David et al., 2019, Pharr et al., 2016, Burley, 2012]. Heitz [2014] defined the class of shape-invariant isotropic distributions for which this construction is always possible. Specifically, this class of distributions has considerable practical advantages:

- Derivation of normalized anisotropic distributions by applying different roughness values along x and y axes. For shape-invariant distributions, this is equivalent to a non-uniform linear scaling of the distribution.
- Extension of the 1D isotropic shadowing function to anisotropic without increasing its dimension.
- Extension of the isotropic sampling equations (including distribution of visible normals, if available) to anisotropy.

Our work generalizes all of these advantages to all microfacet distributions and all 2D invertible transformations, beyond simple scaling. We investigate how the microfacet distribution and the shadowing function of an existing microsurface change when the microsurface is linearly transformed in the tangent plane.

The related previous work is presented in the next section. Our method is derived in Section 4.3: we first investigate which linear transformations of the microsurface result in valid microsurfaces, then we show how to construct the statistics of the transformed microsurfaces and finally we make the connection

of our technique with the classical anisotropy for shape-invariant distributions. In Section 4.4, we demonstrate our technique by constructing anisotropic modifications of shape-variant distributions. To enable the visualization of deforming objects, we show how to track the local deformation of an object using a tension texture.

4.2 Previous work

Kajiya [1985] derived anisotropic reflection models from the equations of electromagnetism. He pointed out that statistical models are desirable, but the anisotropic integrals are "extremely complex". Poulin and Fournier [1990] proposed an anisotropic model based on sampling microscopic cylinders layered onto the surface. Ashikmin et al. [2000] introduced various anisotropic models to microfacet theory and developed matching shadowing equations.

Nagano et al. [2015] studied the complex appearance of human skin under deformation. In addition to the mesostructure detail represented by scanned normal maps, they showed that microstructure dynamics play a vital role in the overall appearance. Measurement of skin samples indicated that stretched skin appears shinier while compressed skin has a rougher look. They incorporated this measurement data into their shader by blurring the stretched portions of the surface and sharpening the compressed ones. From an appearance point of view, this aspect is similar to our work, although we present a mathematical framework for microfacet surfaces that is not based on measurements.

Dupuy et al. [2013] presented a filtering technique for displaced surfaces based on non-centered anisotropic Beckmann distributions. It is a multi-scale representation for mapped surfaces that, similarly to our work, can predict scaled surface appearance. However, our work is focused on microsurface-related behavior for all distributions while their work is targeted at filtering Beckmann surfaces. Non-centered distributions can be used to modulate an arbitrary microsurface by a normal or displacement map, and they can be applied to our model as well. The shifted distribution remains normalized and only the shadowing function requires alteration [Heitz, 2014].

There are several approaches for filtering reflectance from microstructures [Han et al., 2007, Wu et al., 2011, Jakob et al., 2014, Wu et al., 2019], including a large amount of work for rendering glints following the framework of Yan et al. [2014]. A model capable of a broad spectrum of appearances based on tabulated piecewise linear distributions was presented by Ribardière et al. [2019]. Our work is targeted at modifying existing distributions and as such is orthogonal to these methods.

GGX is probably the microfacet distribution that is currently most widely used in practice, as it has been consistently verified to be a better all-around match for acquired data [Trowbridge and Reitz, 1975, Blinn, 1977, Walter et al., 2007, Dupuy et al., 2015] than the Beckmann distribution: the main reason for this seems to be the long tails of the distribution. However, many real materials have even longer tails, and therefore Disney developed the Generalized Trowbridge-Reitz (GTR) distribution as a natural extension to the GGX distribution [Burley, 2012]: this exposes an exponent γ that provides additional control of the tail. This is especially attractive for exponents less than two, which

correspond to heavier tails than those of GGX. For example, a GTR exponent $\gamma = \frac{3}{2}$ results in the Henyey-Greenstein distribution [1941] evaluated at half-angles, while the exponent of $\gamma = 1$ corresponds to the Berry distribution [1923] which is employed in the coating layer of Disney's material [Burley, 2012]. We are not aware of other tail-controllable distributions that are used in the industry. Interestingly, the GTR distribution is shape-variant for all tail exponents $\gamma \neq 2$, and the elliptical anisotropic formulation cannot be integrated analytically in this case [Burley, 2012]. This makes it a practical use case to demonstrate our technique.

Löw et al. [2012] introduce isotropic models suited for glossy reflections that allow control of the tail of the distribution. Work has been done to develop shape-invariant distributions that admit tail control. The STD [Ribardière et al., 2017] was proposed to span Beckmann and GGX distributions although the heavier than GGX tail control was limited. The Hyper-Cauchy distribution was utilized in the optics literature [Wellems et al., 2006] and was later used for BRDF fitting [Butler and Marciniak, 2014].

Ribardière et al. [2019] presented a hierarchical classification of analytic distributions that generalize Beckmann and GGX distributions until they reach the most general Skewed Generalized T-Distribution (SGTD). They note that only the distributions in the lower levels of this hierarchy are shape-invariant, and therefore are guaranteed to have an anisotropic form. With our framework, an anisotropic version of any SGTD can easily be constructed. Notably, the GTR distribution is not part of this classification.

Recently, Barla et al. [2018] presented a technique that combines two BRDFs to provide more control on the tails of the distribution. Similarly, our technique modifies existing BRDFs and can be employed on top of their technique, if their input BRDFs are microfacet-based.

It has been shown that genetic programming could be used to find new analytic BRDFs, including microfacet ones that provide better fits to acquired data compared to existing microfacet BRDFs [Brady et al., 2014]. All of their microfacet distributions are isotropic, and it is unlikely for their technique to generate shape-invariant distributions. Our framework can directly extend them to anisotropy.

Heitz et al. [2016a] introduced Linearly Transformed Spherical Distributions (LTSD) and derived a closed-form formula for them. However, their formula does not lead to normalized microfacet distributions, while our framework for linearly transformed microsurfaces ensures this. In Section 4.3, we derive the Jacobian that ensures the normalization of the transformed microfacet distributions, and then we discuss the differences to the LTSD formulation.

Finally, there are examples of shape-variant distributions used in the industry, like Phong [Walter et al., 2007], Sheen [Estevez and Kulla, 2017], discrete GGX and Beckmann [Atanasov and Koylazov, 2016], and GTR [Burley, 2012].

4.3 Our framework

We take a different approach to altering microfacet distributions. By investigating how linear transformations of the microsurface affect its statistics we arrive

at a simple and general framework for applying modifications to all valid microsurfaces. Table 4.1 shows the respective notation.

M	Microsurface transformation matrix
D_M	Microfacet distribution of the transformed surface
$D_{\mathbf{v}M}$	Visible normals distribution for the transformed surface
G_{1M}	Shadowing function of the transformed surface
P_{22M}	Slope distribution of the transformed surface
$\mathbf{u}$	Transformed and normalized micro-normal
N	Applies linear transformation and normalizes the result
J_N	The Jacobian matrix of the nonlinear transformation N
$\mathbf{m}'$	Sampled micro-normal
$\mathbf{m}'_\mathbf{v}$	Sampled micro-normal, visible from direction $\mathbf{v}$
T	Local tension matrix of a deformed mesh
s	Stretch anisotropy parameter
M_s	Stretch anisotropy matrix
γ	Tail exponent of the GTR distribution

Table 4.1: Notation for linearly transformed microsurfaces.

4.3.1 Transforming the microsurface

Although the microfacets are infinitely small, we can still think of the microsurface as a 3D object which is aligned with the macrosurface. For instance, Heitz et al. [2016b] validated their multiple scattering model by ray-tracing a pre-generated Beckmann surface. So it stands to reason that the microsurface can be transformed by a 3×3 matrix, just like regular geometry. We consider all such transformations that lead to a valid microsurface, which are all invertible matrices of the form

$$M = \begin{pmatrix} a & c & 0 \\ b & d & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{4.1}$$

This is the case since:

- M applies an arbitrary linear transformation to the microsurface in the tangential plane, parallel to the macrosurface, Figure 4.1 b).
- The first two column vectors of M are restricted to the tangential plane because the microsurface must remain aligned with the macrosurface, Figure 4.1 c).
- The first two row vectors of M are also contained in the tangential plane because vertical shear could break the microsurface configuration. Microfacet normals $\mathbf{m}$ could turn to the negative hemisphere, which is below the shaded macrosurface, Figure 4.1 d).
- Scaling the height of the microsurface h times results in the same microsurface statistics as if we scale the xy plane by a factor of $\frac{1}{h}$. This is observed

in the slopes of the transformed surface $-\frac{am_x+bm_y}{m_z}$ and $-\frac{cm_x+dm_y}{m_z}$ where the scaling of the height hm_z is identical to modifying M using $(\frac{a}{h}, \frac{b}{h}, \frac{c}{h}, \frac{d}{h})$. Therefore, we always set the last entry of M to 1.

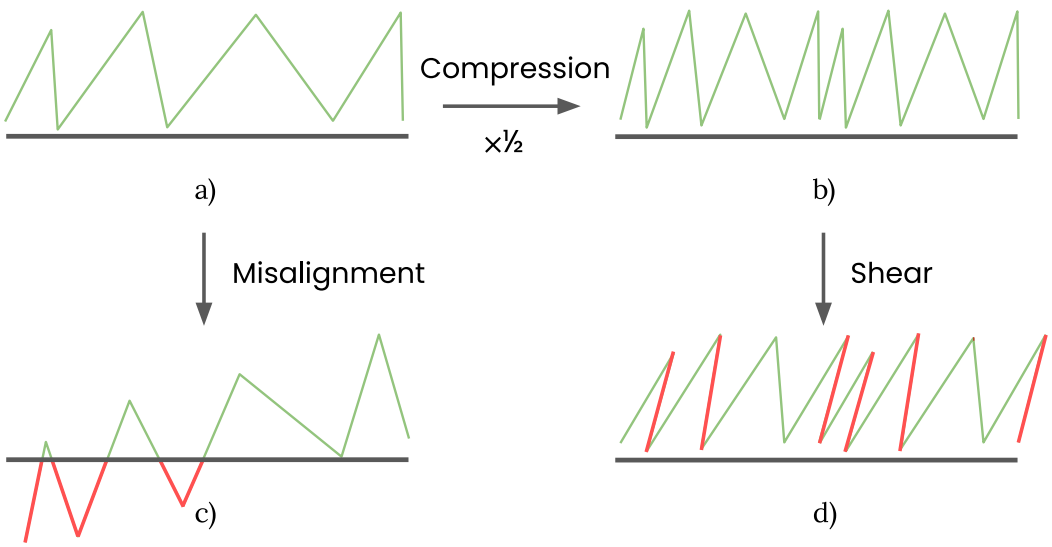


Figure 4.1: **Valid (top row) and broken (bottom row) microsurface configurations. Compression: a)→ b):** Linear transformations parallel to the macrosurface lead to new valid microsurfaces. **Misalignment: a)→ c):** Linear transformations outside the tangential plane break the micro-macrosurface alignment, as some microfacets sink below the macrosurface (red). **Shear: b)→ d):** Vertical shear parallel to the xy plane moves some microfacets into the negative hemisphere (red).

A basic fact in computer graphics is that when an object is transformed using a matrix M, the normal vectors to the surface must be transformed by the inverse transpose of that matrix $(M^{-1})^T$. This transformation is derived by enforcing the transformed tangents to be orthogonal to the transformed normals, as they should be by definition [Pharr et al., 2016]. Indeed this is directly verified

$$((M^{-1})^T\mathbf{m})^T(M\mathbf{t}) = \mathbf{m}^T(M^{-1}M)\mathbf{t} = \mathbf{m}^T\mathbf{t} = 0, \tag{4.2}$$

where $\mathbf{t}$ is a tangent to a microfacet with normal $\mathbf{m}$.

Figure 4.2 demonstrates our method with non-uniform scaling and skew transforms for shape-variant GTR. Note that the effect of stretching in Figure 4.2 (top right) is observed using the elliptical anisotropic formula for shape-invariant distributions, however, in this shape-variant case, the linear stretching of the microsurface does not result in linear scaling in the roughness parameter. The effect of a general 2D linear transform in Figure 4.2 (bottom left), beyond stretching, has not been shown so far even for the simpler case of shape-invariant distributions. Furthermore, in Figure 4.2 (bottom right) we demonstrate our method for a nonlinear 2D transformation. The Jacobian matrix of this transformation is a local linear approximation that we use within our framework.

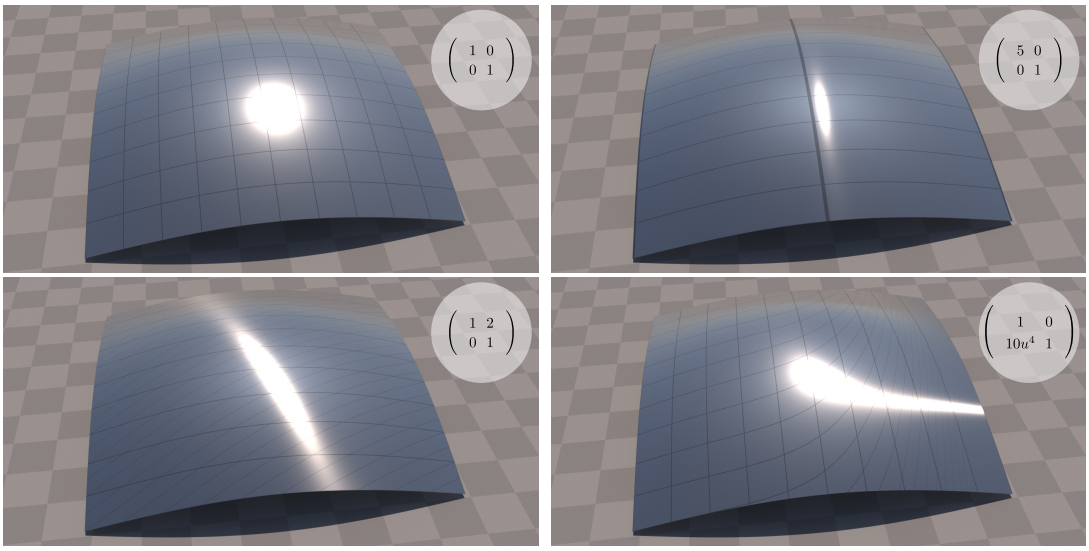


Figure 4.2: **Reflection from deformed microsurfaces.** A demonstration of how different 2D transformations (top right insets) of the microsurface affect the reflection off a rough silver plate. **Top left:** The original unmodified surface with shape-variant GTR distribution with tail $\gamma = 1.7$ and roughness $\alpha = 0.0121$. A 10×10 grid covers the unit texture space to serve as a reference. **Top right:** The microsurface is stretched 5 times horizontally. This makes the surface smoother in that direction, so the highlight is compressed accordingly. **Bottom left:** A skew transform M with $(a, b, c, d) = (1, 0, 2, 1)$, see Equation (4.1). **Bottom right:** An example of a nonlinear transformation: $T(u, v) = (u, v + 2u^5)$. The Jacobian matrix J_T (top right inset) is a local linear approximation of T and we apply it as a local tangential transform in Equation (4.1). Notice that the prolonged shape of the highlight to the right is due to more surface compression in that region.

4.3.2 Statistics of the transformed microsurface

To compute the statistics of the transformed microsurface we transform the arguments of the initial valid microsurface statistics. All arguments are unit vectors so normalization is required after the arguments are transformed.

Shadowing function $G_{1M}(\mathbf{v}, \mathbf{m})$: When the vector $\mathbf{v}$ is transformed along with the microsurface, the shadowing configuration does not change. Note that the micro-normal $\mathbf{m}$ must be transformed with the inverse transpose of M

$$G_{1M}\left(\frac{M\mathbf{v}}{\|M\mathbf{v}\|}, \frac{(M^{-1})^T\mathbf{m}}{\|(M^{-1})^T\mathbf{m}\|}\right) = G_1(\mathbf{v}, \mathbf{m}), \tag{4.3}$$

and therefore we can express the shadowing of the transformed microsurface G_{1M} in terms of G_1 by inverting the matrices before the arguments and normalizing

$$G_{1M}(\mathbf{v}, \mathbf{m}) = G_1\left(\frac{M^{-1}\mathbf{v}}{\|M^{-1}\mathbf{v}\|}, \frac{M^T\mathbf{m}}{\|M^T\mathbf{m}\|}\right). \tag{4.4}$$

This is a generalization of the masking probability invariance [Heitz, 2014]. Indeed, if an object is lit by a directional light and some parts of it are self-shadowed, then a linear transformation applied both to the object and the light direction will preserve the shadowed portions. This is because the linear transformations preserve the projection along the transformed light direction.

Microfacet distribution $D_M(\mathbf{m})$: The input normal $\mathbf{m}$ is transformed similarly, but the resulting distribution is not normalized in general. To find the proper normalization we investigate the nonlinear transformation $N : \mathcal{H}^2 \to \mathcal{H}^2$ that applies a linear transformation to a normal and normalizes the result

$$\mathbf{u} = N(\mathbf{m}) = \frac{M^T\mathbf{m}}{\|M^T\mathbf{m}\|}. \tag{4.5}$$

N is a bijection because M^T is invertible. The product of M^T and the micro-normal $\mathbf{m} = (m_x, m_y, m_z)^T$ is $M^T\mathbf{m} = (am_x + bm_y, cm_x + dm_y, m_z)^T$ and has length

$$\|M^T\mathbf{m}\| = \sqrt{(am_x + bm_y)^2 + (cm_x + dm_y)^2 + m_z^2}. \tag{4.6}$$

Note that the microfacet distribution is two-dimensional: the micro-normal $\mathbf{m}$ can be represented by its first two components m_x and m_y. The z component is projected on the hemisphere $m_z = \sqrt{1 - m_x^2 - m_y^2}$.

To find a new distribution D_M by transforming the argument of D with N we need to normalize by the absolute value of the Jacobian determinant of N [Pharr et al., 2016]

$$D_M(\mathbf{m}) = |\det J_N| D(\mathbf{u}), \tag{4.7}$$

where

$$\det J_N = \begin{vmatrix} \frac{\partial u_x}{\partial m_x} & \frac{\partial u_x}{\partial m_y} \\ \frac{\partial u_y}{\partial m_x} & \frac{\partial u_y}{\partial m_y} \end{vmatrix} = \frac{\partial u_x}{\partial m_x}\frac{\partial u_y}{\partial m_y} - \frac{\partial u_x}{\partial m_y}\frac{\partial u_y}{\partial m_x}. \tag{4.8}$$

We proceed to compute the partial derivatives

$$\frac{\partial u_x}{\partial m_x} = \frac{m_y^2(ad^2 - bcd - a) + m_x m_y(acd + b - bc^2) + a}{\|M^T\mathbf{m}\|^3} \tag{4.9}$$

$$\frac{\partial u_x}{\partial m_y} = \frac{m_x^2(bc^2 - acd - b) + m_x m_y(bcd + a - ad^2) + b}{\|M^T\mathbf{m}\|^3} \tag{4.10}$$

$$\frac{\partial u_y}{\partial m_x} = \frac{m_y^2(cb^2 - abd - c) + m_x m_y(abc + d - da^2) + c}{\|M^T\mathbf{m}\|^3} \tag{4.11}$$

$$\frac{\partial u_y}{\partial m_y} = \frac{m_x^2(da^2 - abc - d) + m_x m_y(abd + c - cb^2) + d}{\|M^T\mathbf{m}\|^3}. \tag{4.12}$$

After simplification of Equation (4.8) we arrive at a concise result

$$\det J_N = \frac{ad - bc}{\|M^T\mathbf{m}\|^4} = \frac{\det M}{\|M^T\mathbf{m}\|^4}. \tag{4.13}$$

We use this result in Equation (4.7) to obtain the microfacet distribution of the transformed microsurface

$$D_M(\mathbf{m}) = \frac{|\det M|}{\|M^T\mathbf{m}\|^4} D(\mathbf{u}). \tag{4.14}$$

Additionally, we provide a Mathematica notebook with the derivation of $\det J_N$ in Appendix D.2, which can be evaluated and downloaded from Wolfram Foundation's Notebook Archive [Atanasov, 2022].

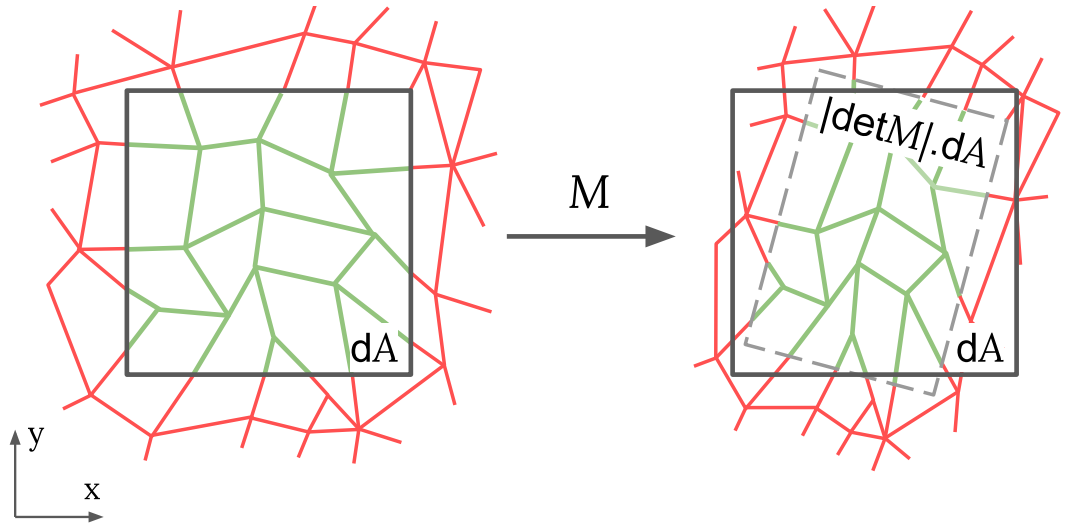


Figure 4.3: **Change of projected area.** Illustration of the microsurface projection onto the xy plane. **Left:** A square patch (black) on the macrosurface, which has a differential area $\mathrm{d}A$. The portion of the microsurface that is inside this patch is colored green while the rest is colored red. **Right:** When this microsurface is transformed using the matrix M, the portion of the microsurface which was previously in $\mathrm{d}A$ is transformed accordingly (gray dashed region). The transform M changes the area by a factor of $|\det M| = |\det M^T|$.

We provide geometrical intuition behind Equation (4.14). The matrix M transforms the xy plane and changes the projected microsurface area by a factor of $|\det M| = |\det M^T|$. The microfacet distribution measures the microsurface differential area relative to the macrosurface differential area, see Figure 4.3. Therefore, this factor compensates for the change in the projected area.

Furthermore, we observe in Figure 4.4 that

$$\|M^T\mathbf{m}\| = \frac{\cos\theta_{\mathbf{m}}}{\cos\theta_{\mathbf{u}}}. \tag{4.15}$$

Therefore, in Equation (4.14) the factor $\frac{1}{\|M^T\mathbf{m}\|^4}$ serves the purpose to change the Jacobian $\frac{1}{\cos^3\theta_{\mathbf{u}}}$ and the inverse projection $\frac{1}{\cos\theta_{\mathbf{u}}}$ in Equation (3.6), both with respect to the transformed micro-normal $\mathbf{u}$, to the Jacobian $\frac{1}{\cos^3\theta_{\mathbf{m}}}$ and the inverse projection $\frac{1}{\cos\theta_{\mathbf{m}}}$ with respect to the original micro-normal $\mathbf{m}$.

Equation (4.14) is similar to the Linearly Transformed Spherical Distributions (LTSD) [Heitz et al., 2016a], however, their formula does not include the inverse projections. Hence, our formulation that is given in Equations (4.14) and (4.1) defines a different family of distributions which we name by analogy *Linearly Transformed Microfacet Distributions (LTMD)*.

Finally, we express the slope distribution of the transformed microsurface P_{22M} from the original slope distribution P_{22} by using, in that order, Equations (3.6),

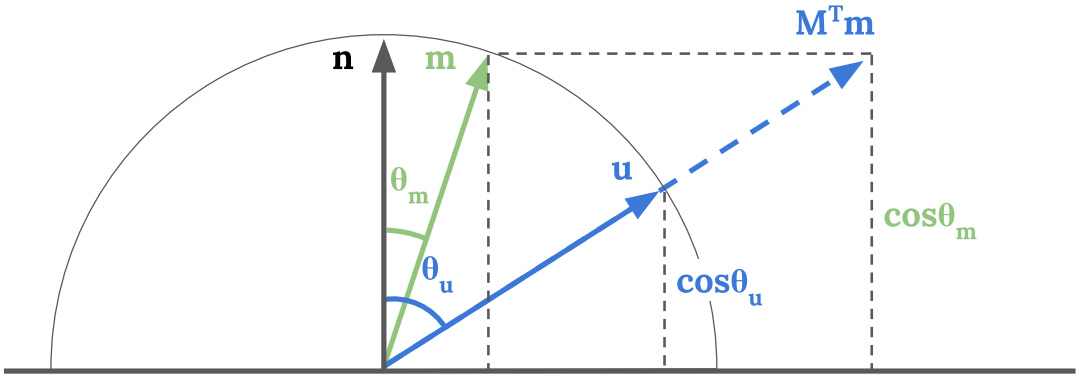


Figure 4.4: **Geometry of micro-normal transformation.** Let $\mathbf{m}$ (green) be a micro-normal from the original microsurface. The transformed vector $M^T\mathbf{m}$ (blue, dashed) has the same z-component as the original vector $\mathbf{m}$ because the transformation M^T acts only on the xy plane. This vector is normalized and labeled $\mathbf{u} = \frac{M^T\mathbf{m}}{\|M^T\mathbf{m}\|}$. From the two similar right triangles with hypotenuses $M^T\mathbf{m}$ and $\mathbf{u}$ we find that $\|M^T\mathbf{m}\| = \frac{\cos\theta_\mathbf{m}}{\cos\theta_\mathbf{u}}$.

(4.14) and (4.15)

$$\begin{aligned} P_{22M}\left(-\frac{m_x}{m_z}, -\frac{m_y}{m_z}\right) &= D_M(\mathbf{m})\cos^4\theta_\mathbf{m} \\ &= \frac{|\det M|}{\|M^T\mathbf{m}\|^4} D(\mathbf{u})\cos^4\theta_\mathbf{m} \\ &= |\det M|\frac{\cos^4\theta_\mathbf{u}}{\cos^4\theta_\mathbf{m}} D(\mathbf{u})\cos^4\theta_\mathbf{m} \\ &= |\det M| D(\mathbf{u})\cos^4\theta_\mathbf{u} \\ &= |\det M| P_{22}\left(-\frac{u_x}{u_z}, -\frac{u_y}{u_z}\right). \end{aligned} \tag{4.16}$$

Consequently, Equation (4.14) can be also derived from the slope distribution.

Sampling micro-normals: to implement practical microfacet materials we need sampling equations for the microfacet distribution. In our work, we assume that the original microsurface before the transformation has sampling equations at least for $D(\mathbf{m})(\mathbf{n}\cdot\mathbf{m})$. Based on these, a micro-normal $\mathbf{u}'$ is sampled

$$\mathbf{u}' \propto D(\mathbf{m})(\mathbf{n}\cdot\mathbf{m}). \tag{4.17}$$

Then it is transformed to the modified microsurface using the inverse transpose of M

$$\mathbf{m}' = \frac{(M^{-1})^T\mathbf{u}'}{\|(M^{-1})^T\mathbf{u}'\|} \propto D_M(\mathbf{m})(\mathbf{n}\cdot\mathbf{m}). \tag{4.18}$$

If the original microfacet distribution has sampling equations for the distribution of visible normals, see Equation (3.10), then we can use them to sample the distribution of visible normals of the transformed surface

$$D_{\mathbf{v}M}(\mathbf{m}) = \frac{D_M(\mathbf{m})G_{1M}(\mathbf{v},\mathbf{m})(\mathbf{v}\cdot\mathbf{m})}{\mathbf{v}\cdot\mathbf{n}}. \tag{4.19}$$

To do that for a given direction $\mathbf{v}$ we first transform it to the space of the original surface

$$\mathbf{w} = \frac{M^{-1}\mathbf{v}}{\|M^{-1}\mathbf{v}\|}, \tag{4.20}$$

generate a visible normal $\mathbf{u}'_{\mathbf{w}}$ using the available equations and then transform it to the modified surface using the inverse transpose

$$\mathbf{m}'_{\mathbf{v}} = \frac{(M^{-1})^T \mathbf{u}'_{\mathbf{w}}}{\|(M^{-1})^T \mathbf{u}'_{\mathbf{w}}\|}. \tag{4.21}$$

Sampling micro-normals and visible micro-normals is illustrated in Figure 4.5.

4.3.3 Numerical validation

In addition to the proof presented in this section, we also validated our technique numerically. Specifically, we applied random transformations M on a large variety of microsurfaces with random parameters and verified that:

- The normalization constraint in Equation (3.5) is fulfilled for all transformed microfacet distributions D_M.
- The shadowing constraint in Equation (3.9) with random directions $\mathbf{v} \in \mathcal{H}^2$ is fulfilled for all transformed shadowing functions G_{1M}.

The list of tested distributions include GTR [Burley, 2012], anisotropic Beckmann and GGX [Heitz, 2014], STD [Ribardière et al., 2017], Phong [Walter et al., 2007], Sheen [Estevez and Kulla, 2017], Discrete GGX distribution [Jakob et al., 2014, Atanasov and Koylazov, 2016]. The source code for this numerical validation is available on the project page (see Attachments).

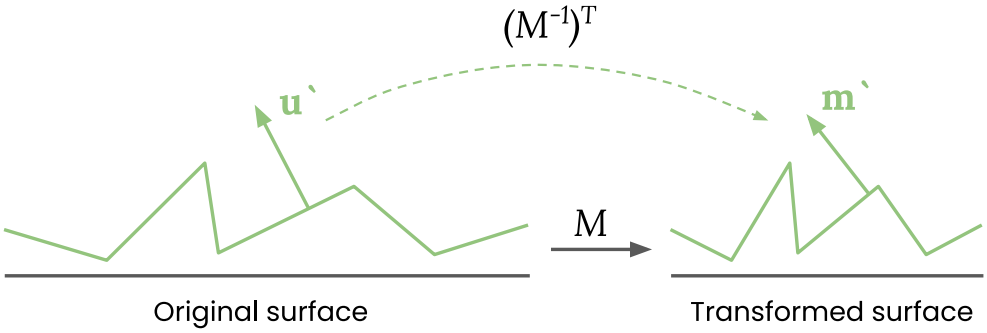
4.3.4 Equivalence with classical anisotropy for shape-invariant distributions

Suppose that we have a shape-invariant isotropic distribution $D(\mathbf{m})$ with roughness α, defined by Equation (3.16). To construct the traditional anisotropic form we replace α with two roughness values α_x and α_y and use Equation (3.17). In our framework, this is equivalent to applying the diagonal transformation matrix

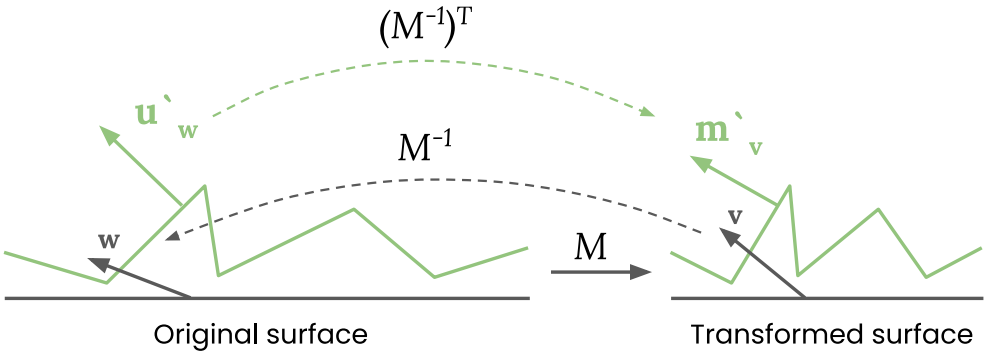
$$M = \begin{pmatrix} \frac{\alpha}{\alpha_x} & 0 & 0 \\ 0 & \frac{\alpha}{\alpha_y} & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{4.22}$$

Then substituting Equation (3.16) in Equation (4.14)

$$\begin{aligned} D_M(\mathbf{m}) &= \frac{|\det M|}{\|M^T \mathbf{m}\|^4} \frac{1}{\cos^4 \theta_{\mathbf{u}}} \frac{1}{\alpha^2} f\left(\frac{\tan \theta_{\mathbf{u}}}{\alpha}\right) \\ &= \frac{\frac{\alpha^2}{\alpha_x \alpha_y}}{\frac{\cos^4 \theta_{\mathbf{m}}}{\cos^4 \theta_{\mathbf{u}}}} \frac{1}{\cos^4 \theta_{\mathbf{u}}} \frac{1}{\alpha^2} f\left(\frac{1}{\alpha}\sqrt{\frac{u_x^2 + u_y^2}{u_z^2}}\right) \\ &= \frac{1}{\cos^4 \theta_{\mathbf{m}} \alpha_x \alpha_y} f\left(\frac{1}{\alpha}\sqrt{\frac{\alpha^2 m_x^2}{\alpha_x^2 m_z^2} + \frac{\alpha^2 m_y^2}{\alpha_y^2 m_z^2}}\right) \\ &= \frac{1}{\cos^4 \theta_{\mathbf{m}} \alpha_x \alpha_y} f\left(\frac{\sin \theta_{\mathbf{m}}}{\cos \theta_{\mathbf{m}}}\sqrt{\frac{\cos^2 \phi_{\mathbf{m}}}{\alpha_x^2} + \frac{\sin^2 \phi_{\mathbf{m}}}{\alpha_y^2}}\right) \end{aligned} \tag{4.23}$$



a) Sampling micro-normals



b) Sampling visible micro-normals

Figure 4.5: **Sampling the transformed NDF and VNDF.** An illustration of how to sample micro-normals on the transformed surface using the sampling equations of the original surface. The original surface is transformed using the matrix M. **a) Sampling micro-normals:** Sample a micro-normal $\mathbf{u}'$ on the original surface and move it to the transformed surface using the inverse transpose of M. **b) Sampling visible micro-normals:** We want to sample a micro-normal on the transformed surface which is visible from direction $\mathbf{v}$. First, we transform the direction $\mathbf{v}$ to the original surface using the inverse of M. Then we take the resulting direction $\mathbf{w}$, along which we sample a visible micro-normal $\mathbf{u}'_\mathbf{w}$. Finally, we move this micro-normal to the original surface using the inverse transpose of M. The result $\mathbf{m}'_\mathbf{v}$ is visible from direction $\mathbf{v}$ and it is distributed according to the distribution of visible normals of the transformed surface $D_{\mathbf{v}M}$ from Equation (4.19).

we get to Equation (3.17) when we first convert $\tan\theta_{\mathbf{u}}$ to cartesian coordinates, express $\mathbf{u}$ in terms of $\mathbf{m}$ from Equation (4.5) and convert back to spherical coordinates, see Appendix B.

An important point is that for shape-variant distributions the scaling of the microsurface is not equivalent to the scaling of the roughness parameter. However, within our framework, scaling operations can always be applied to the microsurface, while formulas for scaling the roughness parameter do not always exist.

4.3.5 Discussion

In summary, for a given microsurface (D, G_1) and a linear transformation M given in Equation (4.1), our technique defines the transformed microsurface (D_M, G_{1M}) using Equations (4.14) and (4.4). It is noteworthy that in our derivation we do not make any assumptions on the input microsurface. Therefore, the input distribution can be both shape-invariant and shape-variant, as well as isotropic or anisotropic. This means that it can be used as a *black box* - it can be implemented in a base class and can be applied regardless of the underlying microfacet distribution, including stochastic [Jakob et al., 2014] or data-driven [Ashikmin et al., 2000, Yan et al., 2014, Ribardière et al., 2019] approaches. Furthermore, the solution is in closed form and its performance depends mainly on the expressions for D and G_1.

4.4 Results

We demonstrate our technique using the standard microfacet BRDF for specular microfacets (Equation (3.24)) with an appropriate Fresnel term for dielectrics and conductors. All materials use the GTR distribution with different tail exponents. Additionally, we provide a video demonstration of our technique (see Attachments).

Dimov [2015] derived an accurate Smith shadowing term for GTR tails $\gamma \in [0, 4]$ that is exact at the integers, which we provide in Appendix C.3.

Due to the lack of a formula for sampling the distribution of visible GTR normals, the sampling efficiency deteriorates with roughness because of the increased shadowing. We propose a sampling strategy to improve the sampling of rough GTR materials in Appendix C.4.

4.4.1 Deforming objects

In Figure 4.6, we apply our framework in a scene with deforming objects. We track the deformations using a tension texture, based on which we derive the local linear transformations and render the corresponding anisotropic microsurface. The purpose of the tension texture is to compute a 2D tension matrix that contains the local surface transformation relative to a predefined "rest" state. To compute this, we need a stable local space for each shading point that is consistent between the rest state and the current deformed state of the mesh. Then, at each shading point, we transform the edges of the currently shaded triangle to this local space both for the deformed mesh and for the "rest" mesh. Finally,

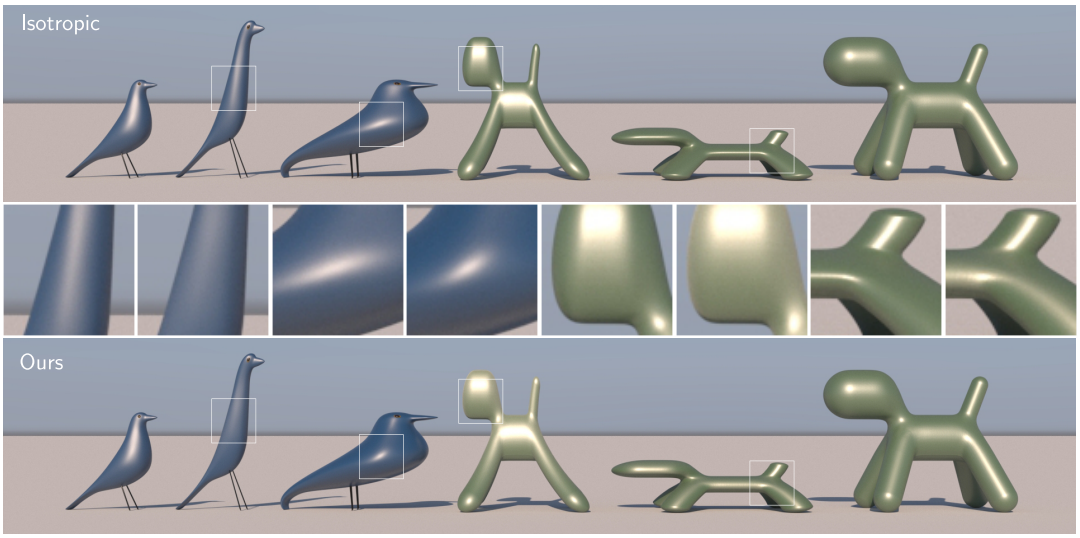


Figure 4.6: **Reflection from deforming objects — comparison between baseline isotropic model vs. our model.** Decorative objects made from an elastic dielectric material with microfacet BRDFs. All birds and all dogs have materials with shape-variant GTR distributions with roughness values $\alpha = 0.25$ and $\alpha = 0.16$, and tail exponents $\gamma = 2.2$ and $\gamma = 1.4$, respectively. We demonstrate how the object's reflections change due to deformation. When the object is stretched, the microsurface becomes smoother, and therefore the reflection highlight becomes sharper. When the microsurface is compressed its reflection spreads out. **Top row:** A standard isotropic material. **Bottom row:** Our method tracks the deformation via a tension texture and computes the corresponding anisotropic reflection. The first and last objects in the composition are at a "rest" state while all objects in between are deformed. **Middle row:** Alternating zoomed-in regions of isotropic (top row) and deformed anisotropic (bottom row). A helpful analogy to understand the highlight changes that occur with our model is a rubber balloon: as it gets inflated (stretched out), it becomes smoother and shinier, see Figure 1.5. Our model can replicate this behavior even for anisotropic stretching. (The bird and the dog models are part of Chaos Cosmos content collection [2023a].)

we compute a transformation in local space that converts the "rest" edges to the current deformed edges.

We use the UV mapping coordinates assigned to the surface as the stable local space. For each shading point, we can compute an orthonormal matrix that converts from world space to the local UVW space for the deformed mesh, and another orthonormal matrix that converts from world space to the local UVW space for the "rest" mesh. Using these two matrices, we can rotate the edges of the currently shaded triangle in the deformed mesh and the edges of the same triangle in the "rest" mesh to local space. Note that in general the local UVW spaces are not orthonormal [Pharr et al., 2016] but we force them to be by preserving the U direction and then orthogonalizing and normalizing the matrices. In this way, we retain the shape of the triangles and only align them with the U direction in the tangent plane. Let $\mathbf{r}_1$ and $\mathbf{r}_2$ be the transformed edges in the local space of the "rest" triangle, and $\mathbf{d}_1$ and $\mathbf{d}_2$ be the transformed edges in the local space

of the deformed triangle. These edges are 2-dimensional because we are only interested in the projection to the tangent UV space. Then the 2×2 tension matrix is $T = DR^{-1}$, where $R = (\mathbf{r}_1\ \mathbf{r}_2)$ and $D = (\mathbf{d}_1\ \mathbf{d}_2)$, see Figure 4.7. For

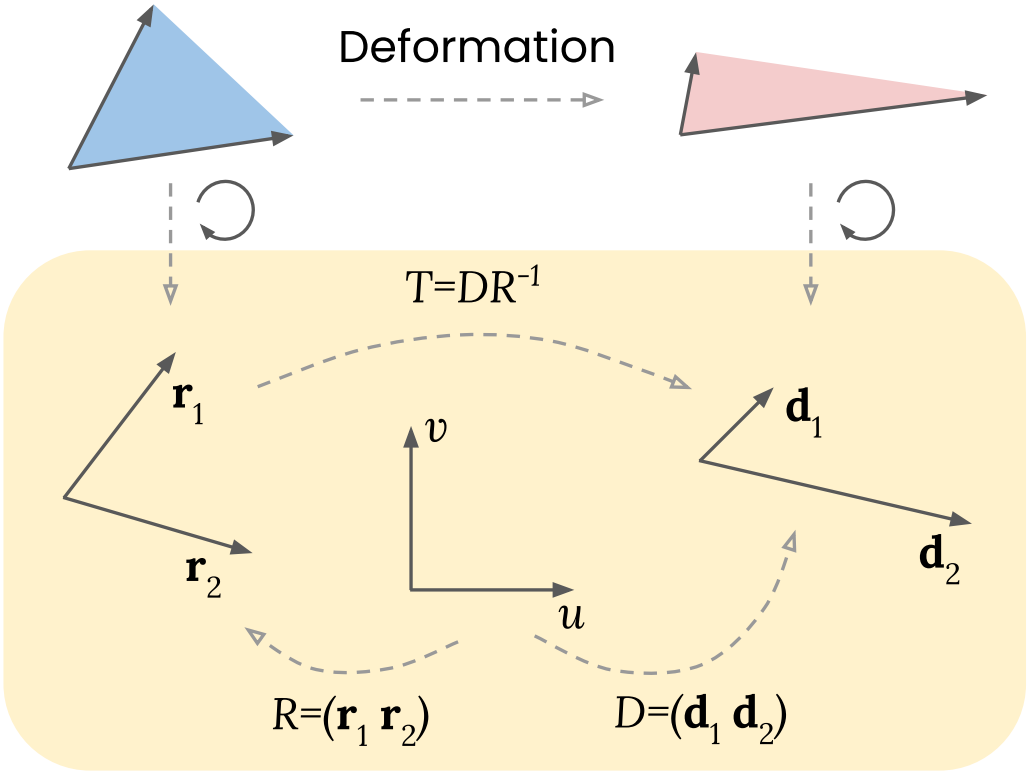


Figure 4.7: **Tracking mesh deformation.** A triangle of the "rest" mesh (blue) and its corresponding triangle in the deformed mesh (red). The edges of these triangles are rotated to UV space (yellow): the edges of the "rest" triangle $\mathbf{r}_1$ and $\mathbf{r}_2$ form the matrix R that converts to the space of the "rest" triangle, and the edges of the deformed triangle $\mathbf{d}_1$ and $\mathbf{d}_2$ form the matrix D that converts to the space of the deformed triangle. Therefore, the matrix $T = DR^{-1}$ encodes the local tension of the deformation.

shading, we embed the tension matrix in the matrix M, see Equation (4.1). The tension texture is applied to the deformed object and keeps a reference to the "rest" object. During texture evaluation, the matrices R and D are computed for the shaded triangle and the tension matrix T is returned.

This method computes per-face tension matrices and requires relatively fine geometry tessellation to achieve smooth results. Another option is to pre-process the "rest" mesh and the deformed mesh and to store a weighted average of the face tension matrices in the vertices. In this case, vertex tension matrices are interpolated along the triangles to facilitate continuous shading.

4.4.2 Stretch anisotropy

We have established that traditional anisotropy can be reproduced as a special case of our framework by using a non-uniform scaling matrix, see Equation (4.22). For our stretch anisotropy, we use a convenient scaling transform that is defined

by a single scaling factor $s \in (-1, 1)$:

$$M_s = \begin{cases} diag\left(\frac{1}{1-s}, 1-s, 1\right), & s \in [0, 1) \\ diag\left(1+s, \frac{1}{1+s}, 1\right), & s \in (-1, 0) \end{cases}. \tag{4.24}$$

For positive values of s it stretches the x-coordinate and shrinks the y-coordinate while for negative values of s, it does the opposite. Moreover,

$$\det M_s = 1,\ M_s^T = M_s,\ M_0 = I \text{ and } M_s^{-1} = M_{-s}. \tag{4.25}$$

Note that a 2D rotation can be concatenated with the matrix M_s, but we choose to keep the matrix simple and perform anisotropy rotation via the shading frame rotation.

We showcase our technique in Figure 4.8. The vases have brushed aluminum material, and all of them have anisotropic shape-variant distributions that do not have analytic equations for elliptical anisotropy, given in Equation (3.15).

4.5 Conclusion

We propose a change in how one goes about the construction of anisotropic microfacet distributions. Performing these derivations based on the transformations that need to be done to the micro-geometry, instead of based on the directional variation of surface roughness, offers desirable advantages: our approach works with all planar linear transformations (including skew), and with all slope distribution functions, regardless of whether they are shape-invariant, or not. This offers artists a wider range of creative control, both in terms of the highlight shapes they can create, and the types of surfaces that can be made anisotropic.

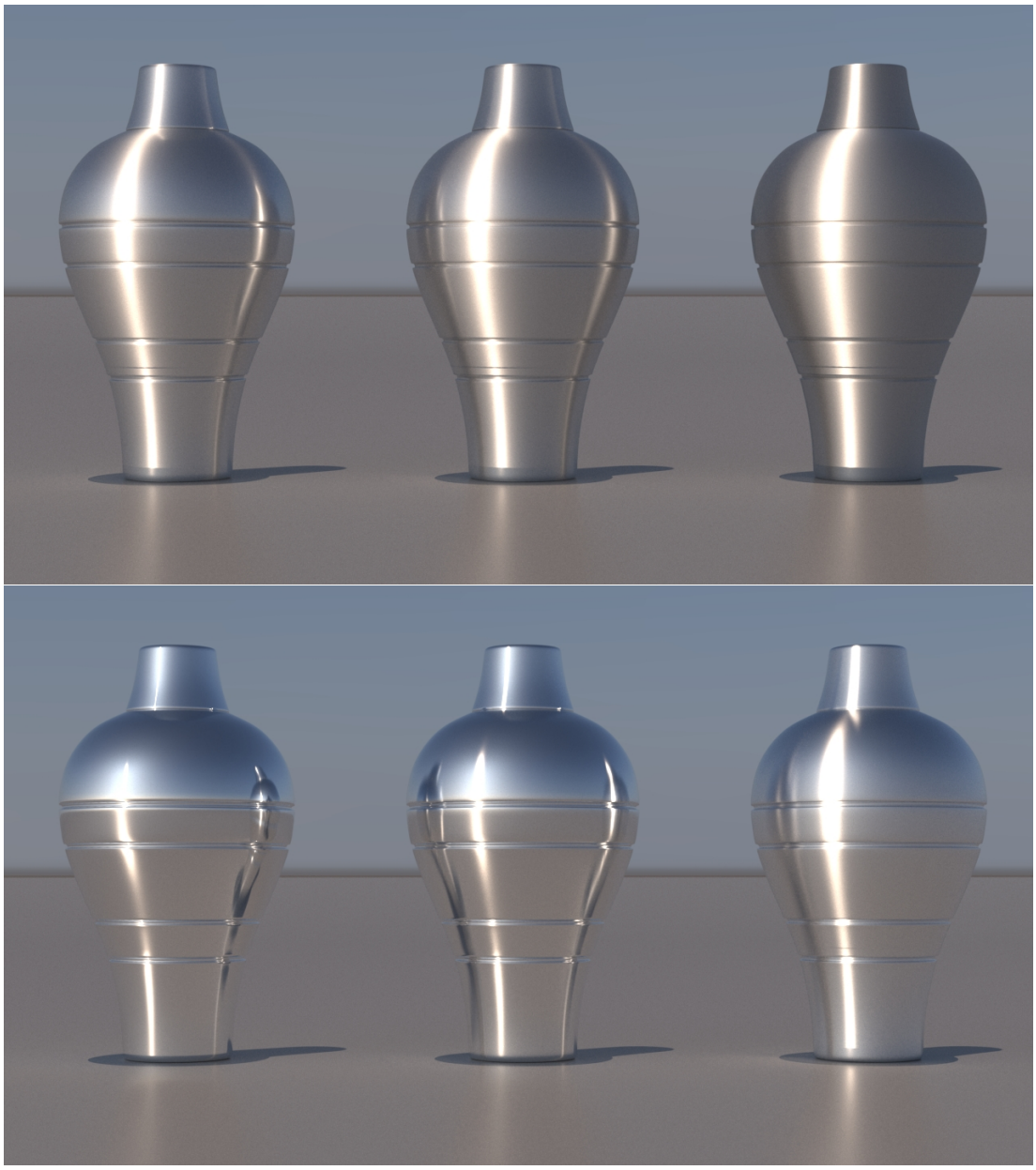


Figure 4.8: **Brushed aluminum vases rendered with (anisotropic) GTR distribution.** Columns from left to right have roughness values $\alpha = 0.01, 0.04, 0.16$. The top row has the Berry distribution $\gamma = 1$ and the bottom row has a tail exponent $\gamma = 3.5$. All materials have shape-variant anisotropic distributions with scaling factor $s = \frac{5}{2}$, see Equation (4.24). (The vase model is part of Chaos Cosmos content collection [2023a].)

Chapter 5

Multiscale Microfacet Model

5.1 Introduction

In this chapter, we investigate the problem of efficient filtering of the *combined BRDF*, which we defined in Chapter 3, Equation (3.27). Filtering techniques like MIP mapping [Williams, 1983] and summed-area tables (SATs) [Crow, 1984] are available for diffuse color textures, but they do not work for normal maps due to the nonlinearity of the reflection operator [Han et al., 2007]. We show that the normal map filtering problem can be solved using a generalized summed-area table known as *integral histogram (IH)* [Porikli, 2005].

Here we develop an accurate and efficient filtering algorithm for Beckmann flake roughness that can be implemented using IH. However, the memory requirements of standard IH implementations make them impractical to use for non-trivial scenes with large and numerous assets. Therefore, we introduce a new optimized form of IH, the Inverse Bin Map (IBM), which is very fast to build, and has modest memory requirements, comparable with MIP maps and SATs.

5.2 Related work

We start with a brief discussion of *Integral histograms (IH)*. Then we review the diverse palette of techniques for rendering reflection from microstructures and place focus on normal-map-based approaches for rendering glints.

5.2.1 Integral histograms (IH)

Integral histograms (IH) were first introduced in the field of computer vision [Porikli, 2005] and became a fundamental tool for image analysis and processing with numerous applications [Ballester-Ripoll and Pajarola, 2019]. Due to their high memory requirements, WaveletSAT [Lee and Shen, 2013] was developed, offering lossless compression of IHs at the cost of reducing the query complexity from constant to logarithmic. Compression rates are commonly around 1:8 [Ballester-Ripoll and Pajarola, 2019].

Recently, Ballester-Ripoll and Pajarola [2019] proposed a lossy compression scheme for IHs based on tensor decomposition with higher compression rates and extended IH queries for arbitrary regions: however, the price for this is slower

retrieval time. In principle, our algorithm or other variations based on the binned BRDF described in this chapter can be implemented with any of these three data structures: classical IH, WaveletSAT and the tensor decomposition scheme. Due to the high number of bins required by our solution, and potentially high-resolution normal map, the memory requirements of the first two are too high for our practical application. The method of Ballester-Ripoll and Pajarola [2019] provides significantly more flexibility in terms of supported query regions, but it is not viable for our problem, due to its pre-processing time which can range from a few minutes up to several hours.

5.2.2 Explicit microstructure modeling

Günther et al. [2005] simulated metallic car paint glitter based on non-filtered procedurally generated normal maps. Weidlich and Wilkie [2008] used a similar approach to render the glitter in aventurescent gems (we provide microscope photographs of a goldstone in Figure A.2). We also employ procedurally generated normal maps to render car paint and demonstrate that filtering is crucial to resolve its inherent sharp glints, see our video in Attachments.

Rump et al. [2008] rendered car paint using measured data. Ershov et al. [1999, 2001] simulated metallic car paint using a statistical model: the main drawback of their approach is that the results are not consistent in animation. Jakob et al. [2014] developed a model based on microfacet theory which uses a stochastic process to compute temporally consistent sparkling. This led to a more practical version of the same underlying model which features optimal importance sampling strategy [Atanasov and Koylazov, 2016]. These stochastic methods can be used to render metallic flakes, but flakes' roughness and sizes, and the transparency of the flake layer cannot easily be included in the model. Zirr and Kaplanyan [2016] presented a real-time approach for rendering sparkling flakes and parallel scratches, however, their method is limited to Beckmann distribution. Later, Chermain et al. [2020] developed a real-time approach to render glitter that additionally converges to the microfacet BRDF for high flake densities.

Methods specialized in the efficient rendering of scratches have been recently developed [Raymond et al., 2016, Werner et al., 2017, Velinov et al., 2018]. Kuznetsov et al. [2019] simulate materials with stochastic nature like flakes and scratches with a pre-trained neural network. All these approaches are limited to specific spatial details and thus have limited expressiveness.

5.2.3 Normal-mapped microstructure

Approaches based on normal maps are more flexible since they can represent arbitrary spatial and directional features. Therefore, efficient implementation of normal map filtering is a very important problem for production rendering systems.

Approaches that approximate the distribution of normals inside the pixel with a single lobe [Toksvig, 2005, Olano and Baker, 2010, Dupuy et al., 2013, Hery et al., 2014] offer artifact-free solutions and are compatible with real-time graphics, but high-frequency detail like sharp sparkling is lost [Yan et al., 2014]. Both Han et al. [2007] and Xu et al. [2017] investigated the combined effect of an

isotropic BRDF and a normal map and provided filtering techniques that approximate the distribution of normals by a small number of lobes.

Notably, Wu et al. [2019] developed a method for prefiltering displacement-mapped surfaces with isotropic BRDFs that accounts for accurate shadowing-masking and interreflections. The method does not support high directional resolutions to render glints from specular surfaces.

A family of accurate approaches for rendering specular normal maps inherits the mathematical model of Yan et al. [2014]. This model represents the NDF as a convolution of Gaussian footprints around the shading points and Gaussian intrinsic roughness lobes around the normal map directions. This definition leads to 4D texture-direction Gaussian queries to evaluate the NDF. Subsequent approaches that are based on the same model improve performance [Yan et al., 2016], compute antialiasing for global illumination effects [Belcour et al., 2017], derive accurate shadowing-masking factors using approximation with anisotropic Beckmann lobes [Chermain et al., 2018], and introduce wave effects [Yan et al., 2018]. All of them have high memory requirements and are based on expensive 4D position-normal queries.

Zhu et al. [2019] developed a method based on the method of Yan et al. [2016] which offers memory reduction for the special case of normal maps with a block structure. Wang et al. [2020] generate an infinite surface from a small example map via by-example blending. Its memory usage is 35MB for a 512^2 map.

Recently, Deng et al. [2022] presented a normal map prefiltering method that offers constant storage and constant queries. This is achieved by enumerating all NDFs and compressing them with a tensor rank decomposition. They report a dramatic performance improvement compared to Wang et al. [2020] and a moderate improvement compared to our method [Atanasov et al., 2021]. While Deng's method benefits from constant queries, our method requires a kd-tree traversal for each query due to our choice to trade some performance for memory efficiency by using the IBM, instead of a classical IH. On the other hand, enumerating and compressing all NDFs is viable only for small tiles that cannot faithfully represent all normal map features. Furthermore, Deng's method has a lengthy precomputation for NDF generation and compression. Also, their method is based on Wang's method [2020] and thus it inherits Wang's repetitive artifacts [Deng et al., 2022]. Importantly, all of these three methods based on map tiles [Zhu et al., 2019, Wang et al., 2020, Deng et al., 2022] are intended to render textures with predominantly stationary structures, and without macroscopic features. In the survey of Zhu et al. [2022], it is evident that both Zhu et al. [2019] and Wang et al. [2020] methods visually diverge from the reference images.

Gamboa et al. [2018] explored the combined filtering of specular normal maps and environment illumination maps. The additional prefiltering of the incident illumination by projecting the environment to a spherical harmonics (SH) basis is a key advantage of their technique. The prefiltering of the SH coefficients is achieved by a spherical histogram, which is an integral histogram that is constructed over spherical bins. This aspect is similar to our solution, although the use of classical IH attributes to the large memory requirements of the method (2.3-2.7GB for 2K maps). In comparison, our proposed method usually uses less than 40MB (for 2K maps), which is at least 60 times less memory. Apart from very high memory requirements for a moderate number of directional bins, the

method requires at least minutes of pre-computation time, which is needed for SH projections of the environment light coefficients. Furthermore, the SH coefficients of area light sources cannot be pre-computed. In theory, they can be calculated at run-time [Wang and Ramamoorthi, 2018], however, Gamboa et al. [2018] do not showcase their method with area lights.

The specular manifold sampling (SMS) method was demonstrated to render glints with modest memory requirements and with similar convergence rates [Zeltner et al., 2020]. The method also has a brief pre-computation: only a LEAN map [Olano and Baker, 2010] is built. However, SMS does not employ an acceleration or prefiltering data structure to find the glints in the pixel footprint, and instead relies on stochastic sampling. This strategy becomes impractical for an increasing number of glints in the footprint [Zeltner, 2021]. In Figure 5.7 we show that SMS fails to render the aggregated subpixel glints when there are a large number of them.

5.2.4 Classification of normal map filtering techniques

Given such a large body of previous work, it is necessary to position our method in the landscape of existing techniques. We restrict our discussion to normal map based techniques since normal maps are widely accepted in the industry and they offer full control over the surface's spatial and directional features (notice that we exclude the tile-based techniques that do not fully capture these features [Zhu et al., 2019, Wang et al., 2020, Deng et al., 2022]). These techniques can be classified into two main categories:

- **Real-time prefiltering.** These are all filtering techniques that are used for the antialiasing of normal maps [Han et al., 2007, Olano and Baker, 2010, Dupuy et al., 2013, Hery et al., 2014, Xu et al., 2017]. They allow real-time performance due to prefiltering based on MIP mapping and consume a low to moderate amount of memory for a small number of MIP maps. They also expose roughness control which is usually in terms of Beckmann roughness. To the best of our knowledge, no filtering technique for normal maps with GGX BRDF has been demonstrated. The main limitation of these techniques is that they are not capable of rendering sharp subpixel glints.

  Bruneton and Neyret [2012] formulate the nonlinear prefiltering problem, offer a classification of the main approaches to tackle it and survey the early techniques.

- **Offline specular filtering.** These are offline rendering techniques for rendering specular normal maps, which enable the accurate rendering of glints [Yan et al., 2014, 2016, Gamboa et al., 2018, Zeltner et al., 2020]. All of these methods, except for SMS, have very high memory requirements and heavy pre-computation. None of them offers roughness control (in Section 5.5.2 we show that Yan's intrinsic roughness is not designed to control the appearance).

  Zhu et al. [2022] survey the recent techniques for rendering glints, mainly focusing on the convolutional framework of Yan et al. [2014]. However, they

also offer performance comparisons with our method [Atanasov et al., 2021]. It is noteworthy, that for the same roughness, our method produces sharper results compared to the convolutional approaches. The reason for this is that convolutional approaches use Gaussian footprint (see Figure 5.5) which results in an additional blur, thus improving their performance. Also, the survey omits the important aspects of performance at different scales and pre-computation times. We detail these distinctions in Section 5.5.

In general, our method belongs to the second category since it is an offline technique for the accurate rendering of glints. However, it is also bridging the gap between the two categories, because our method exposes the Beckmann roughness parameter, has very low memory usage and implements prefiltering. All of these properties are of great importance for practical applications, as we discussed in Section 5.5.1.

5.3 Normal map filtering

In this section, we analyze the challenging problem of accurate and fast filtering of normal maps. The main notation for this chapter is given in Table 5.1.

A	Finite region in texture space around x
f_A^α	Filtered BRDF over A
w_k	Texel weight $\lvert A \cap T_k \rvert / \lvert A \rvert$
H_j	Bin on the hemisphere with index j
B	Total number of bins
W_j	Bin weight $\sum_{k \mid \mathbf{t}_k \in H_j} w_k$
σ	Standard deviation of Beckmann roughness $\left(\alpha = \sqrt{2}\sigma\right)$
B^0	Number of bins within 3σ of a Beckmann distribution
I	2D data array
SAT_I	Summed-area table of I
$\beta,\ \beta^{-1}$	Binning strategy and its inverse
$\mathrm{IH}_{I,\beta}$	Integral histogram of I with binning β
$\mathcal{B},\ \mathcal{B}^{-1}$	Bin Map and Inverse Bin Map
$\mathcal{I}$	Index storing the size and the offset of bin's texel position list
$\mathcal{F}$	Forest of 2D kd-trees for the bins with large texel position lists
L	Maximum number of elements for each leaf in $\mathcal{F}$
σ_r	Yan's intrinsic roughness [2014, 2016]

Table 5.1: Notation for normal map filtering.

5.3.1 Filtered BRDF

We define the filtered BRDF f_A^α by averaging the spatially varying combined BRDF f_x^α over a finite texture space region A around the shading point x:

$$f_A^\alpha(\mathbf{i}, \mathbf{o}, \mathbf{n}) = \frac{1}{|A|} \int_A f_x^\alpha(\mathbf{i}, \mathbf{o}, \mathbf{n}) \mathrm{d}x. \tag{5.1}$$

Note that on both sides of the equation, $\mathbf{n}$ is the normal at the position x. Like all previous filtering techniques, we rely on the assumption of a locally flat geometry. In practice, this is a source of bias, which is sufficiently small when the surface normal $\mathbf{n}$ does not change considerably over the region A. The filtered BRDF can be defined more generally using an arbitrary filter kernel, but we use the constant "box" filter over the region A because this will lead to an efficient implementation later.

Using Equation (3.27) for f_x^α we expand f_A^α by substituting the microfacet BRDF f^α and distributing the integration over the normal map texels T_k

$$f_A^\alpha(\mathbf{i}, \mathbf{o}, \mathbf{n}) = F(\mathbf{i}, \mathbf{h}) \sum_{k=1}^{N} C(\mathbf{i}, \mathbf{o}, \mathbf{t}_k, \mathbf{n}) w_k, \tag{5.2}$$

where

$$w_k = \frac{|A \cap T_k|}{|A|}. \tag{5.3}$$

Texel weights w_k sum up to 1 and they are non-zero if the texels T_k cover the filtering region A.

Therefore, the evaluation of this filtered BRDF requires a loop over all normal map texels that happen to be fully or partially contained in the filtering region A. For a small number of texels in A, this formulation is the most efficient way to compute the filtered BRDF, but as A grows to cover more texels, this formula becomes impractical. For such scenarios, we derive an alternative formulation for f_A^α.

5.3.2 Binned BRDF

We partition the hemisphere $\mathcal{H}^2$ into directional bins

$$H_j \subset \mathcal{H}^2 \tag{5.4}$$

such that

$$H_i \cap H_j = \emptyset, \ \forall i \neq j, \tag{5.5}$$

$$\cup_{j=1}^{B} H_j = \mathcal{H}^2, \tag{5.6}$$

where B is the total number of bins. Each normal from the map belongs to a bin $\mathbf{t}_k \in H_j$, and therefore for a sufficiently fine binning all normals inside a bin are nearly identical. This allows us to group the terms in f_A^α by bin index j

$$f_A^\alpha(\mathbf{i}, \mathbf{o}, \mathbf{n}) \approx F(\mathbf{i}, \mathbf{h}) \sum_{j=1}^{B} C(\mathbf{i}, \mathbf{o}, \mathbf{b}_j, \mathbf{n}) W_j, \tag{5.7}$$

where $\mathbf{b}_j \in H_j$ is a normal in bin j and the bin weights are

$$W_j = \sum_{k | \mathbf{t}_k \in H_j} w_k. \tag{5.8}$$

Consequently, the bin weights W_j also sum up to one like the texel weights w_k. The bin weight represents what portion of the texture space area in the region A has normal in a given bin.

Therefore, if we have an efficient way of computing the bin weights W_j, we can use Equation (5.7) to compute the binned approximation of the filtered BRDF. This can be achieved using the *integral histogram* data structure, which we introduce next.

5.3.3 Integral histograms in 2D

For our exposition we describe 2D IHs, however, they are directly generalized to higher dimensions. We start by defining the related concept of a summed-area table (SAT) [Crow, 1984]. It is a cumulative table of a 2D array used for fast integral look-ups in arbitrary axis-aligned regions. Given a 2D array $I(i, j) \in \mathbb{R}$, the SAT of I is again a 2D array of the same size

$$\mathrm{SAT}_I(i, j) = \sum_{k=0}^{i} \sum_{l=0}^{j} I(k, l). \tag{5.9}$$

Given an arbitrary axis-aligned region $i_0 \leq i \leq i_1,\ j_0 \leq j \leq j_1$ in I, the SAT is used to efficiently look up the sum of I over it

$$\mathrm{SAT}_I(i_1, j_1) - \mathrm{SAT}_I(i_1, j_0) - \mathrm{SAT}_I(i_0, j_1) + \mathrm{SAT}_I(i_0, j_0). \tag{5.10}$$

Note that this is a discrete application of the Fundamental Theorem of Calculus in 2D.

Integral Histograms (IHs) are a natural extension of SATs. For a 2D data set with a specified binning strategy, the idea is to build SATs on the indicator functions of the bins [Ballester-Ripoll and Pajarola, 2019]. Specifically, for a data set I with binning β the indicator functions are

$$I_b(i, j) = \begin{cases} 1, & \beta(I(i, j)) = b \\ 0, & \text{otherwise} \end{cases}, \quad 0 \leq b < B \tag{5.11}$$

where B is the number of bins. The IH of the data set is

$$\mathrm{IH}_{I,\beta} = \{\mathrm{SAT}_{I_0}, \cdots, \mathrm{SAT}_{I_{B-1}}\} \tag{5.12}$$

Consequently, the histogram of any axis-aligned subregion of I can be extracted by evaluating Equation (5.10) for each bin.

In their original form, IHs are ideal due to their fast look-ups, but they have three considerable disadvantages:

- **High memory requirements:** the higher the number of bins the sparser the bin indicator functions are. This redundancy increases the memory footprint proportionally to the number of bins. Therefore, it is not unusual for IHs to be unviable due to them exceeding the available memory for a given task [Ballester-Ripoll and Pajarola, 2019].
- **Slow construction:** the construction speed can be too slow for some applications, especially for a large number of bins.
- **Only axis-aligned look-ups:** traditional IHs with fast look-ups are restricted to axis-aligned rectangle regions. In turn, this would imply that we would have to use axis-aligned pixel footprints, which could be undesirable.

Note that if we build an IH of the normal map with a given binning we can compute the bin weights W_j in Equation (5.7) efficiently for any axis-aligned region in texture space $A^{\perp}$, that additionally do not split texels.

5.4 Our solution

There are practical concerns regarding the direct use of Equation (5.7) with an IH. Although a single bin can be queried cheaply using Equation (5.10), a large number of bins can lead to significant overhead. The other practical challenges stem from the issues of the classical IHs, which are discussed in Section 5.3.3. In this section, we address these issues and describe our filtering algorithm.

5.4.1 Beckmann flake roughness

First, we specify the microfacet BRDF f^α. We use the Beckmann microfacet BRDF with Smith shadowing-masking function. The corresponding equations can be found in Appendix C, Equations (C.1) and (C.2). The Beckmann microfacet distribution D^α is a Gaussian distribution of slopes with standard deviation $\sigma = \frac{\alpha}{\sqrt{2}}$ [Heitz, 2014]. The tails of D^α are exponentially bounded, and therefore, 95% of its microfacets are within 2σ and nearly all of them are contained within 3σ. We selected the Beckmann distribution with this property in mind because the bin weights W_j in Equation 5.7 do not need to be computed for all bins outside of this region. Indeed, these terms of the sum will be multiplied by the tails of D^α and will have a negligible contribution to the BRDF value.

Given a threshold 3σ where we cut the tails, the truncated part of D^α lies in a cone of angle $\theta_0 = \arctan(3\sigma)$ with radius $\sin(\theta_0)$. Subsequently, we can evaluate Equation (5.7) by only considering a smaller number of bins B_0 that sufficiently cover this cone:

$$f_A^\alpha(\mathbf{i}, \mathbf{o}, \mathbf{n}) \approx F(\mathbf{i}, \mathbf{h}) \sum_{j=1}^{B_0} C(\mathbf{i}, \mathbf{o}, \mathbf{b}_j, \mathbf{n}) W_j. \tag{5.13}$$

5.4.2 Binning strategy

We partition the bounding square $[-1, 1]^2$ of the unit disk $\mathcal{D}$ uniformly into $b \times b$ bins, each with index $j \in [0, b^2 - 1]$, see Figure 5.1 (left). Hence, each normal on the unit disk belongs to a single bin. The bins that do not overlap with $\mathcal{D}$ are empty ($B \leq b^2$). Then we can efficiently implement the binning function β and its inverse:

- $\beta(\mathbf{m}) = j$: Given a normal $\mathbf{m} \in \mathcal{H}^2$, we can find the index of the bin j which contains it: $j = \lfloor b(0.5m_x + 0.5) \rfloor + b\lfloor b(0.5m_y + 0.5) \rfloor$.
- $\beta^{-1}(j, \xi_0, \xi_1) = \mathbf{m}$: We can sample a random normal inside a given bin j: $(m_x,\ m_y) = (2(\lfloor j\%b \rfloor + \xi_0)/b - 1,\ 2(\lfloor j/b \rfloor + \xi_1)/b - 1)$, where ξ_0 and ξ_1 are uniform random variables.

Our key idea is to choose the bin resolution b depending on the flake roughness α in such a way that the number of contributing bins B_0 is a small constant. Thus, surfaces with lower roughness will have higher bin resolutions. This can be done in several ways, but we found that the following approach works well in practice. We take a square neighborhood of bins centered at the bin which contains the half vector by taking two bins in each direction, a total of 25 bins.

Let the 3σ cone of radius $\sin(\theta_0)$ is centered around the half vector $(0, 0, 1)^T$. Our goal is to choose b so that the neighborhood of bins sufficiently covers this

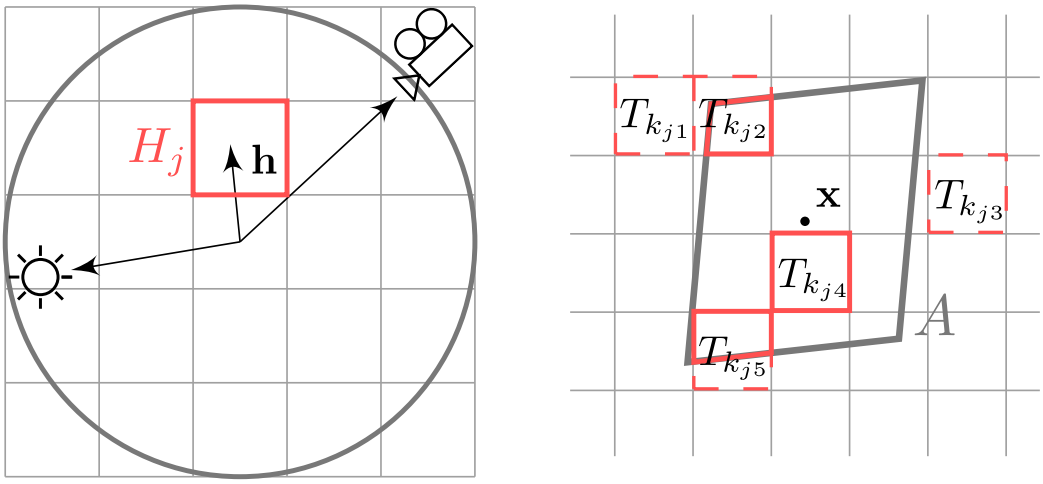


Figure 5.1: **Bins and their corresponding texel positions. Left:** The unit disk, which is partitioned into 5×5 bins. The half vector $\mathbf{h}$ is inside the bin H_j. **Right:** The pixel footprint A in the texture space. All 5 texels corresponding to the j-th bin $\mathcal{B}^{-1}(j) = \{k_{j1}, \ldots, k_{j5}\}$ are drawn in red. Three of them overlap with A, thus contributing to the bin weight W_j.

cone. We set b in such a way that the ratio between the 25 neighborhood bins and the total number of b^2 bins approximates the ratio between the area of the cone's bounding square $4\sin^2(\theta_0)$ and the square of area 4 which bounds the unit disk (i.e., $[-1, 1]^2$):

$$b = \left\lfloor \frac{5}{\sin(\theta_0)} \right\rfloor . \tag{5.14}$$

Lastly, we omit the four corners of this square neighborhood, because they are mostly outside the 3σ cone and end up with $B_0 = 21$, see Figure 5.2.

5.4.3 Evaluation and sampling

Using the binning β we transform the input normal map into a *bin map* $\mathcal{B}$, which is an integer map of bin indices corresponding to normal map elements:

$$\mathcal{B}(k) = \beta(\mathbf{t}_k) = j, \quad \mathbf{t}_k \in H_j. \tag{5.15}$$

See an illustration of the bin map in Figure 5.3. Once the bin map is computed we do not store the original normal map because all necessary components of our algorithm are based on $\mathcal{B}$.

We evaluate the BRDF as follows:

- Find the bin of the half vector $j = \beta(\mathbf{h})$.
- Find all nearby $B_0 - 1$ bins that belong to the neighbourhood of j, see Figure 5.2.
- Compute all B_0 bin weights W_j of these bins.
- For all nonzero bin weights, compute the texel contribution function C at bin centers and accumulate the result, see Equation (5.13).

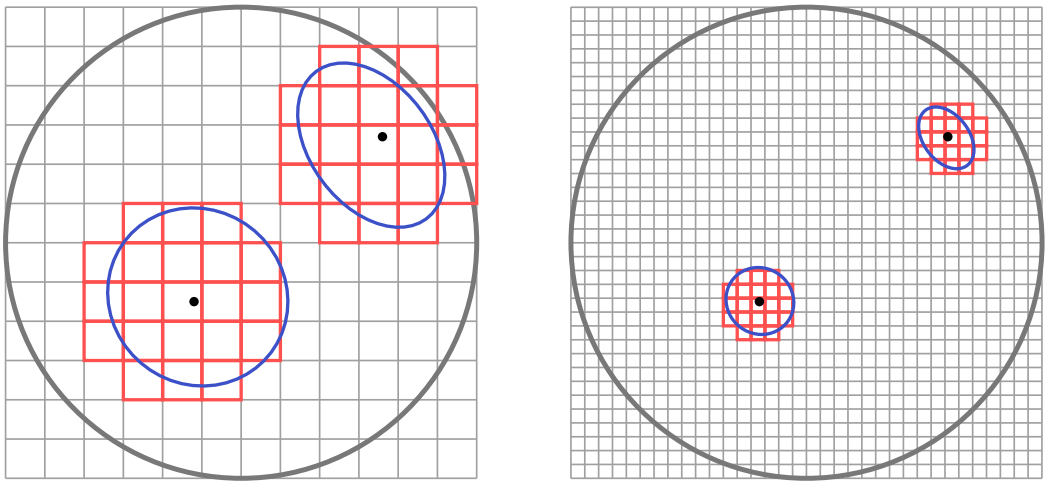


Figure 5.2: **Bin size vs. Beckmann flake roughness.** Shown is the unit disk with entries for two half vectors (black dots), and their corresponding 3σ cones (blue). **Left:** bin resolution 12×12 for Beckmann flake roughness $\alpha = 0.2$ and **Right:** 34×34 for $\alpha = 0.07$. See Table 5.5 for the memory usage of our method for different bin resolutions.

- Apply Fresnel term for conductors or dielectrics.

To use this BRDF in the multiple importance sampling (MIS) framework [Veach and Guibas, 1995], we also provide a sampling technique for this BRDF:

- Sample a random point in A.
- Find the corresponding texel T_k.
- Look up its bin index from the bin map $j = \mathcal{B}(k)$.
- Compute the center normal of this bin $\mathbf{b}_j = \beta^{-1}(j, 0.5, 0.5)$.
- Generate a random normal with Beckmann distribution $\mathbf{m} \propto D^\alpha$ centered at $\mathbf{b}_j$.
- Compute the BRDF direction by reflecting the outgoing direction $\mathbf{o}$ from the generated micro-normal: $\mathbf{i} = \text{reflect}(\mathbf{o}, \mathbf{m})$.

The corresponding probability is

$$p(\mathbf{i}) = \frac{1}{4(\mathbf{i} \cdot \mathbf{h})} \sum_{j=1}^{B_0} D^\alpha(\mathbf{h}, \mathbf{b}_j) W_j. \tag{5.16}$$

The rest of this section describes how we compute the bin weights W_j.

5.4.4 Inverse Bin Map (IBM)

We developed a novel variant of IH data structures that we call *Inverse Bin Map*. Its memory footprint and construction time are practically independent of the number of bins. Additionally, it natively supports arbitrary-shaped query

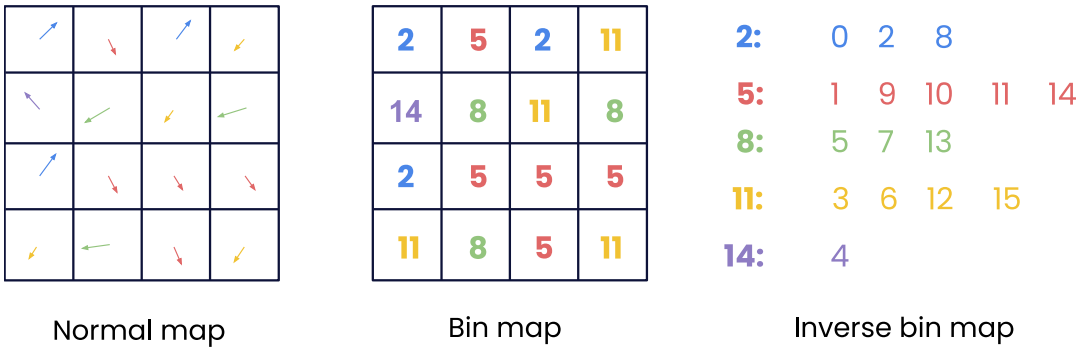


Figure 5.3: **Data structures for sampling and evaluation. Left:** The original normal map. **Middle:** The bin map is a map of bin indices that is created from the original normal map. The normal map is quantized using a binning strategy, see Equation (5.15). As a result, normals that are pointing in similar directions are assigned the same bin index. **Right:** The inverse bin map is built from the bin map, and it consists of concatenated lists of texel positions. Each such list contains all texel positions in the bin map which correspond to a given bin. For example, all texels with normals that are inside the bin with index "2" (cyan) are found in positions "0", "2" and "8" in the bin map.

regions. These critical advantages come at the price of raising the look-up cost to logarithmic, as with WaveletSAT.

A key observation is that in our context sampling and evaluation are inverse operations. Sampling finds the incoming light direction $\mathbf{i}$ which is reflected from a generated microsurface normal, while the evaluation finds all contributing microsurface normals given the direction $\mathbf{i}$. As usual, the inverse problem is the harder of the two. We notice that sampling based on the bin map $\mathcal{B}$ is a very efficient operation, while the evaluation would require looping over all bin map texels inside the filtering region A. Potentially, this is inefficient since the number of texels could be arbitrarily large.

Therefore, we designed a data structure that can act as the inverse of the bin map, to make the reverse operation efficient. Since the bin map $\mathcal{B}$ maps a texel position to a bin index, we define the inverse bin map $\mathcal{B}^{-1}$ as the mapping from a bin index to the list of texel positions of all the texels with the given bin index:

$$\mathcal{B}^{-1}(j) = \{k_{j1}, k_{j2}, \cdots, k_{jn}\}, \quad \mathcal{B}(k_{ji}) = j, \quad i = 1..n, \tag{5.17}$$

where n is the total number of normal map normals in bin j. All lists of positions $\mathcal{B}^{-1}(j)$ are concatenated in a single array of size N (note that $|\mathcal{B}^{-1}| = |\mathcal{B}|$). Essentially, a map of all normal map texel positions that are grouped by bin index, see Figure 5.3.

We use $\mathcal{B}^{-1}$ to compute all bin weights W_j by selecting the subset of texels in $\mathcal{B}^{-1}(j)$ that overlap with A: in Figure 5.1, only texels $\{k_{j2}, k_{j4}, k_{j5}\}$ overlap with A.

Note that in a typical normal map for some bins the list of texels $|\mathcal{B}^{-1}(j)|$ is too large for an efficient linear traversal. Therefore, we construct a 2D hierarchy for each bin j such that $|\mathcal{B}^{-1}(j)| > L$, where L is a fixed leaf size ($L = 10$ is our

implementation).

Two additional data structures accompany $\mathcal{B}^{-1}$:

- **Index $\mathcal{I}$:** Given a bin index, it returns the size of $\mathcal{B}^{-1}(j)$ and an offset. For $|\mathcal{B}^{-1}(j)| <= L$ the offset is the start of the list $\mathcal{B}^{-1}(j)$ in $\mathcal{B}^{-1}$. This list is a single leaf and is queried linearly: all texels $k \in \mathcal{B}^{-1}(j)$ are tested directly for intersection with A and w_k are computed, see Equation (5.3). If $|\mathcal{B}^{-1}(j)| > L$ the offset is the start of the 2D hierarchy in the forest $\mathcal{F}$. $\mathcal{I}$ is implemented as an array of size B.

- **Forest $\mathcal{F}$:** A forest of 2D kd-trees, one for each bin j such that $|\mathcal{B}^{-1}(j)| > L$. During the construction of the forest, we follow several conventions. First, we always split the larger side of the node in the middle, so that the split dimension and split position do not need to be stored. Second, we sort each texel list so that for each tree node its texels are consecutive in $\mathcal{B}^{-1}$. This serves two purposes for the traversal: cache coherence is improved, and the size of each node is implicitly propagated as offsets to the node start and end in $\mathcal{B}^{-1}$. This property provides prefiltering data, so if a node is entirely inside the filtering region A, its total area is immediately returned, see Figure 5.4. Lastly, to achieve memory efficiency and to favor serialization, we pack the whole forest topology in a single integer list.

Usually, IH is used to query the whole histogram with all B bins, or as in our case sub-histogram of B_0 bins together. The inverse bin map is designed with this proposition in mind. Since all kd-trees have the same splitting planes, therefore the same node sizes. We implement traversal for a fixed number of bins that computes the intersections between tree nodes and the region A once for all queried bins. While the tree nodes are axis-aligned, we use a parallelogram approximation of the pixel footprint A based on ray differentials [Igehy, 1999]. In principle, IBM can work with arbitrarily shaped query regions by providing the proper node-region intersection procedures.

5.5 Discussion and results

As pointed out in Section 5.2.4, our method belongs to the family of techniques for the accurate rendering of glints. Here we discuss how our method compares to these previous techniques and provide results.

5.5.1 Glints in production rendering systems

Some properties of normal map filtering techniques are of great importance for practical applications. For example, the integration of accurate glint rendering shaders into production rendering systems requires a reasonable balance of these method properties:

- **Roughness control.** Users usually expect to be able to control the roughness of the surface, regardless of the normal map being used. While the normal map defines the main surface features, the roughness parameter

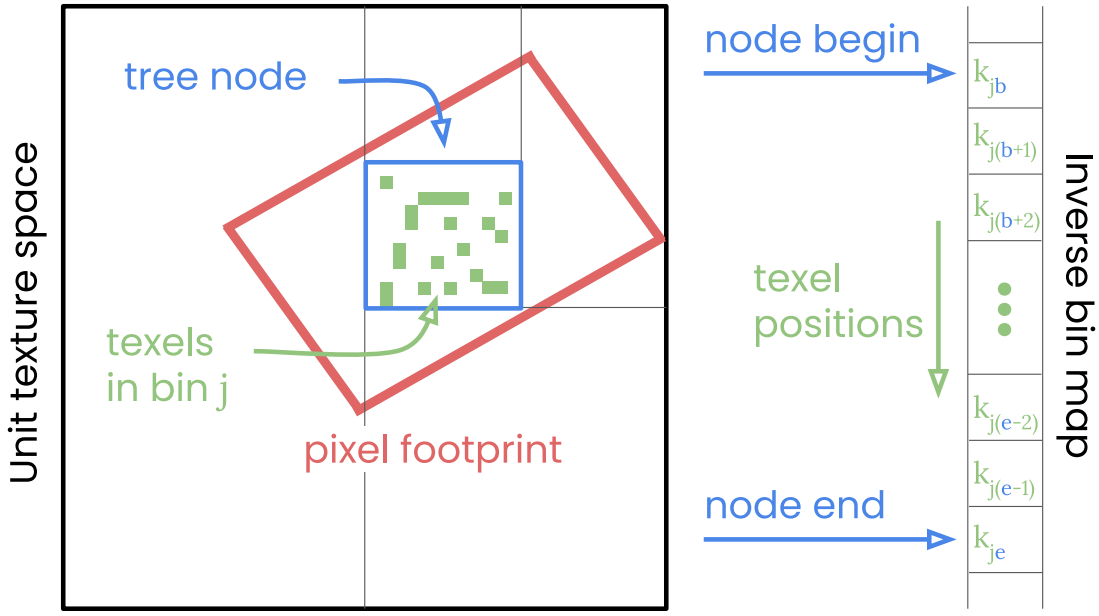


Figure 5.4: **Kd-tree traversal with prefiltering.** A key advantage of our method is its prefiltering, which we illustrate here. The **unit texture space** is drawn on the left, and a part of the **inverse bin map** is on the right. During an evaluation, we want to compute the area of the intersection between the pixel footprint and all texels in bin j. In the figure, we have a large pixel footprint, and the list of texel positions in the inverse bin map corresponding to bin j has a kd-tree in $\mathcal{F}$. If during traversal a tree node (blue) is contained in the pixel footprint, we stop the traversal and immediately return the total area corresponding to the node. This is possible because we sort all texel positions in the inverse bin map $\mathcal{B}^{-1}$ so that each kd-tree node has its texel indices consecutive in the memory layout of $\mathcal{B}^{-1}$. This layout is shown on the right. During traversal, we keep track of the "begin" texel position k_{jb}, which is the first texel included in the node, and the "end" texel position k_{je}, which is the first texel excluded from the node ($k_{j(e-1)}$ position is the last texel included in the node). Thus, the total number of texels in the node is $e - b$, and the total area corresponding to the node is this same number, scaled by the area of a single texel.

allows them to change the appearance of the glints — low roughness values produce very sharp and sparkling glints while higher roughness values decrease the sparkling and produce smoother glints.

- **Instant pre-computation.** It is very common for normal maps to be created in external tools, or to be generated on the fly based on a set of input parameters. Furthermore, production scenes usually have a large number of materials and maps. Therefore, an instant pre-computation for a reasonably large map is needed to ensure quick iterations when choosing the desired appearance. In any case, a few seconds or even minutes of pre-computation for a single normal map is rarely permissible.

- **Low memory usage.** Low memory requirements for a single map are also important since the number of maps is usually very high in some production scenes. In practice, a few MIP maps for a single normal map is considered

reasonable.

- **Multiscale (prefiltering).** All normal map filtering techniques employ a form of filtering – this could be either a specialized data structure as is the case of Yan's method, or a numerical method as is the case of SMS. However, prefiltering is a stronger property. It is related to some filtering information that is pre-computed or organized during the pre-computation stage of the method which is directly used during rendering. Therefore, the prefiltering property is tightly related to the performance of the technique at different scales. The performance of all filtering techniques that do not employ a prefiltering strategy deteriorates for large filtering regions — that is for large *texel-to-pixel ratios*.

  Note that increasing the texel-to-pixel ratio is an extremely common scenario in practical renderer usage. It takes place when:

  – The normal-mapped object is moving away from the camera.

  – The camera is zooming out.

  – The scene is rendered at a lower resolution.

  – The tiling of the texture is increased.

  In our method, the contributing texels in each pixel increase with Beckmann flake roughness. Therefore, our data structure provides prefiltering: the aggregated projected area of texels with similar normals inside the pixel is efficiently computed for high texel-to-pixel ratios.

In Table 5.2, we show which techniques of the class of accurate filtering for glints have these properties. Our method has all of them.

	Gamboa	Zeltner	Yan	Ours
Roughness control	✗	✗	✗	✓
Instant pre-computation	✗	✓	✗	✓
Low memory usage	✗	✓	✗	✓
Multiscale (prefiltering)	✓	✗	✗	✓

Table 5.2: Properties of glint rendering techniques — Gamboa et al. [2018], Zeltner et al. [2020], Yan et al. [2016] and our method [Atanasov et al., 2021].

5.5.2 A remark on Yan's method

The method of Yan et al. [2016] provides two modes of operation: flat Gaussian elements which represent the non-interpolated normal map, and curved Gaussian elements which represent a smooth interpolated surface. We support only flat un-interpolated normal maps, and therefore our surface is very similar to Yan's flat elements, see Figure 5.5.

Our proposed technique aims to provide a practical filtering solution not only for specular surfaces, but also for surfaces with low roughness where the appearance changes dramatically, but filtering is still beneficial. To this end, our

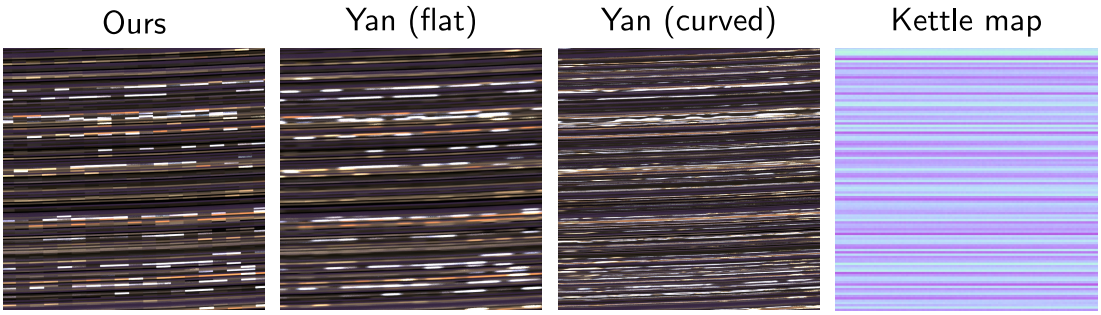


Figure 5.5: **Surfaces of different filtering techniques.** A 50× zoom is applied to the **Kettle scene** [Zeltner et al., 2020] to observe the surfaces of three different models. From left to right: our method, Yan et al. [2016] flat and curved Gaussian elements. The rightmost image is the normal map. Our method does not support normal map interpolation and renders a piecewise flat surface similar to the flat Gaussian elements. Yan's flat elements are blurrier due to the Gaussian footprint whereas our method uses a pixel-wide "box" filter. The original kettle scene can be seen in Figure 5.9.

solution exposes a single parameter, Beckmann roughness, which provides meaningful artistic control. This parameter, which we also refer to as *Beckmann flake roughness*, is conceptually similar to the intrinsic roughness of Yan et al. [2016]. But the intrinsic roughness is designed and demonstrated to work in a small operational range that represents specular surfaces. Surfaces with slightly larger roughness values are out of the scope of Yan et al. [2016] — in these cases the method loses energy, see Figure 5.6. Note that in all comparisons we match our Beckmann flake roughness to Yan's intrinsic roughness using the relation $\alpha = \sqrt{2}\sigma_r$ [Heitz, 2014].

5.5.3 Comparison with the state of the art

The method of Yan et al. [2016] is widely regarded as the state-of-the-art technique for high-resolution specular normal maps [Zeltner et al., 2020, Zhu et al., 2022, Deng et al., 2022]. All comparison results in this subsection are based on the original code, scenes and scripts provided by Zeltner et al. [2020], which includes the original implementation of SMS and Yan's method. We also implemented our method as a Mitsuba 2 BRDF plug-in [Nimier-David et al., 2019]. The source code is available on the project page, see Attachments. Comparisons were rendered on an *AMD Ryzen Threadripper 3970X* machine.

As discussed earlier, we want to investigate the scenario of increasing the number of texels that fall within a typical image pixel. Figure 5.7 shows equal-time comparisons for our method, Yan's method and SMS, for a scene with many texels in each pixel. The SMS method cannot find all subpixel glints, which results in darkening due to lost energy, thus it is excluded from subsequent performance comparisons.

We perform a more detailed comparison between our method and Yan's method by rendering the original scenes from Zeltner et al. [2020], and by increasing the tiling of the input texture 8 times which results in increasing the texel-to-pixel ratio 8^2 times. The results from these comparisons are presented

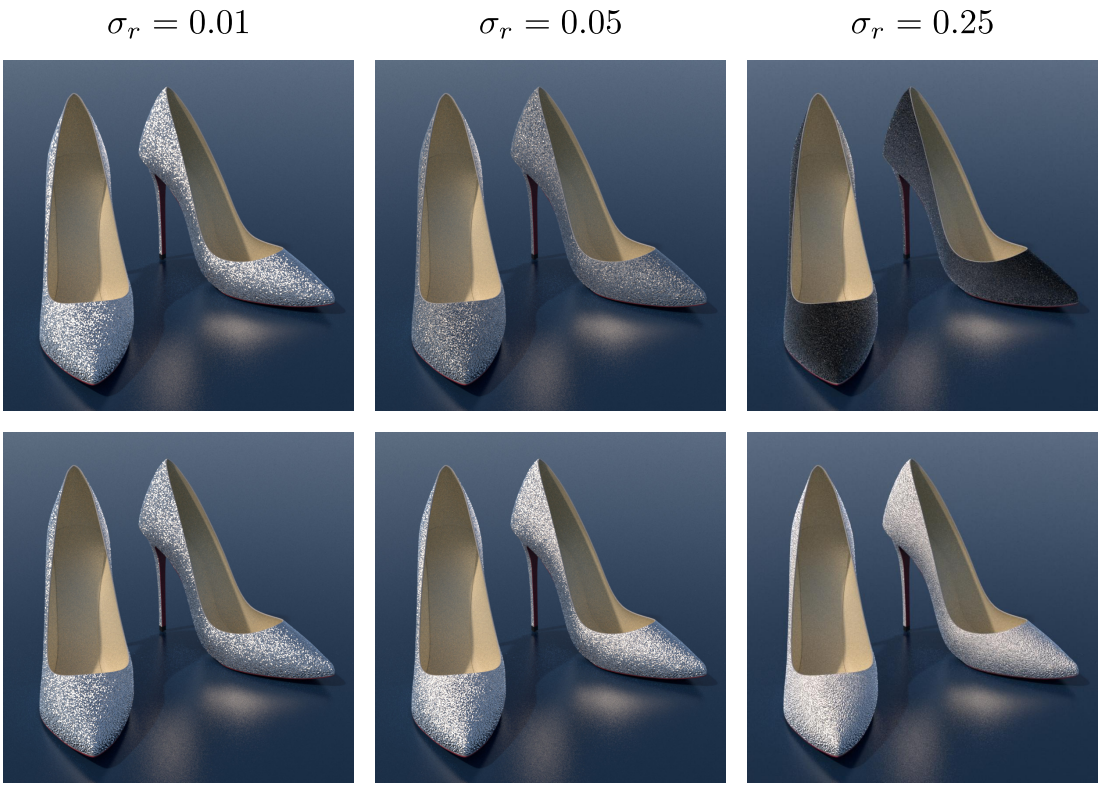


Figure 5.6: **Increasing roughness test.** Renders of the **Shoes scene** [Zeltner et al., 2020] with Gaussian flat elements (top row) [Yan et al., 2016] and with our method (bottom row). The three columns show increasing roughness σ_r $(0.01, 0.05, 0.25)$. Note that the method of Yan et al. [2016] is designed to render specular surfaces and does not conserve energy for increased roughness.

in Figures 5.8 and 5.9. Our method demonstrates very similar convergence rates to Yan's flat elements for equal matched roughness $\sigma_r = 0.005$. The convergence plots for both scenes clearly show a tendency: as the number of texels per pixel grows, Yan's curved elements convergence declines.

Note that both Yan's flat and curved elements use the same hierarchy and intrinsic roughness values. The difference in the performance in "$\times 8$ tiles" scenes (right columns in Figures 5.8 and 5.9) is because, in the case of the curved elements, there are 1-2 orders of magnitude more contributing elements to be processed in comparison with the case of the flat elements. The method of Yan et al. [2016] must locate all Gaussian elements inside the pixel that contribute to the reflection. Then contributing elements are processed individually and weighted against the Gaussian footprint. This is inefficient for high texel-to-pixel ratios and many contributing elements.

We also provide additional results from our method with increased roughness to demonstrate that our convergence is not impeded by the growing number of contributing texels. In fact, it improves slightly.

Additionally, the **Shoes scene** demonstrates that for some normal maps, the appearance of Yan's curved elements can be matched with our method with increased roughness, see Figure 5.8 (**Yan (curved)** and **Ours (rough)**).

Furthermore, our method is $90\times$ more memory efficient and the pre-processing is around $30\times$ faster than Yan's method with flat Gaussian elements, see Tables

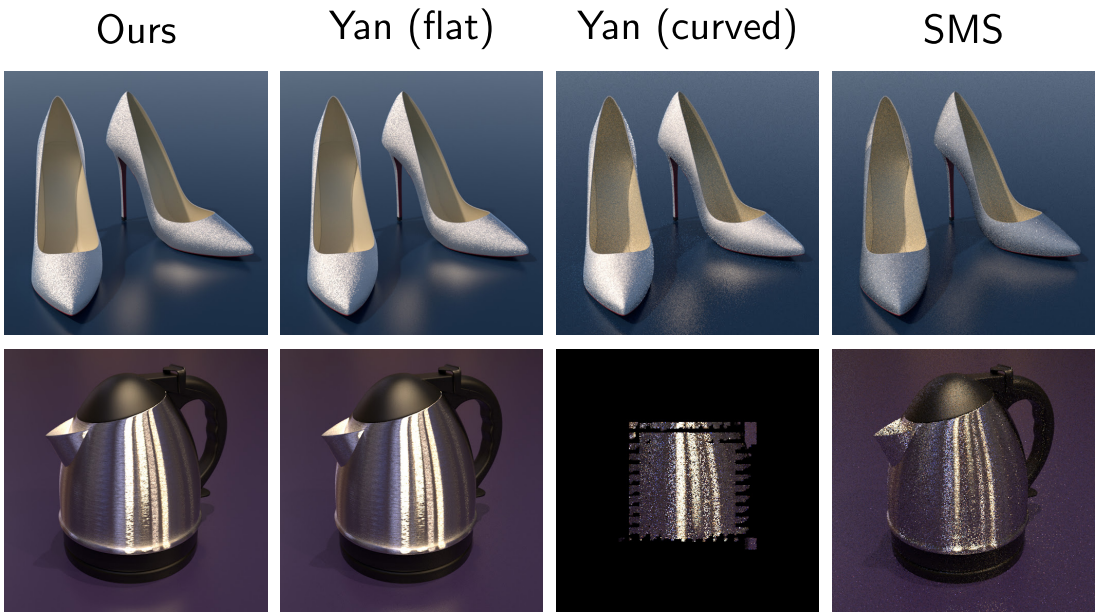


Figure 5.7: **Equal-time comparison with state-of-the-art techniques.** We render **Shoes** and **Kettle scenes** [Zeltner et al., 2020] in resolution 512×512 with four different methods: our method [Atanasov et al., 2021], Yan's flat and curved Gaussian elements [2016] and SMS [Zeltner et al., 2020]. The tiling of the textures is increased $\times 8$ in order to compare the methods for large texel-to-pixel ratios. All images were rendered for 1 minute on an *AMD Ryzen Threadripper 3970X machine*. While our method and Yan's flat elements show comparable performance, the other two approaches have issues. Yan's curved elements become very slow due to the lack of prefiltering. See also the convergence plots in Figures 5.8 and 5.9. This issue has been identified also by Deng et al. [2022]. On the other hand, SMS is unable to find all the glints in the pixel, resulting in substantial darkening.

5.3 and 5.4.

	Ours	Yan (flat)	Yan (curved)	Ours (rough)
Shoes scene	36MB	3.2GB	19.3GB	36MB
Kettle scene	36MB	3.4GB	59GB	36MB

Table 5.3: Memory usage of the methods from Figures 5.8 and 5.9.

Our method has low memory requirements for a wide range of Beckmann flake roughness values, and therefore a wide range of bin resolutions, see Table 5.5. For higher resolutions, the index $\mathcal{I}$ takes more memory, but the trees in the forest $\mathcal{F}$ are shallow and take less memory. For lower resolutions, it is the opposite.

5.5.4 Animated glints

We demonstrate our method with different normal maps and lighting scenarios in our supplementary video (see Attachments).

We have implemented a car paint material with metallic flakes, modeled with a normal map. Flakes orientations are sampled from a GTR distribution [Burley,

	Ours	Yan (flat)	Yan (curved)	Ours (rough)
Shoes scene	0.2s	8.2s	56.6s	0.3s
Kettle scene	0.3s	8.4s	241.3s	0.3s

Table 5.4: Pre-processing times of the methods from Figures 5.8 and 5.9.

Beckmann roughness α	0.0025	0.01	0.04	0.16
Bin resolution b^2	942^2	235^2	59^2	15^2
Wing mirror scene (2K map)	43MB	37MB	36MB	36MB
Car wheel scene (1K map)	15MB	9MB	9MB	8MB
Shoes scene (2K map)	40MB	36MB	36MB	36MB
Kettle scene (2K map)	42MB	36MB	36MB	36MB

Table 5.5: The bin resolution and memory usage of our method for varying Beckmann flake roughness.

2012], and the corresponding Smith shadowing-masking function is applied for G_x [Dimov, 2015]. These equations are given in Appendix C. The paint consists of three layers: a specular coat layer, metallic flakes layer filtered with our algorithm and a base layer with mixed diffuse and glossy terms. The transparency of the flake layer is computed using a single-channel MIP map. When we report the memory for our car paint material we include this map in the total.

Additionally, we provide a control for the roughness of the individual flakes, as we show in the **Wing mirror scene**, Figure 5.10.

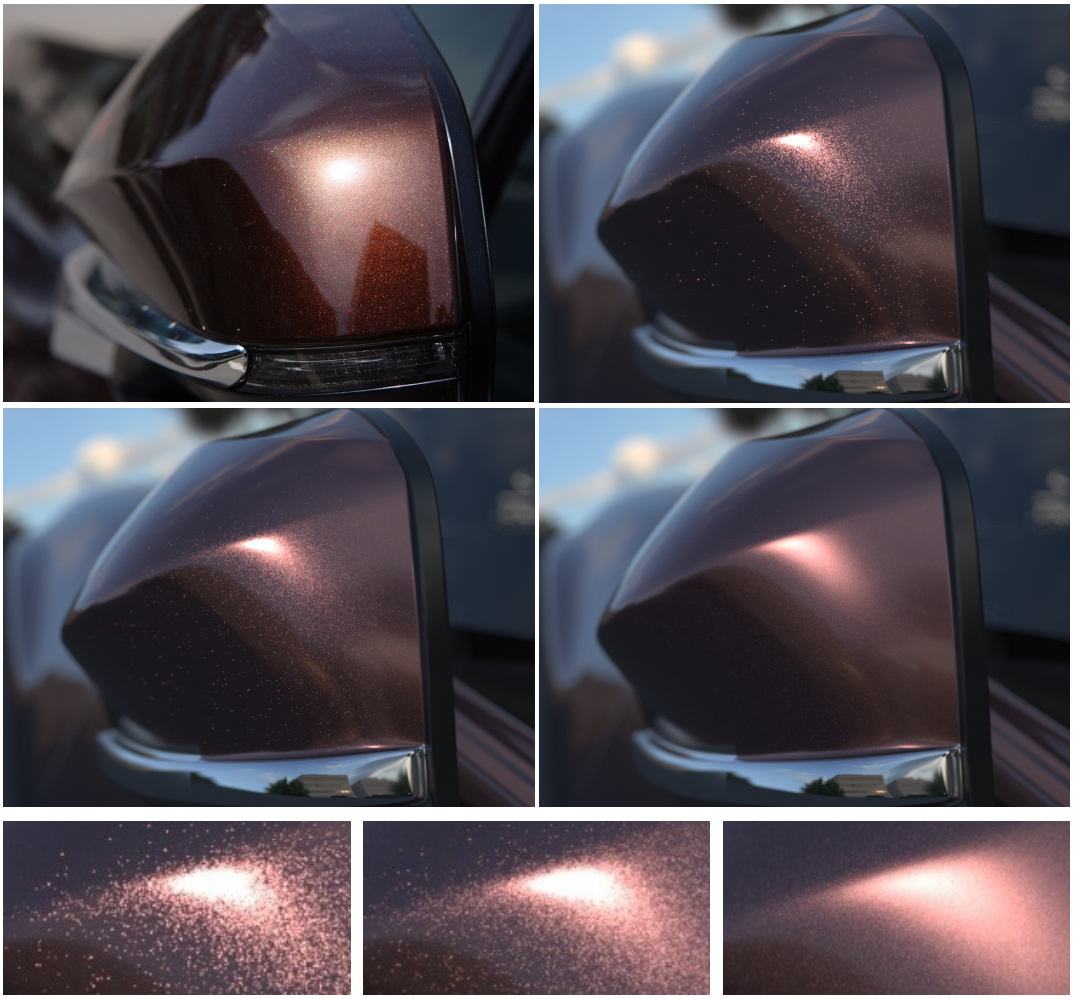


Figure 5.10: **Car paint material with roughness control.** A photograph of a wing mirror (top left) with pronounced glints from metallic flakes that served as an inspiration for our wing mirror scene (all other images). In this scene, the metallic flakes are modeled with a 2K normal map with flakes sampled from a GTR distribution (GTR $\gamma = 1.5$, GTR $\alpha = 0.002$) [Burley, 2012]. Additionally, we provide three renders (top right, middle left, middle right) of this scene with different roughness values of the flakes — $\alpha = 0.0025, 0.01, 0.04$, respectively. The Beckmann flake roughness contributes to the overall appearance and is a useful parameter for artistic control. In the bottom row, we provide three zoomed-in regions from these same renders. Small perturbations of the roughness of the flakes completely change the behavior of the glints. The rendering of such nearly specular surfaces requires some form of filtering, the effect of which is shown in our accompanying video (see Attachments). All the renderings in this figure were done with our proposed normal map filtering algorithm. (The wing mirror scene was created by Tashko Zashev.)

This scene is lit by a sun light and an environment light. In our video, we show that for a range of small Beckmann flake roughness values the filtering is crucial to achieving converged results: renders with equal time stochastic sampling exhibit severe flickering.

Our second scene, the **Car wheel** (Figure 5.11) has a 1K map with scratches that are tiled over the surface to achieve a high texel-to-pixel ratio.

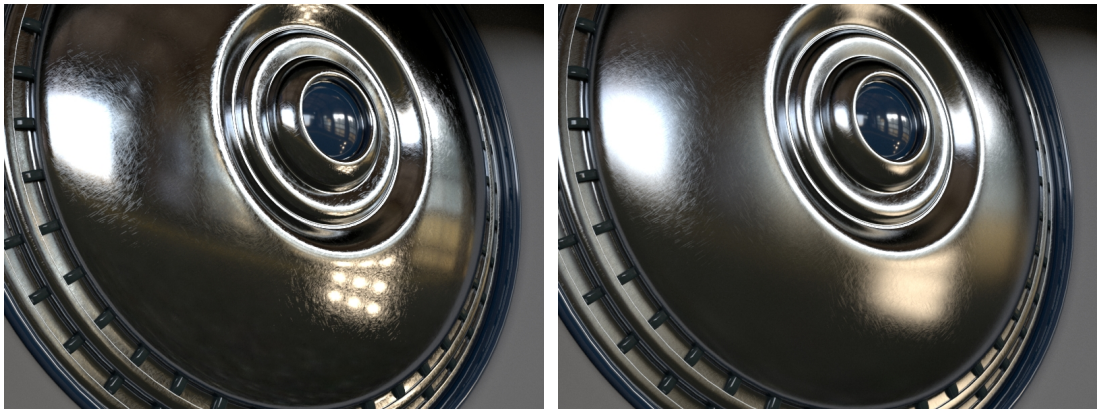


Figure 5.11: **Scratched metal with roughness control. Car wheel scene** with scratch normal map and Beckmann flake roughness 0.01 (left) and 0.04 (right), and filtered with our algorithm. The behavior of this surface in an animation can be seen in the accompanying video: due to our prefiltering, object appearance is temporally stable across frames.

The scene is lit by 9 small area lights and an environment with large light sources. In our video, we observe that the portions of the surface lit by the high-frequency illumination (the 9 small lights) benefit from our filtering technique for low roughness values. As the roughness increases the filtered version is still more stable and some "boiling" can be seen in the stochastic version, however, the benefit of our technique for these scenarios is smaller.

5.6 Conclusion

In this chapter, we presented an efficient normal map filtering method: there are no noticeable pre-computation times, and its memory requirements are very low. Due to prefiltering being applied, our technique does not slow down if large numbers of normal map texels fall within a single pixel: zooming out from a surface with glints does not cause performance issues.

The algorithm we propose filters direct illumination. Our filtering solution is based on an *Inverse Bin Map*: a specialized integral histogram implementation that enables us to perform the necessary lookups with low memory requirements, and at reasonable speeds. We believe that this data structure has the potential to replace IHs in some applications where construction speed, memory efficiency or regions with arbitrary shapes are important.

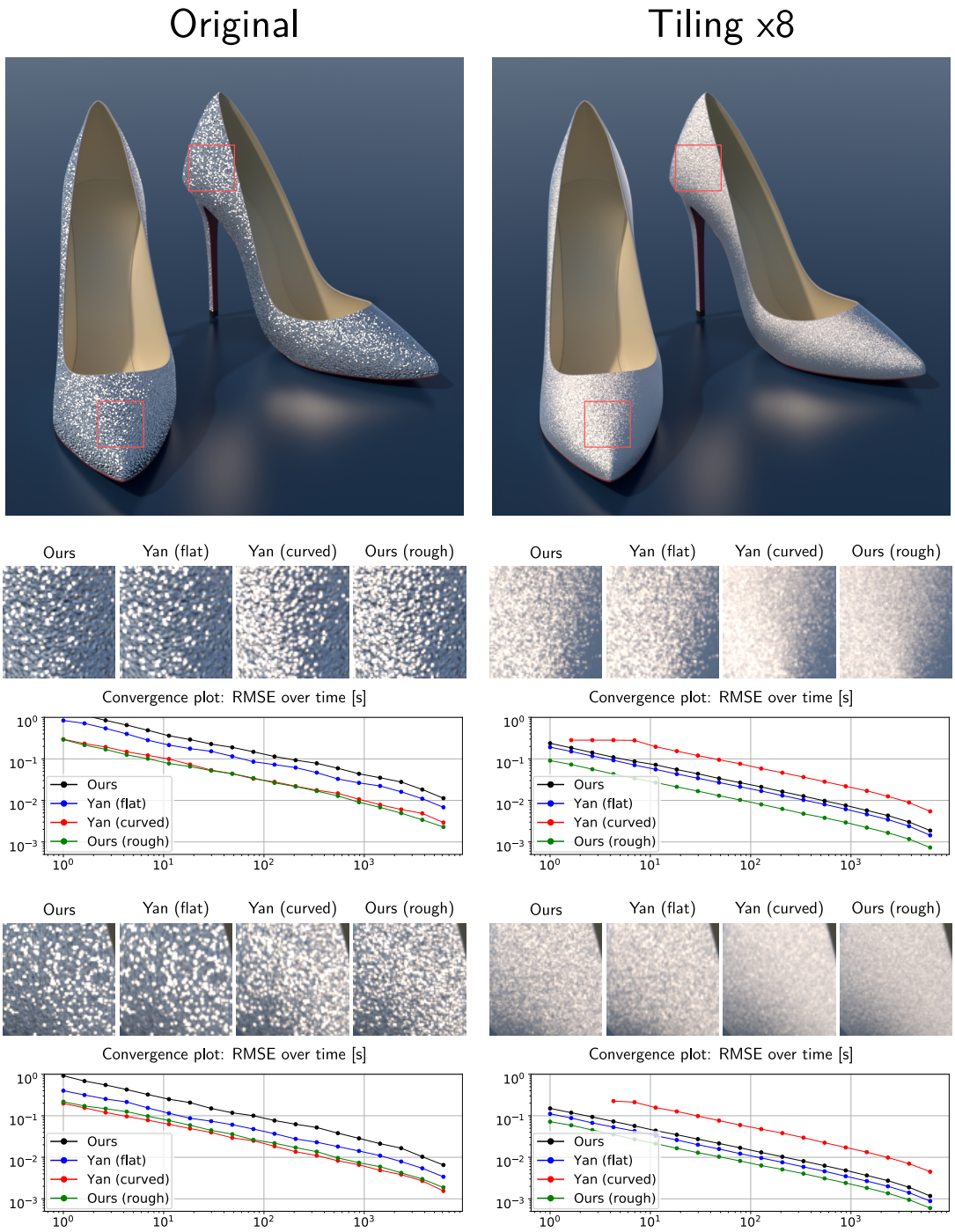


Figure 5.8: **Convergence plots (Shoes scene).** We render two different texture tiles — **left column:** the original scene from Zeltner et al. [2020] and **right column:** modified scene with $\times 8$ tiles of the normal map. These two scenes are rendered in four variations: our method, Yan et al. [2016] flat elements, Yan et al. [2016] curved elements, and our method with slightly increased roughness. The first three variations have equal roughness values $\sigma_r = 0.005$, while the last demonstrates our method with higher Beckmann flake roughness $\alpha = 0.05$ ($\sigma_r = \alpha/\sqrt{2}$), which is not supported by Yan's method. The large images in the top row are rendered with our low roughness variant (the first variant labeled **Ours**). Two regions of size 80×80 are selected from both scenes, and convergence plots are computed for each of them (middle and bottom row). We use the script provided by Zeltner et al. [2020] to compute the plots.

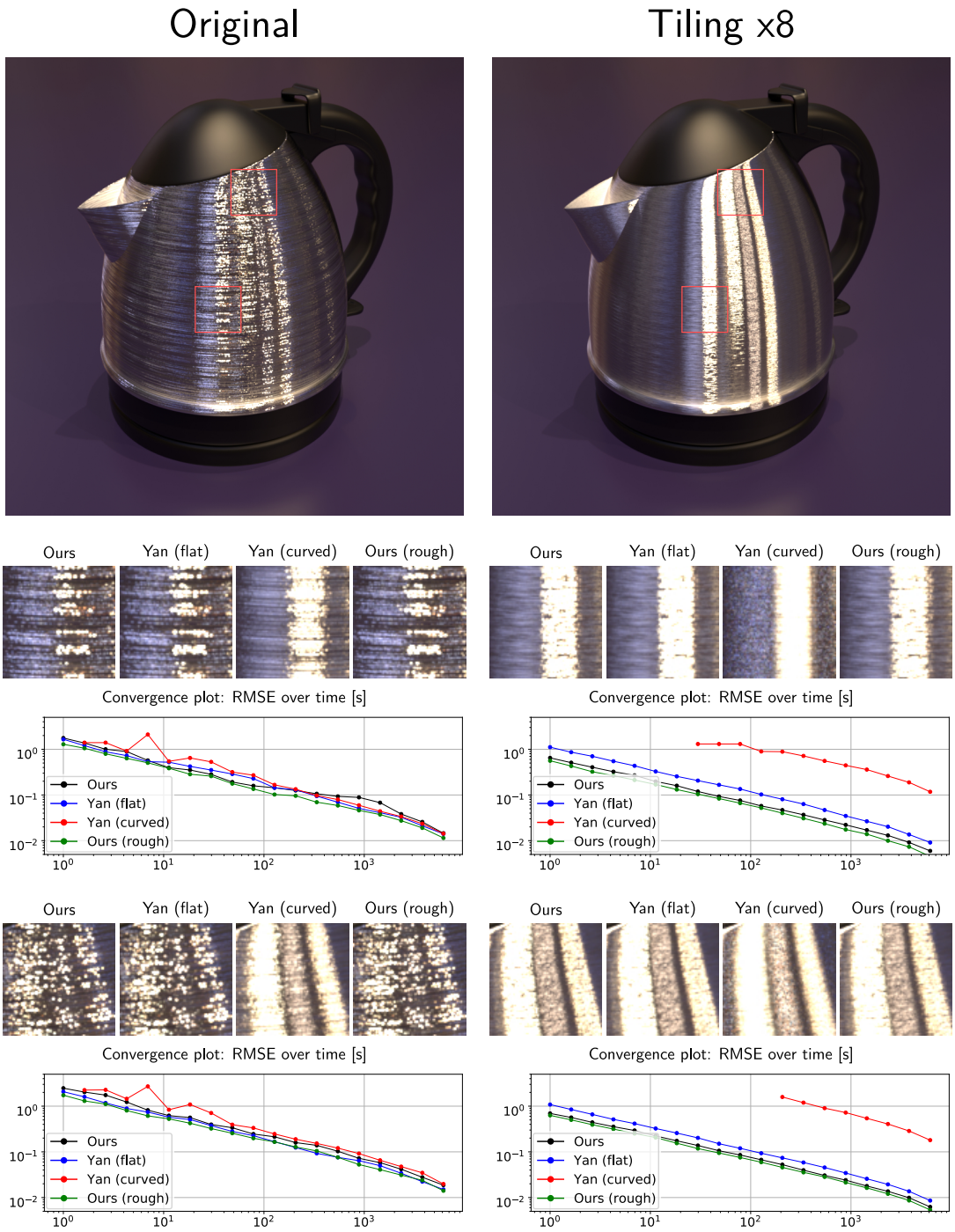


Figure 5.9: **Convergence plots (Kettle scene).** We render two different texture tiles — **left column:** the original scene from Zeltner et al. [2020] and **right column:** modified scene with $\times 8$ tiles of the normal map. These two scenes are rendered in four variations: our method, Yan et al. [2016] flat elements, Yan et al. [2016] curved elements, and our method with slightly increased roughness. The first three variations have equal roughness values $\sigma_r = 0.005$, while the last demonstrates our method with higher Beckmann flake roughness $\alpha = 0.05$ ($\sigma_r = \alpha/\sqrt{2}$), which is not supported by Yan's method. The large images in the top row are rendered with our low roughness variant (the first variant labeled **Ours**). Two regions of size 80×80 are selected from both scenes, and convergence plots are computed for each of them (middle and bottom row). We use the script provided by Zeltner et al. [2020] to compute the plots.

Chapter 6

Conclusion

In this thesis, we have presented two microfacet models that are designed to resemble two distinct visual phenomena — anisotropic surfaces and surfaces with visible microstructure. Both models are tailored to express a wide variety of appearances, and both are very efficient to compute.

Linearly Transformed Microsurface. In Chapter 4, we described a model that transfers all important properties from isotropic distributions to anisotropic distributions — normalization of the microfacet distribution and the shadowing function, NDF and VNDF sampling. It is based on closed-form formulas, thus it is very attractive — analytic microfacet models are usually the most efficient models for computation. With this model for the first time, all microfacet distributions can have anisotropic versions, which are also consistent with existing anisotropic models. We also showed that this model can be utilized to compute the reflection of deforming surfaces.

Multiscale Microfacet Model. In Chapter 5, we presented a practical filtering method that is suitable for high-quality (animated) glints. Among other techniques, it is the first to offer control over the roughness of the individual microfacets, defined by the normal map. We have demonstrated that this auxiliary parameter significantly improves the model's expressiveness. Furthermore, our technique has appealing properties: improved performance in some scenarios and comparable performance in others, instant pre-computation and the lowest memory usage of all methods that filter full normal maps.

6.1 Limitations and future work

We are aware of the following limitations, and we discuss possible improvements.

Our anisotropic model assumes that surface deformations are uniformly transferred to the microsurface geometry. However, the deformations of the microgeometry are highly dependent on the physical properties of the material. For instance, as a balloon is getting inflated not only does its roughness change but also its thickness, see Figure 1.5. Therefore, our model could be used as a reasonable approximation or as a starting point in simulating more complex behaviors. In the future, it will be interesting to explore the problem of deforming skin [Nagano et al., 2015] where our model has the potential to replace run-time convolutions with analytic expressions.

Our normal map filtering algorithm has two main limitations. First, it is

designed to work with Beckmann roughness only. For example, if GGX flake roughness is required, our algorithm must query all hemispherical bins for each evaluation, due to GGX having a non-negligible projected area of microfacets that are oriented in all directions. This will be considerably less efficient than the small number of bins that we query for our Beckmann roughness. Second, our algorithm does not support textured Beckmann flake roughness. Indeed, our bin resolution is chosen in advance, based on a fixed flake roughness value. Perhaps, one can build several IBMs for a range of roughness values, but we leave this for future work. Another avenue for future work would be to develop a hybrid normal map filtering technique that uses fast prefiltering like LEAN mapping [Olano and Baker, 2010] for high-roughness surfaces and our algorithm for the more demanding low-roughness surfaces.

6.2 Production use

Both microfacet models, presented in this dissertation, are currently used in production. They are part of the official installation of Chaos V-Ray.

V-Ray is a production renderer with a rich set of features that is available for many 3D platforms including Autodesk 3ds Max, Trimble SketchUp, Autodesk Maya, Maxon Cinema4D, SideFx Houdini, McNeel Rhinoceros 3D, Autodesk Revit and Blender, and it is spanning a broad spectrum of industries like architecture, automotive, interior design, product design, games, television and film VFX and more [Chaos, 2023b].

The two models are implemented in the following V-Ray material plug-ins:

- **VRayMtl.** This is the main material in V-Ray — it has a wide range of capabilities such as different types of diffuse and glossy reflections, refraction, translucency, cloth's sheen, coat and more [Chaos, 2023d]. The reflection layer is a GTR microfacet BRDF, which prior to our work used to provide non-physical anisotropy due to the lack of analytic anisotropic equations. The anisotropic model described in this dissertation was developed specifically to address this issue.

- **VRayFlakesMtl2, VRayCarPaintMtl2.** *VRayFlakesMtl2* is a single BRDF material that is used as a layer in more complex compound materials. Internally, it generates a flake normal map based on a set of parameters such as flakes' size, density and orientation. Flake orientations are sampled from GTR distribution with tunable roughness and tail exponent. This BRDF is filtered using our proposed multiscale microfacet model. The Flake roughness parameter is exposed to the user interface. *VRayCarPaintMtl2* is a three-layer material: the base layer is a mix of diffuse and GGX lobes, the middle layer is the flakes BRDF and the top layer is a clear (or glossy) coat. The wing mirror scene in Figure 5.10 was rendered with it. The behavior of some *VRayCarPaintMtl2* parameters, along with corresponding renders can be explored at Chaos's documentation page [2023c].

Note that our multiscale model is not currently used as a general normal map filtering technique in V-Ray due to the limitations mentioned above.

Appendix A

Microscope photographs

Microfacet models are capable of predicting a wide range of surface appearances, depending on their microstructure and material properties. To support intuition, we provide a set of microscope photographs of exemplary conductors and dielectrics in Figures A.1 and A.2, respectively. All photos were taken using *Canon D5600* camera and *Olympus SZX7* microscope.

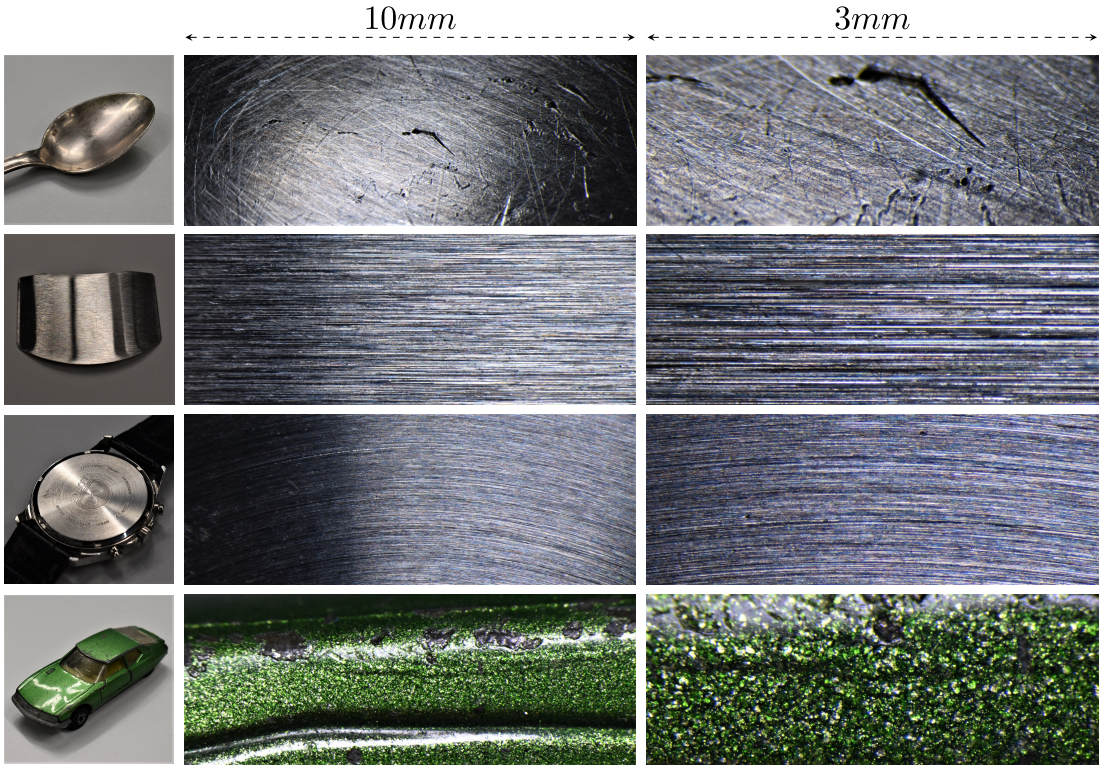


Figure A.1: **Microscope photographs of rough conductors.** From top to bottom: scratched silver spoon; strongly anisotropic brushed steel piece; circular-anisotropic steel back of a watch manufactured using a lathe machine; car toy with metallic paint.

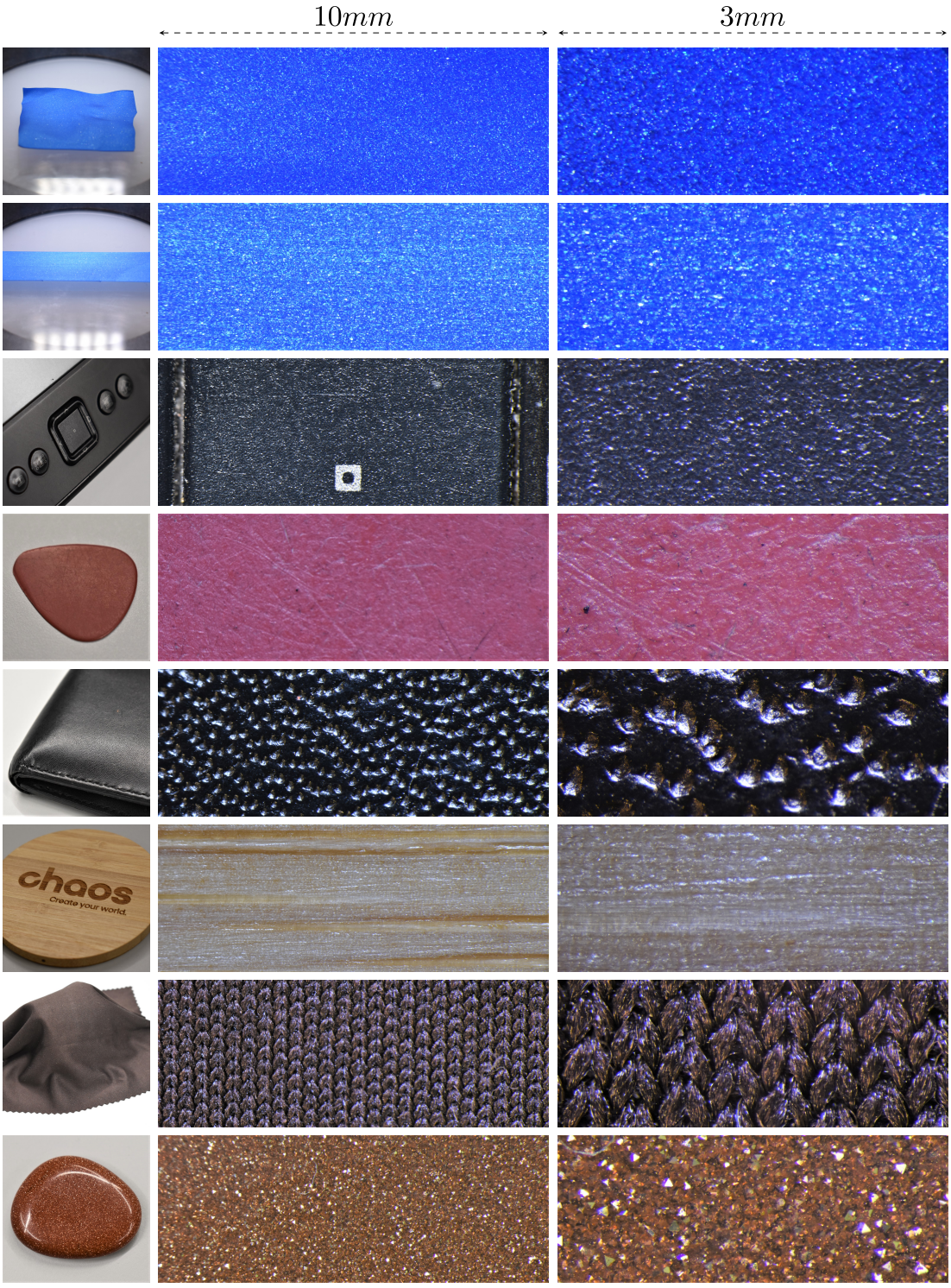


Figure A.2: **Microscope photographs of rough dielectric materials.** From top to bottom: a piece of a rubber balloon (the same from Figure 1.5); the same balloon piece stretched $\times 4$; a button made from rough plastic; a nylon guitar pick; a leather wallet; a weakly anisotropic bamboo wireless charger; a piece of cloth; a goldstone.

Appendix B

Integration on the hemisphere

The material in this appendix is partially based on the book by Pharr et al. [2016]. We denote 3-dimensional vectors with bold small letters $\mathbf{v} = (v_x, v_y, v_z)^T \in \mathbb{R}^3$. The scalar product of two vectors $\mathbf{v}$ and $\mathbf{w}$ is defined as $\mathbf{v} \cdot \mathbf{w} = v_x w_x + v_y w_y + v_z w_z$ and the Euclidean norm of a vector: $\|\mathbf{v}\| = \sqrt{\mathbf{v} \cdot \mathbf{v}}$. The unit hemisphere is

$$\mathcal{H}^2 = \{\mathbf{v} \in \mathbb{R}^3 \mid v_z \geq 0,\ \|\mathbf{v}\| = 1\}, \tag{B.1}$$

and it is isomorphic to the unit disk $\mathcal{D}$: directions $\mathbf{v} \in \mathcal{H}^2$ can be represented on the unit disk by projection $(v_x, v_y) \in \mathcal{D}$, and the z-coordinate can be recovered by the re-projection $v_z = \sqrt{1 - v_x^2 - v_y^2}$.

It is often convenient to express unit directions on the hemisphere in spherical coordinates. Each direction on the unit hemisphere $\mathbf{v} \in \mathcal{H}^2$ can be represented by the pair of angles $(\theta_\mathbf{v}, \phi_\mathbf{v})$, where $\theta_\mathbf{v}$ is the angle between $\mathbf{v}$ and the $\mathbf{z}$ axis (i.e., the up direction) and $\phi_\mathbf{v}$ is the angle between the projection of $\mathbf{v}$ onto the xy plane and the $\mathbf{x}$ axis, see Figure B.1 a). The following equations transform between spherical and Cartesian coordinates:

$$v_x = \sin\theta_\mathbf{v} \cos\phi_\mathbf{v} \tag{B.2}$$

$$v_y = \sin\theta_\mathbf{v} \sin\phi_\mathbf{v} \tag{B.3}$$

$$v_z = \cos\theta_\mathbf{v} \tag{B.4}$$

and

$$\theta_\mathbf{v} = \arccos v_z \tag{B.5}$$

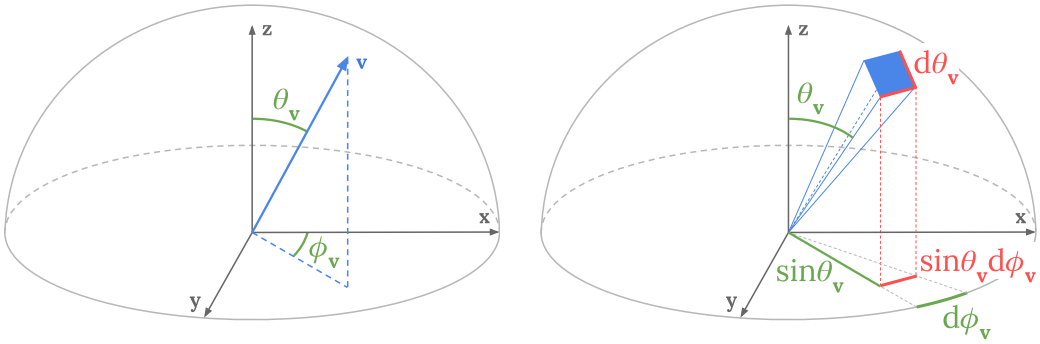
$$\phi_\mathbf{v} = \arctan \frac{v_y}{v_x}. \tag{B.6}$$

Solid angles are a generalization of planar angles: planar angles are measured with the length of the arc that they cut from the unit circle, and by analogy, solid angles are measured with the surface area from the unit sphere that they encompass. Planar angles are measured in *radians*, while solid angles are measured in *steradians* (denoted *sr*).

A real-valued function on the hemisphere is

$$f : \mathcal{H}^2 \to \mathbb{R}. \tag{B.7}$$

To integrate such a function on the hemisphere, we need to define a differential solid angle $\mathrm{d}\mathbf{v}$ around the integration variable $\mathbf{v}$. In spherical coordinates, this



a) Spherical coordinates b) Differential solid angle

Figure B.1: **Illustration of spherical geometry. a) Spherical coordinates:** each direction $\mathbf{v}$ on the (hemi) sphere can be expressed with the pair of angles $(\theta_{\mathbf{v}}, \phi_{\mathbf{v}})$. **b) Differential solid angle:** a spherical quadrilateral (blue) with sides $\mathrm{d}\theta_{\mathbf{v}}$ and $\sin\theta_{\mathbf{v}}\mathrm{d}\phi_{\mathbf{v}}$, the product of which is equal to the differential area of the quadrilateral, see Equation (B.8).

is expressed as the differential area of a spherical quadrilateral, see Figure B.1 b). The first side of this quadrilateral is the arc $\mathrm{d}\theta_{\mathbf{v}}$ and the second side is the arc $\mathrm{d}\phi_{\mathbf{v}}$, downscaled by $\sin\theta_{\mathbf{v}}$. The differential area of the quadrilateral is the product of its two sides

$$\mathrm{d}\mathbf{v} = (\sin\theta_{\mathbf{v}}\mathrm{d}\phi_{\mathbf{v}})(\mathrm{d}\theta_{\mathbf{v}}). \tag{B.8}$$

Therefore, the integral over the hemisphere is expressed as

$$\int_{\mathcal{H}^2} f(\mathbf{v})\mathrm{d}\mathbf{v} = \int_0^{2\pi}\int_0^{\frac{\pi}{2}} f(\theta_{\mathbf{v}}, \phi_{\mathbf{v}})\sin\theta_{\mathbf{v}}\mathrm{d}\theta_{\mathbf{v}}\mathrm{d}\phi_{\mathbf{v}}. \tag{B.9}$$

Consider the simple example of integrating the constant $f(\mathbf{v}) = 1$:

$$\int_0^{2\pi}\int_0^{\frac{\pi}{2}} \sin\theta_{\mathbf{v}}\mathrm{d}\theta_{\mathbf{v}}\mathrm{d}\phi_{\mathbf{v}} = \int_0^{2\pi}\left(-\cos\theta_{\mathbf{v}}\Big|_0^{\frac{\pi}{2}}\right)\mathrm{d}\phi_{\mathbf{v}} = \int_0^{2\pi}\mathrm{d}\phi_{\mathbf{v}} = 2\pi, \tag{B.10}$$

which is equal to the surface area of the unit hemisphere.

Appendix C

Common microsurfaces

In this appendix, we provide microfacet distributions that are relevant to this thesis: Beckmann distribution [Beckmann and Spizzichino, 1963], Trowbridge-Reitz distribution [1975] which was popularized by Walter et al. [2007] with the alias GGX, and the Generalized Trowbridge-Reitz (GTR) distribution [Burley, 2012]. We also provide their corresponding average visibility functions $S^\alpha(\mu)$ (where $\mu = \cot\theta_{\mathbf{v}}$) from which their Smith monodirectional shadowing terms can be constructed using Equation (3.8).

C.1 Beckmann distribution

The Beckmann surface is an early model based on the Gaussian distribution of slopes with microfacet distribution

$$D^\alpha(\mathbf{m},\mathbf{n}) = \frac{1}{\pi\alpha^2\cos^4\theta_{\mathbf{m}}}\exp\left(-\frac{\tan^2\theta_{\mathbf{m}}}{\alpha^2}\right), \tag{C.1}$$

and an approximate average visibility function

$$S^\alpha(\mu) = \begin{cases} \frac{3.535a+2.181a^2}{1+2.276a+2.577a^2}, & a < 1.6 \\ 1, & \text{otherwise} \end{cases}, \tag{C.2}$$

where $a = \mu/\alpha$.

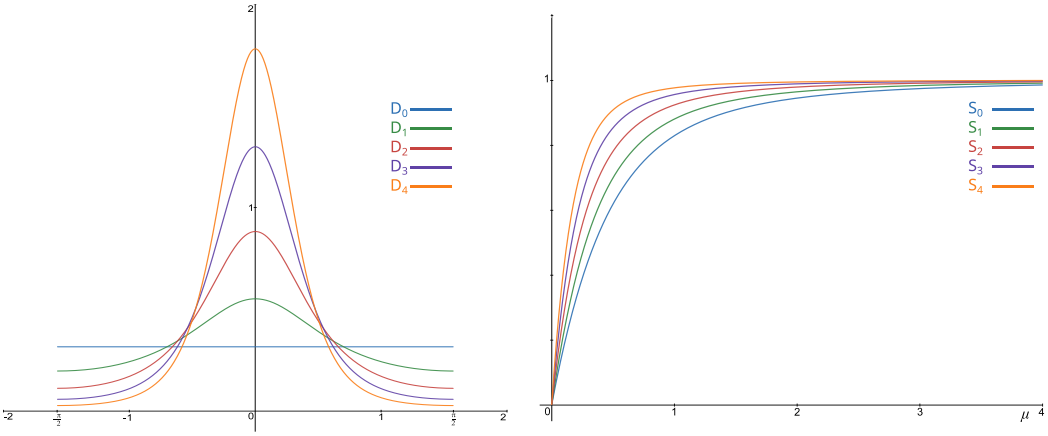
C.2 GGX distribution

The most popular microfacet distribution is GGX:

$$D^\alpha(\mathbf{m},\mathbf{n}) = \frac{1}{\pi\alpha^2\cos^4\theta_{\mathbf{m}}\left(1+\frac{\tan\theta_{\mathbf{m}}}{\alpha^2}\right)^2}. \tag{C.3}$$

Its average visibility is

$$S^\alpha(\mu) = \frac{2}{1+\sqrt{\frac{\alpha^2}{\mu^2}+1}}. \tag{C.4}$$



a) GTR Microfacet distribution D_γ^α b) GTR Average visibility S_γ^α

Figure C.1: **GTR distribution and shadowing plots.** Plots of the microfacet distribution D_γ^α and the average visibility S_γ^α for $\gamma = \{0, 1, 2, 3, 4\}$ and for a fixed roughness α =0.6. Note that $\lim_{\alpha\to 0,\gamma>0}(D_\gamma^\alpha, S_\gamma^\alpha) = (\delta, 1)$ and $\lim_{\alpha\to 1}(D_\gamma^\alpha, S_\gamma^\alpha) = (D_0^0, S_0^0)$.

C.3 GTR distribution

Burley [2012] introduced the Generalized Trowbridge-Reitz (GTR) distribution as a generalization of the GGX distribution which offers additional control on the tails:

$$D_\gamma^\alpha(\mathbf{m}, \mathbf{n}) = \frac{(\gamma - 1)(\alpha^2 - 1)}{\pi(1 - (\alpha^2)^{1-\gamma})(1 + (\alpha^2 - 1)\cos^2\theta_\mathbf{m})^\gamma}, \tag{C.5}$$

where $\alpha \in [0, 1]$ controls the roughness and $\gamma \geq 0$ is the tail exponent. The distribution has a 0/0 singularity at $\gamma = 1$, and it converges to the Berry distribution [Berry, 1923, Burley, 2012] at the limit

$$D_1^\alpha(\mathbf{m}, \mathbf{n}) = \lim_{\gamma\to 1} D_\gamma^\alpha(\mathbf{m}) = \frac{(\alpha^2 - 1)}{\pi(\log\alpha^2)(1 + (\alpha^2 - 1)\cos^2\theta_\mathbf{m})}. \tag{C.6}$$

A plot of the GTR distribution for a fixed roughness $\alpha = 0.6$ and tails $\gamma \in \{0, 1, 2, 3, 4\}$ is presented in Figure C.1 a).

Dimov [2015] followed the derivation procedure of Smith shadowing that is described by Walter et al. [2007]. With this approach, he was not able to find a general analytic formula S_γ^α, perhaps due to the singularity at $\gamma = 1$. Fortunately, he found analytic formulas for a few special values - $\gamma \in \{0, 1, 3, 4\}$. Follows a list of formulas for the GTR average visibility for integer tails, including the GGX average visibility S_2^α for completeness:

$$S_0^\alpha(\mu) = \frac{2}{1 + \sqrt{\frac{1}{\mu^2} + 1}}, \tag{C.7}$$

$$S_1^\alpha(\mu) = \frac{\mu\log\alpha^2}{A_1 - B_1 + \mu\log\left(\frac{\alpha^2(\mu + B_1)}{\mu + A_1}\right)}, \tag{C.8}$$

where $A_1 = \sqrt{\mu^2 + \alpha^2}$ and $B_1 = \sqrt{\mu^2 + 1}$,

$$S_2^\alpha(\mu) = \frac{2}{1 + \sqrt{\frac{\alpha^2}{\mu^2} + 1}}, \tag{C.9}$$

$$S_3^\alpha(\mu) = \frac{4B_3\mu A_3}{\alpha^2(3\alpha^2+1)+2\mu B_3(\mu+A_3)}, \tag{C.10}$$

where $A_3 = \sqrt{\mu^2+\alpha^2}$ and $B_3 = \alpha^2+1$ and

$$S_4^\alpha(\mu) = \frac{2A_4\mu B_4^3}{A_4\mu(B_4^3+\mu^3)+3\alpha^2(\alpha^2(5\alpha^4+2\alpha^2+1)+4\mu^2(2\alpha^4+\alpha^2+1))}, \tag{C.11}$$

where $A_4 = 8\alpha^4+8\alpha^2+8$ and $B_4 = \sqrt{\mu^2+\alpha^2}$.

In Figure C.1 b) we show these shadowing curves for a fixed roughness $\alpha = 0.6$. To compute an accurate approximation for all non-integer tails $\gamma \in (0,4)$ all five integer formulas are evaluated and a cubic spline interpolation [de Boor, 2001] is performed. Natural boundary conditions are suggested to minimize the oscillations.

We have verified that with this approach the constraint in Equation (3.9) is closely met.

C.4 An improved sampling for the GTR distribution

Rendering a microfacet BRDF with a high-roughness GTR distribution has inherent high variance due to the lack of sampling equations from the distribution of visible normals [Heitz and D'Eon, 2014]. The reduced effectiveness of the sampling from D [Walter et al., 2007] is due to increased shadowing. When the roughness is approaching 1, the single scattering term gets darker and more diffuse.

We found that stochastic mixing of the sampling strategy with uniform sampling improves the sampling of rough surfaces. A random number and the roughness α determine whether uniform sampling or sampling from D will be used for the direction. The balance heuristic is finally applied for probability computation [Veach and Guibas, 1995]. Render time comparison can be seen in Figure C.2.

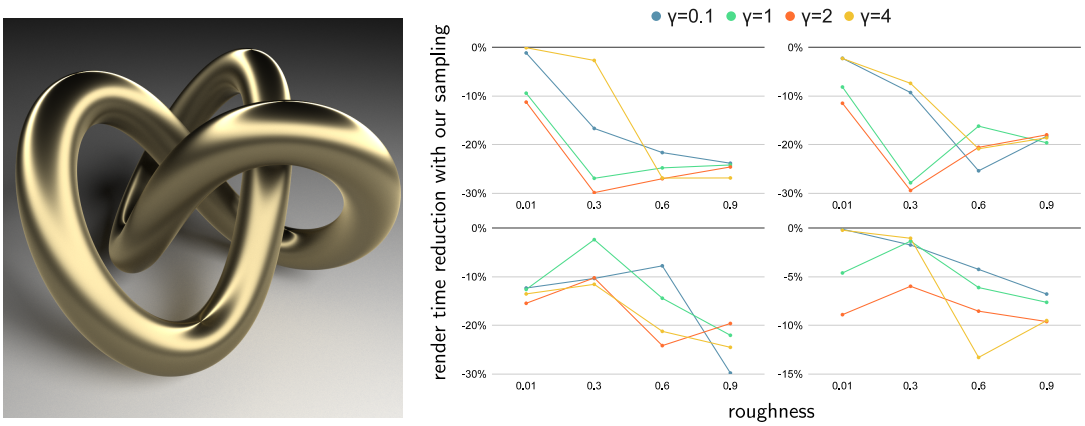


Figure C.2: **Render time comparison between standard NDF sampling and our improved sampling. Left:** Torus knot with GTR roughness $\alpha = 0.3$, tail exponent $\gamma = 1$ and anisotropy $s = 0.75$, lit by a rectangular light. **Right:** Charts for the torus knot scene showing how much render times improve for the same image quality when switching from standard sampling to our sampling described in Appendix C.4. We use a variance-based image sampler which samples each pixel until a given noise level is reached. The top left chart is for isotropic material ($s = 0$) with 95 average samples per pixel (aspp) for the standard sampling vs. 84 aspp for our sampling. The top right, bottom left and bottom right charts represent anisotropic materials with $s = \{0.25,\ 0.5,\ 0.75\}$ and 91 vs. 79 aspp, 95 vs. 89 aspp and 131 vs. 119 aspp, respectively. Therefore, our sampling technique reduces the number of samples per pixel by 10% on average for this scene. Moreover, the charts demonstrate that this technique helps with very different distribution tails: GTR with $\gamma = 0.1$ has a much heavier tail than GGX while the tail of GTR with $\gamma = 4$ is much "lighter", see Figure C.1 a).

Appendix D

Mathematica code

D.1 GTR normalization

In Listing 1, we use Mathematica [Wolfram Research, 2016] to find normalization for the GTR distribution by evaluating Equation (3.5). For the isotropic case, we find the proper normalization from Equation (C.5), while Mathematica is unable to solve the anisotropic case.

Listing 1: Computing normalization coefficients for isotropic and anisotropic GTR.

```
In[1]:= Integrate[Cos[t]*Sin[t]/(a^2*Cos[t]^2+Sin[t]^2)^g,
         {t, 0, Pi/2}, {p, 0, 2*Pi}] (* Isotropic GTR *)
```

$$\text{Out[1]=} \quad \frac{(1-(a^2)^{1-g})\pi}{(-1+a^2)(-1+g)}$$

```
In[2]:= Integrate[Cos[t]*Sin[t]/
          (Cos[t]^2+Sin[t]^2*(Cos[p]^2/a^2+Sin[p]^2/b^2))^g,
          {t, 0, Pi/2}, {p, 0, 2*Pi}] (* Anisotropic GTR *)
```

Out[2]=

D.2 Derivation of the Jacobian for Linearly Transformed Microfacet Distributions

In Listing 2, we provide the Mathematica notebook with the derivation of the Jacobian which is used to normalize Linearly Transformed Microfacet Distributions, see Equations (4.9) - (4.13). This notebook is published in Wolfram Foundation's Notebook Archive [Atanasov, 2022].

Listing 2: Mathematica nodebook [Wolfram Research, 2016] with derivation for Equations (4.9) - (4.13).

In[3]:=
```
(* Length of the transformed micro-normal ||M^Tm||,
where mz^2=1-mx^2-my^2 *)
len[mx_, my_, a_, b_, c_, d_]:=
  Sqrt[(a*mx+b*my)^2+c*mx+d*my)^2+1-mx^2-my^2]

(* Components of the transformed normalized micro-normal u,
where uz^2=1-ux^2-uy^2 *)
ux[mx_, my_, a_, b_, c_, d_]:=
  (a*mx+b*my)/len[mx, my, a, b, c, d]
uy[mx_, my_, a_, b_, c_, d_]:=
  (c*mx+d*my)/len[mx, my, a, b, c, d]
```

In[4]:=
```
(* Partial derivatives of u *)
duxdmx = Simplify[D[ux[mx, my, a, b, c, d], mx]]
```

Out[4]=
$$\frac{b\ my\ (mx - c^2\ mx - c\ d\ my) + a\ (1 + c\ d\ mx\ my + (-1 + d^2)\ my^2)}{(1 - mx^2 - my^2 + (a\ mx + b\ my)^2 + (c\ mx + d\ my)^2)^{3/2}}$$

In[5]:=
```
duydmy = Simplify[D[uy[mx, my, a, b, c, d], my]]
```

Out[5]=
$$\frac{c\ mx\ (-a\ b\ mx + my - b^2\ my) + d\ (1 + (-1 + a^2)\ mx^2 + a\ b\ mx\ my)}{(1 - mx^2 - my^2 + (a\ mx + b\ my)^2 + (c\ mx + d\ my)^2)^{3/2}}$$

In[6]:=
```
duxdmy = Simplify[D[ux[mx, my, a, b, c, d], my]]
```

Out[6]=
$$\frac{a\ mx\ (-c\ d\ mx + my - d^2\ my) + b\ (1 + (-1 + c^2)\ mx^2 + c\ d\ mx\ my)}{(1 - mx^2 - my^2 + (a\ mx + b\ my)^2 + (c\ mx + d\ my)^2)^{3/2}}$$

In[7]:=
```
duydmx = Simplify[D[uy[mx, my, a, b, c, d], my]]
```

Out[7]=
$$\frac{d\ my\ (mx - a^2\ mx - a\ b\ my) + c\ (1 + a\ b\ mx\ my + (-1 + b^2)\ my^2)}{(1 - mx^2 - my^2 + (a\ mx + b\ my)^2 + (c\ mx + d\ my)^2)^{3/2}}$$

In[8]:=
```
(* Notice that the determinant of M is in the numerator and the
length of the transformed micro-normal (see ''len'' above) to the
4-th power is in the denominator *)
Simplify[duxdmx*duydmy-duxdmy*duydmx]
```

Out[8]=
$$\frac{-b\ c + a\ d}{(1 + (-1 + a^2 + c^2)\ mx^2 + 2\ (a\ b + c\ d)\ mx\ my + (-1 + b^2 + d^2)\ my^2)^2}$$

Bibliography

Michael Ashikmin, Simon Premože, and Peter Shirley. A Microfacet-Based BRDF Generator. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, page 65–74, USA, 2000. ACM Press/Addison-Wesley Publishing Co. ISBN 1581132085. doi: 10.1145/344779.344814. URL https://doi.org/10.1145/344779.344814.

Asen Atanasov. Microsurface Transformations: Jacobian2D. Wolfram Foundation Notebook Archive, 2022. URL https://notebookarchive.org/microsurface-transformations-jacobian2d--2022-09-0iry427/. [Online; accessed 27-October-2022].

Asen Atanasov and Vladimir Koylazov. A Practical Stochastic Algorithm for Rendering Mirror-like Flakes. In *ACM SIGGRAPH 2016 Talks*, SIGGRAPH '16, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342827. doi: 10.1145/2897839.2927391. URL https://doi.org/10.1145/2897839.2927391.

Asen Atanasov, Vladimir Koylazov, Blagovest Taskov, Alexander Soklev, Vassillen Chizhov, and Jaroslav Křivánek. Adaptive Environment Sampling on CPU and GPU. In *ACM SIGGRAPH 2018 Talks*, SIGGRAPH '18, pages 68:1–68:2, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5820-0. doi: 10.1145/3214745.3214808. URL http://doi.acm.org/10.1145/3214745.3214808.

Asen Atanasov, Jaroslav Křivánek, Vladimir Koylazov, and Alexander Wilkie. Efficient Multiscale Rendering of Specular Microstructure. In *ACM SIGGRAPH 2020 Talks*, SIGGRAPH '20, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450379717. doi: 10.1145/3388767.3407338. URL https://doi.org/10.1145/3388767.3407338.

Asen Atanasov, Alexander Wilkie, Vladimir Koylazov, and Jaroslav Křivánek. A Multiscale Microfacet Model Based on Inverse Bin Mapping. *Computer Graphics Forum*, 40(2):103–113, 2021. doi: https://doi.org/10.1111/cgf.142618. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.142618.

Asen Atanasov, Vladimir Koylazov, Rossen Dimov, and Alexander Wilkie. Microsurface Transformations. *Computer Graphics Forum*, 41(4):105–116, 2022. doi: https://doi.org/10.1111/cgf.14590. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14590.

R. Ballester-Ripoll and R. Pajarola. Tensor Decompositions for Integral Histogram Compression and Look-up. *IEEE Transactions on Visualization and*

Computer Graphics, 25(2):1435–1446, February 2019. doi: 10.1109/TVCG.2018.2802521.

W W Barkas. Analysis of Light Scattered from a Surface of Low Gloss into its Specular and Diffuse Components. *Proceedings of the Physical Society*, 51 (2):274–295, March 1939. doi: 10.1088/0959-5309/51/2/307. URL https://doi.org/10.1088/0959-5309/51/2/307.

Pascal Barla, Romain Pacanowski, and Peter Vangorp. A Composite BRDF Model for Hazy Gloss. *Comput. Graph. Forum*, 37(4):55–66, 2018. doi: 10.1111/cgf.13475. URL https://doi.org/10.1111/cgf.13475.

Petr Beckmann and Andre Spizzichino. *The Scattering of Electromagnetic Waves from Rough Surfaces*. New York: Pergamon, 1963.

Laurent Belcour. Efficient Rendering of Layered Materials Using an Atomic Decomposition with Statistical Operators. *ACM Trans. Graph.*, 37(4), July 2018. ISSN 0730-0301. doi: 10.1145/3197517.3201289. URL https://doi.org/10.1145/3197517.3201289.

Laurent Belcour, Ling-Qi Yan, Ravi Ramamoorthi, and Derek Nowrouzezahrai. Antialiasing Complex Global Illumination Effects in Path-Space. *ACM Trans. Graph.*, 36(4), January 2017. ISSN 0730-0301. doi: 10.1145/3072959.2990495. URL http://doi.acm.org/10.1145/3072959.2990495.

Eugene M. Berry. Diffuse Reflection of Light from a Matt Surface. *J. Opt. Soc. Am.*, 7(8):627–633, August 1923. doi: 10.1364/JOSA.7.000627. URL http://opg.optica.org/abstract.cfm?URI=josa-7-8-627.

Benedikt Bitterli. The Tungsten Renderer, 2014. URL https://benedikt-bitterli.me/tungsten.html. [Online; accessed 6-September-2022].

Benedikt Bitterli and Eugene d'Eon. A Position-Free Path Integral for Homogeneous Slabs and Multiple Scattering on Smith Microfacets. *Computer Graphics Forum*, 2022. ISSN 1467-8659. doi: 10.1111/cgf.14589.

James F. Blinn. Models of Light Reflection for Computer Synthesized Pictures. *SIGGRAPH Comput. Graph.*, 11(2):192–198, July 1977. ISSN 0097-8930. doi: 10.1145/965141.563893. URL https://doi.org/10.1145/965141.563893.

James F. Blinn. Simulation of Wrinkled Surfaces. In *Proceedings of the 5th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '78, pages 286–292, New York, NY, USA, 1978. ACM. doi: 10.1145/800248.507101. URL http://doi.acm.org/10.1145/800248.507101.

P. Bouguer. *Traité d'Optique sur la Gradation de la Lumière*. Gueriss & Delatouve, 1760. URL https://books.google.bg/books?id=pzYVAAAAQAAJ.

Adam Brady, Jason Lawrence, Pieter Peers, and Westley Weimer. GenBRDF: Discovering New Analytic BRDFs with Genetic Programming. *ACM Trans. Graph.*, 33(4), July 2014. ISSN 0730-0301. doi: 10.1145/2601097.2601193. URL https://doi.org/10.1145/2601097.2601193.

Eric Bruneton and Fabrice Neyret. A Survey of Nonlinear Prefiltering Methods for Efficient and Accurate Surface Shading. *IEEE Transactions on Visualization and Computer Graphics*, 18(2):242–260, 2012. doi: 10.1109/TVCG.2011.81.

Brent Burley. Physically-Based Shading at Disney. 2012.

Samuel D. Butler and Michael A. Marciniak. Robust Categorization of Microfacet BRDF Models to Enable Flexible Application-specific BRDF Adaptation. In Leonard M. Hanssen, editor, *Reflection, Scattering, and Diffraction from Surfaces IV*, volume 9205 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, page 920506, September 2014. doi: 10.1117/12.2061134.

Chaos. Cosmos, 2023a. URL `https://www.chaos.com/cosmos`. [Online; accessed 9-January-2023].

Chaos. V-Ray Renderer, 2023b. URL `https://www.chaos.com/3d-rendering-software`. [Online; accessed 5-January-2023].

Chaos. V-Ray Car Paint 2 Material, 2023c. URL `https://docs.chaos.com/display/VC4D/Car+Paint+2+Material`. [Online; accessed 5-January-2023].

Chaos. V-Ray Material, 2023d. URL `https://docs.chaos.com/display/VMAX/VRayMtl`. [Online; accessed 5-January-2023].

Xavier Chermain, Frédéric Claux, and Stéphane Mérillou. A Microfacet-based BRDF for the Accurate and Efficient Rendering of High-definition Specular Normal Maps. *The Visual Computer*, 10 2018. doi: 10.1007/s00371-018-1606-7.

Xavier Chermain, Basile Sauvage, Dischler Jean-Michel, and Carsten Dachsbacher. Procedural Physically-based BRDF for Real-Time Rendering of Glints. *Comput. Graph. Forum (Proc. Pacific Graphics)*, 39(7):243–253, 2020.

A. W. Christie. The Luminous Directional Reflectance of Snow. *J. Opt. Soc. Am.*, 43(7):621–622, July 1953. doi: 10.1364/JOSA.43.000621. URL `http://opg.optica.org/abstract.cfm?URI=josa-43-7-621`.

R. L. Cook and K. E. Torrance. A Reflectance Model for Computer Graphics. *ACM Trans. Graph.*, 1(1):7–24, January 1982. ISSN 0730-0301. doi: 10.1145/357290.357293. URL `http://doi.acm.org/10.1145/357290.357293`.

Franklin C. Crow. Summed-area Tables for Texture Mapping. In *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '84, pages 207–212, New York, NY, USA, 1984. ACM. ISBN 0-89791-138-5. doi: 10.1145/800031.808600. URL `http://doi.acm.org/10.1145/800031.808600`.

C. de Boor. *A Practical Guide to Splines*. Applied Mathematical Sciences. Springer New York, 2001. ISBN 9780387953663. URL `https://books.google.bg/books?id=m0QDJvBI_ecC`.

N. Dekker, E. J. J. Kirchner, R. Supèr, G. J. van den Kieboom, and R. Gottenbos. Total Appearance Differences for Metallic and Pearlescent Materials: Contributions from Color and Texture. *Color Research & Application*, 36(1):4–14, 2011. doi: https://doi.org/10.1002/col.20586. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/col.20586`.

Hong Deng, Yang Liu, Beibei Wang, Jian Yang, Lei Ma, Nicolas Holzschuch, and Ling-Qi Yan. Constant-Cost Spatio-Angular Prefiltering of Glinty Appearance Using Tensor Decomposition. *ACM Trans. Graph.*, 41(2), January 2022. ISSN 0730-0301. doi: 10.1145/3507915. URL `https://doi.org/10.1145/3507915`.

Rossen Dimov. Deriving the Smith Shadowing Function G_1 for $\gamma \in (0, 4]$. 2015.

Jonathan Dupuy, Eric Heitz, Jean-Claude Iehl, Pierre Poulin, Fabrice Neyret, and Victor Ostromoukhov. Linear Efficient Antialiased Displacement and Reflectance Mapping. *ACM Transactions on Graphics*, 32(6):Article No. 211, November 2013. doi: 10.1145/2508363.2508422. URL `https://hal.inria.fr/hal-00858220`.

Jonathan Dupuy, Eric Heitz, Jean-Claude Iehl, Pierre Poulin, and Victor Ostromoukhov. Extracting Microfacet-based BRDF Parameters from Arbitrary Materials with Power Iterations. *Computer Graphics Forum*, page 10, 2015. URL `https://hal.inria.fr/hal-01168516`.

Sergey Ershov, Konstantin Kolchin, and Karol Myszkowski. Rendering Pearlescent Appearance Based On Paint-Composition Modelling. *Comput. Graph. Forum*, 20, 09 2001. doi: 10.1111/1467-8659.00515.

Sergey V. Ershov, Andrei B. Khodulev, and Konstantin V. Kolchin. Simulation of Sparkles in Metallic Paints. In *Proceedings of Graphicon'99*, pages 121–128, 1999.

Alejandro Conty Estevez and Christopher Kulla. Production Friendly Microfacet Sheen BRDF. 2017.

A.J. Fresnel. *Mémoire sur la loi des modifications que la réflexion imprime à la lumière polarisée*. Académie des Sciences, 1823. URL `https://books.google.bg/books?id=463IvAEACAAJ`.

Luis E. Gamboa, Jean-Philippe Guertin, and Derek Nowrouzezahrai. Scalable Appearance Filtering for Complex Lighting Effects. *ACM Trans. Graph.*, 37(6): 277:1–277:13, December 2018. ISSN 0730-0301. doi: 10.1145/3272127.3275058. URL `http://doi.acm.org/10.1145/3272127.3275058`.

Georgi T. Georgiev and James J. Butler. Long-term Calibration Monitoring of Spectralon Diffusers BRDF in the Air-ultraviolet. *Appl. Opt.*, 46 (32):7892–7899, November 2007. doi: 10.1364/AO.46.007892. URL `https://opg.optica.org/ao/abstract.cfm?URI=ao-46-32-7892`.

Johannes Günther, Tongbo Chen, Michael Goesele, Ingo Wald, and Hans-Peter Seidel. Efficient Acquisition and Realistic Rendering of Car Paint. 2005.

Charles Han, Bo Sun, Ravi Ramamoorthi, and Eitan Grinspun. Frequency Domain Normal Map Filtering. *ACM Trans. Graph.*, 26:28, 07 2007. doi: 10.1145/1275808.1276412.

E. Heitz. Can't Invert the CDF? The Triangle-Cut Parameterization of the Region under the Curve. *Computer Graphics Forum*, 39(4):121–132, 2020. doi: https://doi.org/10.1111/cgf.14058. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14058.

Eric Heitz. Understanding the Masking-Shadowing Function in Microfacet-Based BRDFs. *Journal of Computer Graphics Techniques (JCGT)*, 3(2):48–107, June 2014. ISSN 2331-7418. URL http://jcgt.org/published/0003/02/03/.

Eric Heitz. Sampling the GGX Distribution of Visible Normals. *Journal of Computer Graphics Techniques (JCGT)*, 7(4):1–13, November 2018. ISSN 2331-7418. URL http://jcgt.org/published/0007/04/01/.

Eric Heitz and Eugene D'Eon. Importance Sampling Microfacet-Based BSDFs using the Distribution of Visible Normals. *Computer Graphics Forum*, 33(4):103–112, July 2014. doi: 10.1111/cgf.12417. URL https://hal.inria.fr/hal-00996995.

Eric Heitz and Jonathan Dupuy. Implementing a Simple Anisotropic Rough Diffuse Material with Stochastic Evaluation. Technical report, 2015.

Eric Heitz, Jonathan Dupuy, Stephen Hill, and David Neubelt. Real-Time Polygonal-Light Shading with Linearly Transformed Cosines. *ACM Trans. Graph.*, 35(4), July 2016a. ISSN 0730-0301. doi: 10.1145/2897824.2925895. URL https://doi.org/10.1145/2897824.2925895.

Eric Heitz, Johannes Hanika, Eugene d'Eon, and Carsten Dachsbacher. Multiple-Scattering Microfacet BSDFs with the Smith Model. *ACM Trans. Graph.*, 35(4), July 2016b. ISSN 0730-0301. doi: 10.1145/2897824.2925943. URL https://doi.org/10.1145/2897824.2925943.

L. G. Henyey and J. L. Greenstein. Diffuse Radiation in the Galaxy. *Astrophysical Journal*, 93:70–83, January 1941. doi: 10.1086/144246.

Christophe Hery, Michael Kass, and Junyi Ling. Geometry into Shading. 2014.

Stephen Hill, Stephen McAuley, Alejandro Conty, Michał Drobot, Eric Heitz, Christophe Hery, Christopher Kulla, Jon Lanz, Junyi Ling, Nathan Walster, Feng Xie, Adam Micciulla, and Ryusuke Villemin. Physically Based Shading in Theory and Practice. In *ACM SIGGRAPH 2017 Courses*, SIGGRAPH '17, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450350143. doi: 10.1145/3084873.3084893. URL https://doi.org/10.1145/3084873.3084893.

Nicolas Holzschuch and Romain Pacanowski. A Two-Scale Microfacet Reflectance Model Combining Reflection and Diffraction. *ACM Transactions on Graphics*, 36(4):12, July 2017. doi: 10.1145/3072959.3073621. URL https://hal.inria.fr/hal-01515948. Article 66.

Weizhen Huang, Matthias B. Hullin, and Johannes Hanika. A Microfacet-based Hair Scattering Model. *Computer Graphics Forum*, 2022. ISSN 1467-8659. doi: 10.1111/cgf.14588.

Homan Igehy. Tracing Ray Differentials. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 179–186, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co. ISBN 0-201-48560-5. doi: 10.1145/311535.311555. URL `http://dx.doi.org/10.1145/311535.311555`.

Wenzel Jakob. Mitsuba Renderer, 2010. http://www.mitsuba-renderer.org.

Wenzel Jakob. An Improved Visible Normal Sampling Routine for the Beckmann Distribution. 2014.

Wenzel Jakob, Miloš Hašan, Ling-Qi Yan, Jason Lawrence, Ravi Ramamoorthi, and Steve Marschner. Discrete Stochastic Microfacet Models. *ACM Trans. Graph.*, 33(4), July 2014. ISSN 0730-0301. doi: 10.1145/2601097.2601186. URL `https://doi.org/10.1145/2601097.2601186`.

Henrik Wann Jensen. *Realistic Image Synthesis Using Photon Mapping*. A. K. Peters, Ltd., USA, 2001. ISBN 1568811470.

James T. Kajiya. Anisotropic Reflection Models. *SIGGRAPH Comput. Graph.*, 19(3):15–21, July 1985. ISSN 0097-8930. doi: 10.1145/325165.325167. URL `https://doi.org/10.1145/325165.325167`.

Eric Kirchner, Geert-Jan van den Kieboom, Lan Njo, Rianne Supèr, and Roel Gottenbos. Observation of Visual Texture of Metallic and Pearlescent Materials. *Color Research & Application*, 32(4):256–266, 2007. doi: https://doi.org/10.1002/col.20328. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/col.20328`.

Murat Kurt, László Szirmay-Kalos, and Jaroslav Křivánek. An Anisotropic BRDF Model for Fitting and Monte Carlo Rendering. *SIGGRAPH Comput. Graph.*, 44(1), February 2010. ISSN 0097-8930. doi: 10.1145/1722991.1722996. URL `https://doi.org/10.1145/1722991.1722996`.

Alexandr Kuznetsov, Miloš Hašan, Zexiang Xu, Ling-Qi Yan, Bruce Walter, Nima Khademi Kalantari, Steve Marschner, and Ravi Ramamoorthi. Learning Generative Models for Rendering Specular Microgeometry. *ACM Trans. Graph.*, 38(6):225:1–225:14, November 2019. ISSN 0730-0301. doi: 10.1145/3355089.3356525. URL `http://doi.acm.org/10.1145/3355089.3356525`.

J.H. Lambert. *Photometria*. 1760. URL `https://books.google.bg/books?id=tbM6AAAAcAAJ`.

T. Lee and H. Shen. Efficient Local Statistical Analysis via Integral Histograms with Discrete Wavelet Transform. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2693–2702, December 2013. doi: 10.1109/TVCG.2013.152.

Joakim Löw, Joel Kronander, Anders Ynnerman, and Jonas Unger. BRDF Models for Accurate and Efficient Rendering of Glossy Surfaces. *ACM Trans. Graph.*, 31(1), February 2012. ISSN 0730-0301. doi: 10.1145/2077341.2077350. URL https://doi.org/10.1145/2077341.2077350.

C. S. McCamy. Observation and Measurement of the Appearance of Metallic Materials. Part I. Macro appearance. *Color Research & Application*, 21(4):292–304, 1996. doi: 10.1002/(SICI)1520-6378(199608)21:4⟨292::AID-COL4⟩3.0.CO;2-L. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291520-6378%28199608%2921%3A4%3C292%3A%3AAID-COL4%3E3.0.CO%3B2-L.

C. S. McCamy. Observation and Measurement of the Appearance of Metallic Materials. Part II. Micro appearance. *Color Research & Application*, 23(6):362–373, 1998. doi: 10.1002/(SICI)1520-6378(199812)23:6⟨362::AID-COL4⟩3.0.CO;2-5. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291520-6378%28199812%2923%3A6%3C362%3A%3AAID-COL4%3E3.0.CO%3B2-5.

W. E. K. Middleton and A. G. Mungall. Colorimetry of Bronze Paints∗. *J. Opt. Soc. Am.*, 42(12):969–970, December 1952. doi: 10.1364/JOSA.42.000969. URL http://opg.optica.org/abstract.cfm?URI=josa-42-12-969.

Koki Nagano, Graham Fyffe, Oleg Alexander, Jernej Barbič, Hao Li, Abhijeet Ghosh, and Paul Debevec. Skin Microstructure Deformation with Displacement Map Convolution. *ACM Trans. Graph.*, 34(4), July 2015. ISSN 0730-0301. doi: 10.1145/2766894. URL https://doi.org/10.1145/2766894.

F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis. Geometrical Considerations and Nomenclature for Reflectance. Final Report National Bureau of Standards, Washington, DC. Inst. for Basic Standards., October 1977.

Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. Mitsuba 2: A Retargetable Forward and Inverse Renderer. *ACM Trans. Graph.*, 38 (6), November 2019. ISSN 0730-0301. doi: 10.1145/3355089.3356498. URL https://doi.org/10.1145/3355089.3356498.

Marc Olano and Dan Baker. LEAN Mapping. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '10, pages 181–188, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-939-8. doi: 10.1145/1730804.1730834. URL http://doi.acm.org/10.1145/1730804.1730834.

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*, SIGGRAPH '94, page 239–246, New York, NY, USA, 1994. Association for Computing Machinery. ISBN 0897916670. doi: 10.1145/192161.192213. URL https://doi.org/10.1145/192161.192213.

Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, November 2016. ISBN 9780128006450.

GI Pokrowski. Zur Theorie der Diffusen Lichtreflexion. *Zeitschrift für Physik*, 30 (1):66–72, 1924.

GI Pokrowski. Zur Theorie der Diffusen Lichtreflexion. IV. *Zeitschrift für Physik*, 36(6):472–476, 1926.

Fatih Porikli. Integral Histogram: A Fast Way To Extract Histograms in Cartesian Spaces. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, volume 1 of *CVPR '05*, page 829–836, USA, 07 2005. IEEE Computer Society. ISBN 0769523722. doi: 10.1109/CVPR.2005.188. URL https://doi.org/10.1109/CVPR.2005.188.

Pierre Poulin and Alain Fournier. A Model for Anisotropic Reflection. *SIGGRAPH Comput. Graph.*, 24(4):273–282, September 1990. ISSN 0097-8930. doi: 10.1145/97880.97909. URL https://doi.org/10.1145/97880.97909.

Boris Raymond, Gael Guennebaud, and Pascal Barla. Multi-Scale Rendering of Scratched Materials using a Structured SV-BRDF Model. *ACM Transactions on Graphics*, July 2016. doi: 10.1145/2897824.2925945. URL https://hal.inria.fr/hal-01321289.

Mickaël Ribardière, Benjamin Bringier, Daniel Meneveaux, and Lionel Simonot. STD: Student's t-Distribution of Slopes for Microfacet Based BSDFs. *Computer Graphics Forum*, 36(2), 2017. doi: 10.1111/cgf.13137. URL https://hal.archives-ouvertes.fr/hal-01535614.

Mickaël Ribardière, Benjamin Bringier, Lionel Simonot, and Daniel Meneveaux. Microfacet BSDFs Generated from NDFs and Explicit Microgeometry. *ACM Trans. Graph.*, 38(5), June 2019. ISSN 0730-0301. doi: 10.1145/3338697. URL https://doi.org/10.1145/3338697.

Vincent Ross, Denis Dion, and Guy Potvin. Detailed Analytical Approach to the Gaussian Surface Bidirectional Reflectance Distribution Function Specular Component Applied to the Sea Surface. *J. Opt. Soc. Am. A*, 22(11):2442–2453, November 2005. doi: 10.1364/JOSAA.22.002442. URL http://opg.optica.org/josaa/abstract.cfm?URI=josaa-22-11-2442.

Martin Rump, Gero Müller, Ralf Sarlette, Dirk Koch, and Reinhard Klein. Photorealistic Rendering of Metallic Car Paint from Image-Based Measurements. *Computer Graphics Forum*, 27(2):527–536, April 2008.

Vincent Schüssler, Eric Heitz, Johannes Hanika, and Carsten Dachsbacher. Microfacet-Based Normal Mapping for Robust Monte Carlo Path Tracing. *ACM Trans. Graph.*, 36(6), November 2017. ISSN 0730-0301. doi: 10.1145/3130800.3130806. URL https://doi.org/10.1145/3130800.3130806.

B. Smith. Geometrical Shadowing of a Random Rough Surface. *IEEE Transactions on Antennas and Propagation*, 15(5):668–671, 1967. doi: 10.1109/TAP.1967.1138991.

Jos Stam. An Illumination Model for a Skin Layer Bounded by Rough Surfaces. In S. J. Gortle and K. Myszkowski, editors, *Eurographics Workshop on Rendering*. The Eurographics Association, 2001. ISBN 3-211-83709-4. doi: 10.2312/EGWR/EGWR01/039-052.

Li-Piin Sung, Maria E. Nadal, Mary E. McKnight, Egon Marx, and Brent Laurenti. Optical Reflectance of Metallic Coatings: Effect of Aluminum Flake Orientation. *Journal of Coatings Technology*, 74, 2002. ISSN 1935-3804. URL `https://doi.org/10.1007/BF02697975`.

Michael Toksvig. Mipmapping Normal Maps. *Journal Graphics Tools*, 10:65–71, 2005.

K.E. Torrance and E.M. Sparrow. Theory for Off-Specular Reflection from Roughened Surfaces. *Journal of the Optical Society of America (JOSA)*, 57(9):1105–1114, September 1967. URL `http://www.graphics.cornell.edu/~westin/pubs/TorranceSparrowJOSA1967.pdf`.

T. S. Trowbridge and K. P. Reitz. Average irregularity representation of a rough surface for ray reflection. *J. Opt. Soc. Am.*, 65(5):531–536, May 1975. doi: 10.1364/JOSA.65.000531. URL `http://www.osapublishing.org/abstract.cfm?URI=josa-65-5-531`.

Eric Veach. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford, CA, USA, 1998. AAI9837162.

Eric Veach and Leonidas J. Guibas. Optimally Combining Sampling Techniques for Monte Carlo Rendering. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 419–428, New York, NY, USA, 1995. ACM. ISBN 0-89791-701-4. doi: 10.1145/218380.218498. URL `http://doi.acm.org/10.1145/218380.218498`.

Zdravko Velinov, Sebastian Werner, and Matthias B. Hullin. Real-Time Rendering of Wave-Optical Effects on Scratched Surfaces. *Computer Graphics Forum 37 (2) (Proc. EUROGRAPHICS)*, 37(2), 2018.

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*, EGSR'07, pages 195–206, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association. ISBN 978-3-905673-52-4. doi: 10.2312/EGWR/EGSR07/195-206. URL `http://dx.doi.org/10.2312/EGWR/EGSR07/195-206`.

Beibei Wang, Miloš Hašan, Nicolas Holzschuch, and Ling-Qi Yan. Example-Based Microstructure Rendering with Constant Storage. *ACM Trans. Graph.*, 39(5), August 2020. ISSN 0730-0301. doi: 10.1145/3406836. URL `https://doi.org/10.1145/3406836`.

Beibei Wang, Wenhua Jin, Jiahui Fan, Jian Yang, Nicolas Holzschuch, and Ling-Qi Yan. Position-Free Multiple-Bounce Computations for Smith Microfacet BSDFs. *ACM Trans. Graph.*, 41(4), July 2022. ISSN 0730-0301. doi: 10.1145/3528223.3530112. URL `https://doi.org/10.1145/3528223.3530112`.

Jingwen Wang and Ravi Ramamoorthi. Analytic Spherical Harmonic Coefficients for Polygonal Area Lights. *ACM Trans. Graph.*, 37(4), July 2018. ISSN 0730-0301. doi: 10.1145/3197517.3201291. URL https://doi.org/10.1145/3197517.3201291.

Gregory J. Ward. Measuring and Modeling Anisotropic Reflection. *SIGGRAPH Comput. Graph.*, 26(2):265–272, July 1992. ISSN 0097-8930. doi: 10.1145/142920.134078. URL https://doi.org/10.1145/142920.134078.

Andrea Weidlich and Alexander Wilkie. Arbitrarily Layered Micro-Facet Surfaces. In *Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia*, GRAPHITE '07, page 171–178, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595939128. doi: 10.1145/1321261.1321292. URL https://doi.org/10.1145/1321261.1321292.

Andrea Weidlich and Alexander Wilkie. Modeling Aventurescent Gems with Procedural Textures. In *Proceedings of the 24th Spring Conference on Computer Graphics*, SCCG '08, page 51–58, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605589572. doi: 10.1145/1921264.1921278. URL https://doi.org/10.1145/1921264.1921278.

David Wellems, Steve Ortega, David Bowers, Jim Boger, and Matthew Fetrow. Long Wave Infrared Polarimetric Model: Theory, Measurements and Parameters. *Journal of Optics A: Pure and Applied Optics*, 8(10):914–925, October 2006. doi: 10.1088/1464-4258/8/10/014.

Sebastian Werner, Zdravko Velinov, Wenzel Jakob, and Matthias B. Hullin. Scratch Iridescence: Wave-optical Rendering of Diffractive Surface Structure. *ACM Trans. Graph.*, 36(6):207:1–207:14, November 2017. ISSN 0730-0301. doi: 10.1145/3130800.3130840. URL http://doi.acm.org/10.1145/3130800.3130840.

Wikipedia. Brushed Metal, 2023a. URL https://en.wikipedia.org/wiki/Brushed_metal. [Online; accessed 5-January-2023].

Wikipedia. Lambert's Cosine Law, 2023b. URL https://en.wikipedia.org/wiki/Lambert%27s_cosine_law. [Online; accessed 5-January-2023].

Wikipedia. Fresnel Equations, 2023c. URL https://en.wikipedia.org/wiki/Fresnel_equations. [Online; accessed 5-January-2023].

Lance Williams. Pyramidal Parametrics. *SIGGRAPH Comput. Graph.*, 17(3):1–11, July 1983. ISSN 0097-8930. doi: 10.1145/964967.801126. URL http://doi.acm.org/10.1145/964967.801126.

Inc. Wolfram Research. Mathematica, Version 10.4, 2016. URL https://www.wolfram.com/mathematica. Champaign, IL.

Hongzhi Wu, Julie Dorsey, and Holly Rushmeier. Physically-Based Interactive Bi-Scale Material Design. *ACM Trans. Graph.*, 30(6):1–10, December 2011. ISSN 0730-0301. doi: 10.1145/2070781.2024179. URL https://doi.org/10.1145/2070781.2024179.

Lifan Wu, Shuang Zhao, Ling-Qi Yan, and Ravi Ramamoorthi. Accurate Appearance Preserving Prefiltering for Rendering Displacement-Mapped Surfaces. *ACM Trans. Graph.*, 38(4), July 2019. ISSN 0730-0301. doi: 10.1145/3306346.3322936. URL https://doi.org/10.1145/3306346.3322936.

Chao Xu, Rui Wang, Shuang Zhao, and Hujun Bao. Real-Time Linear BRDF MIP-Mapping. *Computer Graphics Forum*, 2017. ISSN 1467-8659. doi: 10.1111/cgf.13221.

Ling-Qi Yan, Miloš Hašan, Wenzel Jakob, Jason Lawrence, Steve Marschner, and Ravi Ramamoorthi. Rendering Glints on High-Resolution Normal-Mapped Specular Surfaces. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 33(4):116:1–116:9, July 2014. doi: 10.1145/2601097.2601155.

Ling-Qi Yan, Miloš Hašan, Steve Marschner, and Ravi Ramamoorthi. Position-Normal Distributions for Efficient Rendering of Specular Microstructure. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2016)*, 35(4), 2016.

Ling-Qi Yan, Miloš Hašan, Bruce Walter, Steve Marschner, and Ravi Ramamoorthi. Rendering Specular Microgeometry with Wave Optics. *ACM Trans. Graph.*, 37(4):75:1–75:10, July 2018. ISSN 0730-0301. doi: 10.1145/3197517.3201351. URL http://doi.acm.org/10.1145/3197517.3201351.

Tizian Zeltner. *Light Path Gradients for Forward and Inverse Rendering*. PhD thesis, École Polytechnique Fédérale de Lausanne, December 2021.

Tizian Zeltner, Iliyan Georgiev, and Wenzel Jakob. Specular Manifold Sampling for Rendering High-Frequency Caustics and Glints. *ACM Trans. Graph.*, 39 (4), July 2020. ISSN 0730-0301. doi: 10.1145/3386569.3392408. URL https://doi.org/10.1145/3386569.3392408.

Junqiu Zhu, Yanning Xu, and Lu Wang. A Stationary SVBRDF Material Modeling Method Based on Discrete Microsurface. *Computer Graphics Forum*, 38 (7):745–754, 2019. doi: 10.1111/cgf.13876. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13876.

Junqiu Zhu, Sizhe Zhao, Yanning Xu, Xiangxu Meng, Lu Wang, and Ling-Qi Yan. Recent Advances in Glinty Appearance Rendering. *Comput. Vis. Media*, 8(4):535–552, 2022. doi: 10.1007/s41095-022-0280-x. URL https://doi.org/10.1007/s41095-022-0280-x.

Tobias Zirr and Anton S. Kaplanyan. Real-Time Rendering of Procedural Multiscale Materials. In *Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '16, pages 139–148, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4043-4. doi: 10.1145/2856400.2856409. URL http://doi.acm.org/10.1145/2856400.2856409.

List of Figures

List of Tables

List of Abbreviations

MIS	multiple importance sampling
BRDF	bidirectional reflectance distribution function
IOR	index of refraction
NDF	normal distribution function
VNDF	visible normal distribution function
PDF	probability density function
CDF	cumulative density function
GTR	generalized Trowbridge-Reitz (distribution)
LTSD	linearly transformed spherical distributions
LTMD	linearly transformed microfacet distributions
STD	student's t-distribution
SGTD	skewed generalized t-distribution
SH	spherical harmonics
SMS	specular manifold sampling
MIP	multum in parvo (from Latin: much in a small space)
LEAN	linear efficient antialiased normal (mapping)
SAT	summed-area table
IH	integral histogram
BM	bin map
IBM	inverse bin map

Attachments

Article pages

- "Microsurface Transformations" [Atanasov et al., 2022] article page containing the paper PDF, supplementary video and GitHub repository with numerical validation C++ code and the Mathematica notebook from Listing 2:
  https://cgg.mff.cuni.cz/publications/microsurface-transformations/
- "A Multiscale Microfacet Model Based on Inverse Bin Mapping" [Atanasov et al., 2021] article page containing the paper PDF, supplementary video and GitHub repository with the implementation of our method as a Mitsuba 2 plug-in [Nimier-David et al., 2019]:
  https://cgg.mff.cuni.cz/publications/a-multiscale-microfacet-model-based-on-inverse-bin-mapping/

Contents of the supplementary archive

All supplementary materials that are available on our article pages are also included in the accompanying archive, attached to this thesis. In particular, this archive contains the original supplementary videos, both of which have superior quality when compared to the YouTube videos from the article pages. This difference is especially apparent in our glints video since the YouTube compression loses some of the fine details of the sparkling. Therefore, we recommend the reader watch the original videos. The file structure of the archive is given in Listing 3.

Listing 3: Contents of the supplementary archive.

```
archive
├─ microsurface_transformations
│  ├─ video.mp4
│  ├─ teaser.jpg
│  ├─ article.pdf
│  ├─ numerical_validation
│  │  ├─ microsurface_transformations.h
│  │  ├─ math_utils.h
│  │  └─ test.cpp
│  └─ mathematica_derivation
│     ├─ jacobian2d.nb
│     └─ jacobian2d.jpg
├─ multiscale_microfacet_model
│  ├─ video.mp4
│  ├─ teaser.jpg
│  ├─ article.pdf
│  └─ mitsuba_plugin.cpp
├─ LICENSE
└─ thesis.pdf
```