

Real-time Rendering of Measured Materials

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Visual Computing

eingereicht von

Thomas Mühlbacher

Matrikelnummer 0625075

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung
Betreuer: Associate Prof. Dipl.-Ing. Dipl.-Ing. Dr.techn. Michael Wimmer
Mitwirkung: Dipl.-Ing. Dr. Robert F. Tobler

Wien, 13.12.2012

(Unterschrift Verfasser)

(Unterschrift Betreuer)

Erklärung zur Verfassung der Arbeit

Thomas Mühlbacher
Harrachgasse 1/18, 1220 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum)

(Unterschrift Verfasser)

Acknowledgements

I wish to thank the following people and institutions for making this work possible: the VRVis research center in Vienna for allowing me to be a part of the fascinating HILITE-project by incorporating my thesis. In particular, Robert F. Tobler for guidance throughout the entire project, him and Michael Schwärzler for reviewing this thesis and providing invaluable feedback, and Christian Luksch for advice, discussions and technical help during the implementation. The project partners Zumtobel and Hefel Wohnbau for providing material measurement data and test scene geometry.

I would also like to thank my supervisor at the Institute of Computer Graphics and Algorithms at the Vienna University of Technology, Michael Wimmer, for valuable feedback and input during all stages of writing this thesis. The entire institute of Computer Graphics and Algorithms for marvellous teaching, reliability and fast aid in any scientific or organizational affairs. My fellow students, friends and colleagues, especially the Computer Graphics club for enlightening discussions and suggestions.

Personal thanks go out to my family for supporting my studies and finally to my partner M.-Kathleen Jimenez for enduring me during this work-intensive period, reviewing important sections of my thesis and also for supporting and reassuring me in moments of doubt and frustration during this work. Thank you!

Abstract

Interactive walkthroughs of virtual scenes are not only common in fictional settings such as entertainment and video games, but also a popular way of presenting novel architecture, furnishings or illumination. Due to the high performance requirements of such interactive applications, the presentable detail and quality are limited by the computational hardware. A realistic appearance of materials is one of the most crucial aspects to scene immersion during walkthroughs, and computing it at interactive frame rates is a challenging task.

In this thesis an algorithm is presented that achieves the rendering of *static scenes* featuring view-dependent materials in real-time. For walkthroughs of static scenes, all light propagation but the last view-dependent bounce can be precomputed and stored as diffuse irradiance light maps together with the scene geometry. The specular part of reflection and transmission is then computed dynamically by integrating the incident light approximately according to view and local material properties. For this purpose, the incident radiance distribution of each object is approximated by a single static environment map that is obtained by rendering the light-mapped scene as seen from the object. For large planar reflectors, a mirror rendering is performed every frame to approximate the incident light distribution instead of a static environment map. Materials are represented using a parametric model that is particularly suitable for fitting to measured reflectance data. Fitting the parameters of a compact model to material measurements provides a straightforward approach of reproducing light interactions of real-world substances on a screen.

During walkthroughs, the view-dependent part of the local illumination integral is approximated by sampling the representation of incident light while weighting the samples according to the material properties. Noise-free rendering is achieved by reusing the exact same sampling pattern at all pixels of a shaded object, and by filtering the samples using MIP-maps of the incident light representation. All available samples are regularly placed within the specular lobe to achieve a uniform symmetric coverage of the most important part of the integration domain even when using very few (5-20) samples. Thus, the proposed algorithm achieves a biased but stable and convincing material appearance at real-time frame rates. It is faster than existing random-based sampling algorithms, as fewer samples suffice to achieve a smooth and uniform coverage of specular lobes.

Kurzfassung

Interaktive Rundgänge durch virtuelle Szenen sind ein beliebtes Werkzeug zur Präsentation neuartiger Architektur, Einrichtung oder Innenraumbeleuchtung. Für eine hinreichende Szenenimmersion während solcher Rundgänge ist eine realistische Darstellung von Materialien unverzichtbar, diese in Echtzeit zu berechnen ist jedoch eine große technische Herausforderung. In dieser Arbeit wird ein Algorithmus vorgestellt, der das Echtzeit-Rendering von blickwinkelabhängigen Materialien in *statischen Szenen* ermöglicht. Für statische Szenen kann die globale blickpunktunabhängige Lichtausbreitung in einem Vorverarbeitungsschritt berechnet und abgespeichert werden. Lediglich der letzte Schritt, in dem ermittelt wird, wieviel Licht der Szene schließlich das Auge erreicht, wird mithilfe des vorgestellten Algorithmus zur Laufzeit berechnet. Der diffuse Anteil wird direkt aus der vorberechneten Bestrahlungsstärke abgeleitet, der blickpunktabhängige spiegelnde Teil von Reflektion und Transmission wird durch näherungsweise Integration des einfallenden Lichtes gemäß den Materialeigenschaften ermittelt. Die Verteilung des einfallenden Lichts wird für gekrümmte Flächen als *environment map* der eintreffenden Strahldichte gespeichert. Für große ebene Flächen wird die Szene vor der Berechnung jedes Bildes zunächst von einer gespiegelten Kamera aus in einen Zwischenspeicher gerendert, welcher dann als näherungsweise Repräsentation der einfallenden Beleuchtung dient. Materialeigenschaften werden in dieser Arbeit durch ein parametrisches Modell ausgedrückt, welches sich besonders zur Anpassung (Fitting) an reale gemessene Lichtstreudaten eignet.

Zur Laufzeit wird der blickpunktabhängige Teil der lokalen Beleuchtung durch Sampling (Abtasten) der einfallenden Beleuchtungsrepräsentation unter Gewichtung der Samples nach den Materialeigenschaften angenähert. Ein rauschfreies Bild wird durch Wiederverwendung des exakt gleichen Abtastmusters an allen Punkten eines Objekts und zusätzliches Filtern der eingehenden Strahldichte erreicht. Die einfallende Beleuchtung wird regelmäßig, symmetrisch und ausschließlich im Bereich des *specular lobe* abgetastet, um eine möglichst uniforme Abdeckung des wichtigsten Integrationsbereichs auch bei sehr geringer Abtastauflösung zu erzielen. Dadurch produziert der Algorithmus eine physikalisch inkorrekte aber glaubwürdige Darstellung von blickpunktabhängigen Materialien in Echtzeit. Das Verfahren ist schneller als bestehende zufallsbasierte Sampling-Methoden, da weniger Samples nötig sind um eine rauschfreie und gleichmäßige Abdeckung der *specular lobes* zu erzielen.

Contents

Acknowledgements	ii
Abstract	iii
Kurzfassung	iv
Contents	v
1 Introduction	1
1.1 Motivation	2
1.2 Application framework	3
1.3 Overview	4
1.4 Contributions	6
1.4.1 Obtaining a compact material representation from measured data	6
1.4.2 Extension of an existing BRDF model to support transmission	6
1.4.3 Real-time integration of incident lighting	6
1.4.4 Representing and sampling incident light	7
1.5 Organization of this thesis	8
2 Theoretical Foundations	9
2.1 The rendering problem	10
2.2 Material optics	11
2.2.1 Quantifying scattering properties - the BSDF	12
2.2.2 Physical plausibility	12
2.2.3 Diffuse and specular reflection	13
2.2.4 The Fresnel effect	15
2.3 Material measurement	17
2.4 Monte-Carlo integration	18
3 Material Representation	21
3.1 Related work	22
3.2 Direct measurement evaluation for photon bounce simulation	26
3.3 A BSDF model for fast integration	29
3.3.1 Representing transmission distributions	32

3.4	Implementation, results and limitations	34
3.4.1	Fitting the BSDF model	34
3.4.2	Fitting results and limitations	38
4	Material Rendering	49
4.1	Related work	50
4.2	Algorithm overview	54
4.3	Rendering diffuse reflection	57
4.4	Rendering specular reflection on curved surfaces	59
4.4.1	Representing incident light as a cubic environment map	59
4.4.2	Unbiased integration of incident light	63
4.4.3	Approximating the specular integral for real-time walkthroughs	69
4.5	Rendering specular reflection on planar surfaces	85
4.5.1	Representing incident light by a mirror rendering	85
4.5.2	Generating illumination samples using the mirror buffer	88
4.5.3	Deterministic sampling pattern and filtering for real-time walkthroughs	93
4.6	Implementation, results and limitations	96
4.6.1	Implementation and testing environment	96
4.6.2	Performance and visual quality	97
4.6.3	Limitations and future work	106
4.6.4	Exemplary architectural scene walkthrough	108
5	Conclusion	111
	List of Figures	114
	List of Tables	117
	Bibliography	119

Introduction

Materials obtain their distinctive appearance from the way they reflect incoming light. Rough surfaces appear uniformly shaded as they diffuse light equally into all directions, while smooth surfaces sharply reflect their environment. *Measuring* the optical behaviour of a material allows a realistic simulation of light transport, as often desired in computer graphics. After establishing how incident light from certain directions is scattered into all possible outgoing directions, the material can be rendered in arbitrary lighting environments (Fig. 1.1).

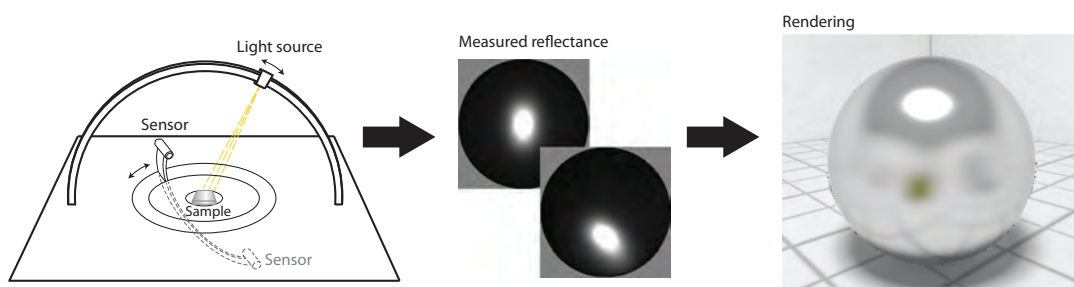


Figure 1.1: A special device is used to establish how incident light is transformed into outgoing light upon encountering a material. From this measured data, the material appearance can then be simulated under different illumination conditions.

When a light source illuminates a scene, light is reflected or transmitted off surfaces several times before reaching the eye. A complete computational solution of this *global* light propagation is in general too expensive for dynamic real-time applications. However, when scenes are assumed to be *static*, the expensive global propagation of light has to be simulated just once, as it does not change over time. The amount and distribution of incident light at each scene point can thus be precomputed and stored together with the scene geometry, e.g. using so-called *photon maps* [Jensen 1996]. Rendering a particular view of the scene then reduces to the local computation of determining which parts of the incident light are scattered into the viewing direction

according to the local material properties. When the view is allowed to be interactively changed at run-time, this *integration of incident light* according to material and viewing direction has to be performed at each pixel, 30-60 times per second.

A complete integration of incident light at each pixel is still not feasible in real-time. Thus, simplifications must be made in a way to optimally balance the performance vs. correctness tradeoff for a specific task. In this thesis, an algorithm is presented that approximates the local illumination integral by considering only the most relevant parts of the incident lighting and by precomputing all diffuse illumination. This requires the entire scene (lighting, geometry and materials) to be static, but achieves a convincing material appearance during real-time walkthroughs. The proposed algorithm enriches static diffuse environments by dynamically computing a local-illumination overlay of measured view-dependent reflection (Fig. 1.2).

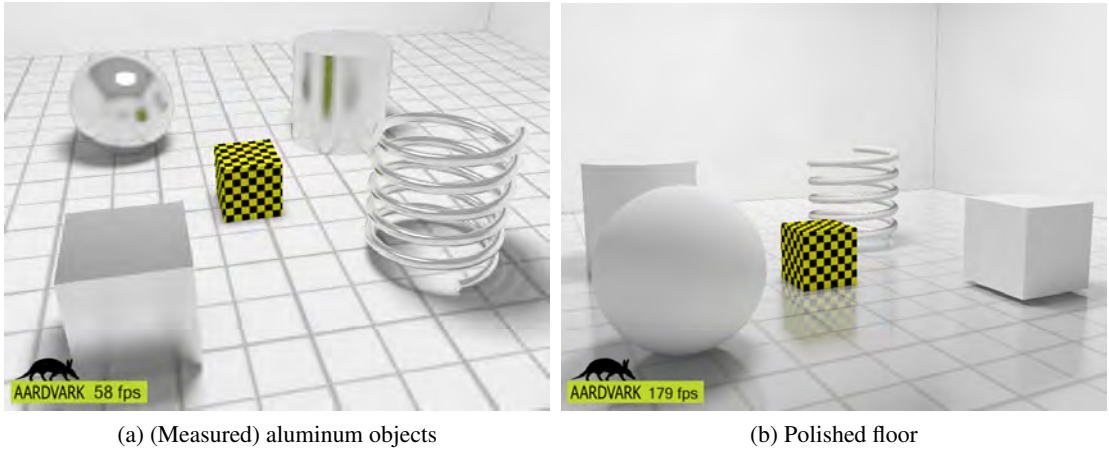


Figure 1.2: Scenes rendered with the proposed algorithm

1.1 Motivation

Measurement yields a realistic quantification of light scattering properties. One application of rendering measured materials is illumination and material prototyping: image synthesis can prove an invaluable tool for previewing hypothetical modifications, when the actual creation of another prototype would outrun one's budget. For such purposes the simulated illumination is hardly allowed to differ from the real-world ground truth, thus all photometric representations and calculations have to be realistic. Due to the high correctness requirements, only parts of this problem are solvable in real-time. The correctness vs. performance tradeoff in rendering can be mitigated by limiting the degrees of freedom at run-time: fixing scene parameters such as geometry, materials, lighting or the view allows precomputation and reuse of light propagation information, thus reducing the workload at run-time. Scene viewing applications for presentation purposes are an example where limiting everything but the view allows the precalculation of the global view-independent illumination. Then, high-quality walkthroughs are possible in

real-time since only the last view-dependent step of light propagation has to be evaluated at run-time.

Another application field of fast material rendering are virtual reality environments or games. Immersion into a virtual world relies on the *suspension of disbelief*, i.e. users should accept a simulated environment as real. A realistic material appearance is just as vital to this goal as detailed geometry or convincing animation.

For dynamic scenes, the representation of incident light has to be updated for every frame. If this is feasible, fully dynamic applications can also benefit from this thesis, as visually convincing materials significantly contribute to the mood and setting of a scene. As render-times are critical even to animation movie and visual effects production, algorithms for fast material rendering are desirable in the film industry as well.

The work of this thesis was initially motivated by a particular scene viewing application that requires *both* real-time performance and illumination correctness to a certain extent. This application will be described in the following section. Beyond that, the presented algorithm is applicable to any real-time application that benefits from a realistic material appearance.

1.2 Application framework

In the course of the HILITE project at the VRVis research center in Vienna, an architectural scene viewing application is being developed that allows real-time walkthroughs through globally illuminated scenes. Since everything but the view is static, multiple bounces of light transport can be precomputed. From the results of this global simulation, representations of local incident light are derived and stored for each scene object. Using these representations, the proposed algorithm then computes the last, view-dependent bounce of light propagation — i.e. it determines the amount of light arriving at the eye by means of reflection and transmission — in real-time.

One of the purposes of the HILITE application is illumination prototyping. If the global illumination precomputation provides physically correct results, the overall light distribution in the scene qualifies for illumination prototyping purposes. Material previewing can be done to a certain extent, as the real-time rendering of view-dependent materials requires substantial simplifications. However, the most significant approximations are made in representing the incident light at each scene point, thus reflections in materials will look slightly tilted or shifted, but still plausible.

Another goal of applications like HILITE is achieving a visually impressive presentation of illumination results. The thesis at hand contributes to this goal, as view-dependent effects in the last bounce of light propagation potentially cause a substantial visual enrichment to otherwise diffusely illuminated scenes (see Figure 1.3). Since the constraint of physical correctness is less tight for this purpose, the simplifications necessary to achieve real-time frame rates are tolerable. The work of this thesis will provide the HILITE project with a basic material rendering framework for the presentation of novel illumination solutions.

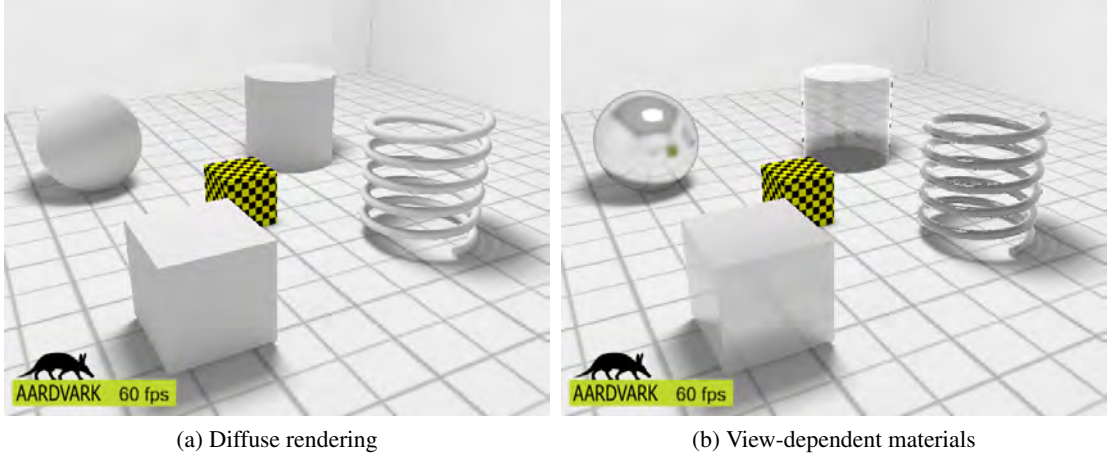


Figure 1.3: Visually enriched rendering due to view-dependent materials

1.3 Overview

In this thesis, an algorithm is presented that computes local view-dependent illumination of static scenes featuring measured materials. The initial input to this thesis is an offline global illumination (GI) simulation, whose results are stored as *light maps* of diffuse irradiance. These light maps are then used for rendering diffuse illumination during scene walkthroughs. The GI-simulation itself is not within the scope of this thesis, but provided by the HILITE application or any other offline renderer, e.g. using *path tracing* [Kajiya 1986], *radiosity* [Goral et al. 1984], [Cohen et al. 1993] or *photon mapping* [Jensen 1996]. Thus, the method proposed in this thesis does not compute global illumination itself; additionally to rendering the light-mapped scene, it computes a high-quality overlay of local specular illumination by sampling the static incident lighting according to the local scattering properties. Materials are represented using an analytic parametric model based on the model of Kurt et al. [Kurt et al. 2010], whose parameters can be either user-specified or estimated by measurement fitting. For dynamically computing the specular part of the local illumination integral according to material and view, two variants of a rendering algorithm are presented in this thesis:

1. **A cube map-based approach for rendering curved surfaces.** A single cubic environment map (cube map) [Greene 1986] is obtained at each scene object by rendering the light-mapped scene as seen from that object. The cube map then serves as an approximate representation of the local incident radiance distribution. If, according to *Heckbert's notation*¹ [Heckbert 1991], the initial GI-simulation computes paths $L(D|S)*E$, baking the $L(D|S)*$ -results into light maps allows paths $L(D|S)*DE$ when rendering the light-mapped scene. With the cube maps at hand, an additional overlay of $L(D|S)*D\tilde{S}E$ can be computed during walkthroughs, where the *tilde* in $\tilde{S}$ means that substantial approximations are made

¹Heckbert's notation describes paths of light in a scene: *L*: light is emitted from a source. *D*: light is transported via diffuse reflection. *S*: light is transported via specular reflection or transmission. *E*: light reaches the eye

during that bounce; a single cube map per object can only provide a rough approximation of correct specular reflection. Note that it is theoretically also possible to obtain the cube maps directly by global illumination, e.g. by directly path tracing the pixels of the cube map. Then, possible paths of the cube map approach would be $L(D|S)*\tilde{S}E$. In this thesis, however, only the light-mapped scene is rendered to obtain the cube maps, as at the time of implementation, the HILITE application could produce GI-light maps, but not GI-environment maps. Thus, in the current implementation, reflected objects do not feature view-dependent materials. Incorporating view-dependent effects into the static cube map generation would enable this without additional run-time cost.

Once a cube map has been obtained at each object, it is reused as an approximative representation of incident light across all points of that object. The specular part of the outgoing radiance is then obtained by sampling the cube map according to current view and local material properties.

2. **A planar reflection rendering approach for rendering large planar surfaces.** Cube maps are not suitable for reuse across large objects, as they correctly capture the environment at a single point in space. For the common case of large *planar* surfaces (e.g. walls, floor), the incident illumination is represented differently in this thesis: before rendering an actual frame, the light-mapped scene is rendered into a buffer as seen from a mirrored point of view [Diefenbach et al. 1997]. This buffer then captures the incident radiance from the perfect mirror direction at each point in the planar reflector. As this additional render pass must not become too expensive, again, only the light-mapped scene is rendered for this purpose in this thesis, which allows evaluating paths $L(D|S)*D\tilde{S}E$ for shading planes. If it is affordable to consider recursive reflections in the mirror rendering, paths $L(D|S)*D\tilde{S}+E$ are possible, but this is not yet implemented in this thesis. Again, the tilde in $\tilde{S}$ indicates substantial approximations in the specular bounce, as the mirror rendering is only correct for a single incident direction at each point of the planar reflector.

The mirror rendering fails to capture incident light from other directions than the mirror reflection direction. For simulating glossy reflections, incident light from slightly off-mirror directions is approximated by sampling neighboring pixels of the mirror buffer representation. For rather narrow specular lobes, this proves to be a plausible approximation (see Figure 1.2b)

Reusing cube maps at multiple points in space or sampling a mirror buffer for off-mirror directions are severe approximations of the actual incident lighting. Since the main goal of the algorithm is a convincing material appearance, such spatial approximations of the incident lighting are more tolerable than approximations of scattering properties; the overall material look does not forfeit plausibility when the reflected environment is slightly tilted or distorted.

Apart from representing materials and incident light, another focus of this thesis is on the *fast integration of incident light*, as the specular part of the outgoing radiance has to be computed dynamically at each pixel, 30 - 60 times per second. Thus, the feasible budget of samples at each pixel is very limited. A substantial part of this thesis deals with the question of how to optimally place and filter these few samples to obtain a smooth, noise-free rendering in real-time.

1.4 Contributions

In this section, the main contributions of this thesis are identified. While the first contribution describes an aspect that is mainly of practical use, the latter three can be termed as scientific contributions.

1.4.1 Obtaining a compact material representation from measured data

In the course of this thesis a material editor was developed that allows automatic fitting of a parametric model to measured material data. Thus, megabytes of measurement tables are reduced to just a handful of parameters that, together with the model, retain the essence of the scattering distribution. A hierarchical brute-force approach is employed to explore the high-dimensional parameter space of possible models, striving for parameter values that minimize the difference from the measurement. Users can steer the fitting into a desired direction by manually constraining certain parameters of the search space. Furthermore, material parameters are directly adjustable with instant visual feedback such as scattering distribution plots or real-time rendering of dummy geometry using the current settings. The material fitting and editing framework is described in Section 3.4.

1.4.2 Extension of an existing BRDF model to support transmission

For the representation of materials, the anisotropic BRDF model proposed by Kurt et al. [Kurt et al. 2010] is employed and extended to represent a transmittance distribution additionally to reflectance. Specifying transmission lobes analogously to reflection allows expressing different levels of blur in the transmission. Thus, clear transparency can be simulated as well as media featuring surface roughness or internal diffusion, such as frosted glass or semi-transparent foil. All details on the extended material model can be found in Section 3.3.

1.4.3 Real-time integration of incident lighting

The main goal of this work is to achieve plausible real-time rendering of architectural indoor scenes featuring measured materials. For a convincing material appearance it must be determined how much light is scattered into the viewing direction, considering all relevant incident directions. From the importance sampling formulae presented by Kurt et al. [Kurt et al. 2010], a uniform sampling scheme is derived to deterministically sample relevant directions of incident light, which, when filtered accordingly, cover the most essential part of the involved light transport. Thus, the proposed algorithm achieves a biased but stable and convincing material appearance at real-time frame rates.

The substantial difference of this method to existing sampling approaches [Colbert et al. 2007], [Kurt et al. 2010] is that samples are taken exclusively within the specular lobe, however, neither randomly nor quasi-randomly, but equally spaced in a fully deterministic way. This decision is motivated by the extremely low sampling budget at run-time (5 - 20 per pixel), since only strategic placement can *guarantee* a perfectly symmetric lobe coverage. To overcome undersampling artifacts, samples are filtered using MIP-maps of the incident light. Thus, reflection

lobes are uniformly covered and a smooth, noise-free result is obtained even when very few samples are taken [Colbert et al. 2007]. Since the integration domain of specular reflection is reduced to the most relevant part (the lobe), performance is gained during walkthroughs at the price of accepting a biased result. Whenever the camera is still, an unbiased solution of the illumination integral is computed by quasi-random importance sampling and blended with the previous rendering result, ensuring fast convergence over time. The rendering algorithm is the main contribution of this thesis and will be described in Chapter 4.

1.4.4 Representing and sampling incident light

As the scene is assumed to be static, the distribution of incident light at each scene point does not change over time and can be precomputed. For all geometry *except large planes*, the incident lighting is approximated by a single static cubic environment map per object, which is obtained by rendering the light-mapped scene as seen from the object. Since in architectural scenes objects can often be assumed to be bounded by planes (e.g. walls, floor), a better estimate of the reflected geometry is obtained in this thesis by intersecting secondary rays with user-specified bounding planes and looking up the incident light from these directions instead (similar to the method of Popescu et al. [Popescu et al. 2006]). Thus, an object on a tiled floor will correctly reflect the pattern of the tiling as if ray-traced (see Figure 1.4).

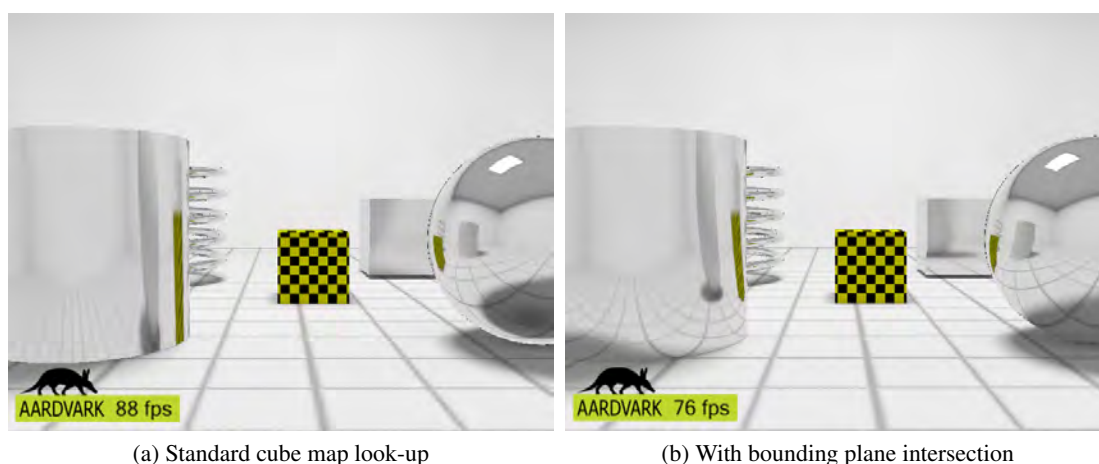


Figure 1.4: Cube map-lookup: intersecting secondary rays with bounding planes like the floor or the ceiling yields a better approximation of the reflected geometry than looking up the secondary ray direction itself

Representing incident light using environment maps works best for small, curved objects. Large planar surfaces such as floors or walls are comparatively large and arbitrarily close to objects reflected in them, which makes them unsuitable for reusing environment maps. For this purpose, the result of an extra rendering pass as seen from a mirrored camera location with respect to the plane is used as an approximation of the incident lighting of large planar surfaces. This representation fails to capture incident light other than from the mirror reflection direction,

but for rather sharp specular reflections, the neighbouring pixels of the mirror rendering prove to be a plausible approximation. In this thesis a projection step is introduced that directly maps angular sampling directions to sampling positions in the mirror rendering buffer. This contribution can be found in Section 4.5.

1.5 Organization of this thesis

This thesis is structured as follows: in Chapter 2, theoretical foundations necessary to understand the presented concepts are explained. Chapter 3 deals with the question of how to optimally represent measured materials for the problem at hand. After reviewing the state-of-the-art in measured material description, two representations used in this thesis are presented and discussed. In Chapter 4, the state-of-the-art in material rendering is briefly reviewed, followed by the main contribution of this thesis: the algorithm to integrate incident light for the real-time rendering of measured materials. Results of the method are presented and discussed. Finally, the thesis concludes in Chapter 5.

Theoretical Foundations

This section introduces theoretical basic concepts required to understand the methods presented in this thesis. After a short definition of the *rendering problem* in Section 2.1, *optical properties of materials* and their relevance in rendering are discussed (Section 2.2). The approach of acquiring material properties by *measurement* is described in Section 2.3, and finally, a short introduction to *Monte-Carlo integration* is given in Section 2.4, as the latter is a crucial concept underlying the rendering method presented in this thesis.

But first, two essential definitions must be given, as their notion is required in all of the following theoretical sections and throughout the entire thesis: *directions* ω are expressed in this thesis as angle pairs (θ, ϕ) in a *spherical coordinate frame*. The *inclination angle* θ states the angular difference between surface normal n and the direction ω , while the *azimuth angle* ϕ is the angle between the projection of direction ω into the local tangent plane and the tangent vector direction (see Figure 2.1). A conic set of directions around a given direction ω can be expressed by a *solid angle*, defined by the size of the subtended area on a unit sphere around the cone’s apex [Szirmay-Kalos 2008]. Figure 2.1 illustrates the concepts of spherical coordinates and solid angles:

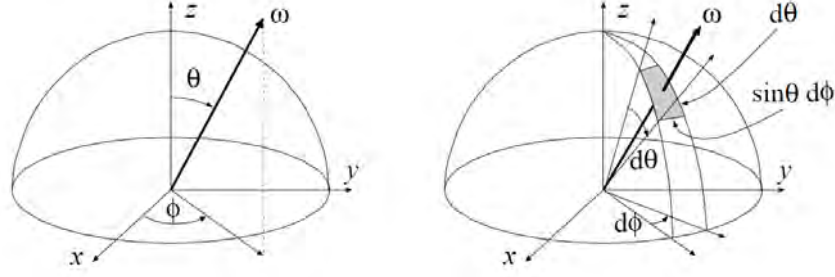


Figure 2.1: Illustration of expressing directions using spherical coordinates (θ, ϕ) (left image) and the concept of expressing sets of directions using solid angles (right image). Note that the surface normal direction is denoted as the z -axis in these images, the local tangent plane by the xy -plane, and the tangent vector direction by the x -axis. Images taken from Szirmay-Kalos [Szirmay-Kalos 2008].

2.1 The rendering problem

Rendering an image of a virtual scene involves determining the amount of light that reaches the eye. For each image pixel it is computed how much light arrives from that pixel's viewing direction ω_o by means of emission, reflection and transmission. As emitted light bounces off surfaces multiple times before possibly reaching the eye, this *global evaluation* of light transport is quite costly. In 1986, J. Kajiya published the *rendering equation*, a fundamental law that fully describes the light transport of global illumination [Kajiya 1986]:

$$L_o(x, \omega_o) = L_e(x, \omega_o) + \int_{\Omega} L_o(h(x, -\omega_i), \omega_i) \cdot f_s(\omega_o, x, \omega_i) \cdot \cos(\theta_i) d\omega_i \quad (2.1)$$

This integral equation quantifies the *exiting radiance* L_o of x , i.e. the amount of light sent from scene point x into the viewing direction ω_o . This amount comprises the all *emitted light* L_e in case x is a light source, plus all light leaving x in direction ω_o by means of reflection or transmission, expressed by the recursive integral across the sphere of directions Ω . The expression $h(x, -\omega_i)$ refers to the scene point seen from x in direction $-\omega_i$, thus, the radiance sent towards x from all scene points visible from x is recursively considered. Light incident at x is attenuated by the function f_s , called *Bidirectional Scattering Distribution Function* (BSDF) which describes how much of the incident light is turned into outgoing light for any combination of directions ω_i, ω_o according to the material properties. Details and a proper definition of the BSDF will follow in the next Section. The additional attenuation factor $\cos(\theta_i)$ accounts for Lambert's law [Lambert 1760], i.e. the fact that an incident light beam's energy is distributed across a larger surface area when incident from an inclined angle, reducing its energy per infinitesimal surface element by a factor $\cos(\theta_i)$. Finally, the expression $d\omega_i$ refers to the differential solid angle defined by an infinitesimal cone around the incident direction ω_i , which gets smaller towards the

poles of the illumination sphere Ω (see Fig. 2.1), in particular: $d\omega_i = \sin(\theta_i) \cdot d\phi d\theta$ [Szirmay-Kalos 2008].

Theoretically, the rendering equation yields the correct global illumination result of each rendered scene point x . However, as the integral is recursive (the integrand contains L_o) with potentially infinite depth, an analytic solution is not feasible. Since light transport is a *contractive* phenomenon, weakening each bounce, terminating the recursion after an adequate number of bounces yields an approximate solution of the rendering equation. Methods achieving this include *path tracing* [Kajiya 1986], *radiosity* [Goral et al. 1984], [Cohen et al. 1993] or *photon mapping* [Jensen 1996]. However, these methods take seconds to hours to complete a single image, and are therefore as such not applicable to the real-time setting of this thesis.

When scenes are static, i.e. geometry, materials and lights do not change over time, substantial parts of the equation can be precomputed and reused for rendering different views of the same scene. Any of the aforementioned global illumination techniques can be employed to compute several bounces of light propagation, and an omnidirectional map of incident radiances $L_i(\omega_i)$ can be stored at each scene point x . Then, determining the amount of outgoing radiance into a viewing direction ω_o can be computed using the much simpler, recursion-free *scattering equation* [Veach 1997] (with x removed for better readability):

$$L_o(\omega_o) = \int_{\Omega} L_i(\omega_i) f_s(\omega_o|\omega_i) \cos(\theta_i) d\omega_i \quad (2.2)$$

With a representation of $L_i(\omega_i)$ and $f_s(\omega_o|\omega_i)$ at each scene point, rendering an arbitrary view of the scene is reduced to integrating products of incident light and BSDF across the sphere of directions. One way of correctly solving the scattering integral is *Monte-Carlo quadrature*, sampling the integrand at a large number of random directions (see Section 2.4). Approximately solving the scattering integral allows reducing the evaluation cost even further, enabling real-time rendering approaches like the one presented in this thesis. The focus of this thesis with respect to the scattering equation is how to optimally represent and evaluate L_i and f_s to allow real-time walkthroughs of static scenes, while keeping the appearance of involved materials as realistic as possible.

Real-time rendering in the context of this thesis means that at the very least 20 - 30 frames per second are computed, such that a human viewer can perceive camera motion as smooth and the system remains responsive at all times, allowing unhindered interaction.

2.2 Material optics

The distinct appearance associated with a material arises from its light scattering distribution. In this section, the notion of the *BSDF* is introduced - a function to quantify this distribution. The term of physical plausibility of BSDFs is defined. Then, the two fundamental types of reflection depending on the surface roughness are described and finally the Fresnel effect and its role in the appearance of a material is explained.

2.2.1 Quantifying scattering properties - the BSDF

The scattering distribution of a material describes in which directions incident light is most likely to leave a surface upon encounter. This optical property is characterized by the Bidirectional Scattering Distribution Function (BSDF) [Heckbert 1991], a four-dimensional function that describes the quotient of differential observed radiance dL_o scattered into direction (θ_o, ϕ_o) per unit of irradiance dE_i incident from (θ_i, ϕ_i) [Veach 1997]:

$$f_s(\theta_i, \phi_i, \theta_o, \phi_o) = \frac{dL_o(\theta_o, \phi_o)}{dE_i(\theta_i, \phi_i)} = \frac{dL_o(\theta_o, \phi_o)}{L_i(\theta_i, \phi_i) \cos(\theta_i) d\omega_i} \quad (2.3)$$

where $d\omega_i$ is the differential solid angle defined by an infinitesimal cone around the incident direction (θ_i, ϕ_i) . The BSDF is proportional to the amount of light scattered in each outgoing direction for any incoming direction of light. It does not, however, directly specify probabilities or percentages; since the denominator of the fraction in Equation 2.3 is not bounded towards zero, the BSDF can become arbitrarily large. Note that formally including spatial dependency (surface point x) or light wavelength λ into the notion of f_s makes it up to six-dimensional. In this thesis, such variations are not included into the notion of f_s for reasons of readability. They can still be incorporated by defining a different 4D-BSDF at each scene point, or, if desired, for each wavelength.

As mentioned before, the BSDF does not directly specify probabilities or percentages. The probability of a light particle (photon) being scattered into a certain direction is defined by the *transfer probability density* [Szirmay-Kalos 2008]:

$$p(\omega_o | \omega_i) = p((\theta_o, \phi_o) | (\theta_i, \phi_i)) = f_s(\theta_i, \phi_i, \theta_o, \phi_o) \cos(\theta_o) d\omega_o \quad (2.4)$$

where $d\omega_o$ is the differential solid angle defined by an infinitesimal cone around the outgoing direction (θ_o, ϕ_o) . For derivation of Equation 2.4 and details please refer to this document: [Szirmay-Kalos 2008]. The BSDF can be seen as the combination of two scattering functions, one for the hemisphere above respectively below the tangent plane of a surface point: the BRDF (Bidirectional Reflection Distribution Function) f_r relates *reflected* differential radiance to irradiance, and gained much more scientific attention in the field of computer graphics than the BTDF (Bidirectional Transmission Distribution Function), which describes the distribution of *transmitted* light.

2.2.2 Physical plausibility

Real-world materials obey the laws of physics. In particular, they satisfy the laws of *reciprocity* and *energy balance*, which is required of BSDFs in order to be *physically plausible*. Reciprocity describes the following symmetry of light scattering [Veach 1997]:

$$\frac{f_s(\theta_i, \phi_i, \theta_o, \phi_o)}{\eta_o^2} = \frac{f_s(\theta_o, \phi_o, \theta_i, \phi_i)}{\eta_i^2} \quad (2.5)$$

where η_i is the refractive index of the medium of the incident ray and η_o the refractive index of the medium of the transmitted ray. Note that when both media exhibit the same refractive index, or when only reflection is involved, the denominator of Equation 2.5 is equal on both

sides of the equation and can be omitted. This corresponds to the well-known notion of BRDF reciprocity. The concept of reciprocity is important in this thesis, since material measurement usually relies on this principle in order to reduce storage requirements and measurement time by omitting measurements of the respective symmetric directions.

Energy balance refers to the fact that the amount of outgoing light cannot be larger than the incoming amount. Parts of the incident lighting might be absorbed by the material and transformed into other forms of energy such as heat. Non-absorbed light is either reflected or transmitted at the surface, but when no additional light is created from other forms of energy, the overall light leaving the surface has to be less or equal to the incident light. The ratio of outgoing light versus incoming light, sometimes called *albedo* of the material, is defined as [Veach 1997]

$$a(\theta_i, \phi_i) = \int_{\Omega} f_s(\theta_i, \phi_i, \theta_o, \phi_o) \cos(\theta_o) d\omega_o \quad (2.6)$$

where Ω is the spherical domain of outgoing directions and $d\omega_o$ is the differential solid angle defined by an infinitesimal cone around the outgoing direction (θ_o, ϕ_o) . Energy balance states that this ratio is bounded by 1 (100%) for all incident directions (θ_i, ϕ_i) . This property is essential to global illumination algorithms whose convergence relies on the contractive nature of light transport.

Splitting up the integration domain of Equation 2.6 into an upper and a lower hemisphere yields the percentages of overall reflected and transmitted light. The *reflectivity* of a material is defined as

$$r(\theta_i, \phi_i) = \int_{\Omega_{HS+}} f_s(\theta_i, \phi_i, \theta_o, \phi_o) \cos(\theta_o) d\omega_o \quad (2.7)$$

while Ω_{HS+} refers to the upper hemisphere ($\theta_o \in [0^\circ, 90^\circ]$). Note that f_s can in this case be replaced by a BRDF f_r , when a material does not involve transmission. Considering the lower hemisphere Ω_{HS+} with $(\theta_o \in [90^\circ, 180^\circ])$ yields the *transmissivity* of the material:

$$t(\theta_i, \phi_i) = \int_{\Omega_{HS-}} f_s(\theta_i, \phi_i, \theta_o, \phi_o) |\cos(\theta_o)| d\omega_o \quad (2.8)$$

Whenever a photon encounters a surface, these percentages define the probability of reflection, transmission and absorption, the latter computed as $1 - (r + t)$. These probabilities can be used in global illumination methods based on photon propagation, e.g. photon mapping [Jensen 1996], for deciding whether a photon survives a bounce and, if so, in which hemisphere it is likely to leave.

2.2.3 Diffuse and specular reflection

There are two basic types of reflection that have received substantial scientific attention: perfectly diffuse reflection and specular (glossy to mirror-like) reflection. In this section these two optical phenomena and their relevance in computer graphics is discussed.

Perfectly diffuse reflection (also called *Lambertian reflection*) refers to the phenomenon of incident radiance being scattered uniformly into all directions. Physical reasons of this behaviour include surface irregularities of approximately the size of the light wavelength, or local subsurface-scattering, i.e. light penetrating the surface, bouncing several times inside and exiting the entered surface into a new direction. Typical examples of diffuse materials include optically rough surfaces such as sand, paper, walls or substances featuring subsurface-scattering such as white marble.

The transfer probability of a Lambertian reflector is $p(\omega_o|\omega_i) = \frac{k_d}{\pi} \cdot \cos(\theta_o)$, i.e. it is more likely for a single photon to bounce into a direction close to the surface normal, but it does not matter from where it arrived [Veach 1997]. The constant k_d is the diffuse reflectivity of the material, i.e. the percentage of incident light leaving the surface due to diffuse reflection. Whenever *any* surface is viewed, the number of infinitesimal surface elements within a viewing ray is proportional to $1/\cos(\theta_o)$. For Lambertian surfaces, this causes the $\cos(\theta_o)$ -terms of transfer probability and viewed area to cancel each other out, yielding uniform radiance into all viewing directions, and thus, a constant BRDF: $f_r(\omega_o|\omega_i) = \frac{k_d}{\pi}$. Note that hardly any real-world materials are perfectly Lambertian, but as this simplified assumption of uniform outgoing radiance facilitates light transport simulation, the Lambertian model is widely used. Figure 2.2 shows a polar plot of the Lambertian BRDF and an example rendering of a diffuse sphere:

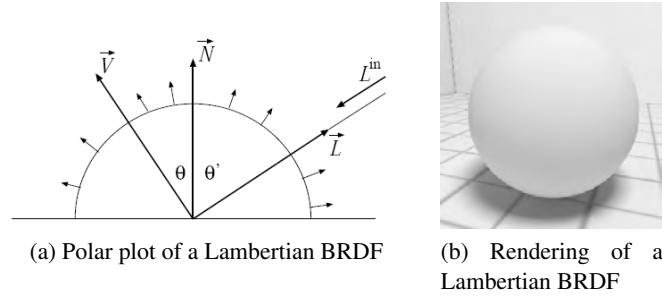


Figure 2.2: A polar plot illustrating the perceived uniform exiting radiance of a Lambertian BRDF. Illustration taken from Szirmay-Kalos [Szirmay-Kalos 2008]. The right image shows an example rendering of a diffuse sphere.

A different optical phenomenon comes into play when surfaces are smooth; the scattering distribution then features a certain directionality, preferring outgoing directions according to the *law of reflection*: $\theta_o = \theta_i$. The extreme example of such *specular reflection* is a mirror, reflecting incident light exclusively into the the perfect mirror direction $\omega_o = (\theta_i, \phi_i + 180^\circ)$. Less polished surfaces feature some roughness, causing reflection to occur mainly in directions similar to the mirror direction. This set of similar directions is called a *specular lobe*, due to its prominent shape when plotting a specular BRDF on a polar plot (e.g. Fig 2.3a). With increasing roughness, specular materials become less mirror-like but rather glossy, and finally almost diffuse (see Fig. 2.3).

There are several models of analytically describing specular reflection distributions, including the Phong-BRDF [Phong 1975], Lafortune-BRDF [Lafortune et al. 1997] or Cook-Torrance-

BRDF [Cook and Torrance 1981]. Lobes can also be anisotropic, when the structure of the surface roughness features directionality in itself, such as a rolling direction in brushed metals. Then, the gloss of specular highlights is more blurry in certain directions than others.

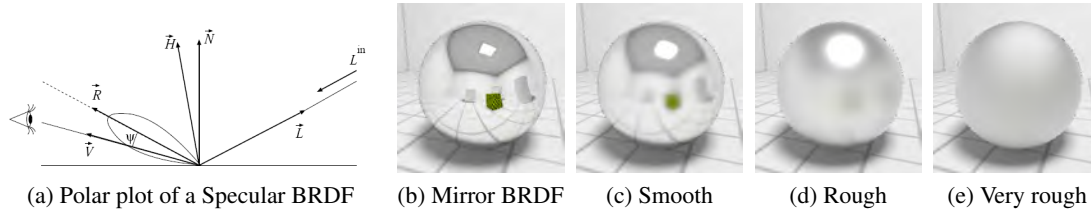


Figure 2.3: A polar plot illustrating the lobe of a specular BRDF. Note how most energy is transferred into directions around the perfect mirror direction, with a steep energy falloff at the borders of the lobe. Illustration taken from Szirmay-Kalos [Szirmay-Kalos 2008]. The rendered images show examples of specular materials with increasing roughness from left to right

The important difference between diffuse and specular reflection in this thesis is that diffuse (Lambertian) reflection is view-independent, and does not need to be recalculated when the view changes, while specular reflection does. As the view is the only degree of freedom during walk-throughs in this thesis, the ability of excluding diffuse reflection from the run-time computation is a prerequisite for the proposed method to run in real-time.

2.2.4 The Fresnel effect

Most materials feature multiple degrees of roughness, resulting in a combination of diffuse and specular reflection. In general, the ratio between the amounts of diffuse vs. specular reflection is not constant, but rather varies for different incident lighting- and viewing directions. Physically, the probability of specular reflection increases towards grazing angles, as photons are less likely to penetrate the surface to be transmitted or reflected diffusely. For instance, a surface of water or a glass window will not be very reflective when viewed at *normal incidence*, i.e. at $\theta \approx 0^\circ$. However, when viewed at grazing angles (glancing incidence, $\theta \approx 90^\circ$), it will sharply reflect the environment due to increased specular reflection.

This behaviour was observed and modeled by J.-A. Fresnel in his *Fresnel equations*, describing the transition of light into a medium of different refractive index. The Fresnel equations thus state which fraction of light is (specularly) reflected and which fraction is transmitted when light encounters a medium transition such as an object's surface. Simulating this effect by directly evaluating the complex Fresnel equations is too costly for real-time purposes, thus, C. Schlick proposed a rational approximation of the Fresnel equations expressing this effect of varying reflectance [Schlick 1994]:

$$F(u) = r_0 + (1 - r_0) \cdot (1 - u)^5 \quad (2.9)$$

where $u = \text{dot}(h, v)$, with h being the halfway-vector between view direction vector v and incident light vector i . The parameter r_0 is the ratio of specular reflectivity at normal incidence

($\theta_i = 0^\circ$) vs. the maximum specular reflectivity at glancing incidence ($\theta_i = 90^\circ$). Thus, a high value of r_0 results in a very weak Fresnel effect, i.e. almost the same amount of specular reflection regardless of the viewing angle, which is the case for most metals. On the other hand, materials with low r_0 are subject to a strong Fresnel effect, featuring specular reflection mainly at grazing angles, such as some plastics, varnish, ceramics, glass or water. Figure 2.4 shows some exemplary plots of Fresnel reflectance for varying angles, and Figure 2.5 shows some example renderings of objects with low to high Fresnel effect from left to right.

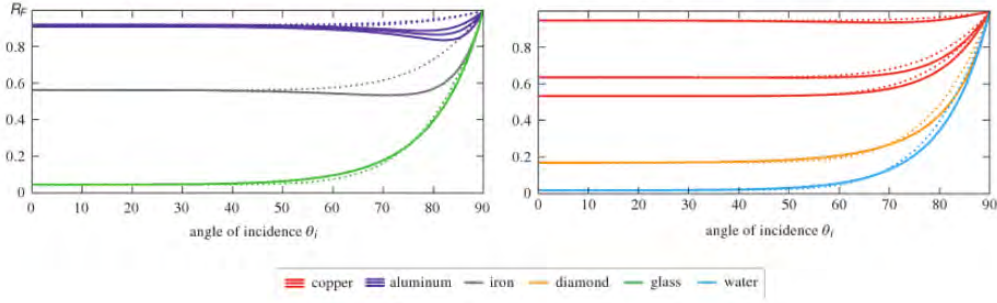


Figure 2.4: Fresnel reflectance for varying angles of incidence. The solid lines are obtained by fully evaluating the Fresnel equations, the dotted lines depict Schlick's approximation. In case of copper and aluminum, the three lines indicate the varying reflectance behaviour for different wavelengths (one line each for R,G,B). Note how glass or water feature a strong Fresnel effect, while the metals' reflectance varies only slightly for different angles. Images taken from Akenine-Möller et al. [Akenine-Möller et al. 2008].

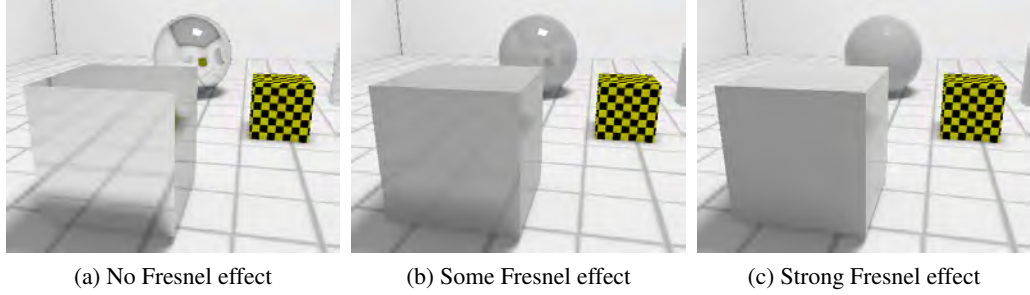


Figure 2.5: Renderings of materials featuring Fresnel effects (Schlick's approximation). The specular reflection of the metal (a) hardly varies for different viewing angles, while in (b) and especially (c), specular reflection occurs mainly at glancing incidence, i.e. at the outer borders of the sphere and the side faces of the cube.

By incorporating Schlick's approximation into the material representation of this thesis, materials featuring strong Fresnel effects can be rendered much more convincingly than without considering this phenomenon.

2.3 Material measurement

When the scattering distribution of a real-world material is required for the use in computer graphics, measuring poses a straightforward alternative to manual design. Measurement is conceptually much easier than modeling, as nothing needs to be known about the material or a model. However, recording a four-dimensional BSDF table is a tedious, time-consuming task that also requires specialized gear, such as a *gonioreflectometer* (e.g. [Ngan et al. 2005], [Ward 1992]).

The measuring procedure using a gonioreflectometer usually works as follows: a material sample is placed in a controlled lighting environment, such as a black, non-reflective dome. Then, the sample is successively illuminated from several representative incident directions, for each of which a mobile sensor is used to measure the radiance scattered by the sample into a dense set of outgoing directions. Thus, the exiting radiance $L_o(\omega_o|\omega_i)$ is measured for a large number of direction pairs. Under the assumption that nothing else but the single (point) light source contributes to the illumination of the sample, the scattering integral is zero for all incident directions but one. Thus, Eq. 2.2 simplifies to

$$L_o(\omega_o|\omega_i) = L_i \cdot f_s(\omega_o|\omega_i) \cdot |\cos(\theta_i)| \quad (2.10)$$

As the exiting radiance of the light source L_i is constant and known, this equation can be rearranged to derive f_s from the measured value of L_o . Thus, a table of measured BSDF-values f_s is obtained and can be used in computer simulations. Non-measured directions can only be reconstructed from the available measurements, therefore care must be taken to record the BSDF at a sufficient resolution to allow reasonable reconstruction. In Chapter 3, methods of efficiently representing measured scattering distributions for particular tasks are discussed.

For the work of this thesis, project partners provide BSDF measurements from their own database. The datasets are delivered as text-file tables containing transfer probabilities instead of BSDFs, i.e. tabulated values of $p(\omega_o|\omega_i) = f_s(\omega_o|\omega_i) \cdot |\cos(\theta_o)|$. From them, BSDF values can be easily derived by dividing by $|\cos(\theta_o)|$. The measured samples are assumed to be spatially and temporally non-varying. Furthermore, the wavelength (color) of the transported light is ignored; just a single-channel intensity value is measured for each direction pair. Figure 2.6 shows an example visualization of a measured anisotropic aluminum sample.

There are two kinds of measurement artifacts present in the data, which are hard to get rid of, regardless of the employed device: first, sensor noise is present especially in directions of low light transport (dark regions). Second, some parts of the hemispherical domain cannot be properly measured due to line-of-sight issues in the recording device; depending on the setup, both the mobile light source and sensor cannot be placed in exactly the same spot, or maybe other parts of the setup occlude or hinder an omnidirectional measurement. This is barely visible in the aluminum dataset (Fig. 2.6), but at the very left side of each subplot is a pitch-black spot that contains no valid information. Diffuse materials suffer more gravely from these shadowing artifacts, as evident later in Figure 3.14a.

A last notable fact regarding the measured tables is that only a relatively small portion of each subplot is covered by specular reflection effects. In the mainly specular aluminum measurement (Fig. 2.6), most measured values are close to zero (black pixels)). When a material features

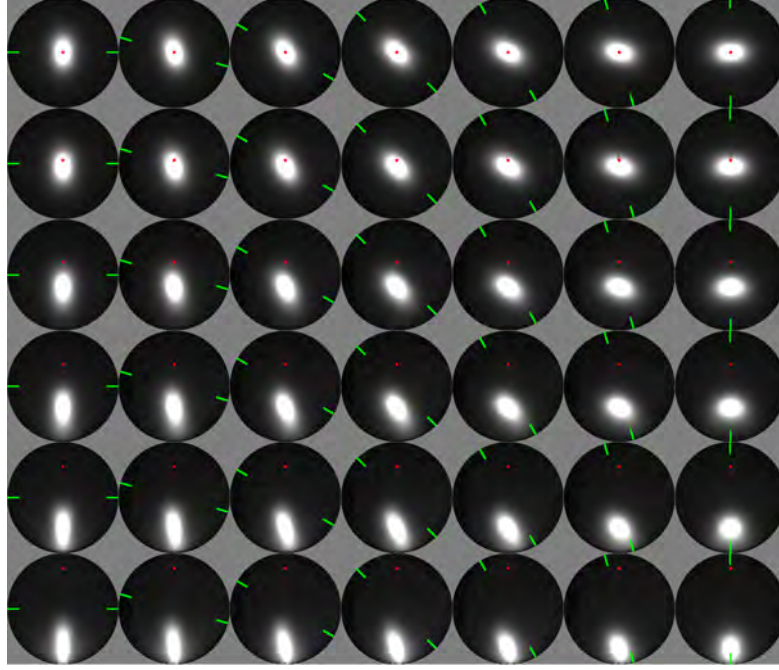


Figure 2.6: This array of 2D-plots visualizes the measured 4D-scattering distribution of a *scatergloss aluminum* sample. Each circular subplot corresponds to a particular incident direction ($\theta_i = \{0^\circ, 5^\circ, 20^\circ, 35^\circ, 50^\circ, 65^\circ\}$ from top to bottom, $\phi_i = \{0^\circ, 15^\circ, 30^\circ, 45^\circ, 60^\circ, 75^\circ, 90^\circ\}$ from left to right). Due to symmetry reasons of this particular material, further ϕ_i -angles do not need to be recorded. Within each circular (polar) plot, pixel intensities denote transfer probabilities into particular outgoing directions, with $\theta_o = 0^\circ$ in the center and $\theta_o = 90^\circ$ at the border of each circle. Note that the cubic root was applied to each intensity to reduce clamping for a better visualization. The light source is always positioned at the top of each plot, denoted by the red dot in each plot. The material sample is rotated by each plot's ϕ_i -angle. The green line indicates the aluminum's brushing direction. Note how the specular highlight blurs much more in directions orthogonal to the brushing direction than parallel to it, as expected of brushed metals. Transmissive materials would feature an additional plot matrix for the transmission hemisphere; this aluminum dataset does not.

considerable diffuse reflection, these off-specular directions are likely to contain the value of the diffuse BRDF part $\frac{k_d}{\pi} \cdot |\cos(\theta_o)|$, as Lambertian reflection happens equally for all outgoing directions. This will come in useful in this thesis when estimating the diffuse coefficient k_d in the course of fitting a model to the measured data as described in Chapter 3.

2.4 Monte-Carlo integration

Integrals can be numerically approximated by treating them as a finite weighted sum of samples across the domain [Szirmay-Kalos 2008]. The sample weighting function is called *quadrature*

rule, stating how samples should be combined to yield the integration result. The most common rules, such as the *trapezoidal rule* or *Simpson's rule* perform a uniform subdivision of the integration domain for sampling. Higher-dimensional integrals are essentially treated as combinations of one-dimensional integrals, subdividing the domain separately along each dimension. A linear increase of the sampling resolution reduces the integration error of such uniform quadrature rules by a factor of $O(n^{\frac{1}{d}})$, where n is the number of samples and d is the number of involved dimensions [Szirmay-Kalos 2008]. Thus, a linear improvement of the approximation requires increasing the number of samples exponentially in d . This *dimensional explosion* makes a numerical solution of high-dimensional integrals using uniform quadrature rules a tedious task.

The idea of Monte-Carlo integration is turning the integration problem into an expected value calculation of the integrand. Suppose the function $f(z)$ is to be integrated across the domain V , and the random domain variable z is uniformly distributed across V with $p(z) = \frac{1}{V}$, then the expected value $E[f(z)]$ can be written as [Szirmay-Kalos 2008]

$$E[f(z)] = \int_V f(z) \cdot p(z) \, dz = \frac{1}{V} \cdot \int_V f(z) \, dz \quad (2.11)$$

Thus, computing the expected value of $f(z)$ and dividing it by $\frac{1}{V}$ yields the integral of $f(z)$ across V . The expected value can be estimated by the arithmetic mean of n independent samples:

$$\hat{E}[f(z)] = \frac{1}{n} \sum_{i=1}^n f(z_i) \quad (2.12)$$

In Monte-Carlo quadrature, n uniform random samples are taken to estimate the expected value of $f(z)$, and then Equation 2.11 is applied to estimate the integral. Increasing the number of samples results in convergence towards the expected integrand. The obtained deviation from the real integral is $O(\frac{1}{\sqrt{n}})$, thus quadratically increasing the number of samples yields a linear decrease of the error [Szirmay-Kalos 2008]. Note that this error bound is independent of the integration domain's dimensionality, making Monte-Carlo integration particularly suitable for tackling high-dimensional integrals. Since no closed form of the integrand needs to be available, as it only needs to be sampled at arbitrary random points, it is also well applicable to integrating discontinuous or discrete functions, as often the case in rendering (e.g. lighting- or visibility maps) [Pharr and Humphreys 2004].

Monte-Carlo quadrature converges towards the expected value at a rate of $O(\sqrt{n})$. A faster convergence can be achieved when the sampling directions are not chosen at total random but deterministically from a *low-discrepancy sequence*. Numbers of a such a sequence are also called *quasi-random numbers*, as they feature some properties of random numbers, but in fact, follow a deterministic sequence. Quasi-random numbers exhibit a lower *discrepancy* than random numbers, meaning that arbitrary subsequences of such numbers are always evenly distributed on average. In other words, even for low numbers of samples, the distribution appears more uniform across the domain than when using random numbers (see Fig. 2.7).

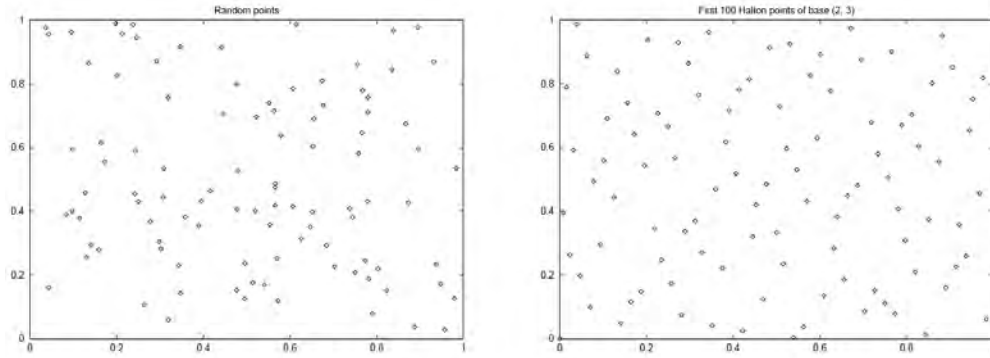


Figure 2.7: Quasi-random numbers are more evenly distributed across the domain than random numbers, regardless of how many samples are taken. Left image: random numbers, right image: quasi-random numbers. Images taken from Szirmay-Kalos [Szirmay-Kalos 2008]

In this thesis, Quasi-Monte-Carlo integration is used to estimate the unbiased *scattering integral* (Eq. 2.2) whenever the camera is still. Then, thousands of samples can be taken over consecutive frames to ensure convergence towards to the correct integration result over a short time. During walkthroughs, sufficient numbers of (quasi-)random samples are not affordable, thus the view-dependent integral is approximated differently while the camera moves (see Chapter 4).

Material Representation

The most relevant aspect of materials in computer graphics is how they interact with incident light. These optical properties determine the familiar appearance of a material, allowing us to recognize a substance under different lighting conditions and environments. One of the most important optical properties is the *scattering distribution*, quantifying how much of the incoming light is scattered into each possible outgoing direction. This property can be completely described by the *Bidirectional Scattering Distribution Function* (BSDF), as explained in Section 2.2. This chapter deals with how to optimally represent the scattering distribution for the problem at hand, focusing on *directional effects* such as gloss, specularly or Fresnel effects. Other optical properties and effects such as polarization, fluorescence or dispersion are not dealt with in the scope of this thesis, as they seem not as crucial to the immersion into architectural indoor scenes as the aforementioned directional phenomena.

There are two common ways of acquiring BSDFs of materials: *measuring* and *manual model specification*. While measuring potentially yields a very realistic quantification of the light transport for the recorded directions, experienced artists may be able to specify essentially the same properties in much less time. For the work of this thesis, *measured* BSDFs are provided by the project partners, since they allow a very realistic quantification and thus qualify for illumination prototyping, and, most importantly, because they already have a large database of measurements at their disposal.

The central question of this chapter is how to optimally represent the measured BSDF tables at hand for the tasks of this thesis. The two main high-level requirements of the HILITE project are:

1. A global lighting simulation that is as *physically correct* as possible
2. An impressive visual presentation of scenes during *real-time walkthroughs* including view-dependent effects and convincing material appearance

These two somewhat contradictory requirements (correctness vs. real-time) are tackled by the following techniques:

1. A precomputed global illumination simulation using the measured material properties, storing the simulation results as irradiance- and incident radiance maps together with the scene geometry
2. An approximative solution of the last view-dependent bounce during real-time walkthroughs, using the physically correct incident light information from the precomputation

During the global illumination simulation the material representation should be as realistic as possible and not introduce any significant systematic error. Thus, the measured tables are employed directly for this stage (see Section 3.2), reconstructing non-measured directions on the fly from the nearest neighbour measurements, guaranteeing an in-between result but no systematic error due to suboptimal model assumptions. During the run-time simulation of the last view-dependent bounce of light transport, the material representation should allow fast integration of incident lighting, and not involve any major reconstruction effort of non-measured directions. Additionally, the correctness requirements are more relaxed in this stage, as the main purpose is a plausible, impressive presentation of the scene during walkthroughs. Thus, a parametric model is fitted to the measured data, allowing the compact storage of many different materials in the graphics memory for rendering at the same time (see Section 3.3). Furthermore, the model allows the derivation of sampling formulae that allow a fast approximation of view-dependent effects, making it very suitable to real-time walkthroughs.

This chapter is structured as follows: first, related work in material representation is reviewed in Section 3.1. Then, the two representations employed in this thesis are elaborated, *direct measurement table evaluation* for the global illumination simulation (Section 3.2), and *a parametric model* for fast integration in the real-time rendering stage (Section 3.3). Finally, results, implementation details and limitations of the representations are presented in Section 3.4.

3.1 Related work

In this section, previous work on representing the scattering properties of materials is reviewed. The discussed approaches can be grouped into three categories, in order of increasing editing flexibility but decreasing generality:

1. Direct storage and evaluation of BSDF measurement tables
2. General BSDF approximation techniques
3. Parametric BSDF Models

Direct table representation: The BSDF tables obtained by measurement directly specify BSDF values (or transfer probabilities, from which the BSDF can be easily derived) for all measured directions of the 4-dimensional angular domain $(\theta_i, \phi_i, \theta_o, \phi_o)$. If the measurement resolution is high enough, these tabulated data can be directly used as look-up tables for rendering. Since no assumptions about the contents of the measured table are made, arbitrary scattering distributions can be represented, making this method the most general material representation.

However, there are several reasons why direct evaluation is suboptimal for real-time rendering: first, tables containing a sufficient number of samples of the 4-dimensional BSDF require a vast (up to gigabytes) amount of storage. Second, systematic edits such as making a material shinier are very hard to incorporate into a measured table. Third, due to time constraints, most measurements are recorded at quite a low resolution, especially in the domain of incoming directions [Rusinkiewicz 1997]. Then, non-measured directions have to be reconstructed on-the-fly (e.g. [Sari 2008]) (or in advance, but then memory requirements would, again, explode). On-the-fly - quadrilinear interpolation of missing directions in the $(\theta_i, \phi_i, \theta_o, \phi_o)$ -domain is the most simple approach, but fails if the measuring resolution is too low, as illustrated in Section 3.2. Rusinkiewicz proposed transforming the measurements such that they are parameterized by *half-angles and difference angles* instead of incident and outgoing angles [Rusinkiewicz 1998], allowing a more efficient representation of specular lobes, as they are always defined around $\theta_h = 0$, enabling better compression and approximation. Additionally, as the lobes of all measured incident directions get aligned, the half-vector parameterization is much more suitable for direct interpolation than the incident/outgoing-parameterization, as illustrated in Section 3.2.

Direct evaluation requires a lot of storage and involves reconstruction of non-measured directions by finding nearest neighbours. Thus, it is inferior regarding real-time evaluation using parallelizable GPU-shaders, when compared to different representations that can be sampled more efficiently: substantial scientific attention has been given to *approximating BSDF measurements with intermediate representations* (categories 2 and 3 of the previous enumeration), striving for advantages such as lower memory consumption, improved editability or faster evaluation and reconstruction. Such derived representations will be discussed in the following.

Basis functions: A popular strategy of representing complex functions like BSDF measurements in computer graphics is representing them in a different, more appropriate *function basis*. The *spherical harmonics*-basis yields a very compact representation when the measured BSDF is smooth, as very few coefficients suffice to convey low-frequency information, and a large number of coefficients can be discarded [Westin et al. 1992], [Cabral et al. 1987]. An alternative basis applicable to hemispherical functions like BRDFs are the *Zernike polynomials*, yielding similar properties and advantages as spherical harmonics [Koenderink and van Doorn, 1998]. Both the spherical harmonics and Zernike polynomials are localized exclusively in the frequency domain, making them inefficient for representing high-frequency information. Band-limiting the signal, i.e. using an insufficient number of coefficients to approximate the signal results in visible *ringing artifacts* [Koenderink and van Doorn, 1998]. To overcome this low-frequency limitation, Schroeder and Sweldens proposed the use of *spherical wavelets*, an orthonormal basis on the sphere that is localized both in the spatial and the frequency domain [Schroeder and Sweldens 1995]. Spherical wavelet-coefficients for a given signal can be obtained using their proposed hierarchical lifting scheme. Due to the hierarchical representation, evaluating a spherical wavelet-representation of a BSDF is logarithmic in the number of coefficients, and thus faster than evaluating a spherical harmonics-representation (linear in the number of coefficients). According to the authors, the main computational cost is determining which spherical wavelet-coefficients to keep and which to discard, but as it has to be done only once per BSDF, it is no real limitation.

Factored decompositions: Another way of reducing the storage requirements of large BRDF/BSDF-tables is approximating them by a separable decomposition into multiple lower-dimensional functions. Such factored representations (e.g. [Kautz and McCool 1999], [Suykens et al. 2003]) can be stored as texture maps and used for interactive rendering. The proposed methods factor the 4D-BRDFs into products of 2D functions comprising very few terms, allowing rather coarse approximations of the underlying measurements, but interactive rendering of measured materials even around the early 2000's. In 2004, Lawrence et al. proposed an importance sampling scheme based on a similar factored representation that allowed using measured BRDFs in Monte-Carlo rendering with reduced sampling variance as opposed to prior sampling methods [Lawrence et al. 2004]. Two years later, Lawrence et al. proposed a *hierarchical decomposition* of spatially varying reflectance measurements into components of lower dimension such as 1D-functions representing directionality or 2D-textures representing spatial variance [Lawrence et al. 2006]. Such *inverse shade trees* can be edited by directly manipulating these low-dimensional functions in the leaves of the tree, allowing interactive editing of real-world spatially varying materials.

A data-driven approach of representing isotropic BRDFs was presented by Matusik et al. [Matusik et al. 2003], where measurement tables are treated as high-dimensional feature vectors with each measured value as one component. Then, dimensionality reduction tools are employed to compress the large measured tables. Editing and derivation of new BRDFs can then be done by linearly combining measured datasets according to some user-defined perceptually meaningful parameters such as *metallic-like*, *dustiness* or *rubber-like*, which are assigned as labels to each material dataset in advance. As the expressivity of this editing approach relies a large labeled database of measurements, the method is de facto limited to isotropic (i.e. highly symmetric) materials, as they require much less space and are much faster to measure than arbitrary substances [Matusik et al. 2003].

Analytic models: Most scientific attention has seemingly been given to parameterized *analytic material models*, as they are very efficient in representing the subclasses of materials they were designed for. The essence of the scattering distribution is captured using a fixed mathematical formula (the *model*), and numeric parameters can be altered to achieve varying appearance within the class of representable materials. Once the model has been chosen, a handful of parameters suffices to uniquely determine an entire 4-dimensional scattering distribution. Thus, parametric models are not only easily modifiable, but also the most compact representation of a particular class of materials, as only a few parameter values need to be stored. On the downside, a particular model can only represent a very narrow subspace of all possible scattering distributions, making this representation also less general in terms of expressivity than the previously mentioned approximation techniques. However, due to the compact storage, continuous evaluability and easy editing process, such analytic models are widely used in many computer graphics applications.

Analytic material models can be grouped into two categories: *physically based models*, with physically meaningful parameters that could theoretically be measured [Matusik et al. 2003], and *phenomenological* (or *empirical*) *models*, with perceptually meaningful parameters that are not necessarily physically justified. Due to their simplicity and fast evaluation, empirical models are especially popular in real-time applications and will be discussed here first. One of the oldest

and best-known empirical models is the *Phong* model [Phong 1975], simulating specular reflection within a lobe around the perfect reflection direction with adjustable degree of shininess. In 1977, Blinn altered the model such that it used the halfway-vector instead of the reflection vector [Blinn 1977], allowing faster evaluation and more accurate specular reflection at grazing angles [Ngan et al. 2005]. Lafortune et al. [Lafortune et al. 1997] proposed a generalized cosine lobe model that allows representing more general lobe shapes by combining multiple cosine lobes, enabling the simulation of effects like off-specular reflection or retro-reflection. Then, some models can be considered as in-between physically based and empirical, as they describe physically justified processes but use severe approximations or violate physics under certain circumstances. Two such examples are the Ward model [Ward 1992] and the Schlick model [Schlick 1994]: in 1992, Ward derived a simplified version of the physically based Cook-Torrance model [Cook and Torrance 1981], which is not physically plausible at grazing angles [Neumann et al. 1999] but allows representing isotropic and anisotropic reflection and is simple enough to be used in real-time applications. Schlick’s BRDF approximates isotropic and anisotropic reflection and Fresnel effects using rational approximations, enabling faster evaluation than similar physically based models. In 2010, Kurt, Krivanek and Szirmay-Kalos proposed a BRDF model that combines the expressivity of the previous two (anisotropy, Fresnel effect), is physically plausible under a larger number of parameter configurations and, due to an additionally introduced parameter, particularly suited for fitting to material measurements [Kurt et al. 2010].

All the aforementioned empirical models are particularly useful due to their simplicity and fast evaluation. Physically based models have a larger theoretical basis and their use even reaches beyond computer graphics. However, their complexity often makes them too expensive to be employed in real-time applications. Torrance and Sparrow introduced the first physically based BRDF model in 1967 [Torrance and Sparrow 1967], simulating a Gaussian distribution of microscale surface irregularities (*microfacets*). Cook and Torrance generalized the Torrance-Sparrow BRDF in that they allowed different microfacet distribution functions [Cook and Torrance 1981]. As opposed to the previously mentioned empirical models, the Fresnel equations are fully evaluated in the Cook-Torrance and Torrance-Sparrow BRDF. An even more complex physically based model was proposed by He et al. in 1991 [He et al. 1991]. It takes into account additional physical parameters such as light polarization, making it one of the most comprehensive but also one of the most computationally expensive models. While the high degree of realism provided by physically based models might be required in predictive rendering or other physical computations, the high evaluation cost is a price too expensive to pay in the real-time context of this thesis.

The BSDF representations reviewed in this section are just a selection of well known and highly discussed models and approaches. Please refer to survey papers on BSDF models (e.g. [Kurt and Edwards, 2009]) or on general BSDF representation (e.g. [Rusinkiewicz 1997]) for a more thorough elaboration. In the following, the two material representations employed in this thesis will be presented in detail, *direct table evaluation for global illumination simulation* and *a BSDF model for fast integration*.

3.2 Direct measurement evaluation for photon bounce simulation

The rendering calculation is split into two parts in this thesis: a precomputed view-independent global illumination simulation, and an approximation of the view-dependent last bounce of light transport at run-time (see Section 4.2). One goal of the global illumination simulation phase is calculating and storing the irradiance at each scene point, such that it can be used for fast rendering of diffuse reflections during walkthroughs. In this section, the material representation used during the multiple-bounce global illumination phase is presented.

The particular global illumination algorithm employed in the HILITE project is based on photon mapping [Jensen 1996]. The algorithm itself is not discussed in detail in this thesis; essentially, the scene illumination is simulated by emitting photons from primary light sources and tracing them through the scene for several bounces.

Whenever a photon hits a surface, its influence on the local irradiance is recorded. Then, it is either absorbed, reflected or transmitted by the surface. Assuming the photon arrived from direction $\omega_i = (\theta_i, \phi_i)$, it must be determined whether it survives the current bounce; the probability of reflection or transmission is defined by the reflectivity and transmissivity of the material. Thus, the probability of absorption, i.e. termination of this photon's lifespan is given by

$$Pr(absorbed \mid incident \text{ from } \omega_i) = 1 - (r(\omega_i) + t(\omega_i)) \quad (3.1)$$

As described in Section 2.2, the reflectivity $r(\omega_i)$ and transmissivity $t(\omega_i)$ for a given direction of incidence can be computed by integrating weighted measured transfer probabilities of the upper and lower hemisphere (Equations 2.7, 2.8). Values of $r(\omega_i)$ and $t(\omega_i)$ can be precomputed and tabulated for all measured incident directions, and interpolated during photon tracing for non-measured incident directions.

Using Equation 3.1, the fate of a photon encountering a surface is decided via *Russian Roulette* ([Arvo and Kirk 1990]), i.e. it survives if a random number between 0 and 1 is larger than the absorption probability in Eq. 3.1. If a photon survives a surface encounter, it has to be given a new direction. As mentioned in Section 2.3, the material measurement tables provided by the project partners directly contain *transfer probabilities* $p(\omega_o \mid \omega_i)$, i.e. probabilities of photons arriving from direction ω_i being scattered into a small interval around a particular direction ω_o . In order to generate a new direction for a photon arriving from direction ω_i , a *rejection sampling* procedure is employed using the measured transfer probabilities (similar to [Sari 2008]). Rejection sampling is a method of producing samples according to a given distribution. In our case, a candidate direction ω_o is generated and only accepted if its probability $p(\omega_o \mid \omega_i)$ is larger than some uniform random number x . Otherwise it is rejected, and a new candidate is generated and tested. Formally, the sampling operates as follows:

ω_i is the direction of the incoming photon

maxPDF = maximum value of $p(\omega_o \mid \omega_i)$ over all ω_o
 x = uniform random float number between 0 and maxPDF

While($x > p(\omega_o \mid \omega_i)$)

Generate new quasi-random candidate direction $\omega_o = (\theta_o, \phi_o)$

ω_o is the new photon direction

Since the candidate directions are quasi-random and uniformly distributed, the loop is guaranteed to terminate, as every direction is eventually generated and the largest possible value of x is contained in the transfer probability measurement table. Thus, a direction with probability equal or larger than x must eventually be found. Using this scheme, photons bounce according to the scattering distribution of the material.

The great benefit of directly using the measured probabilities in this scheme is that arbitrary materials can be represented and nothing else has to be known about the scattering distribution. Scattering effects that are too complicated to be captured by a compact model can be simulated to the extent they are covered by the measurement. One drawback of rejection-sampling the tables is that highly specular materials with narrow lobes (e.g. mirrors) require a much higher number of candidate directions on average before accepting them, as their non-zero domain of the transfer probabilities is very limited. This behaviour is somewhat paradox, since for materials with narrow lobes it is much more obvious, where reflected photons will bounce than for glossy or diffuse materials. Thus, in such cases, different methods of generating new photon directions such as *importance sampling* of a fitted model (see Section 4.4) are likely to produce similar results in much shorter time.

Another drawback of directly using the measurement tables is that not all directions ω_o — and much worse — extremely few directions ω_i are usually measured, thus, the unavailable transfer probabilities have to be reconstructed from the nearest neighbour measurements in the angular domain. The distance between neighbouring measured incident directions in our test-datasets is between five and fifteen degrees in θ and ϕ direction.

A simple way of reconstructing non-measured information is quadrilinear interpolation over $\theta_i, \phi_i, \theta_o, \phi_o$ in the measured intensity domain, i.e. interpolating the transfer probabilities of the nearest neighbour measurements depending on their angular proximity to the queried direction. This means that the transfer probability $p(\omega_o|\omega_i)$ is linearly interpolated between the probabilities of the measured neighbour directions $p(\omega_o|\omega_{i-1})$ and $p(\omega_o|\omega_{i+1})$. Figure 3.1 illustrates this interpolation. The effect of this linear reconstruction is that for a queried direction the neigh-

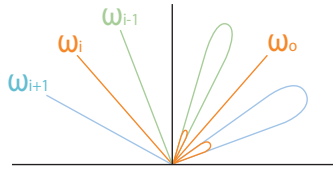


Figure 3.1: The transfer probability of (ω_i, ω_o) is reconstructed by looking up ω_o in the probability distributions of ω_{i-1} and ω_{i+1} and blending between them. Since ω_o is essentially the perfect reflection direction of ω_i in this example, the expected probability would be quite high. But $p(\omega_o|\omega_{i-1})$ and $p(\omega_o|\omega_{i+1})$ are almost zero, since ω_o is far off the perfect reflection directions of ω_{i-1} and ω_{i+1} , and thus, the reconstructed probability is grossly underestimated.

bouncing transfer probability densities are linearly blended but essentially remain at their original angular position. The expected reconstruction result would rather be a new position and shape of the specular lobe and not an intensity-blended darker average of two stationary lobes. The measured signal is not available at a sufficient resolution to be reconstructed by mere interpolation in the intensity domain.

A better alternative for reconstructing non-measured probabilities under the assumption of perfect specular reflection was proposed by Rusinkiewicz [Rusinkiewicz 1998]. The idea is that reconstruction is much more accurate when the lobes are aligned in the angular domain before the quadrilinear interpolation. Thus, the measured tables are reparameterized using half-angles and difference-angles instead of incoming and outgoing angles, thereby expressing the measurements in the so-called *Rusinkiewicz coordinate frame*. The reparameterization is performed as follows: consider the halfway-vector h of ω_i and ω_o . Instead of looking up the probability of ω_o in the measured distributions of ω_{i-1} and ω_{i+1} , a better look-up direction equivalent to ω_o in these distributions is defined as the vector ω_{i-1} (resp. ω_{i+1}) reflected by the half-way vector h . Intuitively, this scheme considers how close the queried outgoing direction is to the perfect reflection direction of ω_i , and interpolates between outgoing directions that are equally close to the perfect reflection directions of the measured incoming directions ω_{i-1} and ω_{i+1} . Thus, the interpolation of the lobe is performed in the angular domain rather than in the intensity domain, yielding a better reconstruction. In Figure 3.2, this reconstruction scheme is illustrated.

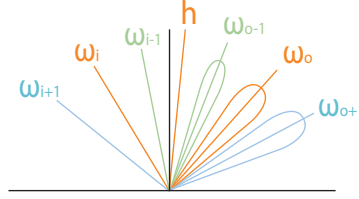


Figure 3.2: The transfer probability of (ω_i, ω_o) is reconstructed by looking up constructed outgoing directions $\omega_{o'-1}$ and $\omega_{o'+1}$ in the probability distributions of ω_{i-1} and ω_{i+1} and blending between them. $\omega_{o'-1}$ is constructed by reflecting ω_{i-1} about the half-way vector of ω_i and ω_o . Thus, the specular lobe is given a new shape and position instead of blending between two immobile specular lobes when using linear interpolation in the intensity domain.

Note that this half-vector-based reconstruction scheme works for all materials where the specular reflection roughly follows the law of reflection ($\theta_i = \theta_o, \phi_i = \phi_o + 180^\circ$). If this is not the case, e.g. with retro-reflective materials, reflecting about the half-way vector does not maintain the proximity to the principal reflection direction, and thus, the scheme fails.

An alternative approach to using the measured transfer probabilities to generate new photon directions is *fitting a mathematical material model* to the measured data, i.e. finding an analytic function f_s that closely approximates the measured BSDF (see Section 3.3). Then, non-measured transfer probabilities can be calculated analytically from $p(\omega_o|\omega_i) = f_s(\omega_o|\omega_i) \cdot \cos(\theta_o)$ (see Section 2.2) and used for rejection sampling, or, even better, *importance sampling* (see Section 4.4). Using a model instead of the measured table in offline photon-bounce simulation is preferable if the involved material is very specular, as the average iteration count of

the rejection sampling procedure becomes arbitrarily high in such cases. Also, models might be preferable when material measurements are very sparse, particularly noisy or feature substantial measurement artifacts. When this is not the case, direct evaluation is the more natural approach as it does not discard any measured information due to fitting.

3.3 A BSDF model for fast integration

One big advantage of the direct representation discussed before is the ability of representing arbitrary scattering distributions. On the downside, measurement tables require immense amounts of storage. Furthermore, the reconstruction effort to determine the BSDF for non-measured directions is not negligible, as it involves finding nearest neighbours and several look-ups just to determine one BSDF value.

Analytic material models remedy these problems at the cost of discarding non-modeled measurement information. They allow the derivation of importance sampling formulae to efficiently generate the most relevant sampling directions, which makes them well-suited to the problem of fast integration, as required in the run-time phase of this thesis' rendering algorithm. Furthermore, material models are very compact (a few parameters instead of thousands of measured values) and allow interactive virtual material design, which is impossible or very impractical when using the direct measurement representation. In short, a BSDF model that reasonably represents a measured dataset comes in useful in many ways for the problem at hand.

Models are well suited to represent the subclasses of materials they were designed for. The following effects should be supported by the model employed for the problem at hand:

- **Separability into a diffuse and a specular term:** Almost all surfaces in architectural indoor scenes reflect light in a partly diffuse fashion (walls, ceilings, some fabrics and textiles, unpolished wood and stone, plants but also plastics or ceramics) and can be partly represented by Lambertian reflectance [Lambert 1760]. As diffuse illumination is view-independent, those large parts can be computed in advance to drastically reduce the run-time workload. For excluding diffuse reflection from the real-time integration it is essential that the BSDF is separable into diffuse and specular reflection terms.
- **Gloss and specularity:** Different degrees of surface roughness result in different levels of specularity. Everything from nearly perfect mirror reflections to rough, glossy reflections should be supported by the model, so that most metals, plastics and ceramics can be represented.
- **Anisotropic reflection:** Due to directional irregularities in the surface roughness the scattering properties of anisotropic materials vary for different incoming ϕ_i - angles. One of the most common anisotropic material classes in modern architectural indoor scenes are brushed metals.
- **Fresnel effect:** For some reflective materials the perceived power of the specular reflection depends on the viewing angle θ_o . Capturing this effect is essential for materials like glass, varnished objects and plastics.

A state-of-the-art model fulfilling all of these criteria was proposed by Kurt, Szirmay-Kalos and Krivanek [Kurt et al. 2010]. Their model only describes reflection behaviour, thus it does not comprise a BSDF, but only the BRDF. Their BRDF allows specifying a diffuse part and a specular part that supports multiple specular lobes:

$$f_r(\omega_i, \omega_o) = \frac{k_d}{\pi} + \sum_{n=1}^{\#lobes} k_{s_n} \cdot f_{r_{lobe_n}}(\omega_i, \omega_o) \quad (3.2)$$

where k_d is the diffuse coefficient describing the percentage of incoming light that leaves the surface by diffuse reflection, k_{s_n} is the specular reflectivity coefficient of each independent lobe and $f_{r_{lobe_n}}$ is the part of the BRDF attributed to the n 'th specular lobe defined in Equation 3.4. The specular reflection model is based on a normalized microfacet distribution [Blinn 1977] and is particularly well suited for fitting to measured data, since it has more parameters and allows more degrees of freedom than most other real-time relevant models. The microfacet distribution $D(h)$ defined by the model is an anisotropic normalized version of the *Beckmann distribution* also used in the Cook-Torrance model [Cook and Torrance 1981]:

$$D(h) = \frac{e^{-\tan^2(\theta_h) \left(\frac{\cos^2(\phi_h)}{m_x^2} + \frac{\sin^2(\phi_h)}{m_y^2} \right)}}{\pi m_x m_y \cos^4(\theta_h)} \quad (3.3)$$

where h is the half-way vector between ω_i and ω_o . Microfacet models assume the surface to consist of tiny mirror-like facets that are tilted with respect to the average surface normal n . For a given half-way vector h , the distribution $D(h)$ defines the number of facets whose normal coincides with h , i.e. the amount of facets mirror-reflecting light into the viewing direction. Statistically, the facets are Beckmann-distributed (similar to Gaussian) around the mean value $\theta_h = 0$ with a standard deviation proportional to the roughness. m_x and m_y are the parameters defining surface roughness in tangent and bitangent direction, where 0 means perfectly mirror-like and larger values indicate a rough heightfield of microfacets, and thus, wider specular lobes. m_x and m_y define the overall shape of the lobe, when $m_x = m_y$ the reflection is isotropic, otherwise anisotropic. Using this distribution, the specular part of the BRDF is defined as:

$$f_{r_{lobe}}(\omega_i, \omega_o) = \frac{F(\omega_o \cdot h) D(h)}{4(\omega_o \cdot h)((\omega_i \cdot n)(\omega_o \cdot n))^\alpha} \quad (3.4)$$

where $F(\omega_o \cdot h)$ is the Schlick-approximation of the Fresnel term described in Equation 2.9, $D(h)$ is the microfacet distribution defined before and $\alpha \in [0, 1]$ is a new parameter introduced by Kurt et al. to allow better control over the shadowing and masking effects of microfacets explained in the following, allowing particular flexibility in representing very smooth and shiny materials. When light arrives from glancing angles, rough surfaces will mask parts of the incident light off and additionally they will shadow some of the reflected light, as explained in Figure 3.3.

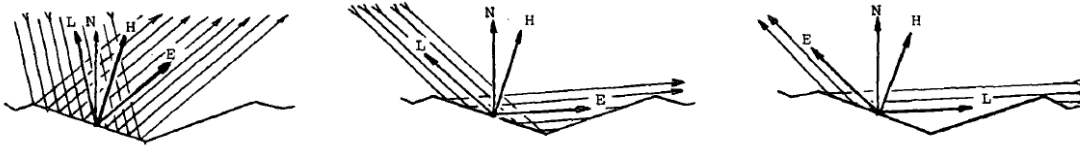


Figure 3.3: In the left image, all of the incident light hitting the microfacet is reflected without hindrance, which is most likely at normal incidence ($\theta_i = 0$). In the middle image, some of the reflected light is shadowed by a neighbouring facet, while in the right image parts of the incident light are masked by another facet. The latter two cases are most likely at glancing incidence ($\theta_i \approx 90^\circ$). Images taken from J. Blinn [Blinn 1977]

The parameter α in Equation 3.4 allows to control the influence of the shadowing/masking-term, which enables a better fitting and representation of very smooth specular surfaces with strongly increasing albedo towards grazing angles [Kurt et al. 2010]. For such surfaces, α will be large, resulting in a very small denominator in Equation 3.4, thus increasing the reflectance at glancing angles. Example renderings with different values of α are shown in Figure 3.4.

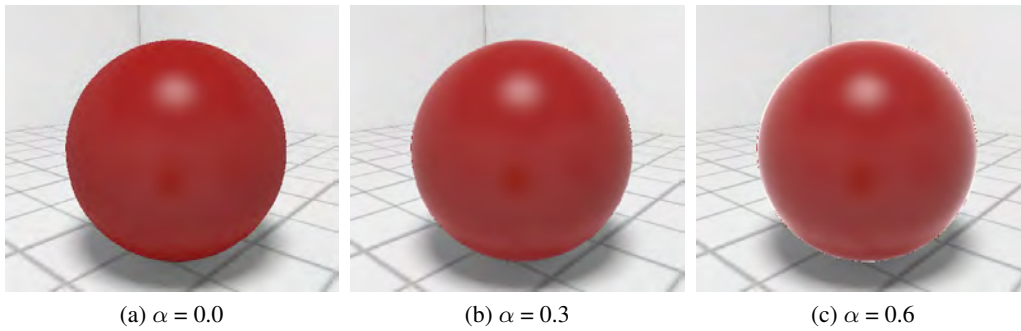


Figure 3.4: Renderings with different values of α . The reflectance gets increased especially at glancing incidence for higher values of α . This allows representing very smooth materials, such as surfaces covered by a thin layer of varnish or extremely polished surfaces.

The price of the flexibility introduced by α is that physical plausibility of the BRDF may be sacrificed, as the denominator in Equation 3.4 gets arbitrarily small for grazing angles when α is non-zero, thus the reflected energy might exceed the incoming energy. For $\alpha = 0$, the model is always physically plausible, for larger values energy conservation depends on the values of both α and the surface roughness parameters $m_{x,y}$ [Kurt et al. 2010]. Nevertheless, specular surfaces as the ones in Figure 3.4 (b) and (c) can be represented more accurately than without this parameter.

The Fresnel term $F(\omega_o \cdot h)$ describes the ratio of light reflection vs. light transmission according to the angle of incidence. In this material model, Schlick's approximation of Fresnel Reflectance [Schlick 1994] is used to calculate the attenuation according to the material reflectance r_o at normal incidence and the dot product of view direction and half-way vector (see

Section 2.2). Thus, varying material reflectance due to the Fresnel effect can be represented as well using this model.

In summary, when this BRDF model is used, the specular reflection of a material is defined by the parameters $m_{x,y}$ (surface roughness), r_0 (reflectance at normal incidence) and α (shadowing/masking bias). For a better approximation of measured data, more than one specular lobe can be used to represent the BRDF. Then, all of the parameters described above must be found individually for each lobe. Furthermore, each specular lobe is assigned a scalar coefficient $k_s \in [0, 1]$ that controls the overall specular reflectivity of the material due to this lobe. A diffuse coefficient $k_d \in [0, 1]$ controls the amount of diffuse reflection. Note that no explicit care is taken that the sum of coefficients k_d and k_s does not get too large to violate energy conservation.

3.3.1 Representing transmission distributions

Using the previously described model, reflection is covered to a satisfying extent for the problem at hand. Taking into account the second hemisphere of incidence that contributes to the material appearance by transmission is explicitly listed as future work by Kurt et al. [Kurt et al. 2010], but not covered by the model so far.

In this thesis the material model is extended to represent transmission through thin transparent media such as foil or glass panes. *Thin* in this context means that refraction, internal reflection and object thickness are ignored. The principal transmission direction ω_t is then equal to the negative incident direction (see Figure 3.5).

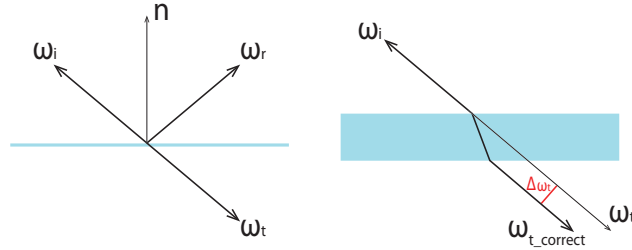


Figure 3.5: Transmission geometry of the BTDF model: An incident ray of light is reflected according to the law of reflection and transmitted according to the refractive indices of the involved media. Since refraction is ignored, the transmission ray is the continuation of the incident ray. This is correct for infinitely thin objects. The offset $\Delta\omega_t$ introduced by objects with non-negligible extent depicted in the right image is ignored. Also, the entrance plane of the optical medium is implicitly assumed to be parallel to the plane of exitance, which is true by definition for infinitely thin objects.

With the aforementioned restrictions, transmitted light essentially passes through transparent objects as if they were not there. However, microfacet theory as mentioned regarding reflection can be directly applied to transmission as well, allowing the specification of transmission lobes, and thus, representing non-perfectly transmissive media such as frosted glass or matt film. A transmission lobe is defined around the principal transmission direction in the same manner as

reflection lobes are defined around the principal reflection direction. Thus, it can be evaluated as if it was a reflection lobe with ω_t as outgoing direction and ω'_i as the incoming direction, where ω'_i is ω_i reflected by the surface plane. This symmetry is illustrated in Figure 3.6:

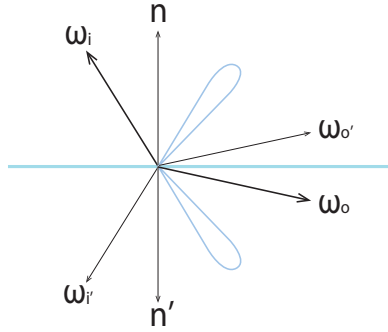


Figure 3.6: Evaluating the transmission lobe for (ω_i, ω_t) : for symmetry reasons the transmission lobe can be evaluated as if it was a reflection lobe on the other side of the surface; vectors with a prime (as in h') are the vectors denoted by the name without prime, mirrored by the surface plane, e.g. $\omega'_t = (\omega_t \text{ mirrored by the surface plane})$

Exploiting this symmetry, the BTDF (Bidirectional Transmission Distribution Function) of the transmission lobe is then defined in analogy to Equation 3.4 as:

$$f_{t_{lobe}} = \frac{(1 - F(\omega_t \cdot h'))D(h')}{4(\omega_t \cdot h')((\omega'_i \cdot n')(\omega'_t \cdot n'))^\alpha} = \frac{(1 - F(\omega'_t \cdot h))D(h)}{4(\omega'_t \cdot h)((\omega_i \cdot n)(\omega'_i \cdot n))^\alpha} \quad (3.5)$$

The Fresnel term of the transmission lobe is $1 - F$, since F is the ratio of reflected vs. transmitted light. Direction vectors with a prime (as in h') are the vectors denoted by the name without prime, reflected by the surface plane, e.g. $\omega'_t = (\omega_t \text{ reflected by the surface plane})$. Note that the only difference of Equation 3.5 over Equation 3.4 is the complementary $(1 - F)$ Fresnel term, as it quantifies the attenuation of transmission and not reflection.

By exploiting this symmetry, transparent rough surfaces such as frosted glass can be represented with relatively low additional implementation effort. This extension to the BRDF model is motivated by the fact that thin transparent objects such as glass panes are quite common in architectural scenes. Solid transmissive objects cannot be represented, since refraction, internal reflection and the extent of the object are not taken into account. Figure 3.7 shows some example renderings achieved with the proposed BTDF extension of the Kurt-Szirmay-Kalos-Krivanek BRDF:

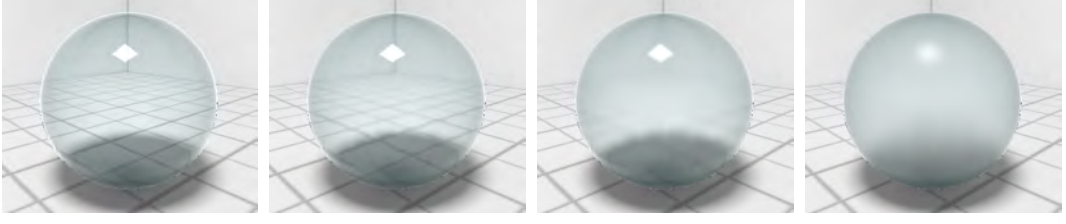


Figure 3.7: Renderings of transmissive materials (glass) with increasing degree of roughness

Finally, the entire BSDF used to represent materials in the work of this thesis is:

$$f_s(\omega_i, \omega_o) = \frac{k_d}{\pi} + \sum_{n=1}^{\#reflectionLobes} k_{s_n} \cdot f_{r_{lobe_n}}(\omega_i, \omega_o) + \sum_{m=1}^{\#transmissionLobes} k_{t_m} \cdot f_{t_{lobe_m}}(\omega_i, \omega_o) \quad (3.6)$$

where k_t is the transmission coefficient which analogously to k_s defines the transmissivity of the material.

3.4 Implementation, results and limitations

In this section the implementation of the model fitting procedure is explained and obtained fitting results are presented. Observed limitations of the model and the fitting procedure are pointed out and discussed.

3.4.1 Fitting the BSDF model

Fitting a model to measured scattering data involves finding parameter values that optimally represent the measurements with respect to a certain error measure. First, the employed error measure is described, then the method of exploring the space of possible parameter configurations is presented. Finally, some results obtained using the presented fitting scheme are shown and discussed.

As stated in Section 2.3, our test-datasets comprise *transfer probabilities* $p_{measured}(\omega_o|\omega_i)$ for several thousand measured direction pairs ω_i, ω_o . For a particular set of parameters, the modeled BSDF can be evaluated using Equation 3.6. Transfer probabilities p_{model} of the model can then be derived directly from the BSDF using Equation 2.4. Good parameter values result in a small difference between $p_{measured}$ and p_{model} for all possible pairs of incoming and outgoing directions.

There seems to be no general consensus in the literature on how to optimally quantify the difference between measured and modeled scattering distributions. Ngan et al. propose using an L^2 -metric to measure the difference between modeled and measured BRDF [Ngan et al. 2005], which was also adopted by others (e.g. [Kurt et al. 2010]). However, squaring the difference severely penalizes outliers, which is not desirable when the measurements contain a considerable level of noise. As a more robust alternative, an L^1 metric is employed in this thesis to measure

the non-squared absolute difference between $p_{measured}$ and p_{model} . The benefit of fitting transfer probabilities over fitting BSDFs is that due to the additional $|\cos(\theta_o)|$ -term the particularly noisy and unreliable measurements at grazing angles are suppressed. The employed L^1 -error metric is stated in Eq. 3.7. The summation in Eq. 3.7 is performed over M , the set of all non-zero scattering measurements. The values of ω_{i_t} , ω_{o_t} and $p_{measured}(\omega_{o_t}|\omega_{i_t})$ for each measurement tuple $t \in M$ are directly obtained from the measurement table.

$$error = \frac{\sum_{t \in M} w(\omega_{o_t}) \cdot |\cos(\theta_{i_t})| \cdot [p_{measured}(\omega_{o_t}|\omega_{i_t}) - p_{model}(\omega_{o_t}|\omega_{i_t})]}{\sum_{t \in M} w(\omega_{o_t})} \quad (3.7)$$

The weight w is the solid angle correction term $\sin(\theta_{o_t})$, as measurements at normal incidence cover a smaller solid angle and thus should contribute less to the metric, when the measurement resolution is uniform in θ_o, ϕ_o . The additional $|\cos(\theta_{i_t})|$ -weighting ensures that the comparison happens under the assumption of uniform incident radiance [Ngan et al. 2005]. Using this metric, a scalar error value is assigned to each possible parameter combination, and a global minimum can be pursued.

Finding an optimal model now corresponds to minimizing Equation 3.7 across the space of possible parameter configurations. There are many ways of exploring the high-dimensional parameter space (see literature on *non-linear optimization*, e.g. [Ruszczynski 2006]). As the focus of this thesis is on the rendering of the model rather than on the efficient identification of good parameters, a simple *brute-force* approach is employed to search for an adequate model. This decision is mainly motivated by the implementation simplicity of the method (essentially a nested loop). Exhaustive methods like brute-force search suffer from the *curse of dimensionality*, i.e. the number of candidate solutions increases exponentially in the number of involved dimensions, when densely covering the search space. Fitting a single lobe of the employed model requires finding five parameters, resulting in a 5-dimensional search space. When n values are tested for each parameter, the number of candidate solutions is n^5 , as each parameter combination theoretically represents a possible model. To tackle this combinatorial complexity, the search is performed in a hierarchical manner, as described in the following, and illustrated in Figure 3.8:

1. **Initialization:** the user defines the search space by devising a reasonable interval $[lower_p, upper_p]$ for each parameter p
2. **Classical brute-force step:** n equidistant values within $[lower_p, upper_p]$ are tried for each each parameter, resulting in n^P error metric evaluations, where P is the number of involved parameter dimensions. The parameter configuration yielding the minimum error within the n^P tried models is identified.
3. **Refinement:** the search space for the next iteration is reduced to the P -dimensional hyperrectangle of extent $\frac{upper_p - lower_p}{n-1}$ in each direction. Continue with step 2, exploring the reduced parameter space. Terminate when a predefined number of iterations has been performed.

Figure 3.8 illustrates this hierarchical exploration for a two-dimensional parameter space. The complexity of this hierarchical brute-force approach is exponential in P , as in each iteration the error metric is evaluated n^P times. However, due to the hierarchical refinement a very dense sampling around each iteration's minimum can be obtained, even for small n . Additional iterations yield only a linear increase of the run-time.

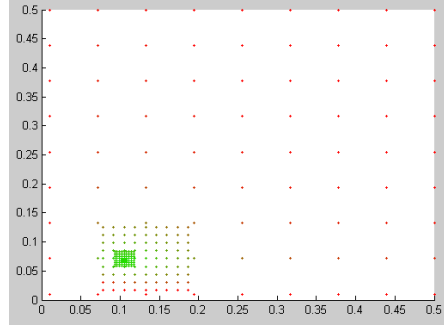


Figure 3.8: The 2-dimensional search space is iteratively explored by trying an equidistant 9×9 grid of parameter combinations. After identifying the combination yielding the lowest error, the search space is reduced to a smaller hyperrectangle centered about the previously found minimum, and again evaluated at 9×9 equidistant combinations. The image shows how the tested parameter configurations densify around the previously found minimum across four refinement iterations. The color of the dots depicts the obtained deviation from a measured dataset using the metric in Eq. 3.7, with red = high error, green = low error

Convergence of the method towards the global minimum of Eq. 3.7 is *not* guaranteed. It converges if (and only if) the global minimum is retained within the reduced search space during each refinement iteration. This is the case if the global minimum's zone of attraction is large enough to include a sampled point. In theory, this can be ensured by sufficiently increasing the sampling density n . For small n , however, convergence is only likely if the scalar error field does not feature high-frequency variations between sampled parameter configurations that are missed during the sampling. Intuitively, the scheme converges if the fitting error is not very sensitive to small changes of any parameter. Analyses of the high-dimensional error field for two exemplary material datasets show that for $n \approx 10$ this assumption seems to hold to a sufficient extent; Figure 3.9 shows a selection of 2D-minimum error projections of the 5-dimensional error field when fitting a single specular lobe to the white foil and scattergloss-aluminum datasets. In combination, the depicted 2D-projection plots suggest a 5D-scalar field that is smooth enough to be explored by hierarchical sampling.

It must be noted that even though the scalar error field appears benign, it is still possible that an entire region of lower error is skipped due to insufficient sampling, and that the approach gets stuck in a local minimum. A possible amendment to this problem could be randomly jittering the sampling positions, and running the fitting procedure multiple times, but an investigation of this is left for future work.

A more severe problem of this purely automatic approach is that the found parameter configuration resulting in minimal error does not necessarily yield visually satisfying results. This

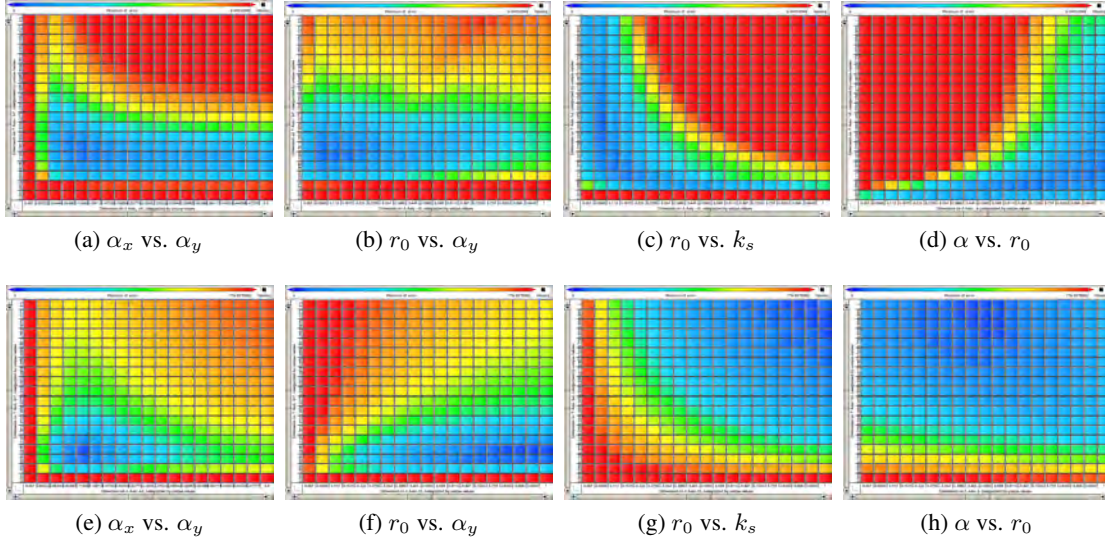


Figure 3.9: These images show 2D-minimum intensity projection slices through the 5D scalar error field when fitting a single specular lobe to the white foil dataset (top row) and to the scattergloss aluminum dataset (bottom row). The smooth variation of the error suggests a low-frequency scalar function that is safe to be hierarchically sampled. Colors range from blue (small error) via green and yellow to red (large error).

is particularly likely when the measured scattering phenomena are not entirely expressible by the fitted model. For instance, materials featuring multiple scales of roughness cannot be represented well with a single specular lobe. In this case, the procedure will minimize the most salient specular peak, ignoring the remaining phenomena that contribute less to the difference metric. To counteract this, the method allows manual specification of the initial search space, as stated in step 1 of the hierarchical fitting procedure. Thus, users understanding the model parameters can utilize this feature to steer the fitting in a desired direction. It was found that best results can be achieved by an interactive feedback-loop, where the algorithm is used to establish an initial solution, followed by manual parameter changes or additional automatic variations of single parameters while fixing all others. Note that fixing some model parameters and thus reducing the dimensionality P of the search space accelerates the fitting procedure by magnitudes, allowing the use of the brute-force method even in interactive fitting sessions.

One parameter that is always removed from the brute-force workload is the diffuse coefficient k_d . Since diffuse and specular reflection are different phenomena captured by different terms of the model, they can be dealt with independently. The diffuse coefficient k_d represents the fraction of irradiance that leaves the surface by diffuse reflection. Since this value is assumed to be the same for all incoming and outgoing directions (Lambertian reflection), $\frac{k_d}{\pi}$ is very likely to be a common value in the measured BRDF table, as the majority of measurements contains only diffuse reflection (see Section 2.3). Thus, building a histogram of measured BRDF values allows a statistical estimation of $\frac{k_d}{\pi}$, as it is very likely to be found in the bin containing the

largest number of values. Excluding the estimation of the diffuse coefficient from the brute-force approach allows keeping its run-time one order faster than including it.

When multiple specular lobes are fitted, a pure brute-force approach gets too expensive even for small sampling resolutions. Each lobe of the employed model is described by five parameters, resulting in a run-time of $O(n^{10})$ when fitting two lobes at once. Thus, the following modification to the brute-force search is proposed when fitting multiple lobes: first, a single lobe is fitted to the specular part of the measurements using brute-force optimization as described above. Then, this fitted lobe is subtracted from the specular measurement, leaving behind only the differences that are not captured by the first lobe. A second lobe is then fitted to these remainders. Note that for some directions the first lobe might overestimate the measurement, resulting in values below zero after subtraction. These values must be clamped to zero, otherwise the second lobe might not be fitted at all since all parameter configurations might result in a larger error than without a second lobe. Due to the clamping, the sum of both lobes is very likely to be larger than the measurement. Thus, the second lobe can now be subtracted from the original specular measurement, and the first lobe is fitted again, replacing the original first lobe. This decoupled fitting procedure is iterated several times. Note that the overall run-time of this modified fitting procedure is still $O(n^5)$, as no more than five parameters are estimated at a time, as opposed to $O(n^{10})$ when using a joint brute-force approach to search for all 10 parameters at once.

3.4.2 Fitting results and limitations

In the following, results of fitting the described model to three exemplary datasets using the aforementioned fitting procedure are presented and discussed: a rather glossy, slightly anisotropic aluminum (Alanod 2000 GS), a partly transmissive white foil and a very anisotropic, more specular aluminum (Miro 5). Figure 3.10 shows exemplary renderings of these three datasets to give an impression of their appearance before detailed fitting results for each of the three materials are discussed.

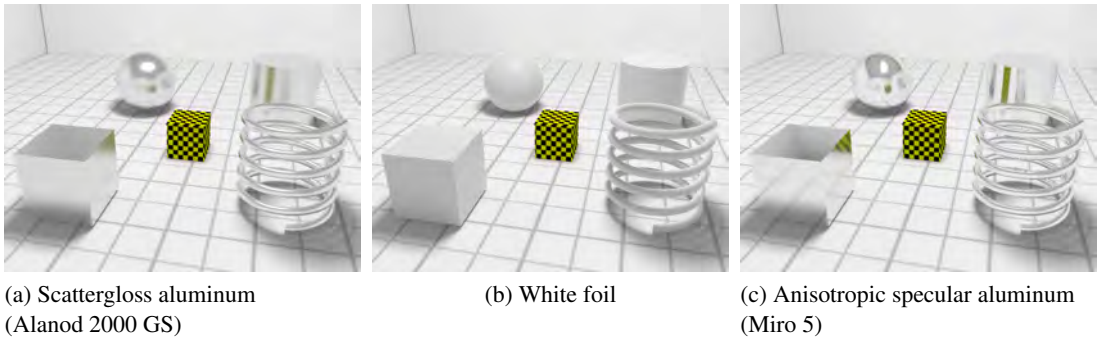


Figure 3.10: Exemplary renderings of the best found models for the three tested datasets.

Alanod 2000GS - scattergloss aluminum This dataset contains measurements of a slightly anisotropic aluminum sample featuring mainly glossy specular reflection. Two obvious scales

of roughness are evident from the measured scattering distribution: apart from the main specular peak, a very soft glossy lobe in the same direction causes a more diffuse-directional scattering, probably giving the material its name *scattergloss* (see Fig. 3.11a). Fitting a single reflection lobe to the measurement only captures the more specular of the two lobes, as omitting the softer lobe causes a less severe violation of the error metric than the other way around. Fitting two lobes however manages to capture both scales of roughness, as shown in Figures 3.11 and 3.12.

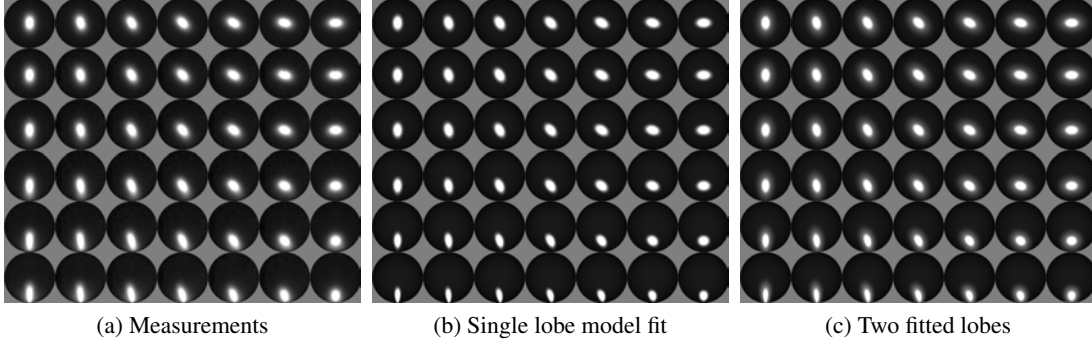


Figure 3.11: Transfer probability measurements of the scattergloss aluminum dataset next to fitted models using 1 and 2 specular reflection lobes. Note how both degrees of roughness are captured better when fitting two lobes than when fitting only one. How to read this visualization: within each plot, the 6 x 7 subplots show different incoming directions ($\theta_i = \{0^\circ, 5^\circ, 20^\circ, 35^\circ, 50^\circ, 65^\circ\}$ from top to bottom, $\phi_i = \{0^\circ, 15^\circ, 30^\circ, 45^\circ, 60^\circ, 75^\circ, 90^\circ\}$ from left to right). The light source is always positioned in the top of each subplot, and the brushing direction of the surface varies across the subplots from horizontal on the left to vertical on the right. For details on this type of stacked plot please refer to Section 2.3. Note that the cubic root was applied to the transfer probabilities for better visualisation.

The Alanod scattergloss data allows very good fitting, as all of its important scattering effects are expressible using the described model. Both scales of roughness can be captured well when fitting two specular lobes. But even one lobe suffices to capture the most essential part of the scattering distribution, which is desirable for fast rendering (as each additional lobe linearly increases rendering cost, see Section 4.6). Table 3.1 shows fitting errors and albedo-values obtained by the one- and two-lobe fitted models.

Number of lobes	k_d	k_s	α_x	α_y	r_0	α	L^1 -error	Albedo
Single lobe model	0.003	0.9	0.11	0.072	0.89	0.43	0.0179	0.787
Two lobes model	0.003	0.9 0.12	0.11 0.25	0.07 0.17	0.87 0.89	0.45 0.6	0.0142	0.878

Table 3.1: Fitting errors and albedo values obtained when fitting one or two lobes to the Alanod 2000GS scattergloss dataset

As evident from the table, the average deviation from the measured table decreases when

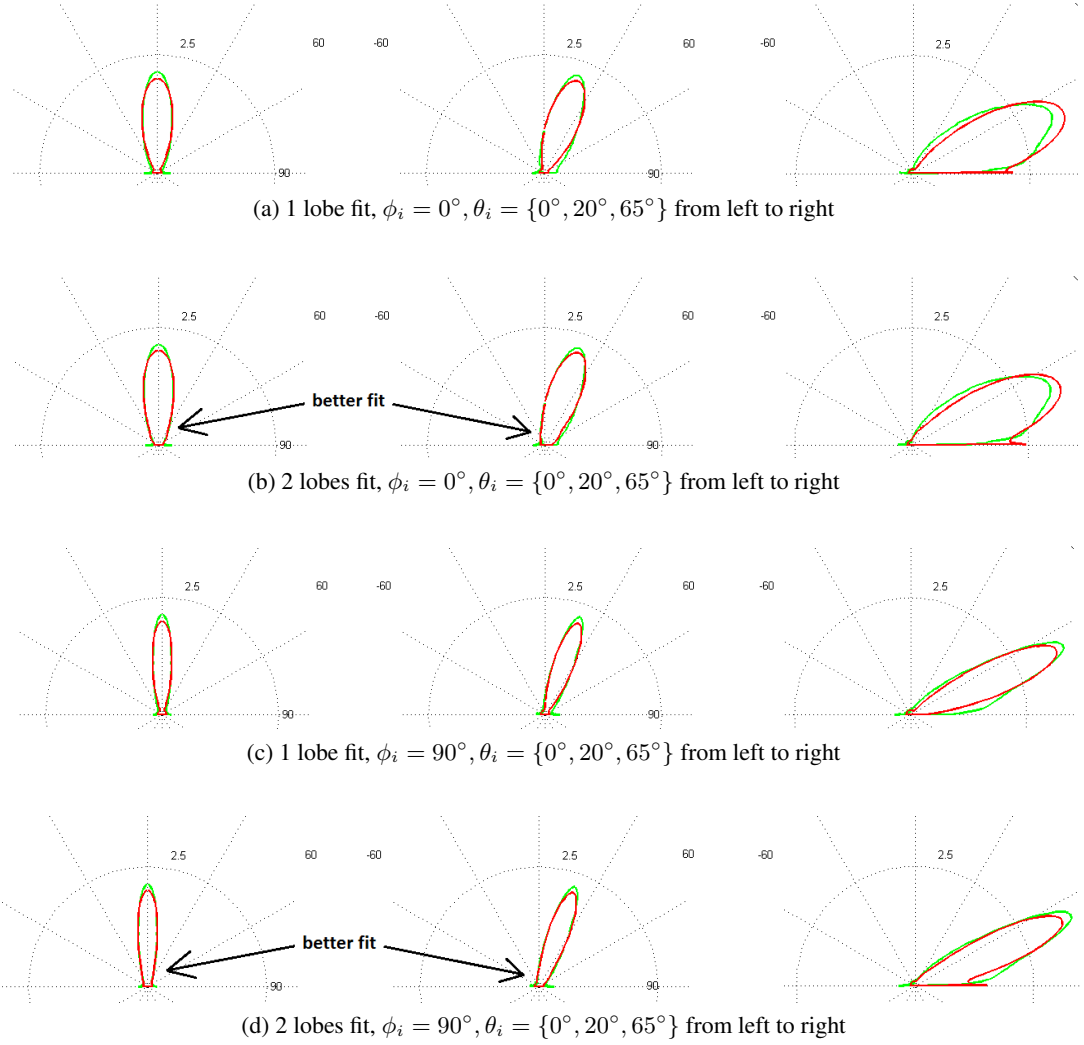


Figure 3.12: Transfer probability plots of varying θ_i in the plane of incidence, for $\phi_i = 0^\circ$ (a),(b) and for $\phi_i = 90^\circ$ (c),(d). Measurements are plotted in green, model fits in red. Plots of the single-lobe-fit are shown in (a) and (c), plots of the two-lobes-fit in (b) and (d). Note how both degrees of roughness are captured better when fitting two lobes than when fitting only one. Note that the cubic root was applied to the transfer probabilities for better visualisation.

fitting two lobes, as the second scale of roughness gets captured by the additional lobe. The estimated albedo of the measurements is 0.87 (obtained using Eq. 2.6), which is underestimated when fitting a single lobe, but approximated well when fitting two lobes. When one is limited to a single lobe for rendering performance reasons, the k_s -parameter can be increased to obtain a similar albedo, achieving the expected overall brightness of the material without changing the characteristic shape of the scattering distribution. Finally, Figure 3.13 shows renderings of the one-lobe and the two-lobe model, using the real-time rendering algorithm presented in Section

4.4.

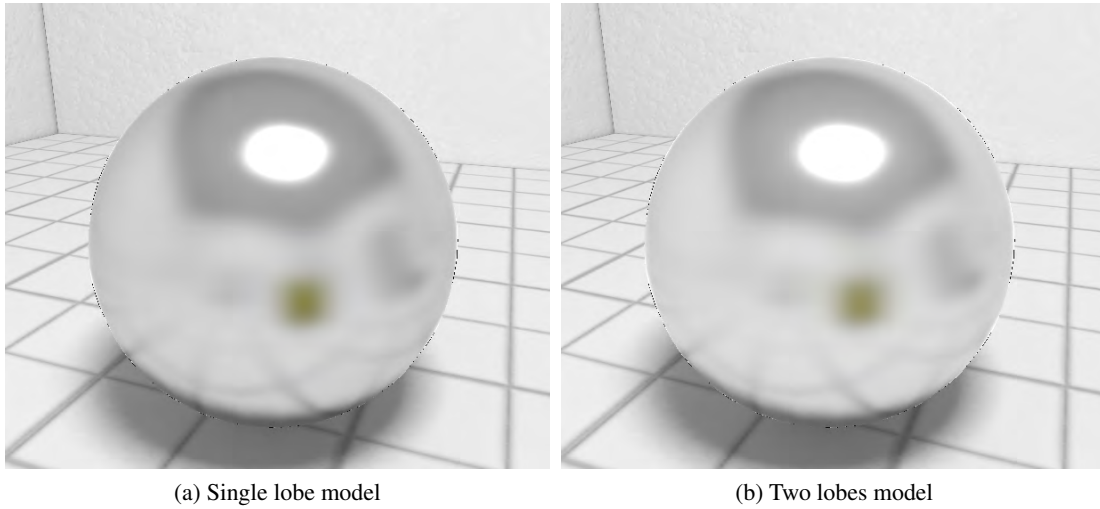
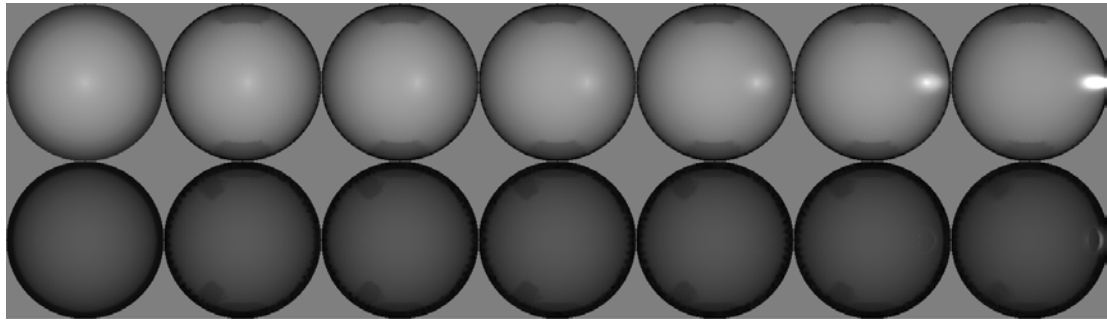


Figure 3.13: Renderings of the single lobe and two lobes model fitted to the Alanod 2000GS scattergloss dataset, using the rendering algorithm described in Section 4.4. Note that the two lobes model is slightly brighter, as its albedo is higher due to the additional lobe, and slightly blurrier, although differences other than the brightness are hardly visible as the second lobe’s k_s parameter is only 0.12.

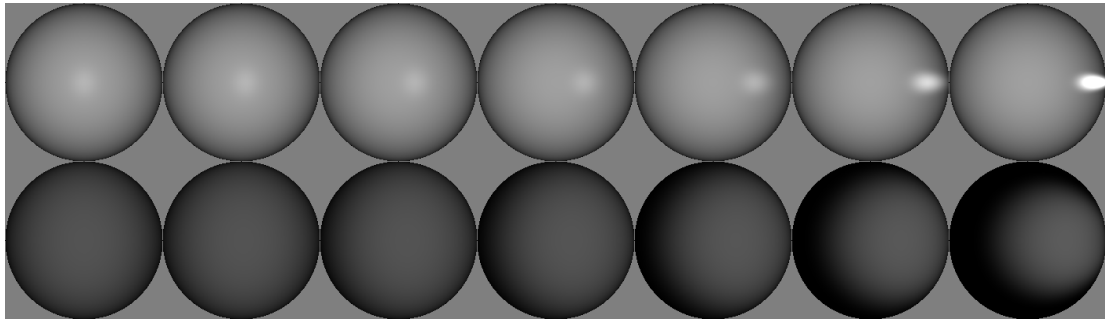
White foil This second test-dataset contains measurements of a partly transmissive, isotropic material that features mainly diffuse reflection and a strong Fresnel effect. The transmission is blurry, and white foil is thin enough to safely ignore object thickness and refraction (see Section 3.3.1). Thus it is expected to be an example where the proposed BTDF-extension of the microfacet-model works well. Figure 3.14 compares the transfer probability densities of measurement and model of the isotropic sample. One lobe is fitted to describe the directional reflection, and another lobe is fitted to represent the transmission distribution.

Similar to the Alanod scattergloss material, most featured effects can be expressed well using the model. A second reflection lobe would help to add the small specular bump that becomes evident for larger θ_i -angles in Figures 3.14 and 3.15. The measured transmission distribution seems to be without any directionality, thus a diffuse (constant) BTDF would capture this particular material better than the fitted directional lobe. Adding an additional diffuse term to the BTDF would be an easy future extension to express such materials even better. However, as the involved transmission is very blurry and rather dark, the mismatch does no real harm in this particular case. Table 3.2 shows the found model parameters, obtained fitting errors and reflectivity/transmissivity values of the model. For the measurement, the reflectivity is 0.77 and the transmissivity is 0.10, resulting in an overall 87 percent of incoming light to be transported.

The average deviation from the measurement is very low, and overall, the fit is very close. The fact that the transmission distribution is not perfectly matched does little harm, as its energy is very low and the light gets diffused so much that no difference could be seen in renderings



(a) Measurements



(b) Fitted model (1 lobe per hemisphere)

Figure 3.14: Transfer probability measurements (a) of the white foil dataset next to the fitted model using one lobe per hemisphere (b). As this material is isotropic, measurements only need to be taken for different θ_i -angles. The measured incoming directions are $\theta_i = \{0^\circ, 5^\circ, 19^\circ, 33^\circ, 47^\circ, 61^\circ, 75^\circ\}$ from left to right. The upper sub-row of each plot shows the reflection hemisphere, while the second (lower) sub-row depicts the transmission hemisphere which is only present in such transmissive datasets. The dark regions on the outer borders of each measurement sub-plot are measurement artifacts, as these directions could not be recorded due to technical limitations in the measurement process. Note how the intensity of the specular lobe increases towards grazing angles due to Fresnel effects, while the diffuse contribution slightly decreases (barely visible). The BRDF part is captured very well by the model, while the BTDF is too diffuse to be properly captured by a directional lobe. Note that the cubic root was applied to the transfer probabilities for better visualisation.

anyway if the distribution was slightly different. One reason why the reflectivity of the model is 3 percent higher than the measured reflectivity are the dark measurement artifacts evident in Figure 3.14, causing the measured material to be quantified darker than it actually is. Finally, Figure 3.16 shows renderings of the the model with (a) and without (b) the transmission lobe. The only obvious visible difference are the additional ten percent of transported light, resulting in a brighter image. As the transmission distribution is very diffuse, the scene from behind the material is blurred too much to be recognized. Thus, the slightly mismatched BTDF does no harm to the rendering.



Figure 3.15: Transfer probability plots of varying θ_i in the plane of incidence, $\theta_i = \{0^\circ, 5^\circ, 19^\circ, 33^\circ, 47^\circ, 61^\circ, 75^\circ\}$ from left to right. Measurements are plotted in green, model fits in red. Regarding reflection, the diffuse term and the specular lobe match the measurement very closely, except for the larger angles of incidence (right half), where an additional scale of roughness becomes visible, causing a sharper specular peak that is not captured by the one reflection lobe. The transmission term is too diffuse to be optimally captured by a directional lobe, especially at glancing incidence. Note that the cubic root was applied to the transfer probabilities for better visualisation.

Number of lobes	k_d	k_s	α_x	α_y	r_0	α	L^1 -error	Refl./Trans.
Reflection lobe	0.76	0.78	0.124	0.136	0.029	0.5	0.0059	0.807
Transmission lobe		0.67	0.25	0.17	0.75	0.6	0.0469	0.093
Both lobes together							0.0047	0.901

Table 3.2: Fitting errors and reflectivity/transmissivity values obtained when fitting a reflection lobe, a transmission lobe or both of them to the white foil dataset

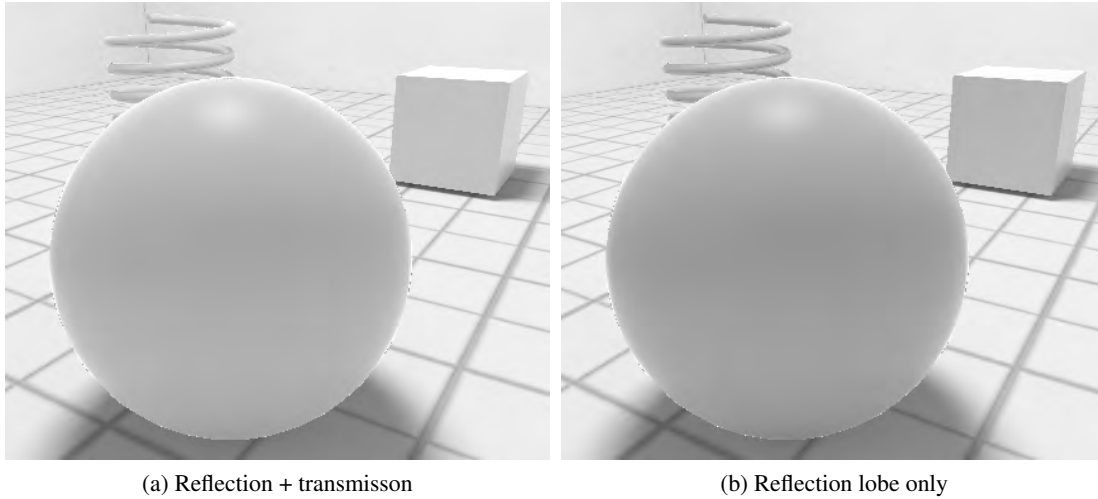


Figure 3.16: Renderings of the model fitted to the white foil dataset. In the left (a), the model is rendered using both the reflection and the transmission lobe. Note the difference in brightness to the image without transmission lobe (b), as roughly 10 percent of the incoming light are additionally transported. However, the transmission distribution is too blurry to allow recognizing any features of the scene behind the surface.

Miro 5 - aluminum This last of three tested datasets contains measurements of a very anisotropic brushed aluminum sample that features more than two scales of roughness. Thus, one or two lobes cannot possibly express all features of its scattering distribution. In the current implementation, only one or two lobes can be fitted and evaluated. Implementing an arbitrary number of lobes would be an easy future extension to allow better fits of complex materials like this dataset. Fitting a single lobe using the L^1 -error metric (Eq. 3.7) will result in approximating only the one distribution feature that influences the metric most, i.e. the maximum specular peak. Thus, found solutions using too few lobes are likely to be too sharp and specular, as glossier parts contributing less to the metric are ignored. This is not necessarily the optimal parameterization human users would choose, when the overall appearance of the material should be conveyed. Fitting the Miro 5 dataset is one such example, where one or two lobes cannot capture all features of the scattering distribution, as shown in Figures 3.17 and 3.18.

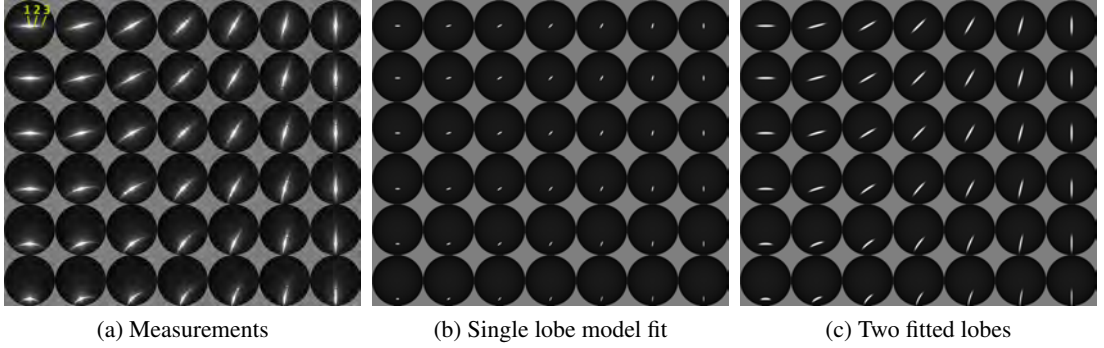


Figure 3.17: Transfer probability measurements of the Miro 5 brushed aluminum dataset next to fitted models using 1 and 2 specular lobes. This material features at least the three degrees of roughness shown in the left image: one rather sharp, less anisotropic specular lobe (1), one more anisotropic specular lobe (2), and one weak, glossy and extremely anisotropic lobe (3). Fitting one lobe only captures the most salient specular lobe in the middle, while an additional second lobe is able to capture the second degree. For the third, blurry anisotropic feature a third lobe would be required. How to read this visualization: within each plot, the 6 x 7 subplots show different incoming directions ($\theta_i = \{0^\circ, 5^\circ, 20^\circ, 35^\circ, 50^\circ, 65^\circ\}$ from top to bottom, $\phi_i = \{0^\circ, 15^\circ, 30^\circ, 45^\circ, 60^\circ, 75^\circ, 90^\circ\}$ from left to right). The light source is always positioned in the top of each subplot, and the brushing direction of the surface varies across the subplots from vertical on the left to horizontal on the right. For details on this type of stacked plot please refer to Section 2.3. Note that the cubic root was applied to the transfer probabilities for better visualisation.

As expected, the found models fail to resemble the overall scattering distribution of this dataset. The extremely anisotropic blurry lobe is totally ignored during the automatic fitting.

Table 3.3 states the automatic fitting errors for the Miro 5 dataset, and Figure 3.19 shows renderings of the identified one-lobe and the two-lobe model. As mentioned before, experienced users can steer the fitting in a desired direction if they deem the automatic fit inadequate. For

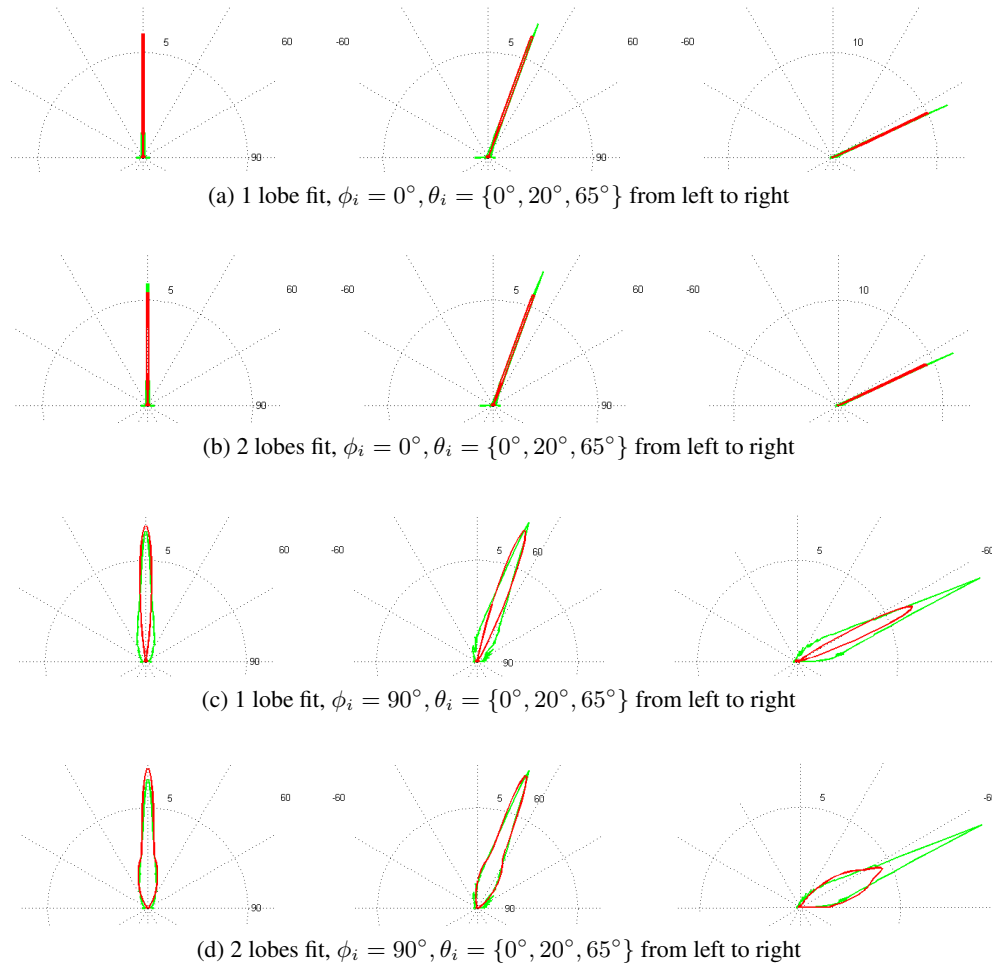


Figure 3.18: Transfer probability plots of varying θ_i in the plane of incidence, for $\phi_i = 0^\circ$ (a),(b) and for $\phi_i = 90^\circ$ (c),(d). Measurements are plotted in green, model fits in red. Plots of the single-lobe-fit are shown in (a) and (c), plots of the two-lobes-fit in (b) and (d). Note how two lobes are able to capture the measured distribution much better than one lobe, especially in the bottom rows (d) vs. (c). Still, fitting results of this dataset are worse than for the 2 other example datasets, as the entirety of features cannot be captured using one or two lobes. Note that the cubic root was applied to the transfer probabilities for better visualisation.

presentation purposes, users might prefer conveying the strong anisotropic blur of the Miro 5 aluminum over minimizing the numerical average error. Figure 3.19c shows the rendering of a completely user-defined two-lobe model that achieves this goal, resembling the real-world material better but resulting in larger L^1 -errors. Fitting three or more lobes might allow a similar appearance while keeping the error even lower than the currently best two-lobe fit, but this has not been implemented yet. Note that user-interaction with the fitting procedure requires experience with the used model as well as a clear conception of how the material should look

Number of lobes	k_d	k_s	α_x	α_y	r_0	α	L^1 -error	Albedo
Single lobe model	0.003	0.98	0.0068	0.0297	0.84	0.49	0.0654	0.815
Two lobes model	0.003	0.57	0.0046	0.0267	0.94	0.5	0.0499	0.944
		0.54	0.013	0.1199	0.84	0.38		

Table 3.3: Fitting errors and albedo values obtained when fitting one or two lobes to the Miro 5 brushed aluminum dataset

like. Implementing a direct renderer of measurement tables might allow inexperienced users to understand the material better, facilitating successful interaction.

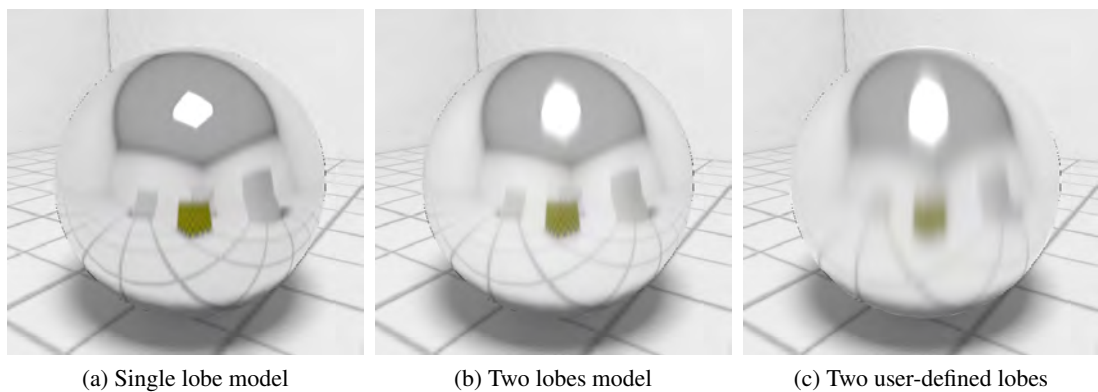


Figure 3.19: Renderings of 'Miro 5 brushed aluminum' models using one or two lobes. The left image is far from the expected results, as all the blurry features have not been captured by the single lobe fit. The middle image (automatic 2-lobe fit) gets closer to the expected anisotropy, but the directional blur is still underestimated. The right image shows a fully user-defined parametrization, which gets closer to the expected appearance, but results in a much higher error metric, as the most salient peak is omitted and replaced by the extremely anisotropic blurry feature.

The bottom-line illustrated by these fitting experiments is that the automatic fitting procedure using the L^1 -metric is able to find optimal models when the allowed degrees of freedom (essentially the number of lobes) suffice to capture all features of the scattering distribution. In such cases, the found models coincide well with what human users would perceptually accept as good fits. As soon as the fitting procedure has to decide which features to drop, it will always strive for the most salient peaks, as omitting them would violate the error measure most severely. This is why user-interaction is incorporated into the fitting process, in that users can manually change parameters and narrow the searched ranges of the parameter space in order to steer the results into the desired direction (e.g. Fig. 3.19c). A useful future extension might be incorporating perceptual aspects into the metric, as it would probably cause the automatic fitting results to match the user's expectations more closely. Providing the user with some form of direct rendering using the measurement tables might also greatly help understanding the measurements

and might even allow a perceptually based fitting procedure, considering differences of rendered images rather than transfer probability values. As the main focus of this thesis is on the fast rendering of the found models, rather than on the fitting, all of this is left for future work.

A limitation of the model rather than the fitting procedure is that it is not able to express the diffuse coefficient k_d as a function of the incident direction. Materials featuring a strong Fresnel effect such as the white foil dataset often decrease in diffuse reflectivity as the specular reflection increases towards grazing incident angles. This would be an easy future extension. Another limitation is that transmission distributions can only be specified by the means of specular lobe parameters that were originally designed for reflection distributions. A more extensive analysis using a larger database of transmissive materials needs to be done to investigate the adequacy of these parameters to express BTDFs.

In summary, we have seen that the fitting procedure described in this section achieves good results when the number of scattering distribution features does not greatly exceed the allowed degrees of freedom. The model itself is very expressive due to its many parameters and the ability to fit multiple additive lobes. In Chapter 4 a method is presented that allows the approximative integration of incident light according to the found models in real-time, achieving a convincing material appearance in interactive applications such as dynamic walkthroughs of static scenes.

Material Rendering

The familiar appearance of a material is caused by the particular way it interacts with incident light. This appearance can be simulated by determining how much of the incident light is scattered into the viewing direction according to the material's scattering properties. The scattering distribution of materials has been thoroughly discussed in the previous chapter, and a flexible representation has been achieved by the proposed BSDF model. Thus, the not yet discussed ingredients to a material rendering method comprise a *representation of the incident light* at every rendered scene point, and a procedure of *integrating the incident light according to the BSDF*. These two aspects will be discussed in this chapter.

As scenes are assumed to be totally static in this thesis, the distribution of incident light at each scene point does not change over time. The global propagation of light within the scene can thus be precomputed in an advance step, and representations of incident light can be derived from this precomputed view-independent illumination data. For rendering a particular view of the scene, the outgoing radiance into the viewing direction at each pixel can then be determined by evaluating the scattering equation (Eq. 2.2), however, the entire spherical integral can in general not be solved at interactive rates. In the real-time application of this thesis, absolute rendering correctness is less important than maintaining interactive frame rates, while keeping the material appearance as realistic as possible within the available run-time budget. This allows introducing approximations during walkthroughs such as considering only the most relevant subset of incident directions for integration, or employing hardware filtering instead of dense sampling for approximating the scattering integral. In summary, splitting the scattering equation into independent parts, conquering the static load in advance and *approximating* the remaining dynamic part at run-time is the strategy of how the rendering algorithm presented in this thesis achieves a convincing material appearance in real-time.

This chapter is structured as follows: first, related work in material rendering is reviewed and its applicability for the problem at hand is investigated (Section 4.1). Then, an overview of the proposed rendering algorithm is given in Section 4.2, before each of the algorithm's steps is elaborated separately in Sections 4.3 - 4.5. Finally, implementation details, limitations and results of the proposed method are presented in Section 4.6.

4.1 Related work

In image synthesis there is a general trade-off between rendering quality and rendering time. Global illumination solutions such as path tracing [Kajiya 1986], radiosity [Goral et al. 1984], [Cohen et al. 1993] or photon mapping [Jensen 1996] produce realistic images, but may take minutes to hours to complete a single frame when rendering complex scenes. However, all but the last bounce of the involved light propagation are view-independent, allowing to reuse intermediate results when re-rendering the exact same scene from a different point of view.

Precomputation-based methods (see [Ramamoorthi 2009] for a survey) rely on this principle, as they completely separate the view-independent light transport calculations from the evaluation of how much light arrives at the camera in the last bounce. Thus, arbitrarily complex global illumination methods can be employed to precompute the overall light transport in static scenes in advance. Then, methods like the one presented in this thesis can be employed to quickly render the scene from arbitrary views, as only the last bounce of light propagation has to be evaluated at run-time.

One of the most widespread precomputation-based methods is *light mapping*: a global illumination method is employed to obtain and store the (undirected) irradiance at every surface point in the static scene, e.g. as a light map texture. At run-time, the irradiance of a surface point can be read by a simple (interpolated) look-up. Diffuse reflection can then be calculated very quickly from the irradiance by multiplying it with the constant diffuse BRDF of the surface (see Section 4.3). As the entire light propagation is simulated in advance, complex diffuse effects can be displayed in real-time. Quake II (1997) was the first commercial interactive application to use light mapping for incorporating high-quality diffuse lighting effects into a game [Akenine-Möller et al. 2008], nowadays many 3D games use it. In this thesis, precomputed light maps are employed to render all local diffuse reflection, allowing to spend most of the available computation time on view-dependent effects.

View-dependent phenomena such as specular reflection or transmission cannot be simulated using a single accumulated irradiance value at each surface point, as the contribution of particular incident directions varies as the view changes. Thus, the incident light has to be integrated according to the current view and the scattering properties of the material in each frame during walkthroughs. In static scenes, at least the distribution of incident light does not change and its representation can be reused over time.

Environment mapping (or *reflection mapping*) [Blinn and Newell 1976] [Greene 1986] refers to the technique of capturing the entire environment, i.e. the incident light from all directions as seen from a single projection center (point in 3D-space) and storing it in a texture, enabling fast run-time retrieval of the incident radiance from arbitrary directions for rendering. For a given viewing direction, the relevant directions ω_i of incoming light can be derived (e.g. using the law of reflection) and their corresponding incident radiance can be looked up directly in the environment map.

The environment map representation of directional incident radiances is correct at exactly one point P in space (its projection center). Storing environment maps at several points in space requires a lot of memory, especially when taken at high resolution. In real-time rendering, a single environment map is often reused for several points P' in space (e.g. an entire object). This

approximation of the incident radiance yields an error that becomes particularly visible when the reflected environment is close, as nearby objects feature a larger parallax than distant ones. The shorter the distance of P' to P , the smaller the potential difference in the incident lighting can be, making this approximation especially suitable for objects of low extent with respect to the distance of the environment (see Section 4.4.1 or [Akenine-Möller et al. 2008]). Most real-time applications like games do not require a perfectly correct rendering result anyway, and they largely benefit from the view-dependent reflection/transmission effects that can be achieved with this technique. One straightforward application of environment maps is rendering perfectly specular (i.e. mirror-like) reflection or transmission, where the outgoing radiance in the viewing direction essentially depends on the incoming light from one single incident direction (see Section 2.2.3). With the surface normal at hand, this relevant incident direction ω_i can be derived from the viewing direction ω_o using the law of reflection ($\theta_i = \theta_o, \phi_i = \phi_o + 180^\circ$). Then, mirror-like surfaces can be shaded with a single radiance query to the environment map.

Glossy reflection occurs when incident light from several directions (e.g. within a lobe) contributes to the outgoing radiance. This effect can be achieved by sampling the environment map in these relevant directions according to the lobe of the material and combining the sampled results, obtaining the familiar blur in glossy reflections. This can be thought of as *filtering* the environment map according to the material, or *convolving the environment map with the BRDF* [Miller and Hoffman 1984]. When the material properties and incident lighting are static, this filtering of the environment map can be done in a preprocessing step for certain classes of BRDFs: Heidrich and Seidel proposed prefiltering a conventional specular environment map with a kernel according to the Phong-BRDF [Heidrich and Seidel 1999], Kautz and McCool extended the prefiltering concept to arbitrary isotropic lobes [Kautz and McCool 2000]. At each pixel, the obtained prefiltered environment maps contain the integration result of the incident radiances multiplied by the BRDF within the filter kernel. Thus, they allow approximating glossy reflection at real-time frame rates, as rendering still requires only a single look-up in the filtered environment map. However, the prefiltering is done with a single kernel, corresponding to the same single specular lobe for all directions. Thus, the lobe is assumed to be of exactly the same shape for every viewing direction and geometry, as they are not considered during the prefiltering. As a result, lobes do not get thinner and sharper near the horizon, and close to horizons light from below is taken into account as well, as the filter does not know anything about viewing angle or geometry.

To overcome this problem, Kautz and McCool suggest representing the BRDF by a superposition of multiple lobes, for each of which the environment map is prefiltered independently [Kautz and McCool 2000]. Then, a particularly shaped filter kernel for a given view and geometry is achieved by summing up several prefiltered lobes, yielding an approximation of the filtering result with the actual BRDF lobe. Generally, a collection of environment maps can be stored to account for varying lobe parameters such as different viewing angles or tangent frames, and in-between environment maps can be reconstructed, allowing to extend the prefiltering approach to more general material phenomena such as anisotropy or view-dependent lobe shapes [Kautz et al. 2000], [Kautz and McCool 2000]. However, when storing a large number of environment maps for rendering a single object, one's graphics memory might reach its limits. In applications like games, where the overall impression of a material should be con-

veyed but absolute correctness is not a requirement, one might accept the artifacts of employing a single prefiltered environment map, saving memory and allowing faster updates of the incident radiance representation in dynamic environments.

Generally, the shape of the BRDF varies for different viewing angles and tangent frames. An alternative to prefiltering the incident lighting is performing the integration of $light \cdot BRDF$ at run-time, on-the-fly. This has the advantage of not requiring substantial extra storage, but when real-time frame rates are desired, the filtering has to be performed very fast. In 2007, Colbert and Krivanek proposed *BRDF-proportional importance sampling with environment map filtering* [Colbert et al. 2007], where the incident lighting is sampled more densely in relevant directions with high BRDF, and sparsely in the less contributing regions. Samples are taken in a *MIP-mapped* environment map [Williams 1983], an image pyramid constructed from the original environment map, where the pyramid levels correspond to increasing numbers of subsequent average downsampling iterations. When taking a sample in a higher MIP-map level, the average (i.e. the approximative integral) over several incident directions is considered at once. In Colbert and Krivanek's method the filter kernel size (i.e. the MIP-map level) is inversely proportional to the sampling density obtained by the importance sampling, such that sparsely sampled regions are filtered stronger than densely sampled parts, obtaining an even filtering across the domain of directions. Using their method, arbitrary BRDFs can be rendered at interactive frame rates as long as they allow the derivation of importance sampling formulae, which are required by the proposed sampling scheme. This makes this method very flexible and so far the most appropriate foundation to build this thesis on. However, since the sampling is performed at each pixel at run-time according to the current view, geometry and material, only a small number of samples can be taken when targeting real-time frame rates. The authors report achieving real-time rates when rendering a single object using 40 samples at a resolution of 512x512 pixels, but in the application of this thesis the entire scene may be made from complex materials and is required to be rendered at very high resolution, maybe even for projection setups using multiple projectors. Therefore, their sampling scheme is modified in this thesis to achieve comparable results with lower numbers of samples, or in other words, to achieve images of the same quality at real-time frame rates even for high resolutions.

Representing the distribution of incident light as an environment map is particularly suited for surfaces of high curvature, since for near-planar surfaces, the surface normals and thus the considered incident directions are very similar across neighbouring pixels. As a result, a small portion of the environment texture is being mapped to a rather large region in the final image, introducing visible texture magnification when the map resolution is not extremely high [Akenine-Möller et al. 2008]. Furthermore, planar surfaces are often very large in architectural scenes (e.g. walls, floors,...), and reusing environment maps across large objects tends to yield large errors in the incident radiance, especially when there are nearby objects that get reflected in the plane. Thus, there are several reasons for employing a different representation of incident lighting for large planar surfaces.

When a planar mirror is viewed from a certain direction ω_o , the law of reflection yields the relevant incident direction ω_i by $\theta_i = \theta_o, \phi_i = \phi_o + 180^\circ$. The incident radiance from that direction can be derived at once for the entire plane by rendering the scene into a buffer from a second point of view, obtained by mirroring the actual camera position and view about the

planar reflector [Diefenbach et al. 1997] (see Section 4.5.1). For shading pixels in the actual view, the incident radiances from the perfect mirror directions can then be read directly from the buffer.

Bastos et al. proposed precomputing several such *incident radiance maps* [Bastos et al. 1999], i.e. rendered 2D-images plus depth as seen from several positions *behind* the planar reflector. These images then contain the incident radiances for several possible views in front of the plane. Views in-between can then be reconstructed by reprojecting the nearest pre-computed incident radiance maps. After reconstructing the map for a particular view, it can be blurred according to the material, allowing approximative glossy effects. Even though this precomputation-based method achieves high-quality results at high frame rates for architectural walkthroughs, the storage requirements for the multiple radiance maps per reflector are immense and the run-time reconstruction of in-between maps is quite costly.

A simpler, more dynamic approach of rendering planar reflections is rendering only the single relevant incident radiance map from behind the plane before rendering each frame. [Diefenbach et al. 1997], [Kilgard 1999], [McReynolds et al. 2000]. As the mirror rendering is performed again in each frame for the current view, the obtained reflections feature correct motion parallax and local reflection, as well as no magnification artifacts if the resolution of the mirror rendering is the same as for the actual rendering, as opposed to static environment maps. Multi-pass rendering also allows simulating several recursive reflection bounces [Diefenbach et al. 1997], [Nielsen and Christensen 2002].

Planar reflection rendered on-the-fly dynamically yields the correct incident radiance for every point on the planar surface, however, only for the perfect mirror direction. Thus, sharp (mirror-like) specular reflections can be rendered very effectively, but incident light from other directions is not captured as such. However, glossy reflections can be approximated by filtering the incident radiances in the mirror rendering buffer before reading them for rendering [Fuhrmann et al. 2004], [Nielsen and Christensen 2002], [Bastos et al. 1999], as it makes sense to assume that neighbouring pixels in the mirror rendering buffer contain incident radiances from directions within the specular lobe (see Figure 4.27 in Section 4.5.1). A uniform filter kernel can only approximate the gloss effect very roughly, as the amount of blurring should vary with the distance between reflector and reflected object [Fuhrmann et al. 2004]. In the same fashion as previously mentioned when filtering environment maps, the shape of the specular lobe and thus of the filter kernel varies with the viewing angle and the local geometry (tangent frame), and this should also be considered during the filtering.

Conceptually, a correct filtering can be achieved by projecting the specular lobe into the mirror rendering buffer, obtaining a set of pixels that contain relevant incident radiances from directions within the lobe, and sampling these pixels. The previously described method of Colbert et al. [Colbert et al. 2007] allows generating angular samples (directions ω_i) within a lobe. In this thesis, a projection scheme is presented that directly maps angular samples to pixels in the mirror rendering buffer, allowing a different per-pixel sampling of the buffer according to the material properties, view and local geometry.

A final concept that should be mentioned in this section is representing the incident lighting and material properties using special basis functions that allow faster evaluation or more compact storage. When the incident light is stored as a 2D texture containing radiance values, it is

represented in the *pixel basis*, i.e. it can be interpreted as a linear combination of pixel basis functions (being non-zero only at a single texel P_{xy}) and coefficients (the radiance value at P_{xy}). Storage requirements are linear in the number of texels, and calculations such as convolution with the BRDF also operate on single texels. A widespread alternative representation is the *Spherical Harmonics basis* (SH basis), which is localized in the frequency domain and can be thought of as the spherical analogon to sines and cosines (details on the SH basis and its uses can be found in this document [Green 2003]). As the SH basis functions have a much larger support than the pixel basis, very few coefficients instead of millions of pixels suffice to retain the low-frequency information of spherical functions. When both the lighting and the BRDF are expressed in the SH basis, the expensive spherical convolution operation can be expressed as a dot product of vectors with relatively few coefficients [Sloan et al. 2002]. Thus, low-frequency environments comprising diffuse and slightly glossy materials can be rendered in real-time, as shown for example by Sloan et al. in their work on *Precomputed Radiance Transfer* [Sloan et al. 2002]. On the downside, representing information of higher frequency such as very specular materials requires the use of much larger numbers of coefficients (quadratic in the number of frequency bands used). Since specular materials like metals, glass, varnished wood or certain plastics are very common in architectural scenes, using Spherical Harmonics would strongly limit the flexibility of material representation in the application of this thesis.

The use of other basis functions such as *Wavelets* was proposed to overcome the low-frequency limitation of Spherical Harmonics [Ng et al. 2003] [Ng et al. 2004]. The accompanying proposed rendering approaches [Ng et al. 2003] [Ng et al. 2004] achieve high quality images but require substantial precomputation effort and storage, and do not achieve real-time frame rates.

For its simplicity, all rendering components are expressed in the pixel basis in this thesis, but the potential performance gain makes investigating appropriate basis functions a promising direction of future work.

4.2 Algorithm overview

As shown in the previous section, a diverse variety of material rendering algorithms has been presented in the past. The algorithm presented in this thesis is based on several of the mentioned concepts and methods, combining and also extending them where they alone are insufficient for the task at hand. In this section an overview of the presented material rendering approach is given, and the structure of the remaining chapter is outlined.

Rendering a scene point P involves determining how much of the incident light is scattered into the viewing direction ω_o according to P 's material properties. Even though scenes are static in this thesis, the final outgoing radiance L_o cannot simply be precomputed as a single value for every point P , as it also depends on the viewing direction ω_o , which changes dynamically during walkthroughs. Therefore the final radiance calculation has to be performed on-the-fly. Computing the outgoing radiance of a scene point P using the scattering equation requires two inputs:

1. A representation of the incident light at P , defined for all incident directions ω_i

2. A material representation that states how incident light is turned into outgoing radiance, defined for all incoming directions ω_i and arbitrary outgoing directions ω_o

As presented in the previous chapter, the material representation is achieved in this thesis by a parametric *Bidirectional Scattering Distribution Function* (BSDF) that accounts for diffuse reflection, several aspects of specular reflection, and, to a certain extent, transmission. It can be analytically evaluated for arbitrary combinations of incoming and outgoing directions. If a set of parameters has been established at each scene point, e.g. by measurement fitting, this second requirement of material rendering is met by the BSDF model.

Determining the overall exiting radiance of point P in the viewing direction ω_o can then be achieved by integrating (summing up) products of incident light and BSDF over all incident directions. Thus, the representation of the incident lighting has to be available at each scene point for the entire sphere of directions. There are two options to achieve this, and both are employed to a certain extent in this thesis: either the incident radiances are computed on-the-fly (computationally expensive), or, as the scene is static, they can be precomputed and stored at each scene point for run-time access (storage-intensive).

Precomputing maps of incident radiance at each scene point exploits the fact that the scene is static, but requires massive storage amounts for complex scenes. In this thesis the memory requirements are kept manageable by precomputing the omnidirectional incident light information *only once per object* at a representative point P (e.g. its mass center) and reusing it across all points P' of the object. This approximation works particularly well for arbitrarily shaped objects that are small with respect to the distance of the reflected environment (see Section 4.4). But even for larger objects the errors often go unnoticed when the surface curvature is high, as the reflected environment changes rapidly across neighbouring screen pixels, making spatial errors hard to recognize. Thus, precomputed incident radiance maps are employed in this thesis for *rendering specular reflection on curved surfaces*.

On-the-fly computation of the incident radiance requires no extra storage, but it does not exploit the fact that the scene is static. As it does not reuse any lighting information spatially across scene points, it can be employed for objects of arbitrary size. However, computing the incident radiance for several incoming directions per pixel in each frame is very slow, thus it can only be calculated for a small relevant subset of directions according to the current view. In this thesis, on-the-fly calculation of incident radiance maps is employed only for *rendering specular reflection on large planar surfaces* (walls, floors, etc.), as a single mirror rendering pass before computing the actual frame provides a relevant portion of incident lighting for all pixels of a planar object at once (see Section 4.5).

But even once the incident light information is available for a large number of directions at each scene point, the integration of products ($light \cdot BSDF$) is an expensive calculation in itself. To achieve the desired frame rate, two measures are taken in this thesis: first, the integration is split into a view-independent (diffuse) and a view-dependent (specular) part. As light is additive, summing up the results of both parts yields the overall outgoing radiance. Since the diffuse part does not change when walking through the scene, it can be entirely precomputed for each scene point, reducing the run-time effort to computing the view-dependent specular part. And second, since specular reflection features a certain directionality, e.g. occurring mainly within a *lobe* around the mirror reflection direction, the relevant integration domain is de facto much

smaller than the entire sphere of directions. In this thesis specular reflection is approximated by considering only this most relevant part of the incident light, achieving a biased but convincing material appearance at real-time frame rates.

In summary, the strategy of the proposed algorithm is dividing the light transport into a static (view-independent, diffuse) part and a dynamic (view-dependent, specular) part, conquering the static part in a high-quality precomputation step, and approximating the dynamic, view-dependent part at run-time. Different representations and methodologies are employed for rendering specular reflection on curved surfaces and on large planar surfaces. Figure 4.1 sketches the outline of the algorithm.

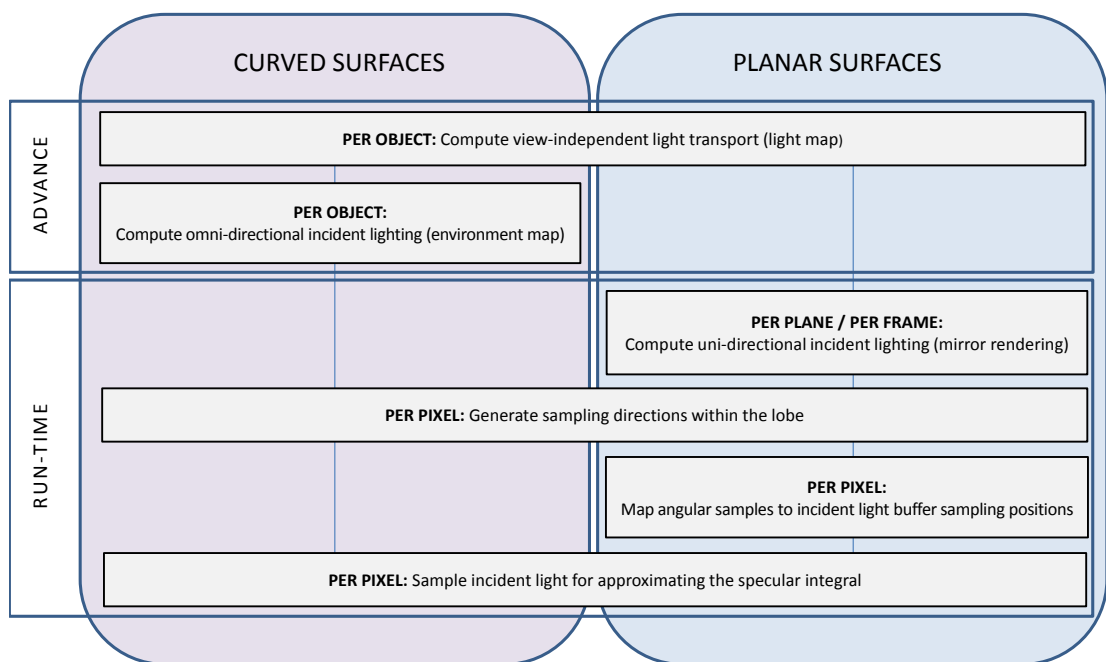


Figure 4.1: Outline of the proposed algorithm: a successive listing of precomputation and run-time steps to achieve material rendering of curved and planar surfaces. Diffuse (view-independent) reflection is rendered using light maps, specular reflection is approximated by sampling the incident light representation according to the material properties

In the following, the major parts of the algorithm are elaborated in separate sections: first, the comparatively easy task of rendering precomputed diffuse reflection is described (Section 4.3). Then, the approach for rendering view-dependent specular reflection and transmission is presented. Two different ways of representing incident light for this algorithm are discussed: precomputing *cubic environment maps* [Greene 1986] for rendering specular reflection on curved

surfaces 4.4, and on-the-fly *planar reflection rendering* [Diefenbach et al. 1997] for rendering large planar reflectors. (Section 4.5). Finally, results of the algorithm are presented and limitations are discussed in Section 4.6.

4.3 Rendering diffuse reflection

When light encounters a surface that is optically rough or features local subsurface-scattering, it is diffused into several outgoing directions (see Section 2.2.3). In this thesis, the term *diffuse reflection* refers to purely *Lambertian reflection*, while every directional, view-dependent reflection is categorized as *specular reflection*. This means that the overall perceived luminance of such a diffuse surface is constant for all viewing directions, as the cosine terms of transfer probability and perceived area cancel each other out (see Section 2.2.3). As a consequence, the appearance of a purely Lambertian surface only depends on the diffuse reflectivity k_d and the (view-independent, directionless) irradiance E of the surface. Since the irradiance does not change over time in static scenes, it can be precomputed and stored as a single value per point with the scene geometry; in this thesis it is stored as a *light map*. During walkthroughs, diffuse reflection can then be calculated from the precomputed irradiance very fast; if the diffuse reflectivity k_d of the material is also fixed, the last computation of obtaining outgoing radiance from irradiance can be performed in advance as well. When k_d varies with the view (e.g. due to Fresnel effects), it makes sense to store only irradiance values and to derive the outgoing radiance from them at run-time.

In practice, hardly any surface is purely Lambertian. Still, most materials are at least partly diffuse, and as specular and diffuse reflection can be dealt with independently, these diffuse parts can be eliminated from the run-time calculation of materials with mixed reflectance behaviour as well. When a Lambertian surface is shaded, the outgoing radiance can be calculated from the (non-directional) surface irradiance E as follows:

$$L_o(\omega_o) = \frac{k_d}{\pi} \cdot E \quad (4.1)$$

where k_d is the *diffuse reflectivity* (aka *directional-hemispherical reflectance* or *diffuse coefficient*), a percentage that states how much of the irradiance is reflected diffusely by the surface. The division by π accounts for the fact that the outgoing energy of $k_d \cdot E$ is uniformly distributed over a hemisphere of directions whose surface area is π . Equation 4.1 follows directly from the

classical scattering equation 2.2:

$$\begin{aligned}
 L_o(\omega_o) &= \int_{\Omega_{HS}} f(\omega_o|\omega_i) L_i(\omega_i) \cos(\theta_i) d\omega_i = \\
 &= \int_{\Omega_{HS}} f(\omega_o|\omega_i) E_i(\omega_i) d\omega_i = \\
 &= \int_{\Omega_{HS}} \frac{k_d}{\pi} E_i(\omega_i) d\omega_i = \\
 &= \frac{k_d}{\pi} \int_{\Omega_{HS}} E_i(\omega_i) d\omega_i = \\
 &= \frac{k_d}{\pi} \cdot E
 \end{aligned} \tag{4.2}$$

where $f(\omega_o|\omega_i)$ denotes the Lambert BRDF which is $\frac{k_d}{\pi}$ and $E_i = L_i \cdot \cos(\omega_i)$ is the irradiance produced by the incident radiance from direction ω_i . The hemispherical integral of $E_i \cdot d\omega_i$ results in E .

Given the fact that the irradiance of a scene point never changes in a static scene, a fully fledged global illumination algorithm can be employed to obtain physically correct irradiance values, which, in this thesis, are stored in a three-channel RGB light map. Then, the scene can be walked through interactively with only Equation 4.1 to evaluate at every pixel, reading the irradiance with a single interpolated look-up from the light map. Figure 4.2 shows some examples rendered at very high frame rates using this procedure:



Figure 4.2: Diffuse illumination rendered using light maps

4.4 Rendering specular reflection on curved surfaces

Contrary to diffuse reflection, specular reflection and transmission depend on the viewing angle and cannot be precomputed in the same fashion. A mirror will always reflect different things when looked at from different directions. Thus, the integration of incident light has to be calculated on-the-fly, according to the current viewing direction.

A standard way of performing this integration in offline rendering is evaluating the scattering equation using Monte-Carlo quadrature, sampling the illumination according to the BRDF thousands of times at each pixel. This method converges towards the correct integration result as more and more samples are taken, but it is not applicable to real-time rendering, as large numbers of samples are required to obtain smooth results, which is not feasible when the camera moves. Therefore, in this thesis the specular integral is approximated during walkthroughs by sampling the illumination using a *deterministic regular sampling pattern* and hardware-filtering, similar to the method of Colbert et al. [Colbert et al. 2007]. As opposed to their quasi-random method, the employed sampling pattern is perfectly regular, such that only 5-20 filtered samples suffice to obtain a smooth material appearance that is similar to the Monte-Carlo result, making the approach very well suited for real-time walkthroughs. Whenever the camera is still, additional random samples are taken and blended with the previous rendering result, ensuring convergence to an unbiased integration result over time.

This section starts off with a discussion on the representation of incident light employed for sampling the illumination integral (Subsection 4.4.1). Then, as the proposed algorithm approximates the scattering equation and fully evaluates it when the camera is still, the Monte-Carlo- and Quasi-Monte-Carlo-approaches for correct integration are recapitulated (4.4.2), as they are required to understand the following approximations. Finally, the random-based sampling schemes are modified to achieve similar results by deterministically approximating the integrand using only 5-20 samples per pixel during real-time walkthroughs (4.4.3).

Please note that directional transmission is computed in the same way as specular reflection in this thesis, defined as a lobe around the principal transmission direction instead of the mirror reflection direction (Section 3.3.1). For the sake of simplicity, all references to the term *specular reflection* in this section also directly apply to directional transmission.

4.4.1 Representing incident light as a cubic environment map

For realistic rendering of materials it does not suffice in general to consider a few discrete point light sources, as light from all incident directions might contribute significantly to the appearance. A mirror, for instance, does not only reflect light sources but the entire environment. For integrating over the entire sphere of directions, the optimal representation is a quantification of incident light for every possible incoming direction. This is provided by cubic environment maps (CEMs) [Greene 1986], as they capture the incident radiance for each direction as seen from a projection center. As scenes are assumed to be static, CEMs can be computed in advance and stored with the scene geometry for run-time access.

In general, the incident lighting varies at each scene point. A single CEM captures the incident radiance as seen from one particular scene point. Storing one cubic environment map (512x512 pixel per face, RGB, 8bit per channel) requires about 4.5 MB, which is too much

for storing CEMs densely across the scene, since textures, light maps and geometry also need to fit into the graphics memory. In the following, an approach is presented that approximates the incident lighting using only a single CEM per object, which is storage-wise feasible for moderately complex scenes with dozens to hundreds of objects.

The main goal of this thesis is to achieve fast, convincing rendering of materials; a material should be clearly recognizable from its appearance. For this purpose it is crucial that the material characteristics are represented accurately, but it is less important that the reflected geometry is spatially correct, as illustrated in Figure 4.3. Thus, approximations in the incident light representation are more acceptable than material approximations.

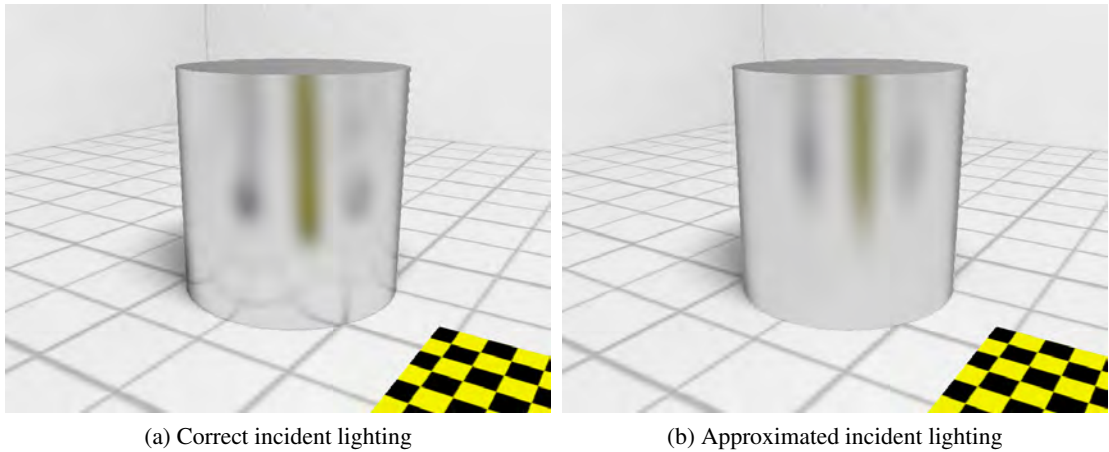


Figure 4.3: Material plausibility does not suffer too much when the representation of incident lighting is approximated

That being said, a single CEM is used in our approach to represent the incident light at all points of an entire object. It is obtained in advance by rendering the light-mapped scene as seen from the object's bounding box center. When a reflective object is rendered at run-time, secondary rays are generated from viewing rays using the law of reflection ($\theta_i = \theta_o, \phi_i = \phi_o + 180^\circ$). The standard way of looking up the incident radiance of a secondary ray is to look up the ray direction as if the secondary ray originated from the CEM projection center, thus neglecting the distance $\Delta\omega_i$ between projection center and actual reflective surface point (see Figure 4.4). In other words, the reflected geometry is assumed to be infinitely far away from the projection center.

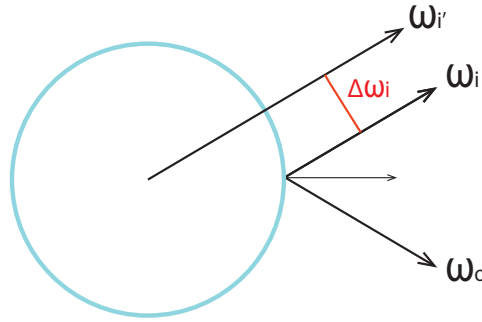


Figure 4.4: Standard cubic environment map look-up: A secondary ray ω_i is looked up as if it originated from the projection center ($\omega_{i'}$). Thereby, the extent of the object is ignored, introducing a spatial error Δ_{ω_i} into the reflected image that aggravates with increasing distance between reflective surface point and projection center

Instead of assuming the reflected geometry to be distant, a better approximation of the incident lighting could be achieved if the actual intersection point of a secondary ray with the scene geometry was known. However, intersecting secondary rays with the scene geometry as done in ray-tracing takes too long for complex scenes. In this thesis, the intersected geometry is approximated by exploiting an advantageous property of the application domain: architectural indoor scenes are very often *bounded by planes* such as walls, the floor or the ceiling. Thus, a lot of secondary rays are likely to hit such planar surfaces. Intersecting secondary rays with these bounding planes, regardless of what they actually hit, and looking up the direction vector from reflective surface point to the found intersection point often yields a better approximation of the correct reflected geometry than the standard look-up (Figure 4.4). This intersection is illustrated in Figure 4.5.

Each secondary ray has to be assigned a new look-up direction, thus, the bounding plane approximation should totally enclose the object such that each possible ray hits at least one plane. The minimum number of planes that can (almost) achieve this, is two, when aligned in parallel like floor and ceiling or two opposing walls. Then, the only undefined directions are those parallel to the planes, which is practically not visible in the image as it is a set of infinitely small extent. However, as this mapping of old directions to new ones is not continuous, discontinuities are visible where the intersected plane changes.

Furthermore, it is ignored that a ray might intersect a different surface before hitting a bounding plane, e.g. furniture or other interior that blocks the direct line of sight to the plane. Thus, the reflection of objects that do not really coincide with the bounding planes will look distorted. This artifact is especially visible when the distance between the ignored object and the plane intersection point is large (see Figure 4.6).

Using this bounding-plane-intersection scheme, the actual bounding plane geometry gets reflected correctly, resembling ray-traced images. The appearance of reflective objects on floors or tables reflecting the floors/tables greatly benefit from this trick (for more result images refer to Section 4.6). Reflected objects protruding from these planes get distorted, but as the selection

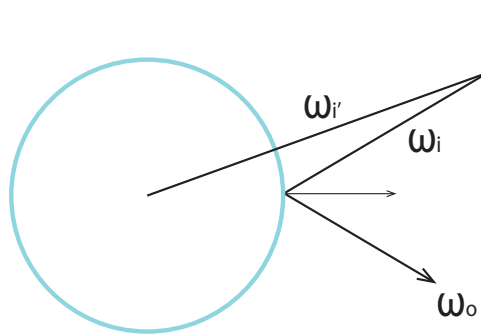


Figure 4.5: When an object is assumed to be bounded by planes, as commonly the case in indoor scenes, secondary rays can be intersected with these bounding planes. The vector pointing from the cube map projection center to the found intersection point is then used as a new look-up direction $\omega_{i'}$.

and specification of the bounding planes is up to the users, a configuration of planes can be chosen to minimize visible artifacts for the most interesting viewing directions. Specifying more than two planes to intersect with linearly increases both the intersection cost but also the number of correctly reflected planar objects. Then, the nearest of all found intersection points is used for looking up the incident radiance from the CEM.

An even better approximation of the intersected geometry might be achieved by using *distance impostors* [Szirmay-Kalos et al. 2005], a method of iteratively finding the actual intersection point using only depth information from the CEM. In general scenes, however, several iterations and look-ups are needed, which might sacrifice too much performance, as intersection points have to be found for each sample ray, i.e. 5-20 times per pixel. Intersecting sample rays with a handful of bounding planes as suggested before is feasible in real-time. But since distance impostors suffer much less from distortion artifacts, and they are particularly suited to scenes containing many planes, they might be a reasonable future extension to the employed method.

Another similar method of approximating scene geometry for reflections was proposed by Popescu et al. in 2006 [Popescu et al. 2006]. For computing reflections, each object is replaced by a *billboard*, and secondary rays are intersected with these billboards instead of the real geometry. Diffuse radiance exiting into the secondary ray's direction can then be directly read from the billboard texture. The bounding plane intersection proposed in this thesis can be seen as a simplified special case of their method, where only a very sparse set of objects (the bounding planes) is intersected with. Intersecting dozens of billboards or even employing depth-map search for the real intersected geometry — as also proposed in their paper — is not affordable in real-time when multiple sampling rays are shot per pixel. However, they also propose a *morphing scheme* to obtain smooth transitions between different intersected geometries, which would also be applicable to the method proposed in this thesis. As another future extension, the aforementioned distortion could be overcome by incorporating their morphing scheme.

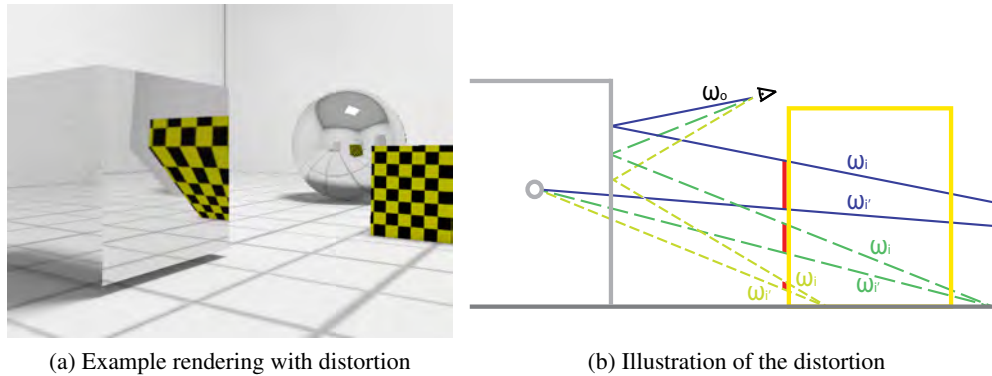


Figure 4.6: Since the reflected geometry is approximated by a set of planes (floor + ceiling in this image), all reflected objects other than these planes will look distorted. Note how the reflection of the floor is perfectly correct, but the yellow cube and the back wall get sheared. The greater the distance between reflected object and plane (e.g. between yellow cube and floor), the larger the distortion, as the actual intersection point then varies substantially from the found planar intersection point. The lower part of the reflected yellow cube gets warped since these rays hit the floor very far behind. The upper half gets less warped, as the distance between yellow cube and ceiling is much larger, compared to the distance between mirror cube and yellow cube. The discontinuity in the reflected yellow cube occurs where the secondary rays suddenly change from intersecting the floor to intersecting the ceiling. Close to this discontinuity, the rays hit the bounding planes extremely far away, thus, distortion is strongest (note the increasing offset (indicated in the right image in red) between actual intersection point and approximated intersection point as the reflection rays get more and more parallel to the bounding planes, i.e. close to the discontinuity)

4.4.2 Unbiased integration of incident light

As stated in the previous section, incident light is represented using a single cubic environment map per object. Rendering an object involves integrating the incident light according to the local material properties. This subsection deals with the *unbiased integration of incoming light*, which is employed in this thesis only for the iterative improvement of the rendering over time when the camera is still. The approximative deterministic integration used for real-time walkthroughs will be described in the next subsection. Reading this subsection first is encouraged, as it introduces required concepts and motivates the need for a faster approximation. Readers familiar with (Quasi-)Monte-Carlo quadrature in rendering might want to skim over this subsection and continue with subsection 4.4.3, as 4.4.2 does not contain any novel contributions.

Evaluating the specular integral using Monte-Carlo quadrature

Rendering specular reflection involves integrating products of incident light and specular BRDF (or BTDF). Equipped with the analytic representation of the specular BRDF ($f_{r_{lobe}}$) as described

in Section 3.3 and the CEM-representation of incident light (L_i) from the previous subsection, the integral can finally be evaluated for particular viewing directions. At each pixel, the viewing direction ω_o is different and the integration is performed independently using the specular scattering equation:

$$L_{o_{spec}}(\omega_o) = \int_{\Omega_{HS}} L_i(\omega_i) f_{r_{lobe}}(\omega_o|\omega_i) |\cos(\theta_i)| d\omega_i \quad (4.3)$$

where Ω_{HS} is the hemisphere of directions around the surface normal, $|\cos(\theta_i)|$ accounts for Lambert's cosine law and $d\omega_i$ is the differential solid angle around ω_i defined as $\sin(\theta_i) d\theta_i d\phi_i$, as the integrated intervals densify towards the poles by a factor $\sin(\theta_i)$. $f_{r_{lobe}}$ is the specular part of the BRDF (or BTDF, if the integration is performed for transmission).

Such integrals are usually evaluated numerically, by sampling the integrand and summing up the (possibly weighted) samples. A standard way to do this in offline rendering is Monte-Carlo quadrature, where samples are not taken in a regular grid but at random locations across the integration domain. In Monte-Carlo quadrature, Equation 4.3 is treated as a calculation of the expected value of the integrand, which can be estimated by averaging independent random samples that follow a probability distribution p [Szirmay-Kalos 2008]:

$$L_{o_{spec}}(\omega_o) = \frac{1}{n} \sum_{i=1}^n L_i(\omega_i) \cos(\theta_i) \frac{f_{r_{lobe}}(\omega_o|\omega_i)}{p(\omega_i|\omega_o)} \quad (4.4)$$

where n is the number of samples, and the quotient of $f_{r_{lobe}}$ and p is the weight of the sample. When sampling the sphere uniformly, the probability distribution function (pdf) p is constant, as each sample is equally probable. The benefits of introducing randomness are discussed in Section 2.4. On the downside, random sampling introduces noise when the samples used for integration at neighbouring pixels are uncorrelated. According to the *law of large numbers*, increasing the number of samples yields a closer estimate of the actual integration result. This reduces the noise, but as the evaluation time increases, it gets less applicable for fast rendering.

Reducing the variance by importance sampling

As most materials feature a certain scattering directionality, e.g. reflecting mainly according to the law of reflection, not all incoming directions contribute equally to the integration result. This is accounted for by the sample weight $f_{r_{lobe}}/p$, but when sampling uniformly, a lot of samples might be taken that, after weighting, do not contribute at all. Optimally, the sampling resolution in significant regions should be high and taking insignificant samples should be avoided. This is the idea of *importance sampling*: random sampling directions are not generated uniformly, but rather according to their probability as stated by the BRDF. Thus, incoming directions with a higher BRDF are more likely to be considered (see Figure 4.7). To ensure the integration result not to become biased by favouring certain regions, the probability distribution used in the sample weights is no longer constant but corresponds to the local sampling density (see Equation 4.9).

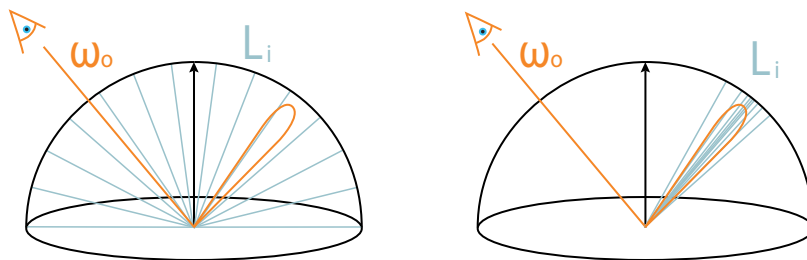


Figure 4.7: Left image: when sampling L_i uniformly, samples are taken in a regular grid, with a constant pdf as all directions are equally probable. Note that the specular lobe is grossly undersampled when the overall sampling resolution is low. Right image: importance sampling yields a higher sampling resolution in regions with a high BRDF, thus fewer samples suffice to obtain better results.

Specular reflection occurs mostly in directions within lobes around the perfect reflection direction. By favouring these most relevant regions, the integration domain is de facto reduced. This consequently decreases the sampling variance across neighbouring similar surface points, and thus, the noise is reduced, as shown in Figure 4.8. In other words, fewer samples suffice to achieve integration results of the same quality.

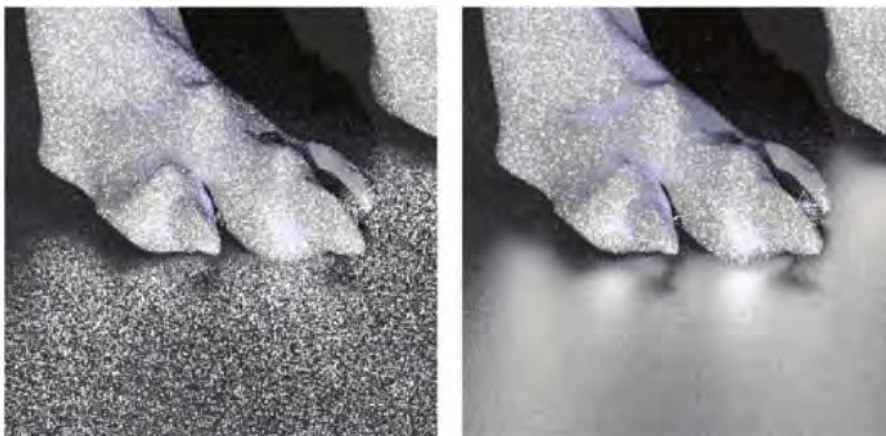


Figure 4.8: Importance sampling reduces the variance in the estimation of the scattering integral by favouring directions with a large BRDF. Left image: uniform sampling, right image: importance sampling. Both images use the same number of samples. Images taken from Pharr and Humphreys [Pharr and Humphreys 2004]

With an importance sampling strategy at hand, samples are efficiently taken in the most relevant parts of the integration domain. But how are these *important* sampling directions generated? The rejection sampling scheme described in Section 3.2 fulfills this purpose, but is not really suitable for real-time rendering. A better alternative are the analytic importance sampling

formulae presented along with the BRDF model [Kurt et al. 2010], on which our representation of $f_{r_{lobe}}$ is based. They are transformation functions that map uniform random variables $\xi_1, \xi_2 \in (0, 1]$ to new sampling directions that roughly follow the distribution specified by the BRDF. For the BRDF model by Kurt et al. [Kurt et al. 2010], which is also employed in this thesis, the transformation formulae first map ξ_1, ξ_2 to a half-way vector, that, along with a given viewing direction ω_o , yields sampling directions ω_i . The first step is [Kurt et al. 2010]:

$$\phi_h = \arctan\left(\frac{m_y}{m_x} \tan(2\pi\xi_2)\right) \quad (4.5)$$

$$\theta_h = \arctan\left(\sqrt{\frac{-\log(\xi_1)}{\frac{\cos^2 \phi_h}{m_x^2} + \frac{\sin^2 \phi_h}{m_y^2}}}\right) \quad (4.6)$$

where m_x, m_y are the material roughness parameters in tangent and bitangent direction (see Chapter 3.3). Thus, a pair of uniform numbers ξ_1, ξ_2 is mapped to a half-way direction (θ_h, ϕ_h) , from which a vector form of h can be computed using (see Fig. 4.9)

$$h = [\sin\theta_h \cos\phi_h, \sin\theta_h \sin\phi_h, \cos\theta_h] \quad (4.7)$$

Finally, an incoming sample direction ω_i can be derived using the equation

$$\omega_i = 2(\omega_o \cdot h)h - \omega_o \quad (4.8)$$

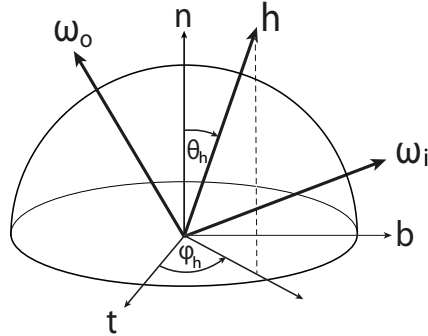


Figure 4.9: Illustration of θ_h, ϕ_h : h is the half-way vector between ω_i and ω_o . θ_h, ϕ_h are its spherical coordinates with respect to the local tangent frame (t, b, n) . When $(\theta_h, \phi_h) = (0, 0)$, h coincides with n , and the outgoing direction ω_o is the perfect mirror reflection direction of the incident direction ω_i .

By using these formulae, sampling directions ω_i are obtained that can be used to estimate the integral in Equation 4.4. As uniform directions are mapped to new, importance-weighted directions, the pdf in Equation 4.4 must be adapted in order to guarantee unbiased integration. Densely sampled regions must obtain lower weights, as large numbers of samples are likely to be generated in similar directions. Walter's notes on the Ward BRDF [Walter 2005] feature a general framework for deriving sampling probabilities from a given transform such as the one

described above. Using these instructions, the pdf used in Eq. 4.4 for the new transformed directions is [Kurt et al. 2010]:

$$p(\omega_i|\omega_o) = \frac{e^{-\tan^2(\theta_h)(\frac{\cos^2(\phi_h)}{m_x^2} + \frac{\sin^2(\phi_h)}{m_y^2})}}{4\pi m_x m_y \cos^3(\theta_h) (\omega_o \cdot h)} \quad (4.9)$$

where h is the half-way vector between ω_i and ω_o (see Fig. 4.9). Note that the pdf is similar to the BRDF (Equation 3.4), but not exactly the same. There is no notion of the Fresnel term or the parameter α in the pdf, thus, these parameters will not influence the sampling distribution. However, this does not matter, as samples are weighted by $f_{r_{lobe}}/p$ during the integration, and $f_{r_{lobe}}$ contains these effects.

In summary, Equations 4.5 to 4.8 can be employed to generate directions for sampling the incident lighting according to the pdf of the material. The obtained sampling directions are then used to estimate the illumination integral using Equation 4.4. By using the importance sampling scheme to sample the incident lighting the variance across pixels is considerably reduced.

Faster convergence using Quasi-Monte-Carlo quadrature

Monte-Carlo quadrature converges towards the expected value of the integrand as the number of samples increases. The convergence speed of the estimated integral is $O(\frac{1}{\sqrt{n}})$, which means that for reducing the noise (the variance) by a factor k , the number of samples has to be increased by a factor k^2 [Szirmay-Kalos 2008].

Quasi-Monte-Carlo rendering methods are essentially the same as their Monte-Carlo counterparts, but use quasi-random numbers instead of random numbers. Thus, they obtain a faster convergence than Monte-Carlo methods, as the coverage of the integration domain is constantly more even (see Section 2.4). Figure 4.10 shows a comparison of rendering the same scene using different numbers of samples in Monte-Carlo quadrature vs. Quasi-Monte-Carlo quadrature. Quasi-Monte-Carlo quadrature produces smoother results for any given budget of samples. Thus, in this thesis, Quasi-Monte-Carlo quadrature is employed when the camera is still, to ensure fast convergence towards the correct integration result over consecutive frames.

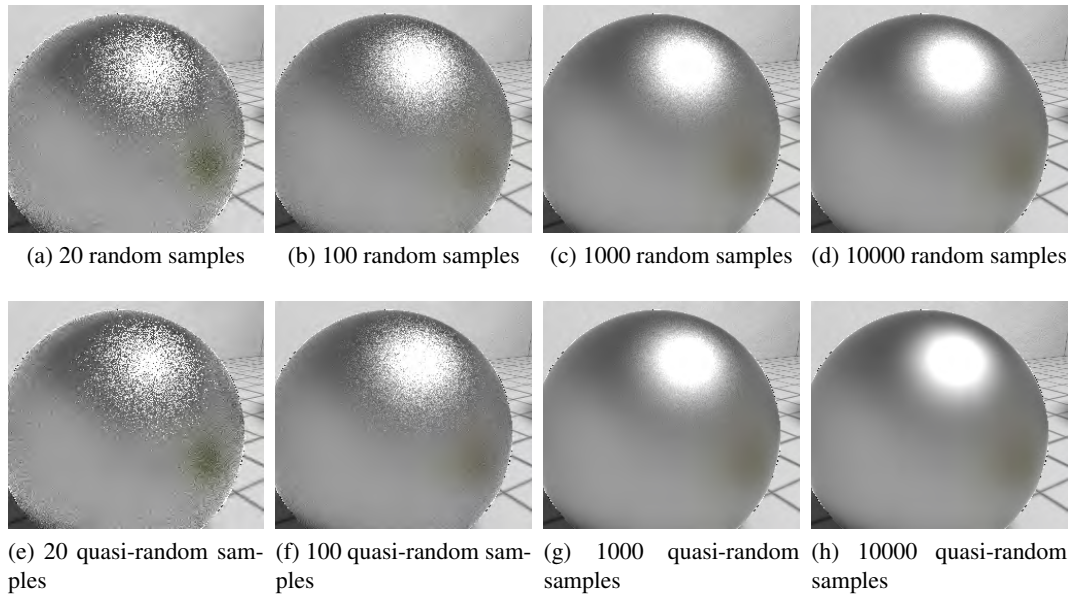


Figure 4.10: Comparison of the variance obtained by Monte-Carlo (top row) vs. Quasi-Monte-Carlo (bottom row) importance sampling. At low sampling rates the achieved images look similar, as the number of samples is grossly insufficient, which cannot be amended by placing them more cleverly. At higher sampling rates the difference becomes evident, especially at the border of the specular highlight.

Due to the real-time constraint, the number of allowed samples per pixel is limited to 5-20, which, even with Quasi-Monte-Carlo importance sampling, yields a noise level that is still noticeable, as evident in Figure 4.10. The methods described in this subsection produce correct integration results, if infinitely many samples are taken. Since the CEM-representation of incident light is discrete and contains limited information, the number of samples must be sufficient to capture the entire hemisphere of represented incoming directions. Still, the number required to obtain noise-free images is way too high to employ this method for real-time rendering. Thus, the method is employed in this thesis only when the camera is still: in each frame, a fixed number of quasi-random importance samples is taken and blended with the previous rendering result. This distributes the workload of thousands of samples over consecutive frames and allows visual convergence towards the correct integration result within one or two seconds. Note that the representation of incident light is just an approximation, thus, the *overall rendering result* will never become correct, but the integration of the light information as specified by the CEM does.

For real-time walkthroughs, the method is not directly applicable, since 10 - 20 samples per pixel result in spatial and temporal noise, when different independent samples are taken in each frame at every pixel. To obtain images that are spatially and temporally smooth, the integration result must be correlated across neighbouring pixels and across consecutive frames. The approach achieving this will be described in the next subsection.

4.4.3 Approximating the specular integral for real-time walkthroughs

The key to achieving a noise-free image within the restricted real-time budget of 5-20 samples is correlating the sampling patterns of all pixels across an object [Colbert et al. 2007]. The easiest way to do so is to reuse the *exact same sampling pattern* for estimating the integral at each pixel.

A sampling pattern can be derived using the importance sampling formulae (Eq. 4.5 to 4.8), transforming a single set of uniform random numbers $\{\xi_{1_i}, \xi_{2_i}\}$ to a set of sampling directions that will be reused at each pixel. As the numbers $\{\xi_{1_i}, \xi_{2_i}\}$ are uniformly distributed, the generated directions are likely to be distributed according to the pdf in Equation 4.9, comprising mainly good directions within the lobe. However, this depends on chance, as it is not strictly excluded that good directions are sparse, especially when using small numbers of random samples, resulting in visible systematic error due to suboptimal lobe coverage.

A better alternative proposed by Colbert and Krivanek [Colbert et al. 2007] is using *quasi-random numbers* for $\{\xi_{1_i}, \xi_{2_i}\}$. They are deterministic numbers from a *low-discrepancy sequence*, which means they are distributed very evenly across the domain, regardless of the number of samples. After transformation, they are distributed with respect to the importance pdf. Since quasi-random numbers are always uniformly distributed, important directions are likely to be generated even when a small number of samples is used (see Fig. 4.11b).

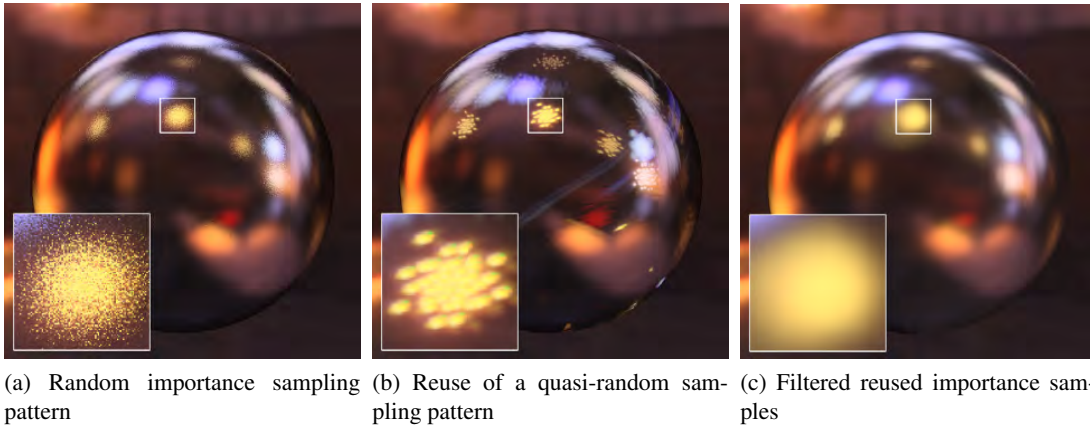


Figure 4.11: Random vs. deterministic importance sampling using 40 pixels. In (a), the sampling pattern varies across pixels, as new random numbers are used for importance sampling at each pixel, as described in the previous section. Noise is introduced by the variance of the sampling pattern. In (b), the sampling pattern is the same at each pixel, obtained by transforming quasi-random numbers using the importance sampling formulae of the material. The visual result is a weighted superposition of tilted CEMs, as the CEM is coherently sampled 40 times. The number of samples is too low to densely cover the lobe; aliasing effects occur that manifest as repeatedly copied light sources. In (c), the coherent samples of (b) are additionally filtered with a filter size according to the sample probability, achieving a smooth result when using 40 samples. Images taken from Colbert et al. [Colbert et al. 2007]

As evident in Figure 4.11b, importance-transformed samples are more concentrated in the middle of specular highlights. The number of samples used in this figure is too low to densely cover the specular lobe. Colbert and Krivanek overcome this problem by filtering the samples with a filter size inversely proportional to the pdf of a sample. In very dense regions near the lobe center, additional samples are very likely and thus, a smaller filter kernel suffices, whereas less probable directions need to be filtered more gravely, as similar directions are unlikely to occur again. The filtering is achieved by taking *MIP-mapped* samples of the environment map. Each MIP-map level contains an averaged version of four pixels of the lower level. Intuitively, taking MIP-mapped samples corresponds to sampling a cone of directions at once instead of a single direction. By using this filtering scheme, they can suppress the undersampling artifact, as shown in Figure 4.11c.

As Colbert and Krivanek report, 40-50 samples suffice to cover specular lobes more or less densely, when filtered. With our budget of 5-20 samples, it is likely, however, that parts of the lobe remain uncovered (see Fig. 4.12). The lower the number of samples, the larger the probability that unequal coverage becomes visible, especially at the less probable outer regions of a lobe. With that few samples, a BRDF-driven sampling density is likely to yield a suboptimal distribution, even when feeding quasi-random numbers to the transformation. Thus, a sample placement strategy is devised in the following that guarantees uniform lobe coverage even for very small numbers of samples.

Equally spaced samples for uniform lobe coverage

Quasi-random sampling distributions are not guaranteed to be visually uniform when using very few samples. The alternative employed in this thesis is taking *equally spaced samples within the angular domain of the specular lobe*. For isotropic lobes, samples are strategically placed in an elliptical, rotationally symmetric pattern (along iso-lines of θ_h). For highly anisotropic lobes, samples are regularly placed along the line of main elongation (iso-lines of ϕ_h). This strategic placement ensures that the coverage of the lobe is perfectly symmetric, regardless of the actual number of samples taken (see Figure 4.12).

For equally distributing samples within the domain of the specular lobe, the lobe's angular extent must be estimated. A threshold must be identified, beyond which the integration domain of the incident illumination is discarded. This subsection deals with the identification of this threshold, i.e. determining the (light transport-wise) most relevant part of the integration domain, which is going to be sampled during walkthroughs.

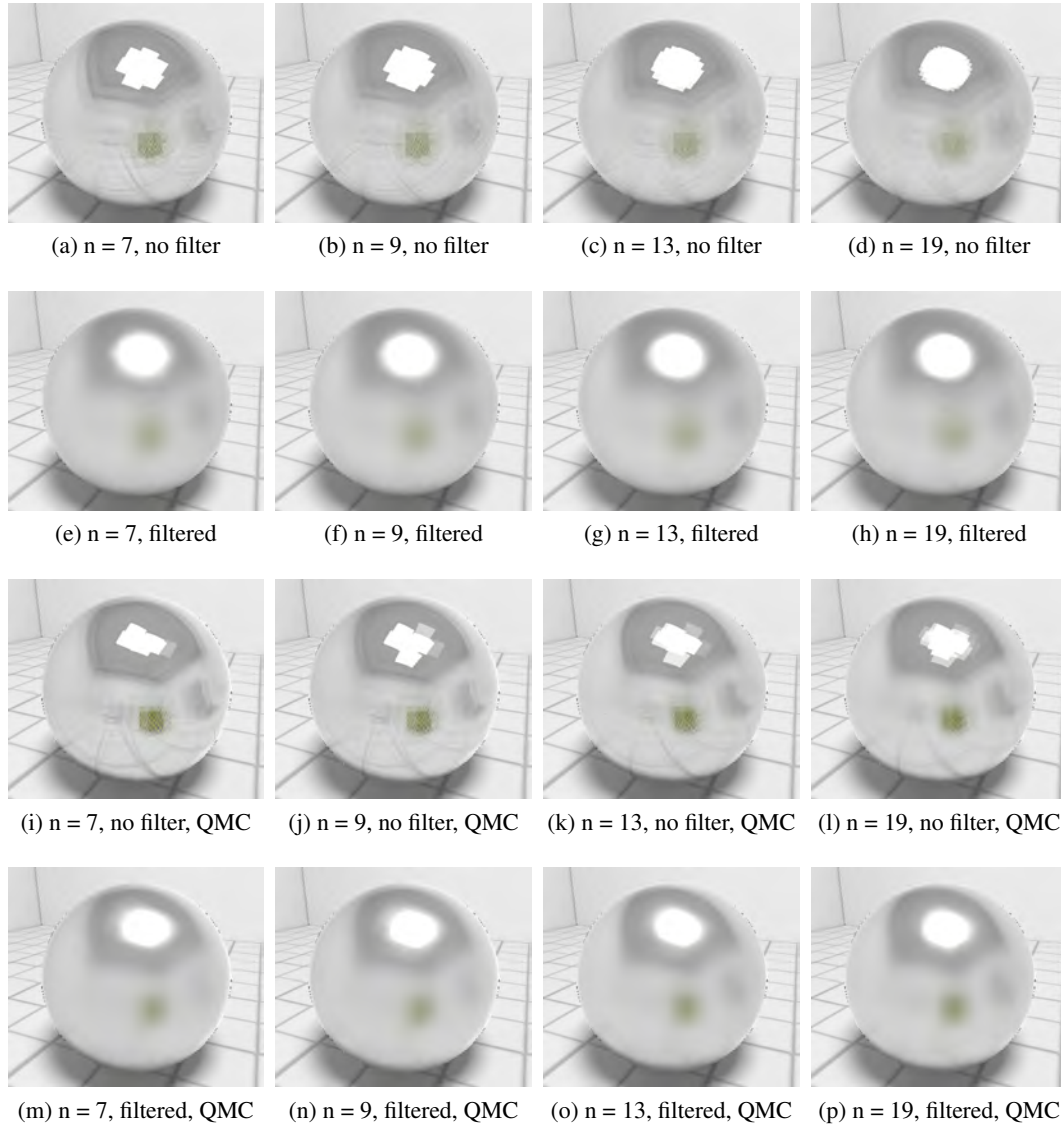


Figure 4.12: Equally spaced samples (rows 1,2) vs. Quasi-Monte-Carlo placement (rows 3,4). With equally spaced samples, the lobe coverage is always uniform and symmetric, while a QMC-distribution with importance sampling appears visually imbalanced when using very few samples. In rows 1 and 3, filtering is deactivated to better illustrate the distribution. The approximated lobe is isotropic with $m_x = m_y = 0.15$

The extent of the considered lobe region can be specified by a particular θ_h threshold, defining the highest allowed angular deviation from the perfect mirror direction in the lobe center (with $\theta_h = 0$). Directions with smaller θ_h than this threshold are considered to lie *within the lobe*. This threshold angle will be estimated in the following using the importance transformation

of Eq. 4.6. This transformation maps uniform numbers $\xi_1 \in (0, 1]$ to θ_h -angles of directions, such that the produced directions are distributed according to their *importance* as stated by the scattering probability. A plot (Fig. 4.13) of this importance transformation shows that directions near the lobe center are more likely to be generated than directions near and beyond the lobe border, as a much larger part of the $(0, 1]$ -source range is mapped to lower values of θ_h . Thus, thresholding this importance can be interpreted as defining ‘lobe membership’ for directions: identifying a value of ξ_1 below which the generation probability of directions approaches zero yields an estimate of the limit angle θ_{thresh} at the lobe border (see Fig. 4.13).



Figure 4.13: This plot shows how numbers $\xi_1 \in (0, 1]$ (on the x-axis) are transformed to half-angles $\theta_h \in (0, \pi/2]$ (on the y-axis) when using the non-linear importance transformation function in Eq. 4.6 for different lobe parameters (different colors). At $\xi_1 = 0$, the resulting θ_h is $\pi/2$, which ensures that no directions are excluded for unbiased integration, but in fact, the portion with significant importance ends before ξ_1 reaches 0. Identifying the θ_{thresh} -angle for a particular ξ_1 -value close to zero, where the slope of the transformation curve starts to get very steep, yields an estimate of the lobe extent worth sampling.

The limit angle θ_{thresh} of the considered lobe region can be obtained by inserting a limit value of $\xi_{1_{border}}$ (close to zero) into Eq. 4.6:

$$\theta_{thresh} = \arctan\left(\sqrt{\frac{-\log(\xi_{1_{border}})}{\frac{\cos^2 \phi_h}{m_x^2} + \frac{\sin^2 \phi_h}{m_y^2}}}\right) \quad (4.10)$$

It is empirically found that a threshold value of $\xi_{1_{border}} = 0.01$ results in a θ_{thresh} -angle that lies close enough to the lobe border and yields visually satisfying results. With this quan-

tification of the lobe extent, the angular domain inbetween can be uniformly sampled during walkthroughs. This can now be done by linearly mapping the range $[0, 1]$ to the range between 0 and the determined threshold:

$$\theta_h(\xi_1) = (1 - \xi_1) \cdot \theta_{thresh} \quad (4.11)$$

Note that θ_{thresh} does not depend on ξ_1 , thus it is essentially constant for a given ϕ_h , view and lobe parameters. We now have a transformation formula that maps uniform numbers $\xi_1 \in (0, 1]$ to uniformly spaced half-angles θ_h within the lobe. This uniformity-preserving linear transformation is illustrated in Figure 4.14.

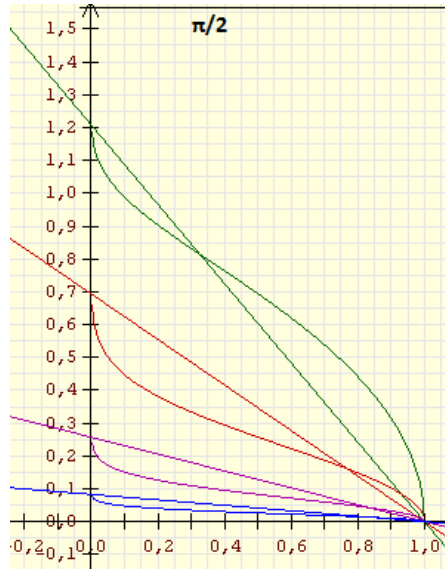


Figure 4.14: Plot of the non-linear importance transformation function for θ_h for several surface roughness values (different colors), and their linear approximations. Using the linear transformation, only directions within the lobe can be generated, as the integration domain is no longer $[0, \frac{\pi}{2}]$, but clamped to $[0, \theta_{thresh}]$.

Using Eq. 4.10, the θ -angle of samples can be obtained. For ϕ_h , the same transformation as in the original importance sampling transformations (Eq. 4.5) is employed. Equation 4.5 is non-linear, as it contains trigonometric functions, however, the distortion introduced by them is desirable, as it ensures even spacing when anisotropic lobes are sampled (see Fig. 4.15). As an anticipatory remark regarding the integration using these samples, please note that the desirable distortion introduced by the ϕ -mapping is not accounted for in the sampling pdf, which is assumed to be uniform in both θ and ϕ . For isotropic lobes, this is actually the case, as then, the $\tan$ and $\arctan$ functions in Eq. 4.5 cancel each other out, and it becomes a linear mapping. For strongly anisotropic lobes, the suggested sampling patterns place their samples along constant iso-lines of ϕ_h , so this distortion does not manifest itself at all. Only when the suggested isotropic ring-patterns are used for anisotropic lobes, this non-uniformity of ϕ_h will erroneously be ignored in the pdf, which then introduces approximation error.

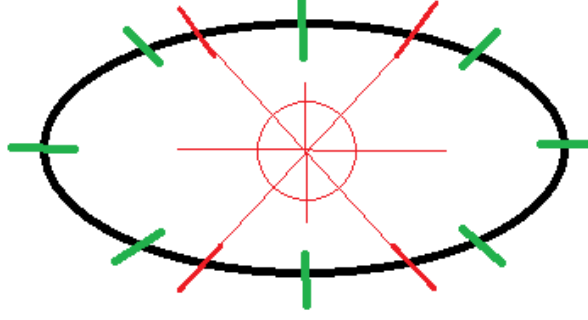


Figure 4.15: Illustration of the (desirable) distortion effect introduced by the non-linearity of Eq. 4.12: when taking 8 samples whose ϕ -angles are equally spaced, the anisotropy of the lobe is ignored, resulting in non-uniformly spaced sampling directions (red). As Equation 4.12 compensates for anisotropy in a non-linear fashion, the resulting sampling directions are evenly spaced (green).

In summary, we have established a scheme that allows quantifying the lobe extent using the importance transformations of the employed material model. Furthermore, we devised a uniformity-preserving transformation scheme that allows the construction of sampling patterns in ξ_1, ξ_2 -space, and then maps these numbers to sampling direction patterns in a particular specular lobe (see Fig. 4.16 for an illustration of the mapping). These transformations are (Eq. 4.12 is the same as Eq. 4.5, but Eq. 4.13 was devised in this subsection):

$$\phi_h = \arctan\left(\frac{m_y}{m_x} \tan(2\pi\xi_2)\right) \quad (4.12)$$

$$\theta_h = (1 - \xi_1) \cdot \theta_{thresh} \quad (4.13)$$

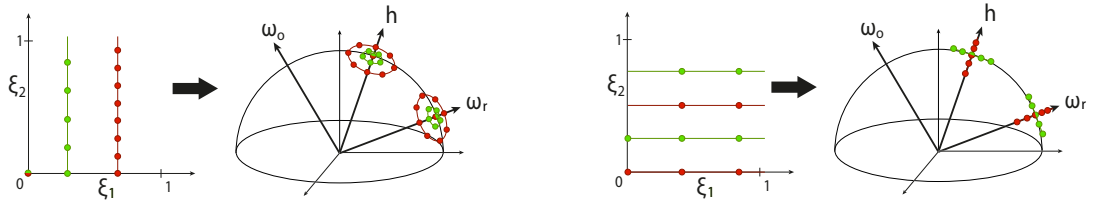


Figure 4.16: Illustration of the uniformity-preserving transformation formulae (Equations 4.12, 4.13). Iso-lines of ξ_1 are transformed to ellipses around the half vector h (left image), while iso-lines of ξ_2 are mapped to lines of fixed ϕ_h (right image). Note that, in the right image, a single iso-value of ξ_2 is mapped to a single value of ϕ_h , yielding a line that only spans one side of the central sample. Thus, we need two iso-values to obtain the depicted lines spanning both sides ($\phi, \phi + 180^\circ$). Together with a particular ω_o , the obtained directions around h yield sampling directions within the lobe, i.e. samples around the mirror reflection direction ω_r .

Finally, it must be noted again that by taking samples exclusively within the specular lobe, large parts of the incident illumination are ignored in computing the outgoing radiance. However, as the specular BRDF is almost zero beyond the lobe, the error introduced by this approximation is negligible next to other simplifications of the method (reuse of a single CEM, small numbers of samples, etc.).

Regular sampling patterns for isotropic and anisotropic lobes

With the uniformity-preserving transformations at hand, uniform sampling patterns can simply be constructed in the (ξ_1, ξ_2) -space, and then transformed to the angular domain of the lobe using Equations 4.12, 4.13, 4.7 and 4.8. In this thesis, two particular patterns are constructed – a circular one for isotropic lobes and a linear one for strongly anisotropic lobes.

In the case of isotropic reflection the lobe should be covered uniformly in all ϕ -directions. This is achieved by taking one sample in the perfect reflection direction (lobe center) and by placing all other samples in a ring around the lobe center, as shown in Figure 4.17a. In the case of strongly anisotropic reflection, when the highlight is rather linear than circular, this ring-configuration would get squeezed together and equal spacing is no longer ensured. Thus, such cases of strong anisotropy are dealt with differently: all samples will be equally spaced along the main elongation axis of the highlight (see Figure 4.17b).

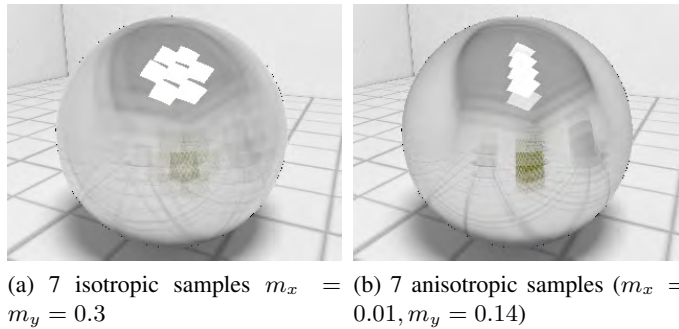


Figure 4.17: Illustration of the equal sample placement: for isotropic lobes, one sample is placed at the lobe center and all others are arranged in a ring around the center. For strongly anisotropic reflection, all samples are equally spaced linearly along the major elongation axis of the highlight. Note that in the right image the two outermost samples are barely visible due to the BRDF-weighting, which is low near the border of the lobe. Filtering is deactivated for a better illustration of the patterns.

To establish which particular numbers must be fed to the modified transformation formulae to obtain these patterns, the structure of the mappings is investigated more closely. Figure 4.18 shows a color plot of how numbers $\xi_1, \xi_2 \in [0, 1]$ are mapped to the angular domain.

Let's look at the case of high anisotropy first: samples should be equally spaced along a single line following the main elongation axis of the highlight. A line within the lobe is achieved for fixed ϕ -angles and varying θ -angles, i.e., fixed ξ_2 and varying ξ_1 . Reading directly from the

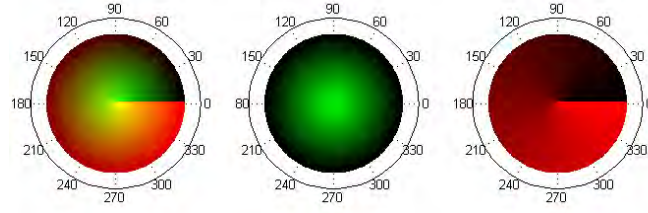


Figure 4.18: Mapping structure of the modified transformation formulae: the left plot shows how numbers ξ_1, ξ_2 are mapped to sampling directions in the angular domain of the specular lobe. The filled circles depict the domain of the specular lobe, with ϕ as the polar angle and θ as the radius. The color of each pixel indicates which pair of ξ -numbers was mapped to obtain that particular direction. The number ξ_1 , which is mapped to the θ -angle is plotted in green, while the number ξ_2 (mapped to ϕ) is plotted in red; for instance, a yellow pixel indicates that $(\xi_1, \xi_2) = (1, 1)$. The middle and right plots each depict the mapping of θ (green) and ϕ (red) separately, for a better distinguishability. It is evident that the perfect reflection direction at the lobe center is obtained when $\xi_1 = 1$ (saturated green), and that the smaller the value of ξ_1 , the larger the distance from the lobe center. ξ_2 is the angular parameter along iso-lines of ξ_1 , where $\xi_2 = 0$ is mapped to $\phi = 0$ and $\xi_2 = 1$ is mapped to $\phi = 2\pi$. Note that in anisotropic cases the circles become ellipses, but the mapping structure and thus the color distribution within the elliptical lobe would be the same.

plot in Figure 4.18 yields the following distribution scheme for one extreme anisotropic case ($m_x \ll m_y$), given a number n of samples:

- Place the first half of samples ($k = 0 \dots \lfloor \frac{n}{2} \rfloor$) at $\xi_1 = \frac{2k+1}{n}, \xi_2 = 0.25$
- Place the other half of samples ($k = \lfloor \frac{n}{2} \rfloor + 1 \dots n - 1$) at $\xi_1 = 2 - \frac{2k+1}{n}, \xi_2 = 0.75$

For example, in case of $n=5$ this produces the pairs $(\frac{1}{5}, 0.25), (\frac{3}{5}, 0.25), (\frac{5}{5}, 0.75), (\frac{3}{5}, 0.75), (\frac{1}{5}, 0.75)$, where the fractions for ξ_1 yield the equidistribution of θ , as 1 yields the perfect reflection direction in the lobe center and 0 yields the lobe border. The fixed ξ_2 -values 0.25 and 0.75 yield the fixed ϕ -angles of 90° and 270° . For the second anisotropic case ($m_x \gg m_y$) the scheme looks very similar:

- Place the first half of samples ($k = 0 \dots \lfloor \frac{n}{2} \rfloor$) at $\xi_1 = \frac{2k+1}{n}, \xi_2 = 0.0$
- Place the other half of samples ($k = \lfloor \frac{n}{2} \rfloor + 1 \dots n - 1$) at $\xi_1 = 2 - \frac{2k+1}{n}, \xi_2 = 0.5$

And finally, for the isotropic case, all but the central sample should be placed in a ring around the middle, i.e. on a particular ξ_1 -iso-line:

- Place one sample at the lobe center $\xi_1 = 1, \xi_2 = 0$
- Place the other samples ($k = 1 \dots n - 1$) at $\xi_1 = radius, \xi_2 = \frac{k-1}{n-1}$

where *radius* determines the particular radius (i.e. distance from the lobe center) of the sampled ring. It is empirically established that $radius = 0.6$ (= 40% off-center) yields a visually similar result to the unbiased integration. Too small or too large radii result in an under- or overestimation of the specular highlight.

For a particular specular lobe, it can now be decided via a threshold of $\frac{m_y}{m_x}$ which of the three distributions should be used. The ring-pattern should not be used for strongly anisotropic lobes, since as it gets squeezed together, formerly opposing samples get very close to each other, wasting sampling resolution. Alternatively to thresholding, some kind of blending between the distributions depending on m_x and m_y can be employed to allow a smooth transition between the patterns.

As a final remark, please note that the produced relative sampling distribution with respect to the lobe is exactly the same regardless of the surface normal and viewing angle. For rough surfaces with wide specular lobes, up to 50% of the generated samples might fall below the geometric horizon at grazing angles, as the sample placement is not aware of geometry. However, this is easily detected during sampling since $\dot{dot}(\omega_i, n)$ is negative for such samples. Whenever such directions are generated, that cannot possibly contribute to the outgoing radiance by means of reflection, they are discarded and not used in the integration at all.

Filtering the evenly spaced samples

In order to avoid aliasing artifacts caused by low sampling (see Fig. 4.11b), each sample is filtered using MIP-maps, similar to the method of Colbert and Krivanek [Colbert et al. 2007]. Sampling the CEM without filtering can be thought of as considering the incident light from an infinitely small cone around the sampling direction. As the MIP-mapped version of the CEM additionally contains averaged downsampled versions of the CEM, a MIP-map-filtered sample considers a wider cone of incident lighting. Roughly speaking, a MIP-mapped sample considers several adjacent incoming directions at once, resulting in a filtered look-up.

As our sampling distribution is uniform within the lobe, the *angular* filter kernel size is the same for all samples. Optimally, the kernel size is chosen such that the entire highlight gets covered by the union of all samples, while the overlapping of samples is kept low, as shown in Figure 4.19.

Even though the angular filter size of all samples is the same, they cover different pixel areas when projected onto the CEM. In order to obtain the correct filter kernel size for a sample, the number of CEM-pixels subtended by the sample has to be estimated. This is done in two steps: First, the area subtended by the entire lobe at unit distance is estimated, and divided by the number of samples. Then, this estimated sample area is projected onto the surrounding cube on which the CEM is defined, to obtain each individual sample area in terms of CEM pixels. Thereby, the cross-section of a specular lobe is always assumed to be elliptical, as illustrated in Fig. 4.20 (left). The projection of a single sample (depicted in red) onto the surrounding cube is illustrated in Fig. 4.20 (right).

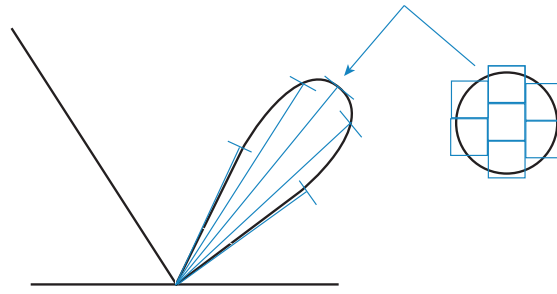


Figure 4.19: This illustration shows how the specular lobe is covered by MIP-mapped samples. Ideally, the pyramidal cones defined by MIP-mapped samples should densely cover the lobe without overlapping. The right image shows the lobe coverage as seen from above, looking into the perfect reflection direction. Larger numbers of samples allow a closer approximation of the elliptical lobe cross-section by square look-ups.

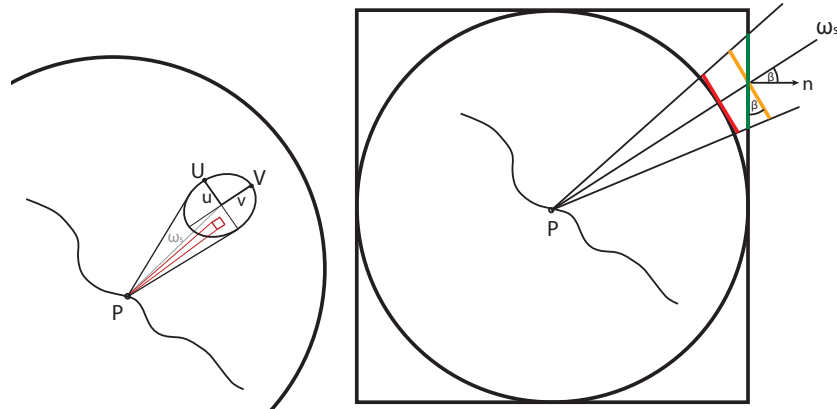


Figure 4.20: Left image: The cross-section of the lobe at unit distance from the shaded point is elliptical. Its vertices U, V can be derived using Eqns. 4.12 and 4.13, allowing the calculation of the elliptical lobe area. Dividing the ellipse area by the number of samples yields an estimate of the sample area at unit distance (red). Right image: 2D-side view of projecting a sample onto the CEM. The sample area at unit distance (red) is projected onto the surrounding CEM-cube to obtain a measure for the number of covered CEM-pixels for the sample. In the course of this projection, the sample area increases with the distance between unit sphere and unit cube (orange vs. red), as well as with the angular difference between the cube's face normal and the sampling direction ω_s (green vs. orange).

First, the area of the lobe-cross-section ellipse that is tangent to the unit sphere in the lobe center direction ω_C must be computed. The transformation formulae in Eq. 4.12 - 4.13 are employed to obtain the lobe border vectors $\omega_U = \frac{P\vec{U}}{\|P\vec{U}\|}$ and $\omega_V = \frac{P\vec{V}}{\|P\vec{V}\|}$, where P is the shaded scene point (= unit sphere center) and U, V are vertices of the ellipse. With ω_U and ω_V , the

angles α_U and α_V between lobe center direction ω_C and ω_U and ω_V are computed using

$$\alpha_U = \text{acos}(\omega_C, \omega_U) \quad \alpha_V = \text{acos}(\omega_C, \omega_V) \quad (4.14)$$

Now the lengths of the ellipse semi-major and -minor axes u and v are obtained using

$$u = \tan(\alpha_U) \quad v = \tan(\alpha_V) \quad (4.15)$$

With these lengths the area of the elliptical cross section is computed using $A = uv\pi$. Now the area of a single sample at unit distance is estimated by dividing the unprojected ellipse area A by the number of samples: $A' = \frac{A}{n_{\text{samples}}}$. Then, each sample area is projected separately onto the cube.

When projecting an area from the unit sphere to the surrounding cube, the area is distorted and gets larger. Two factors are involved: first, the distance d between the sphere and the cube in the projection direction causes a quadratic increase of the area (orange vs. red in Fig. 4.20). This distance is found by intersecting the sampling direction ω_S with the cube faces. Second, the tilting angle β between the projection direction ω_S and the normal vector of the cube-face introduces a distortion according to a factor $\frac{1}{\cos(\beta)}$. Thus, the area of a sample on the CEM cube is:

$$A'_{\text{cube}} = A' \cdot \frac{d^2}{\cos(\beta)} \quad (4.16)$$

Finally, the number of CEM-pixels covered by this area is obtained using

$$\text{pixels}_{\text{sample}} = A'_{\text{cube}} \cdot \frac{N}{4} \quad (4.17)$$

when N is the number of pixels per face of the CEM (e.g. 512^2). Since the CEM-cube surrounds the unit sphere, the cube-face area is 4. Thus, $\frac{N}{4}$ is the number of pixels covering an area of 1. The MIP-map level corresponding to a pixel number of $\text{pixels}_{\text{sample}}$ can now be calculated using $\log_4(\text{pixels}_{\text{sample}})$, as the number of averaged pixels increases by a factor 4 in each level of the MIP-map pyramid.

Equations 4.14 - 4.17 yield an appropriate MIP-map level, i.e. filter kernel size for each sample. Since each sample is projected separately, its area is always estimated in the correct CEM-face, even when the lobe spans two or three adjacent faces.

Figure 4.21 shows a comparison of rendering results using the MIP-map filtering scheme described in this subsection.

Approximating the specular reflection integral with regular uniform sampling

As the samples have been distributed and filtered, the last question is how to integrate incident light using only these particular directions. In the work of this thesis, the purpose of the regular sampling approach is approximating the integration result for real-time walkthroughs. When the camera is still, unbiased Quasi-Monte-Carlo integration using the standard importance sampling from Subsection 4.4.2 is employed to ensure convergence towards the correct integration result within several frames.

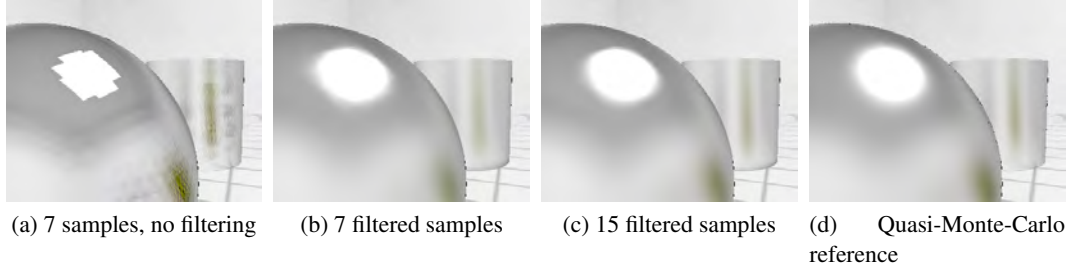


Figure 4.21: Comparison of filtering results. In (a), 7 samples are taken without filtering. In (b), the MIP-map filtering is applied to produce smooth results. When using only 7 samples, the filtering is strong and MIP-mapping artifacts appear, such as the shape of the quadratic filter kernel becoming visible; at the same time, the sampling pattern itself is still slightly visible near the highlight borders when using very few samples. In (c) the number of samples increases, which means denser coverage, weaker filtering (lower MIP-map levels) and fewer visible artifacts. When comparing (b) and (c) to the Quasi-Monte-Carlo reference rendering (d), the overall appearance of the material is quite similar.

The ultimate goal is to achieve a material appearance during real-time walkthroughs that is very similar to the correct integration result obtained when the camera is still. However, the samples are no longer distributed according to the importance pdf, statistically concentrating near the lobe center, but deterministically placed in a uniform way. Thus, we can not simply employ the importance-pdf in Monte-Carlo integration, but have to adapt the pdf in order to match the uniform sample distribution during walkthroughs. A new pdf – hence called p' – must be formulated and used in Equation 4.4, yielding:

$$L_{ospec}(\omega_o) = \frac{1}{n} \sum_{i=1}^n L_i(\omega_i) \cos(\theta_i) \frac{f_{rlobe}(\omega_o|\omega_i)}{p'(\omega_i|\omega_o)} \quad (4.18)$$

As the samples are generated uniformly, p' must be constant. It can thus be extracted from the sum:

$$L_{ospec}(\omega_o) = \frac{1}{n \cdot p'} \sum_{i=1}^n L_i(\omega_i) \cos(\theta_i) f_{rlobe}(\omega_o|\omega_i) \quad (4.19)$$

If in the integration was performed across the entire hemisphere of directions, the constant p' would be $\frac{1}{\pi^2}$, as then $\int_{HS} p' = 1$, which is a necessary requirement for any pdf. However, as only a part PT of the hemispherical domain HS is actually sampled (the part covered by the specular lobe), p' must be zero for directions beyond PT , as these directions can no longer be generated. In order to still yield 1 when integrated across the hemisphere, p' must be A_{HS}/A_{PT} for directions within PT , as it is only non-zero at A_{PT}/A_{HS} percent of the integration domain. Thus, we have to estimate A_{HS}/A_{PT} according to the currently sampled lobe, or in other words, we want to quantify the inverse ratio of *the area of the hemisphere actually covered by the lobe*.

In the following, it is shown how this ratio can be estimated by sampling another known pdf, namely the *importance pdf* p used in MC-importance sampling.

By its definition as a pdf, we know that the hemispherical integral of pdf p (Eq. 4.9) is also 1. As stated in the previous subsection, we only sample parts of the hemisphere that clearly lie within the lobe, i.e. directions with non-negligible pdf and BRDF. Accordingly, we can define a new function, p_{PT} , which equals p for directions within PT , and zero beyond. Thus, the hemispherical integral of p_{PT} is $1 - \epsilon$, with ϵ being the integral over those parts of p that lie beyond PT . However, by construction, ϵ is almost zero, and thus, $\int_{HS} p_{PT} \approx 1$. These integrals over p and p_{PT} are illustrated in Figure 4.22.

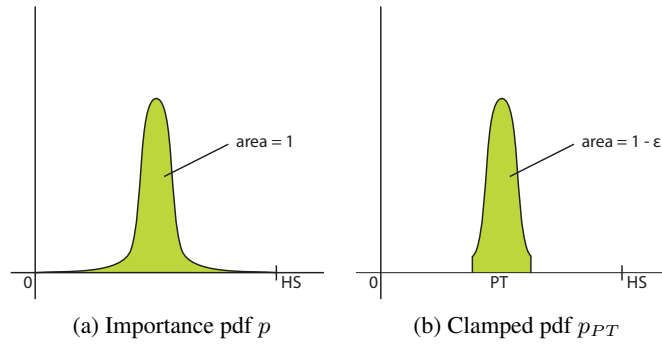


Figure 4.22: Simplified 1D-illustration of the hemispherical integral of the importance pdf p (left), and the clamped importance pdf p_{PT} (right). The discarded area ϵ decreases, as the size of the clamped domain PT is increased towards the entire hemisphere, HS .

We can estimate $\int_{HS} p_{PT}$ using a Monte-Carlo estimator, taking n uniform samples within the hemisphere, and we know that the result must be close to 1:

$$\int_{HS} p_{PT} = \frac{1}{n} \sum_{i=1}^n p_{PT}(\omega_i) \approx 1 \quad (4.20)$$

The sum in Eq. 4.20 can be split up into two partial sums, one considering n_{PT} out of n samples that lie within PT , and one considering $n - n_{PT}$ samples beyond PT . As p_{PT} is 0 beyond PT , the second partial sum can be discarded, yielding:

$$\frac{1}{n} \sum_{i=1}^n p_{PT}(\omega_i) = \frac{1}{n} \sum_{i=1}^{n_{PT}} p_{PT}(\omega_i) = \frac{1}{n} \sum_{i=1}^{n_{PT}} p(\omega_i) \approx 1 \quad (4.21)$$

As n samples are taken uniformly within the hemisphere, and n_{PT} of these samples lie within PT , $n_{PT} = A_{PT}/A_{HS} \cdot n$. Rearranging this expression and inserting it into Eq. 4.21 yields

$$\frac{1}{n_{PT}} \frac{A_{PT}}{A_{HS}} \sum_{i=1}^{n_{PT}} p(\omega_i) \approx 1 \Rightarrow \frac{1}{n_{PT}} \sum_{i=1}^{n_{PT}} p(\omega_i) \approx \frac{A_{HS}}{A_{PT}} = p' \quad (4.22)$$

This means that the average importance pdf p across a number of uniform samples within the lobe (PT) is an estimator of A_{HS}/A_{PT} , or in other words, an estimator of our new constant pdf p' . The estimation error decreases as PT gets closer to HS , i.e., it is dependent on how much of the integration domain is discarded in Eq. 4.11. Currently, the cut is performed at $\xi_{1_{border}} = 0.01$ in Eq. 4.11, which results in a barely visible ϵ , and thus, a result that matches the converged importance sampling reference solution quite closely (see Figure 4.24).

We can insert Eq. 4.22 into Eq. 4.19, yielding our new integration formula for uniformly sampling the lobe (note that n in Eq. 4.23 is equal to the former n_{PT} , as all samples are taken within the lobe):

$$L_{o_{spec}}(\omega_o) = \frac{1}{\sum_{i=1}^n p(\omega_i|\omega_o)} \sum_{i=1}^n L_i(\omega_i) \cos(\theta_i) f_{r_{lobe}}(\omega_o|\omega_i) \quad (4.23)$$

Before showing some result images and comparing them to the importance sampling reference, two more things are pointed out: first, it is empirically found that, for the particular sampling patterns we use, an even better result can be achieved when each sample is additionally weighted with a factor $f_{r_{lobe}}/p$. In order not to bias the overall sampled energy after re-weighting, the resulting sample sum is then no longer divided by $\sum p$, but by $\sum p \cdot \frac{f_{r_{lobe}}}{p} = \sum f_{r_{lobe}}$ instead. A thorough investigation of why this new weighting ameliorates the result is left for future work, but experimental results show that for our particular sampling patterns, this redistribution of sampled energy yields a result that is consistently closer to the importance sampling reference than Eq. 4.23, as shown in Figure 4.24.

The second and last thing that needs to be pointed out is that due to the fixed sampling pattern that is reused at each pixel, continuous horizon lines become visible where a sample is taken at grazing angles, almost below the horizon (see Fig. 4.23). The BRDF and importance pdf get extremely large at grazing angles, which results in a local darkening near the horizon line of a sample, since within the quadrature sum the almost-zero $\cos(\theta_i)$ -term reduces the large sample weight, but in the overall division by the sum of p or $f_{r_{lobe}}$ it remains large. The artifact is illustrated in Figure 4.23.

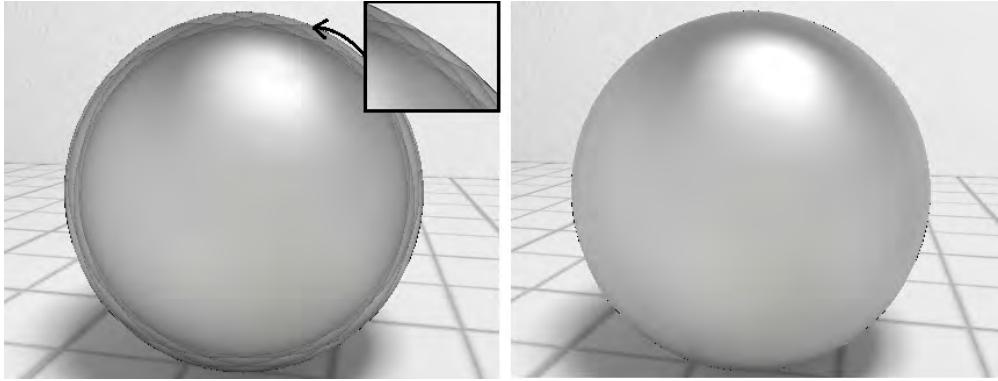


Figure 4.23: The left image shows the visible horizon lines of samples close to grazing angles. These are lines where the BRDF and importance pdf of a sample approach infinity, which results in local darkening, as the large sample weight gets divided by but not fully multiplied with. In the numerator of Eq. 4.23 it gets attenuated by $\cos(\theta_i)$, while in the denominator outside of the sum it is not. Therefore, the weighting term is extended by an additional $\cos(\theta_i)$, to decrease the influence of samples at grazing angles, removing the artifact as depicted in the right image.

In order to remove the visible horizon lines, each sample is additionally weighted with a second $\cos(\theta_i)$ -term in order to reduce its contribution near the horizon, as shown in Fig. 4.23. Again, to maintain the overall energy, the additional weighting has to be included in the weight-sum by which the whole quadrature sum is divided. With this final modification, we have two possible integration formulae (Eqns. 4.24, 4.25), of which the first one is plausible and has been fully derived in this section, whereas the second one is empirically found but yields a consistently closer approximation of the reference solution (see the difference images and PSNR (*peak signal to noise ratio*) values in Fig 4.24). Thus, only the second formula will be employed and evaluated in this thesis, but the first one is more founded and almost as close to reference, as shown in Figure 4.24 on the next page.

$$L_{ospec}(\omega_o) = \frac{1}{\sum_{i=1}^n p(\omega_i|\omega_o) \cdot \cos(\theta_i)} \sum_{i=1}^n L_i(\omega_i) \cos(\theta_i)^2 f_{r_{lobe}}(\omega_o|\omega_i) \quad (4.24)$$

$$L_{ospec}(\omega_o) = \frac{1}{\sum_{i=1}^n f_{r_{lobe}}(\omega_o|\omega_i) \cdot \cos(\theta_i)} \sum_{i=1}^n L_i(\omega_i) \cos(\theta_i)^2 \frac{f_{r_{lobe}}(\omega_o|\omega_i)^2}{p(\omega_i|\omega_o)} \quad (4.25)$$

In summary, the previously presented method allows approximating the specular integral using very low numbers of samples, achieving a similar material appearance as the Quasi-Monte-Carlo approach. More result images, implementation details and limitations to the method can be found in Section 4.6.

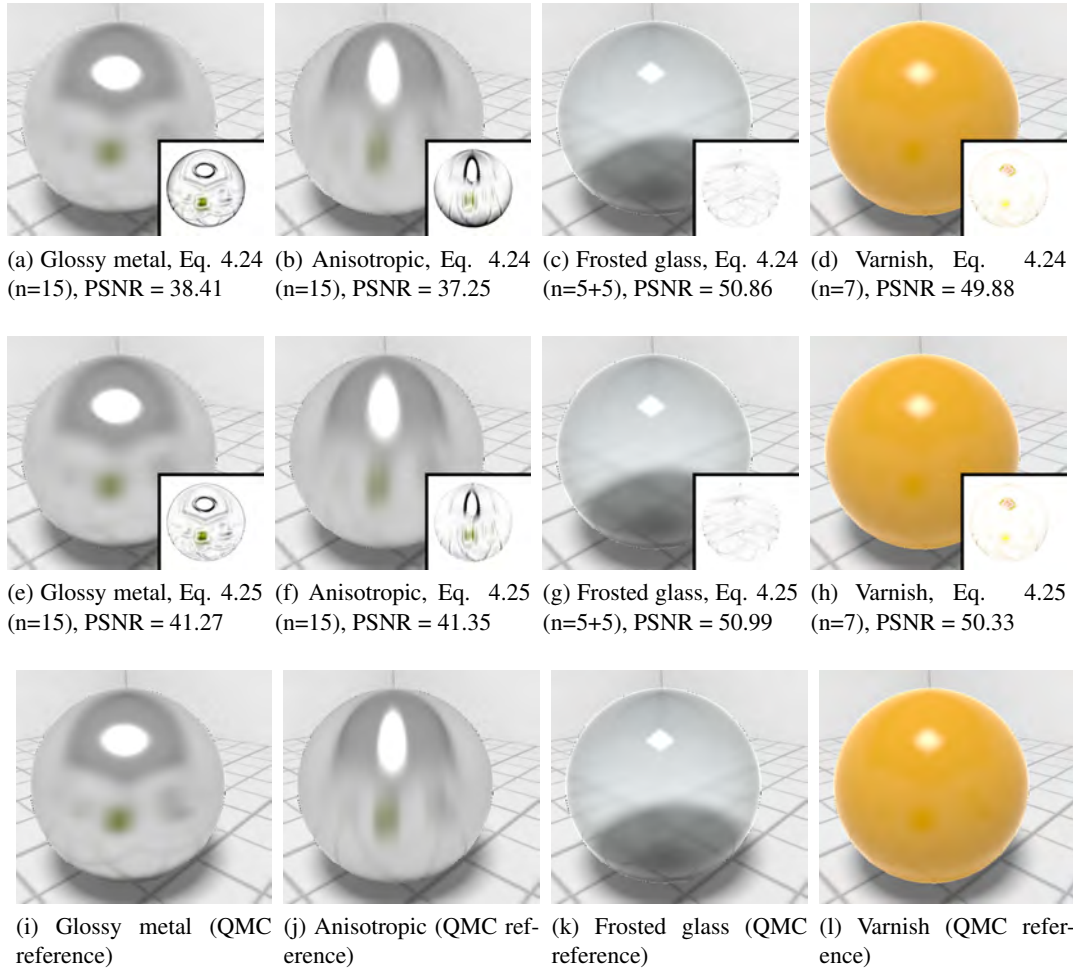


Figure 4.24: Results produced by the approximative integration using n samples within the lobe, together with difference images and PSNR values with respect to the reference integration: the first row shows results using the integration formula of Eq. 4.24, the second row shows results of Eq. 4.25, and the third row shows results of the converged Quasi-Monte-Carlo approach (reference). For the frosted glass rendering, $n = 5 + 5$ means that 5 samples were taken to approximate the reflection lobe, and 5 additional ones were taken for the transmission lobe. Difference images are multiplied by a factor 15 for a better visualization. The PSNR values show that 4.25 yields a consistently better approximation of the reference than Eq. 4.24.

4.5 Rendering specular reflection on planar surfaces

The rendering approach described in the previous section is applicable to specular surfaces of arbitrary geometry, and particularly suited to surfaces of high curvature. However, the cubic environment map representation of incident light fails to reasonably capture the lighting environment across very large objects, as it is only valid at a single point in space. In this section, a different approximative representation of incident light is discussed that allows rendering specular reflection in large *planar* surfaces in real-time. Essentially, the incident light is calculated according to the view by rendering the scene into a buffer texture from a mirrored point of view. First, this representation of incident light is discussed, then the scheme for sampling and filtering this representation is presented.

4.5.1 Representing incident light by a mirror rendering

Representing the incident light using a cubic environment map (CEM) is straightforward, as it is defined in the angular domain and can be sampled directly for integration. However, it is only a correct representation of incident light at one particular point, namely the projection center of the CEM. If one CEM is reused for entire objects that extend way beyond this single point, wrong things get reflected. For small objects the error might not be noticed, especially when the reflections are not perfectly specular but glossy. The larger the object is compared to the distance of the reflected environment, the worse the error gets.

An example in indoor scenes where a single CEM-representation of incident light fails totally are *large planar surfaces* such as walls or the floor. Imagine a single point on the floor for which the incident light is captured and stored in a CEM. Close to that point the incident light is represented well. But there might be an object close to this point, behind which the floor will be shaded totally wrong as the incident light should include the back-side of the object and not the front as captured by the CEM (see Fig. 4.25). This is an extreme example, but generally, the incident light varies too much across large floors or walls to be captured by a single CEM.

A different approximative representation of incident light is employed in this thesis for such large planar surfaces: before rendering the actual frame, an extra rendering pass is performed as seen from a virtual camera location, obtained by mirroring the real camera location and view about the planar surface [Diefenbach et al. 1997]. The plane itself must be omitted in the mirror rendering, as it would occlude the scene. After the mirror rendering, the buffer contains the incident radiance from the perfect reflection direction at each pixel of the actual view, as illustrated in Figure 4.26.

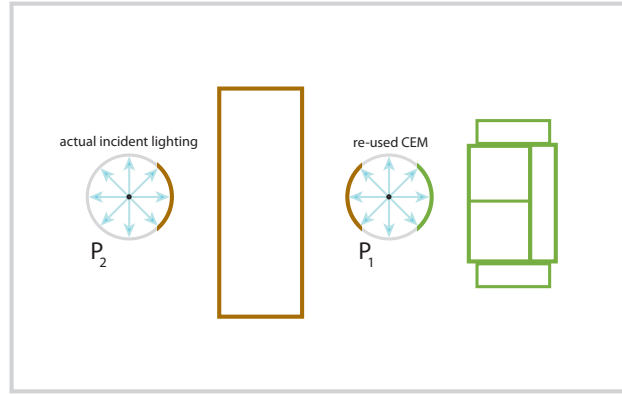


Figure 4.25: Illustration of the error when a single CEM is reused for large objects. The CEM correctly stores the incident light at point P_1 in the center of the room. When point (P_2) on the floor behind the brown cupboard is shaded using the (P_1)-CEM, it gets wrongly illuminated as the green sofa is not even visible from P_2 . The correct hypothetical CEM for P_2 is depicted, and it differs significantly.

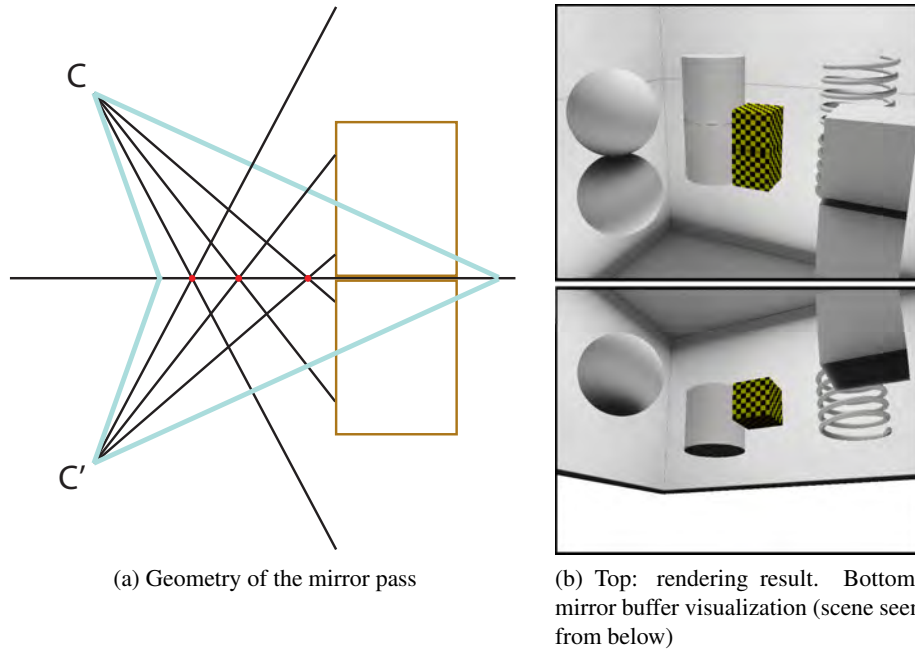


Figure 4.26: Illustration of the mirror rendering: the scene is viewed from the (real) camera position C , the field of view is given by the bright blue lines. When shading the red points of the reflective floor, the reflection rays are determined by the law of reflection ($\theta_i = \theta_o, \phi_i = \phi_o + 180^\circ$). As evident in the construction, the incident radiance from these mirror directions can be captured by rendering the scene (without the plane) first from position C' . Assuming the mirror rendering buffer has the same resolution as the actual rendering viewport, the incident radiance of a point in the plane is found in the buffer at the exact same screen-space position of the plane point, with flipped vertical axis. The lower right screenshot shows the result of the mirror rendering pass, the upper right screenshot shows the result when rendering the actual scene using the mirror buffer as incident radiance representation for the ground plane.

The mirror rendering buffer will be used as the representation of incident light for calculating the specular reflection integral. With a cubic environment map, the substantial approximation is that it is reused across an object but *only valid at a single point in space*. With the mirror rendering representation, incident lighting is represented correctly at each point across the plane, but *only for the perfect mirror direction*. But as specular lobes are defined around this perfect mirror direction, radiance information from the neighbouring pixels in the buffer can be used as an approximative substitute for incident directions within the lobe. For rougher surfaces with wide specular lobes this approximation gets error-prone, since sampling rays might hit scene geometry that is occluded in the mirror view. Only light information that is visible in the mirror rendering may contribute to the approximation of the specular integral. However, for very specular surfaces such as polished or varnished floors or tables, it is reasonable to assume that large parts of the lobe are represented within the mirror buffer (see Figure 4.27).

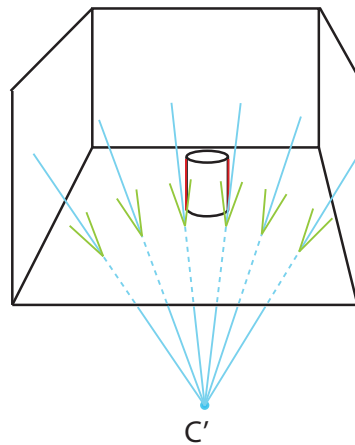


Figure 4.27: The mirror rendering buffer contains the incident radiance for each point in the plane, but only for the perfect mirror direction. For narrow specular lobes, it is likely that the incident radiance for other directions within the lobe can be found in the neighbouring pixels of the mirror look-up position. However, this is not the case when lobes (depicted in green) would intersect geometry that is not visible from the virtual camera C' , e.g. the red regions on the sides of the cylinder. The wider the specular lobe, the larger such undefined regions get, and artifacts become visible, as wrong radiances are read.

Technically, rendering the scene from a mirrored point of view should involve a fully fledged illumination of the mirrored scene, in order to yield correctly illuminated reflected objects. And even reflected objects may reflect view-dependent surfaces, and so on. Allowing several view-dependent bounces substantially increases the rendering costs of each frame. In this thesis, only the last bounce is view-dependent. Thus, only diffuse reflection is considered when rendering objects in the mirror pass for real-time walkthroughs. When the camera is still, more than one view-dependent bounce could be resolved without performance considerations (e.g. [Diefenbach et al. 1997]), but this has not been implemented in the course of this thesis.

4.5.2 Generating illumination samples using the mirror buffer

With the mirror-buffer rendering at hand, the incident radiance at each point P in the reflective plane is available, but only from the mirror reflection direction ω_r . For very specular planes, it can be assumed that the incident radiance of similar directions $\omega_i \approx \omega_r$ can be found in the neighbouring pixels of the buffer. As we already have a scheme to generate angular sampling directions ω_i according to the material (Section 4.4), a projection scheme is needed that maps these angular samples ω_i to look-up positions $(x, y)_{lookup}$ in the mirror buffer.

For shading a point P of the planar reflector, the incident light representation must be sampled. After generating an angular sampling direction ω_i , it is mapped to a mirror buffer sample using the following scheme:

1. Construct a sampling ray from P in direction ω_i . Find the closest intersection point P_I of the sampling ray with the scene geometry
2. Project the intersection point P_I into the view of the mirror camera to obtain look-up coordinates (see Fig 4.28)

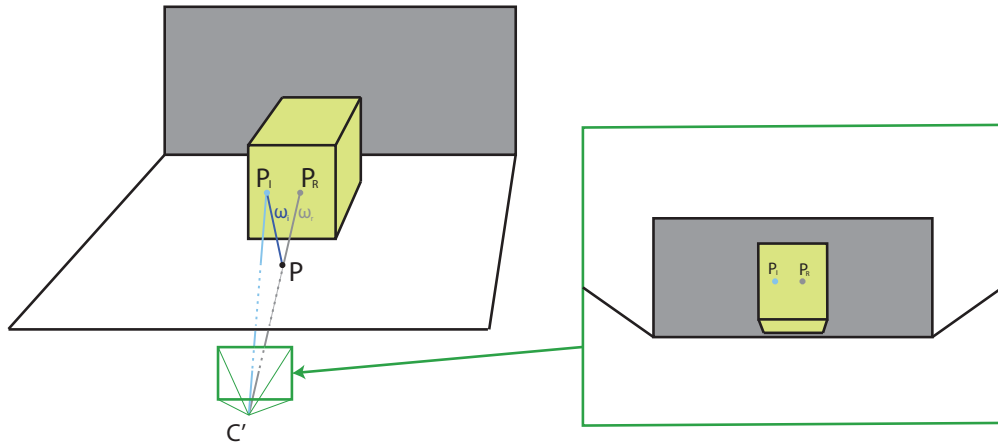


Figure 4.28: When shading the point P in the ground plane, incident light from sampling directions ω_i (dark blue) is taken into account. For obtaining the incident radiance from ω_i , the nearest intersection point P_I of the ω_i -ray with the scene has to be found. Then, P_I is projected into the view of the mirror camera, to obtain 2D-coordinates for reading the exiting radiance of P_I in the mirror buffer. The right image shows the scene as seen from the virtual mirror camera C' . P_R is the reflected scene point in the lobe center direction ω_r , for which the exiting radiance is always captured correctly.

The second step is easy, since all parameters of the virtual mirror camera are known. By multiplying the world-space coordinates of the point P_I with the mirror camera's *view*- and

projection-matrix, performing perspective division and normalizing the coordinates to $[0,1]$, the look-up position $(x, y)_{lookup}$ in the mirror buffer is obtained. It might happen, though, that the point P_I is not visible in the mirror camera's view due to occlusion. But if it is visible, this projection of P_I into the mirror buffer yields the correct incident radiance from that direction. For narrow specular lobes, occlusion will not occur too much, since in this case $\omega_i \approx \omega_r$, and P_R is always visible.

Much more intricate is the first step of the scheme: intersecting the sampling ray with the scene to obtain the point P_I , which illuminates P from the sampled direction. Actually intersecting the ray with the scene geometry as in ray-tracing is not fast enough when scenes are complex, and it has to be done for each sample, i.e. several times per pixel. Therefore, the point P_I is approximated using information from the mirror-buffer as follows:

- During the mirror buffer rendering not only the color information (i.e. incident radiance) is stored, but also the surface normals and world-coordinate positions of the mirror-rendered scene points. Thus, the exact world positions P_R and surface normals are known for all perfect reflection samples ω_r .
- We *assume* that all other sampling rays within the lobe will intersect the scene in points P_I that lie in the tangent plane of P_R (see Fig 4.29). This is true when P_R lies in planar surfaces, which are very common in architectural scenes. For arbitrary surfaces, the error of this assumption is small when the specular lobe is narrow, as the sampling directions ω_i are then close to the lobe center direction ω_r , and locally, most surfaces are planar.
- The sampling ray is intersected with the tangent plane of P_R to obtain P_I .

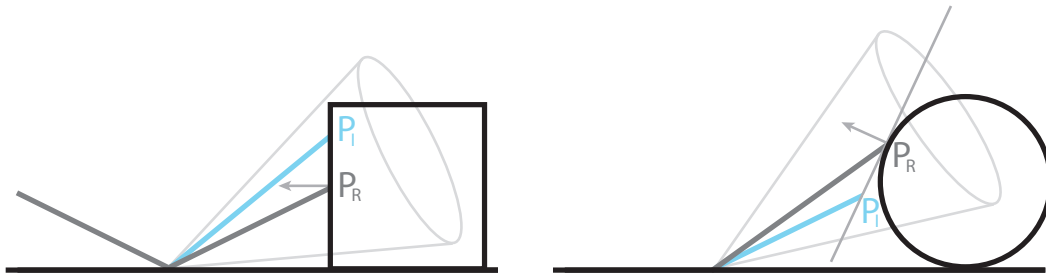


Figure 4.29: The world position P_R and the surface normal of the reflected scene geometry are known at the perfect reflection direction ω_r (dark gray). The scene is then locally approximated by the tangent plane of P_R to obtain approximative positions P_I where other sampling rays (blue) within the lobe intersect the scene. This assumption is correct for planar reflected surfaces, for arbitrary surfaces the error depends on local scene curvature and specular lobe width.

After P_I is obtained, it is projected into the view of the virtual camera C' to obtain mirror buffer coordinates. By using this scheme, each angular sampling ray is mapped to a buffer

look-up position to approximate the incident radiance from that direction. As the geometry that illuminates a point from ω_i is assumed to lie in the tangent plane of P_R , best results are obtained when this is actually the case, e.g. for walls, tables, cupboards, doors and all other roughly planar objects.

On the downside, the approximation of the intersected geometry and thus also the buffer look-up position strongly depend on the tangent plane obtained when sampling the perfect mirror direction P_R , as all samples at a pixel are assumed to intersect this particular tangent plane. When this tangent plane changes significantly across neighbouring pixels, e.g. at the border of an object, visual discontinuities appear as the sampling is suddenly performed in a wholly different plane. Fig. 4.30 illustrates this discontinuity, and Fig. 4.31 shows a screenshot of this artifact.

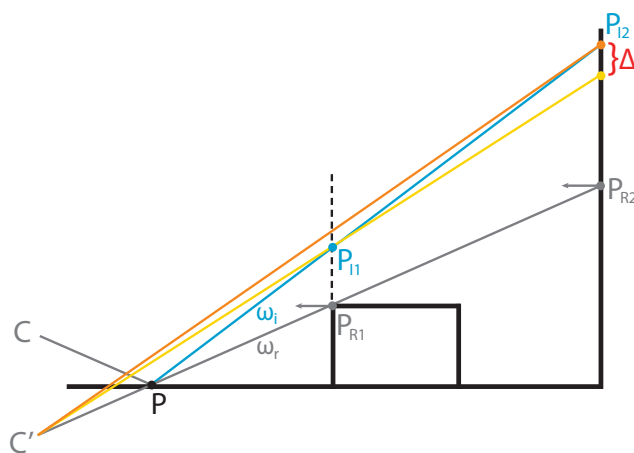


Figure 4.30: Illustrating the consequence of the tangent plane discontinuity: point P is shaded using the incident radiance from ω_i (blue line). When the mirror reflection ray ω_r intersects the black box in P_{R1} , the tangent plane is the side face of the box, yielding intersection point P_{I1} for the sampling direction ω_i . The obtained radiance seen from C' in the direction of P_{I1} is the yellow dot on the back wall. At the neighbouring pixel, when a point next to P is shaded, the mirror reflection ray no longer hits the box but the back wall instead (intersection point P_{R2}). The sampling ray ω_i intersects a different tangent plane now, namely the back wall. The new intersection point P_{I2} now yields the orange dot on the back wall as seen from the virtual camera C' . The red delta on the back wall is the perceived jump caused by this discontinuity.

Close to such depth discontinuities in the mirror buffer, the assumption of P_I lying in the tangent plane of P_R is likely to be violated, as while the perfect mirror ray ω_r might still hit an object, a slightly different sampling ray ω_i might already miss it and hit a different object at different depth instead. Thus, assuming P_I to lie in the tangent plane of P_R is not the best strategy near object discontinuities. Instead of taking all samples in the same tangent plane of P_R , the projection scheme is extended such that each sample ray is intersected with a different tangent plane to estimate P_I as follows (the first two steps are identical to the previous scheme, the third is new):

1. Construct a sampling ray from P in direction ω_i . Intersect this ray with the tangent plane of P_R to obtain a first estimate of P_I .
2. Project the intersection point P_I into the view of the mirror camera to obtain look-up coordinates $(x, y)_{lookup}$
3. **Look up the world position and normal which are seen from the mirror camera in that buffer position $(x, y)_{lookup}$, and intersect the sampling direction ω_i with this plane to obtain a new estimate of P_I . Then repeat step 2 and look up the incident radiance from the new buffer position.**

The assumption behind this additional intersection step is that whatever the mirror camera sees at $(x, y)_{lookup}$ defines the tangent plane in which P_I lies. When a sampling ray ω_i slightly misses an object even though P_R is within its boundaries, the intersection point with the tangent plane of P_R as seen from the camera will already lie beyond the object, yielding a new tangent plane to intersect ω_i with. Regarding Fig. 4.30, the yellow dot would no longer be used directly for shading, but instead, its tangent plane is intersected again by the sampling ray ω_i , yielding the orange point, a much better estimate of the correct P_I . This point is then projected into the mirror camera view and the obtained radiance is used for shading. Through extending the projection scheme by this third step, the orange dot would be found regardless of whether the mirror reflection ray hits the box or the back wall. Figure 4.31 shows a screenshot of how the additional intersection step decouples the obtained radiance from the tangent plane of P_R .

With this extended projection scheme at hand, angular sampling directions are mapped to sampling positions in the mirror rendering buffer. The radiance obtained from the buffer can still be wrong in three ways: first, P_I is just an approximation of the exact scene point that a sampling ray would hit. Second, P_I might be occluded in the mirror camera's view, then, the obtained radiance is the occluder's instead of P_I 's. Third, the radiance values in the mirror buffer are exiting radiances in directions $-\omega_r$. The exiting radiance in directions $-\omega_i$ is only equivalent if the object reflectance is purely diffuse, i.e. view-independent. The latter one is true by definition in this thesis, as only diffuse reflection is considered during the mirror rendering, regardless of whether object materials are different. Occlusion is only a real problem for wider specular lobes, as near object borders large parts of the lobe will intersect geometry behind the object that is occluded in the mirror view. But for narrow lobes, all assumptions are likely to hold, and the neighbouring pixels of each P_R are likely to contain valid P_I -information. Again, keep in mind that the main goal of view-dependent effects in this thesis is capturing the overall material appearance, not absolute correctness.

That being said, some exemplary results of the projection scheme are depicted in Figure 4.32. A very large number of quasi-random importance samples were generated in the angular domain and projected to the mirror rendering buffer. Images like these are achieved when the camera is still, as thousands of samples can only be taken over consecutive frames. Note that the entire rendering result will never get correct over time, as both the representation of incident lighting and the sample projection scheme involve substantial approximations, but the integral converges. Nevertheless, some important effects of glossy planar reflection are captured very well.

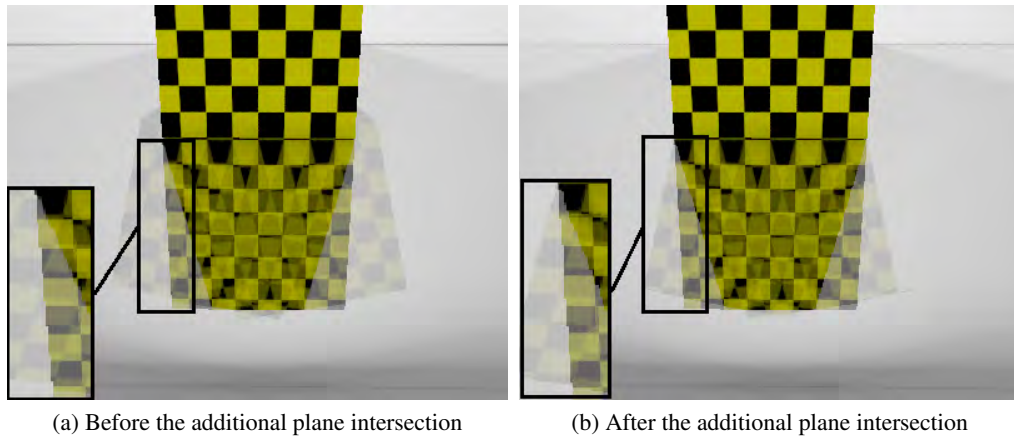


Figure 4.31: This figure shows how the additional intersection step improves the estimation of P_I (filtering is deactivated). At each pixel in the two images, the same three angular samples are taken within the specular lobe. Thus, the expected result would be three continuous sharp reflections of the checkered cube in the plane. In the left image, P_I is assumed to lie in the tangent plane of P_R . The discontinuity of the outer sample reflections is visible near the border of the mirror-reflected cube since the tangent plane of P_R suddenly changes from the cube to the back wall. Note how the inclined yellow and black quads do not align perfectly near the border. With the additional intersection step (right image), all ω_i are now intersected with the tangent plane of the point they yielded in the mirror buffer, which results in a much better approximation of P_I that appears continuous even at depth discontinuities of P_R .

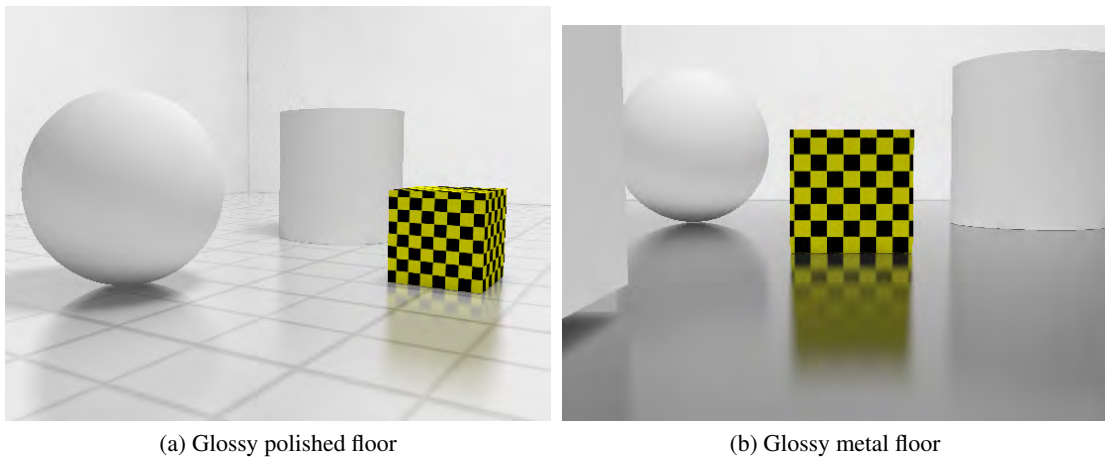


Figure 4.32: Non-real-time renderings of scenes with a glossy floor using thousands of samples per pixel and the proposed projection scheme. Note how the blur increases with the distance between reflector and reflected geometry, as the lobe's cross-section area quadratically increases with this distance.

With thousands of samples at each pixel, soft glossy transitions can be achieved at object- or depth discontinuities. When shading neighbouring pixels along a line orthogonally to the depth discontinuity, the number of sampling rays hitting the objects varies smoothly, yielding a correctly blurred transition (e.g. at the border of the yellow cube). Furthermore, the effect of increasing blur with increasing distance between reflector and reflectee is naturally produced by the projection scheme. As thousands of samples can never be taken in real-time, images like those in Fig. 4.32 can only be achieved over consecutive frames, when the camera is still. During walkthroughs, a deterministic sampling approach using low numbers of samples and hardware filtering is employed to approximately achieve images of similar quality. This approximative solution will be discussed in the next subsection.

4.5.3 Deterministic sampling pattern and filtering for real-time walkthroughs

In Subsection 4.4.3 deterministic uniform sampling patterns were presented that cover a specular lobe with very few samples in the angular domain, allowing the approximative solution of the specular integral in real-time. Theoretically, these patterns can be directly employed for sampling the mirror buffer instead of an environment map. However, it turns out that the presented circular sampling pattern for isotropic lobes is not optimal when rendering large planar surfaces with low numbers of samples. Smooth transitions across depth discontinuities as depicted in Figure 4.32 can only be achieved with sufficient sampling resolution orthogonal to the discontinuity. The circular pattern is suboptimal for this purpose in multiple ways: regardless of the object- or depth discontinuity's orientation, the sampling resolution orthogonal to it is always higher at the outer border of the lobe than near the energy-wise more important lobe center (see Fig. 4.33a). Furthermore, all samples of the ring contribute with the same intensity, which negatively affects the frequency response of the involved low-pass filtering (the ring is essentially a box filter). Thus, the sampling pattern is modified by placing multiple rings of samples with different radii around the lobe center, concentrating the sampling density near the center and achieving a better frequency response (the rings now behave like a tent filter), which aids in achieving the desired blur across discontinuities.

The rendering in Figure 4.34 shows the effect of the new sampling pattern when using 10 samples. Note how the modified pattern captures the shape and smooth fade-out of the highlight much more effectively.

The multiple-rings sampling pattern achieves a more effective coverage of the lobe when depth discontinuities are involved. However, in the angular space the new pattern is less uniform (concentrating towards the lobe center) and, due to the low number of samples per ring, the lobe is less densely covered. This is why the multi-ring pattern is not employed when rendering *curved surfaces* using the environment map representation of incident light, which is defined directly in the angular domain and does not contain any similar discontinuities in the found scene intersection points. Finally, note that the linear sampling patterns for strongly anisotropic lobes do not need adaptation for the mirror buffer sampling, as their resolution is uniform and the samples' contribution already decreases towards the borders.

Now that the samples have been placed, hardware filtering is used analogously to Subsection 4.4.3 to achieve a smooth blurring of the samples according to the material roughness. Again, the filtering is based on a MIP-mapped representation of the incident lighting. After rendering

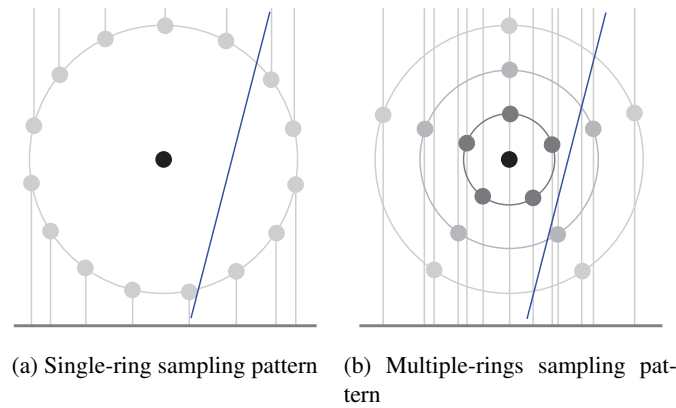


Figure 4.33: The perceived sampling distribution of the circular pattern (a) is concentrated near the ring of outer samples. Placing the samples in multiple concentric rings (b) round the center before projecting causes a concentration near the energy-wise more important lobe center. As a desirable side-effect, the continuously fading blur in glossy reflections can be rendered more effectively as samples are now taken at different radii with fading contribution. As the pattern is swept across the discontinuity (blue line) when shading neighbouring pixels, the contribution from beyond the discontinuity fades in more smoothly with the multiple-rings pattern, the same way as a tent-filter produces a smoother signal than a box-filter.

the scene to a mirror buffer each frame, the buffer's contents are MIP-mapped on the GPU (downsampled, averaging 4 pixels in each iteration), allowing to consider incident radiances from an interval of directions with a single look-up. MIP-map levels (filter kernel sizes) are chosen adequately such that the union of filtered samples ideally captures the entire lobe.

The derivation of the filter kernel size for each sample is performed similarly to filtered environment map sampling: conceptually, the extents of the entire lobe are projected into the mirror buffer, estimating the pixel area that should be covered by the union of samples. This is done by obtaining the 2D-buffer-look-up positions of the horizontal and vertical extreme directions within the lobe using the procedure enumerated in Subsection 4.5.2. The obtained 2D-positions in the buffer are assumed to be the vertices of an approximately elliptical area of the lobe's projection into the buffer. Thus, the area of pixels covered by the entire lobe can be approximated using the ellipse area formula $A = ab\pi$. Dividing the number of subtended pixels by the number of samples yields an approximate number of covered buffer pixels per sample, which is then used as a kernel size to filter each sample. Note that the actual shape of the geometry hit by the lobe is not necessarily elliptical when projected into the mirror buffer, especially when the lobe spans multiple objects of different curvature at different depth. Thus, this assumption is likely to introduce discontinuities of the filter kernel size close to borders of reflected objects. Again, there is less potential for error when lobes are sharp, as in a narrow angular scope the margin of possible geometrical variation is bounded.

In summary, specular reflection on planar surfaces is approximated by first rendering the diffuse scene into a mirror buffer, then sampling this buffer multiple times at each shaded pixel

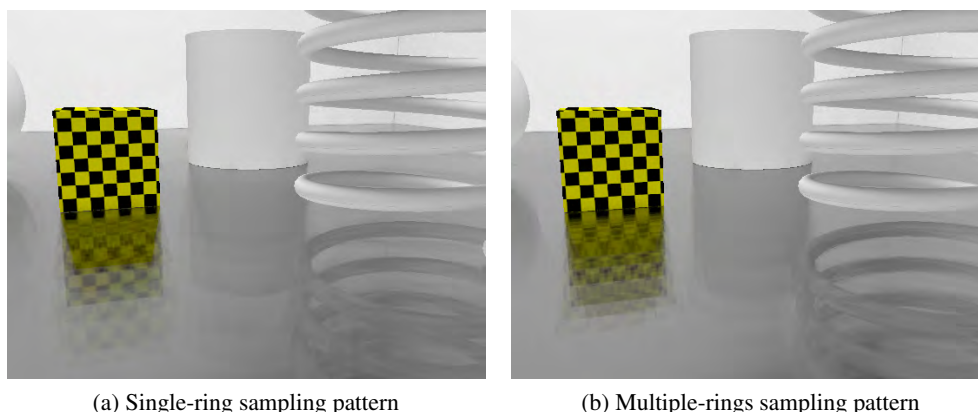


Figure 4.34: The left image (a) shows a rendering of the reflective planar floor using 10 samples and the single-ring pattern (1 center, 9 ring). Filtering is deactivated for a better visualization of the pattern. Note how the (mirror) sample in the lobe center dominates the rendering, as its BSDF is much higher than for the ring samples. Furthermore, the sampling density is unnecessarily high at the low-energy ring of samples, and no samples are taken in-between the ring and the center. In the right image (b), the 9 ring-samples are distributed in three rings with varying radius, of three samples each. The resulting sampling distribution captures the shape and smooth fade-out of the highlight much more effectively.

of the planar reflector, filtering the samples using MIP-maps to obtain a smooth approximation of the specular integral. The involved approximations work best for very narrow specular lobes, as then the underlying assumptions are least violated. Figure 4.35 shows some example renderings obtained with the described real-time approximation technique, more results and a more thorough discussion of problematic cases will follow in Section 4.6.

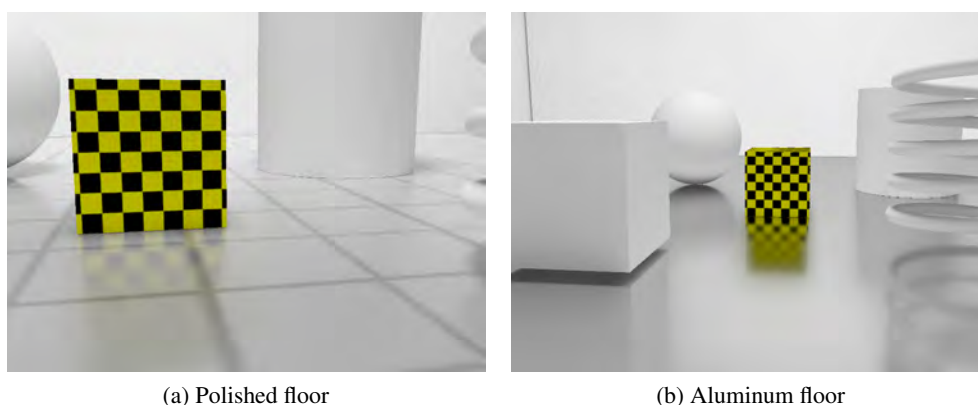


Figure 4.35: Some exemplary results of the planar reflection rendering technique using 16 deterministic samples, obtained at real-time frame rates

4.6 Implementation, results and limitations

In this chapter a regular sampling algorithm was presented that achieves a biased approximation of the specular integral in real-time. This section will briefly expose details on the implementation and testing environment (4.6.1). Then, the performance and visual quality of the method are evaluated for several example materials and different numbers of samples (4.6.2). Problematic cases of both the curved-surface and the planar-surface approach and general limitations of the method are discussed (4.6.3). Finally, rendering results from a typical architectural scene walk-through are shown to validate the applicability of the algorithm to real-world scenarios (4.6.4).

4.6.1 Implementation and testing environment

The employed rendering procedure operates in two steps: first, all view-independent light transport is precomputed off-line for a given static scene. Then, during run-time walkthroughs, the view-dependent integral is approximated using the global illumination input data and the sampling algorithm proposed in this chapter. The off-line computation part is not within the scope of this thesis, but provided by the HILITE-framework: precomputed high-quality light maps and cubic environment maps serve as the irradiance/incident radiance input across the static scene. The computation of the outgoing radiance at each viewed pixel, i.e. sampling and integrating the incident light is entirely performed in a fragment shader program. The shader is essentially a loop computing the weighted sum of all incident radiance samples, approximating the intersected geometry and filter kernel size of each sample as described in the previous sections. Material parameters are set once per object, but theoretically, they could also be provided as textures to cause spatially varying scattering behaviour.

The code for material fitting and editing, scene loading and user input handling is written in Visual Studio using C# and DirectX 10, shaders are written in HLSL. The rendering algorithm performance is measured on a consumer-level PC with an Intel I5-2500 CPU (4x 3.3Ghz), 8GB RAM and NVIDIA G-Force GTS 450 GPU (GPU price $\approx$ EUR 100,- in 2012). The system is later actually going to run on the fastest graphics cards available, thus, real-time performance on consumer level equipment should imply even faster rendering on more powerful systems.

4.6.2 Performance and visual quality

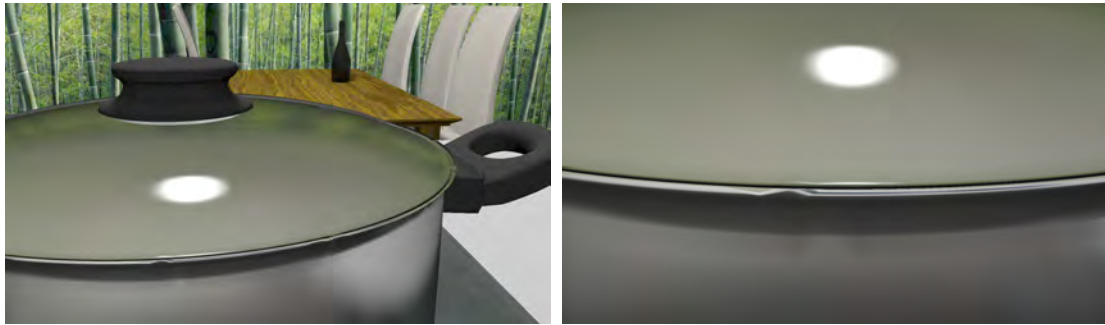
As the method of this thesis is designated for real-time applications, it is crucial that the frame rate never drops below a certain threshold around 20 - 30 fps. Depending on the available computational power and remaining time next to other tasks, the number of illumination samples per pixel must be chosen low enough to guarantee full interactivity. In this subsection the achieved performance for varying numbers of samples is investigated, both for the curved surface- and the planar surface rendering approach. They differ, as the on-the-fly computation of incident light as well as the sampling scheme itself are more expensive in the planar surface rendering method.

Performance-wise it does not matter if n samples are taken within the same lobe or distributed across multiple lobes. It is also irrelevant which material is simulated by the model, as the BSDF will yield different results and different directions will be sampled, but the sampling procedure and thus the performance is exactly the same. Therefore it suffices to give a single performance plot of average frames per seconds for different numbers of samples for each method.

Let's consider the curved surface approach first. When using n samples to shade a pixel, the following steps have to be performed n times per fragment, i.e. for each sample ω_i (for details see Section 4.4:

- The BSDF is evaluated for $(\omega_o|\omega_i)$
- The geometry intersected by the sampling ray is estimated using the bounding plane intersection to obtain a cube map look-up position $\omega_{i'}$
- The area of cube-map pixels covered by the sample $\omega_{i'}$ is determined to estimate an adequate filtering level
- The filtered incident radiance from $\omega_{i'}$ is read, weighted, and added to the estimated view-dependent integral

Then, the view-dependent sum is divided by the sum of weights and combined with the diffuse part to yield the final approximation of the illumination integral. The run-time workload at each pixel is quite high, and linear in the number of samples. As this fragment program is the most expensive part at run-time, the overall performance is determined by the number of shaded fragments in each frame. In typical architectural walkthroughs such as the one in Subsection 4.6.4, around 50 - 100 percent of the frame are filled with specular material pixels. Other large parts like walls, carpets and furniture are often mainly diffuse. The performance chart in Figure 4.38 states two frame rate graphs obtained when rendering a 1680 x 1050 frame completely full of view-dependent pixels, as well as filled approximately half with view-dependent pixels (see Fig. 4.36). During walkthroughs, the actual workload varies somewhere in-between. No culling techniques or other acceleration methods were employed to collect these performance measurements.



(a) Half screen filled with view-dependent pixels

(b) Screen full of view-dependent pixels

Figure 4.36: Performance benchmark scene of the cube map approach

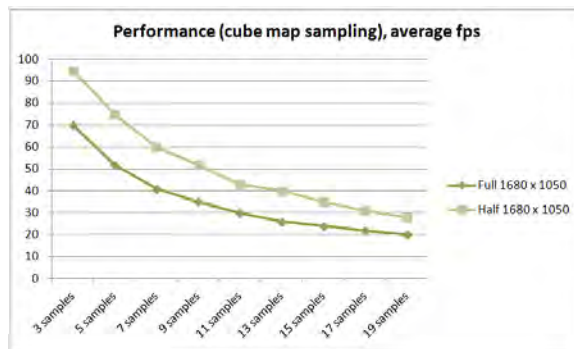


Figure 4.37: Average fps graph

	Full	Half
3 samples	70	95
5 samples	52	75
7 samples	41	60
9 samples	35	52
11 samples	30	43
13 samples	26	40
15 samples	24	35
17 samples	22	31
19 samples	20	28

Table 4.1: Average fps table

Figure 4.38: Two graphs indicating the average performance when rendering a full 1680 x 1050 screen of view-dependent pixels (dark green), and a screen about half specular, half purely diffuse pixels (light green). The average frame rate never drops below 20 fps, even when rendering a full screen using max. 19 samples on a consumer GPU.

The actual number of samples required to obtain a smooth, convincing appearance depends on the roughness of the material, as wide glossy lobes result in a larger integration domain that must be covered by filtered samples. Experimental renderings (such as Figures 4.39 - 4.41 on the following pages) showed that more than 19 samples are never necessary when rendering substances that are between specular and moderately glossy. In fact, materials featuring sharp specular reflection can be rendered using 1-5 filtered samples per pixel, and 7-10 samples suffice for slightly blurry reflection. More than 10 samples are only needed for very glossy substances, for instance the 'chrome'-material in Figure 4.39. Figures 4.39 - 4.41 also depict difference images, comparing the obtained results with a converged importance sampling reference. As a numerical error assessment, PSNR (peak signal to noise ratio) values are stated for each rendering.

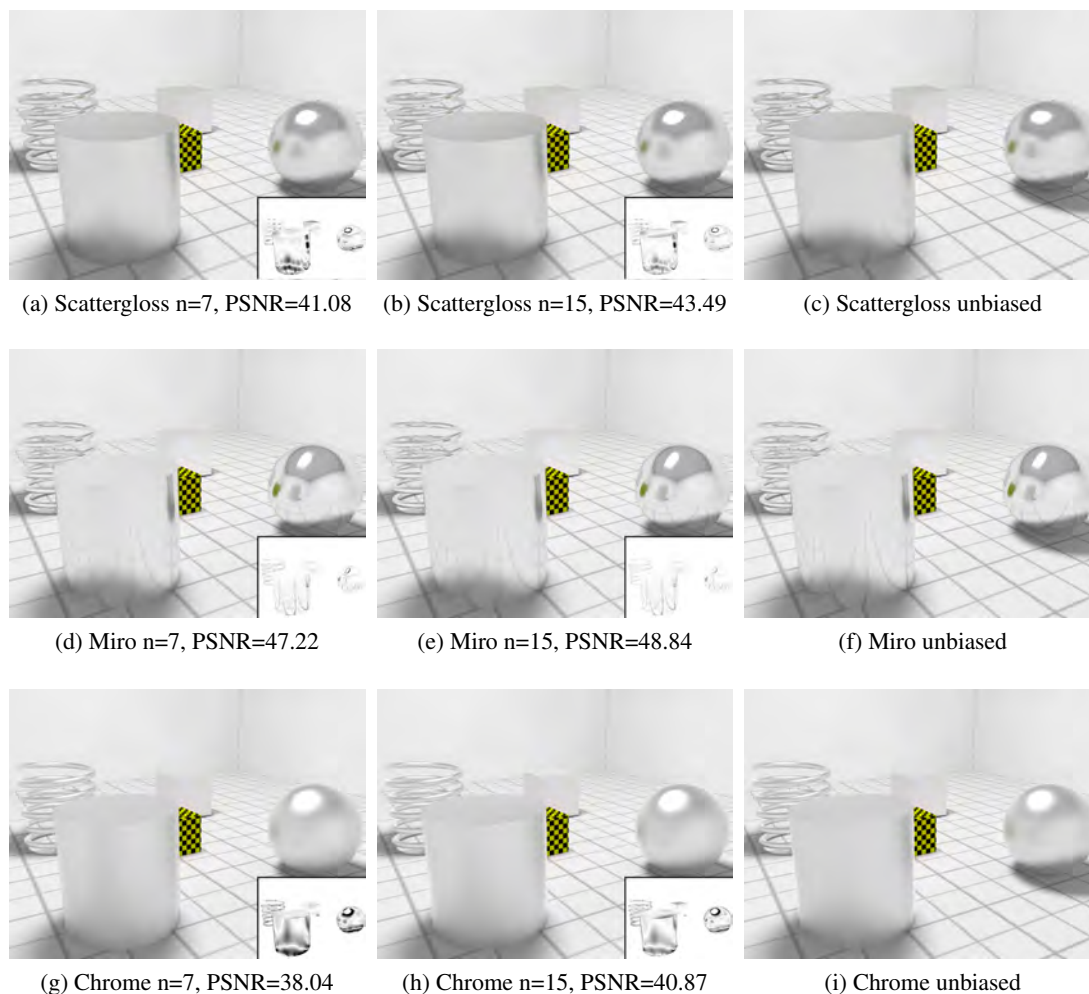


Figure 4.39: Metal rendering results using 7 (left) and 15 (middle) deterministic samples, vs. the unbiased random importance sampling result (right). Note how wide specular lobes result in artifacts caused by undersampling and MIP-mapping when using too few (7) samples. Difference images are multiplied with a factor 15 for better visualization.

Insufficient numbers of samples result in aliasing artifacts that cannot be amended by excessive MIP-map filtering, as evident in the left column of the metal renderings in Fig. 4.39. Such blurry materials require 10 - 15 samples to look convincing, especially when looked at from a close distance. But remember that whenever a viewer decides to take a very close look, camera motion is likely to cease, which results in taking thousands of samples and converging to the unbiased result very quickly, as random importance samples are blended with the previous result.

Figure 4.40 shows partly diffuse / partly specular substances that can be rendered using extremely few samples (3 - 7), as the featured specularity is rather sharp. Such materials enrich

the appearance of a scene compared to a purely diffuse representation at very low additional sampling cost:

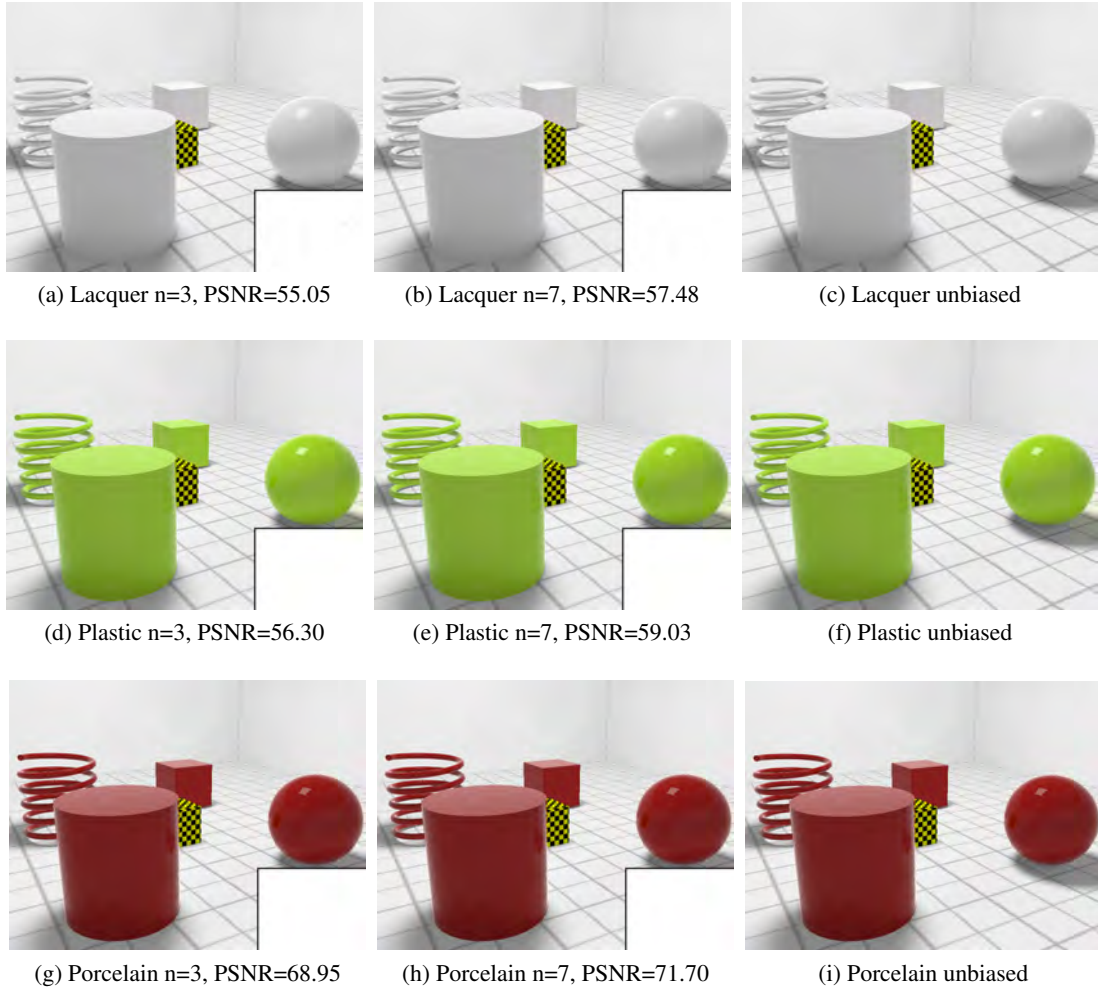


Figure 4.40: Insulator material renderings using 3 (left) and 7 (middle) deterministic samples, vs. the unbiased random importance sampling result (right). Narrow specular lobes require very few samples, resulting in no visual difference if larger numbers of samples are used. Thus, incorporating such substances enriches a scene but has little negative impact on the performance. Difference images are multiplied with a factor 15 for better visualization.

Finally, Figure 4.41 shows example renderings of transmissive substances. The denoted number of samples is split into a half covering reflection and another half covering transmission.

Note how 8 samples (4 per lobe) in Fig. 4.41 already result in a very similar appearance as taking more samples. The strong distortion in the glass rendering (bottom row) is caused by the bounding plane intersection, as all secondary rays are specified to either intersect the floor or the ceiling in this scene, which results in a correctly transmitted floor tiling pattern, but all

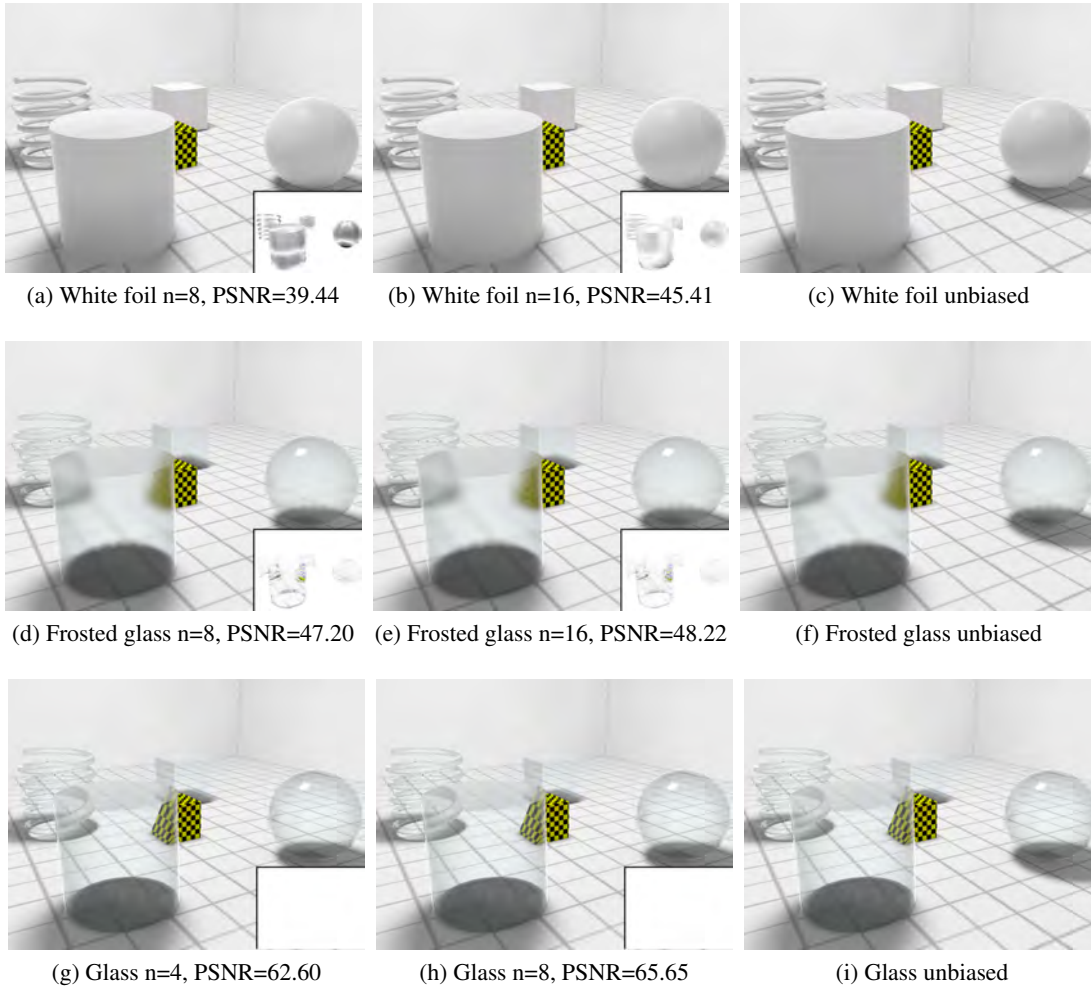


Figure 4.41: Transparent material results using 4 or 8 (left) and 8 or 16 (middle) deterministic samples, vs. the unbiased random importance sampling result (right). The distortion in the glass rendering is due to the bounding plane intersection, as all transmission rays are assumed to intersect the floor. Difference images are multiplied with a factor 15 for better visualization.

objects protruding from the floor get distorted. When rendering the blurry frosted glass (middle row), this artifact is not as disturbing, but when very sharp transmission is involved, transparency could be simulated better by depth-sorting the scene objects and the use of blending. Please note again that *unbiased* in this context means *unbiased sampling*, not unbiased rendering. There are substantial systematic errors in representing the incident light at each scene point, causing resistant bias such as the aforementioned distortion.

The bottom-line of these results is that the number of required samples is proportional to the gloss / blur of the material. Disturbing visible artifacts only occur in very glossy materials, as evident in the difference images in Figures 4.39 - 4.41. View-dependent substances that are

rather specular to moderately glossy can be rendered interactively using 3 to 19 samples at high screen resolution on a standard consumer GPU. The bounding plane intersection trick yields correct secondary illumination when secondary rays actually hit these bounding planes but causes distortion of objects protruding from these planes. The most expensive part of the run-time workload is the fragment shader sampling the incident radiance n times. Thus it helps to draw cheap, purely diffuse surfaces first to reduce potential overdraw of expensive view-dependent fragments by early z-buffering. Generally, the algorithm could be sped up by acceleration techniques that help to avoid drawing unnecessary view-dependent fragments, e.g. occlusion culling.

Finally, Figure 4.42 shows close-up renderings, comparing the quality of the proposed approach to an unbiased reference integration and to the related method of Colbert et al. [Colbert et al. 2007]. As very few samples are taken in our approach, the filtering required to overcome undersampling artifacts is very strong, especially when wide glossy lobes are involved. Thus, the resulting material appearance of the proposed method is a little blurrier than the reference solution. Loss of detail is the price we have to pay for limiting ourselves to very few samples, as they have to be filtered severely to conceal aliasing sufficiently. Colbert et al. use importance sampling to estimate the integrand, causing samples to concentrate near the lobe center. Thus, their filter kernel size is chosen proportionally to the inverse of a sample's probability, as near the lobe center, more samples are likely to be taken and, thus, less filtering is required there. Similar to our approach, they reuse the same sampling pattern across all pixels of an object, but construct the pattern in a quasi-random (QR) fashion. Depending on the actual samples of the QR-sequence, this is *likely* to produce a rather symmetric coverage of the lobe.

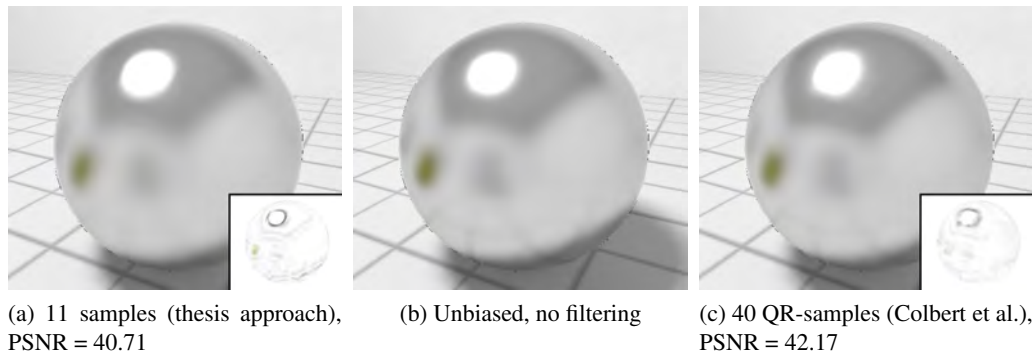


Figure 4.42: Comparison of results achieved using 11 deterministic samples per lobe and filtering as described in Section 4.4 (left image), the unbiased integration using thousands of quasi-random samples and no filtering (middle), and 40 quasi-random samples with pdf-proportional filtering [Colbert et al. 2007] (right). The filtered solutions feature a stronger blur than the reference (e.g. floor tiling), but the overall appearance is similar. The PSNR of the right image is higher than the one of the left, but so is the number of samples. Difference images are scaled by a factor 15 for a better visualization.

Next, some results of the planar surface rendering approach are investigated. Essentially, the performance is expected to be similar but a little worse than the cube map sampling approach, as

the approximate incident light representation (mirror buffer) is updated every frame, and samples have to be projected into the domain of the incident light buffer. After rendering the scene each frame from a mirrored point of view first, the contents of this mirror buffer have to be MIP-mapped, i.e. an image pyramid must be constructed that allows dynamic filtering of samples later. These additional tasks degrade the performance, reducing the allowed sample budget when maintaining high frame rates. Figure 4.44 shows again two frame rate graphs, one for rendering a full screen filled completely and one for the screen filled half with view-dependent pixels of a planar floor:

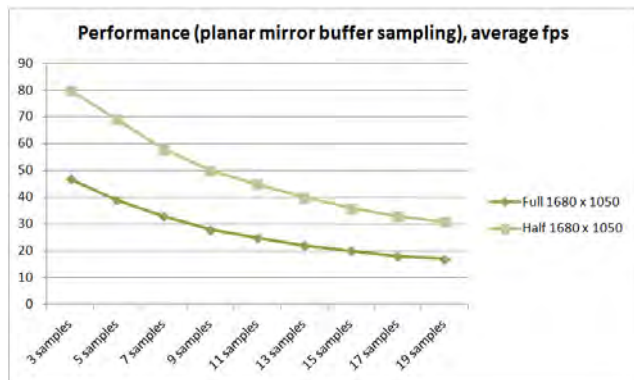


Figure 4.43: Average fps graph

	Full	Half
3 samples	47	80
5 samples	39	69
7 samples	33	58
9 samples	28	50
11 samples	25	45
13 samples	22	40
15 samples	20	36
17 samples	18	33
19 samples	17	31

Table 4.2: Average fps table

Figure 4.44: Two graphs indicating the average performance when rendering a full 1680 x 1050 screen of view-dependent pixels (dark green), and a screen about half specular, half purely diffuse pixels (light green), which is more probable in most architectural indoor scenes. Note how the performance stays interactive even when taking 10 - 19 samples, but the achieved rates are slightly worse than for the curved surface approach, due to the additional workload of rendering the scene into a mirror buffer first, and the more extensive sampling scheme.

The performance test depicted in Figure 4.44 involves rendering a single planar reflector, i.e. a single additional mirror rendering pass each frame. This pass is usually inexpensive if only the relevant (potentially reflected) scene parts are drawn, and only diffusely shaded. Still, the resulting buffer contents have to be MIP-mapped in each frame to allow filtered look-ups during shading. As additional planar reflectors require additional MIP-mapped mirror rendering passes in each frame, the performance is further degraded when rendering multiple view-dependent planes.

As evident in Figure 4.44, smooth real-time walkthroughs can be achieved when using around 3 - 10 samples. This means that there is a material blurriness limit entailed by the sampling budget, as strong filtering of wide lobes only looks convincing at higher sampling rates. However, as presented in Section 4.5, the approximative incident light representation is only valid for narrow specular lobes anyway, as it correctly captures incident radiance from one direction at each pixel of the plane. Thus, the planar reflection method works best for rather specular surfaces for two reasons, performance and correctness.

The planar reflection rendering approach achieves convincing results only for sufficiently sampled lobes. Images feature motion parallax and distance-proportional blurring of the reflected scene, as shown in Figure 4.45. But as lobes get wider and materials get glossier, sampling, filtering and light representation are insufficient to allow results that are visually equivalent to the unbiased reference integration.

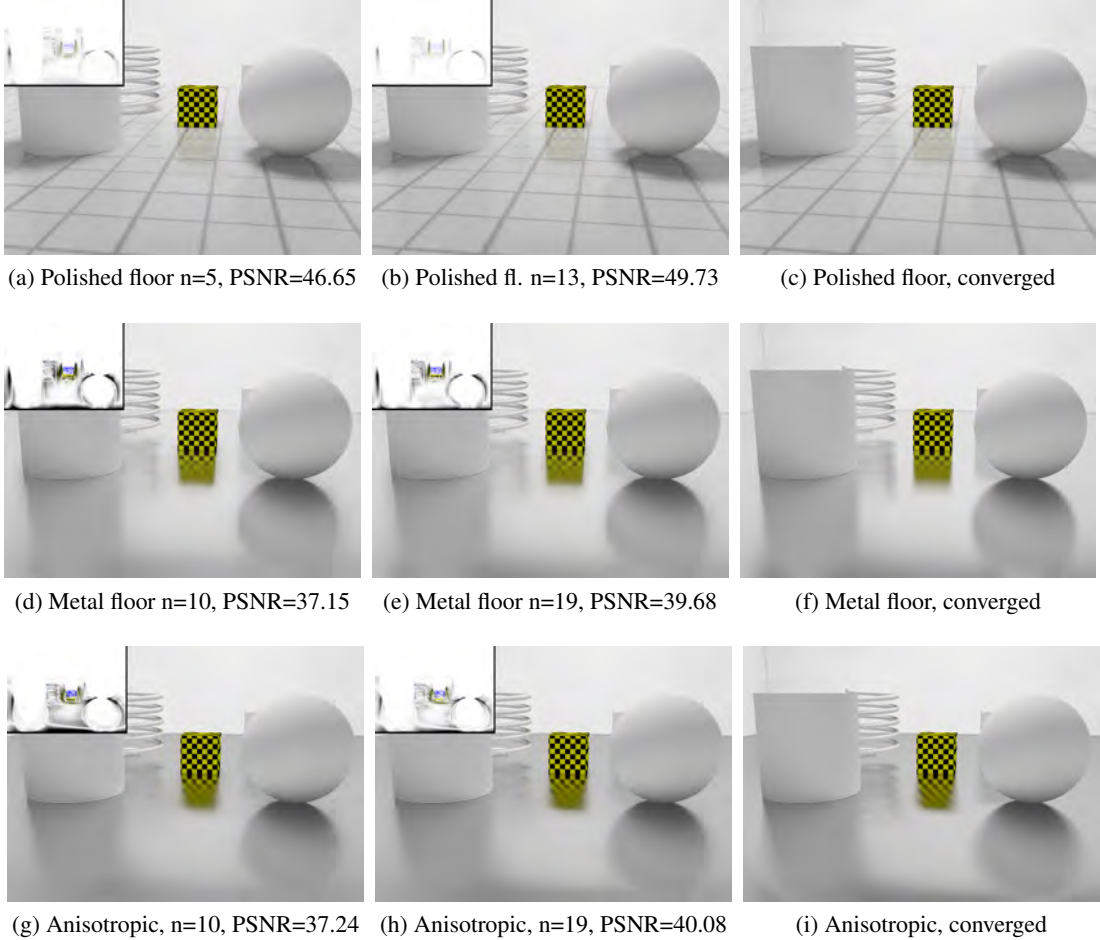


Figure 4.45: Example renderings of the planar surface rendering approach. Note how the blur increases with the distance between reflector and reflected object. The method works best for materials featuring sharp specular reflection, as very few samples produce convincing results then (top row). The glossier the lobe gets, the more error-prone the incident light representation becomes. Furthermore, larger numbers of samples are required to reproduce a smooth glossy blur (middle and bottom row). The right column shows the reference images obtained by quasi-random importance sampling the specular scattering integral using thousands of samples over consecutive frames. The difference images are scaled by a factor 15 for a better visualization.

Some glossy materials that can be rendered well by the curved surface approach feature

lobes that are too wide to be rendered smoothly using the planar method. Figure 4.46a shows a rendering of a plane made from the measured blurry Alanod aluminum. Aside from the under-sampling artifacts below the yellow cube and the sphere, another problem becomes evident in this figure: as the filter kernel size is not determined differently for each sample, but only once per pixel by projecting the lobe's extents to the pixel's reflected tangent plane, there are blurring discontinuities at depth edges of the mirror buffer. This suddenly changing blur filter size manifests as visible contours of perfectly mirrored objects, such as the faint but sharp cylinder or yellow cube shape around the otherwise blurry reflection. During the progressive refinement when the view is still, this can be amended by averaging the found tangent plane coordinates across all previously found samples at a pixel before computing the next sample, emulating a smooth transition, as shown in Figure 4.46b. This trick is also what causes the softly fading edges in the right column of Fig. 4.45. During walkthroughs, such soft transitions cannot be simulated using 3-19 samples. Furthermore, as MIP-mapping is just an approximation of sums across arbitrary pixel-subsets of the mirror buffer, artifacts such as the blocky filter kernel shape or flickering during motion are sometimes noticeable at stronger filtering levels using few samples. Summed-area tables [Crow 1984] might help to amend these issues, but are probably not as fast to compute every frame, as MIP-maps are implemented already in hardware. Still, this might be an extension worth investigating in the future.

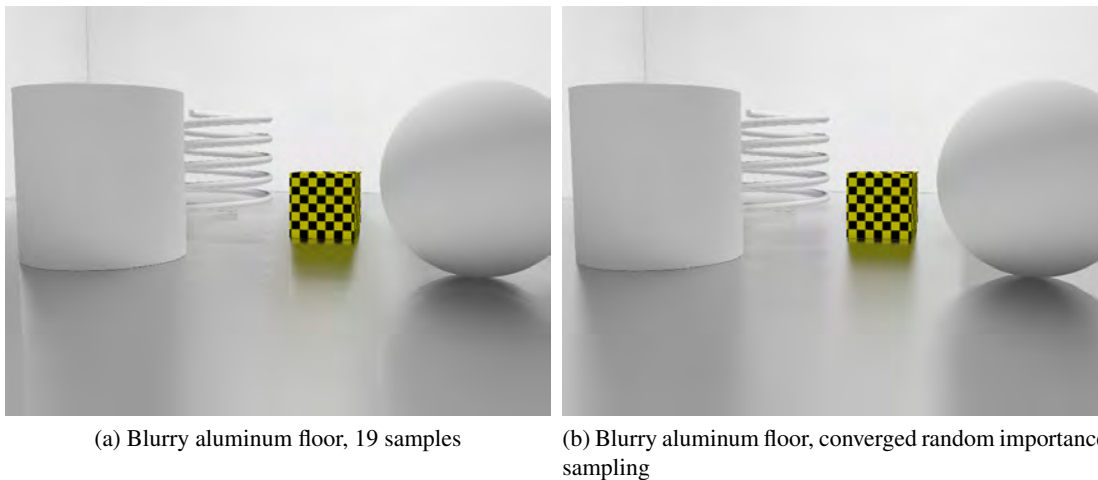


Figure 4.46: Renderings of a very glossy Alanod aluminum floor. Due to the relatively low number of samples, undersampling / aliasing artifacts are visible in the left image that are not amended by excessive MIP-map filtering. Furthermore, discontinuities of the filter kernel size are visible at depth contours, causing the blurry reflections to be framed by faint sharp discontinuities. In the right image, thousands of random importance samples were taken over consecutive frames. To amend the filtering discontinuity, the tangent planes in which the reflected geometry is assumed are averaged across all previously taken samples at each pixel, resulting in smooth transitions of filter level and reflection.

The bottom-line of these planar rendering results is, that again, sharp reflections caused by

narrow lobes can be rendered efficiently, as the integration domain is smaller and the incident light representation is more likely to be valid then. For wide lobes it is not certain that the reflected geometry is entirely visible in the mirror buffer, and also, the sampling scheme requires more samples to produce smooth results. To allow MIP-map filtering, the contents of the updated planar incident radiance buffer must be MIP-mapped every frame, limiting the allowed number of planar reflectors to very few, if high frame rates are to be maintained. In summary, the applicability of the planar approach is more limited than the curved surface approach, but for the particular – not so rare – case of partial sharp specular reflection in planes, the gained visual impact is worth its while (e.g. parquet floors, rubber floors, tiles, varnished tables, wet floors etc.).

4.6.3 Limitations and future work

Before pointing out general limitations of the proposed methods, let's consider what *can* be achieved using the presented approach. First, keep in mind that the entire view-independent calculation is independent of the presented sampling algorithm. Thus, arbitrarily expensive baked global illumination can be displayed without additional cost during walkthroughs, such as high-fidelity shadows, color bleedings and caustics. These effects are just as crucial to a visually impressive scene presentation as the last view-dependent bounce.

Using the high-quality illumination input data, the proposed algorithm then simulates the last bounce of light propagation by sampling approximative incident radiance representations at each rendered scene point. The involved scattering integral can be evaluated for arbitrary materials whose scattering behaviour is modeled by lobes around the perfect reflection direction. Other reflectance behaviour such as retro-reflection is not covered per se by the employed material model. Different models can be used to simulate different effects, as long as they allow the derivation of similarly efficient deterministic sampling schemes.

Regarding the performance vs. quality tradeoff of the sampling, a central observation is that the smaller the involved integration domain is, the better the view-dependent integral can be approximated using small numbers of samples. Materials featuring narrow lobes can be simulated convincingly at very low rendering cost using either of the proposed methods. Large integration domains caused by wide glossy lobes require much more samples to be covered smoothly, as the imperfection of the involved MIP-map filtering becomes particularly noticeable at stronger blurring levels. More sophisticated GPU-filtering techniques such as a hardware implementation of summed-area tables [Crow 1984] would allow faster approximative integration across larger domains at better visual quality. Meanwhile, a future amendment to render very glossy materials using the cube map sampling approach could be prefiltering the static cube map with a filter of a reasonable minimum kernel size across all possible incoming/outgoing directions. At run-time, the integration would then require fewer look-ups and less excessive MIP-map filtering to achieve a similarly blurred appearance, featuring fewer artifacts and improved performance. But for now, very glossy view-dependent materials pose a problematic case for real-time integration using few samples.

To overcome the problem of considering only the direction (but not the origin) of secondary rays in environment map shading, a bounding plane intersection step was proposed to estimate the position of the reflected geometry, obtaining a better look-up position. This results in correct,

ray-tracing-like reflection when the reflected objects actually coincide with the tested planes, but causes considerable distortion in the reflection of objects protruding from these planes. Whenever the distortion becomes noticeable, one can either specify more planes to test against or disable the plane-intersection step completely, getting rid of local distortions at the price of a globally incorrect reflection, as the environment is then assumed to be infinitely distant. Distortions are particularly disturbing when reflections are sharp and clear, but as soon as the material gets glossier or the reflector's geometry is very curved, distortions are much less noticeable. More sophisticated estimations of the reflected geometry such as *distance impostors* [Szirmay-Kalos et al. 2005] sound promising, but might degrade the walkthrough performance significantly, as the involved computations have to be performed for each sample.

Approximating the incident radiance of an entire object by a single environment map works best for objects that are small with respect to the distance of their reflected surroundings. When a cube map is rendered from the mass center of an object, the object itself has to be omitted as not to occlude the environment. This makes rendering self-interreflections of concave objects impossible. An option to overcome this would be splitting concave objects into multiple convex parts, rendering a cube map for each part and omitting only the respective part from the cube map rendering. More sophisticated methods such as storing compressed *radiance transfer functions* (including visibility and interreflection) at each vertex as in PRT [Sloan et al. 2002] would yield better quality, but again, cause a significant performance hit, as the involved tests would have to be performed once per sample.

The number of illumination samples per pixel is currently chosen by the user and set as a uniform parameter for each object. As the required number for a smooth appearance mainly depends on the size of the integration domain (material roughness) and surface curvature (as high curvature makes artifacts hard to notice), this number could theoretically be estimated automatically. Just as the material parameters can theoretically be stored in textures to achieve spatially varying reflectance, a varying number of samples could also be determined and stored in an extra texture channel.

Finally, another limitation of the current implementation is that only the last bounce is view-dependent, and all representations of incident light only contain diffuse, view-independent light information. A metallic object would thus look totally diffuse when looked at through a mirror. As stated in Section 1.3, directly rendering global illumination into the cube maps would overcome this limitation for the cube map approach, since the captured environment would then feature static specular objects as seen from the cube map center. Fully updating the incident radiance representations according to the view in each frame is not feasible in real-time. Obtaining the planar mirror buffer using the full-fledged specular approach would theoretically be possible, but it would cut down the frame rate by fifty percent, as a second expensive frame would have to be computed before rendering each actual frame. On the other hand, multiple view-dependent bounces could be employed when the camera is still by recursively updating the involved incident light representations (see e.g. [Nielsen and Christensen 2002]). As soon as the camera starts to move, the view-independent representations would have to be employed again.

4.6.4 Exemplary architectural scene walkthrough

This final subsection shows screenshots from an exemplary walkthrough of an architectural scene. The baked global illumination information was precomputed externally, material parameters were specified by hand. The floor is the only planar reflector rendered by the planar approach, couches, pillows, the fruit bowl, tv screen, vases, sink, tap, pot, balance and champagne bottle are rendered using the cube map sampling approach. All remaining furniture, walls and ceiling are rendered purely diffuse. The achieved frame rate during the walkthrough is between 30 and 40 fps. Most view-dependent objects are rendered using around 7 samples.

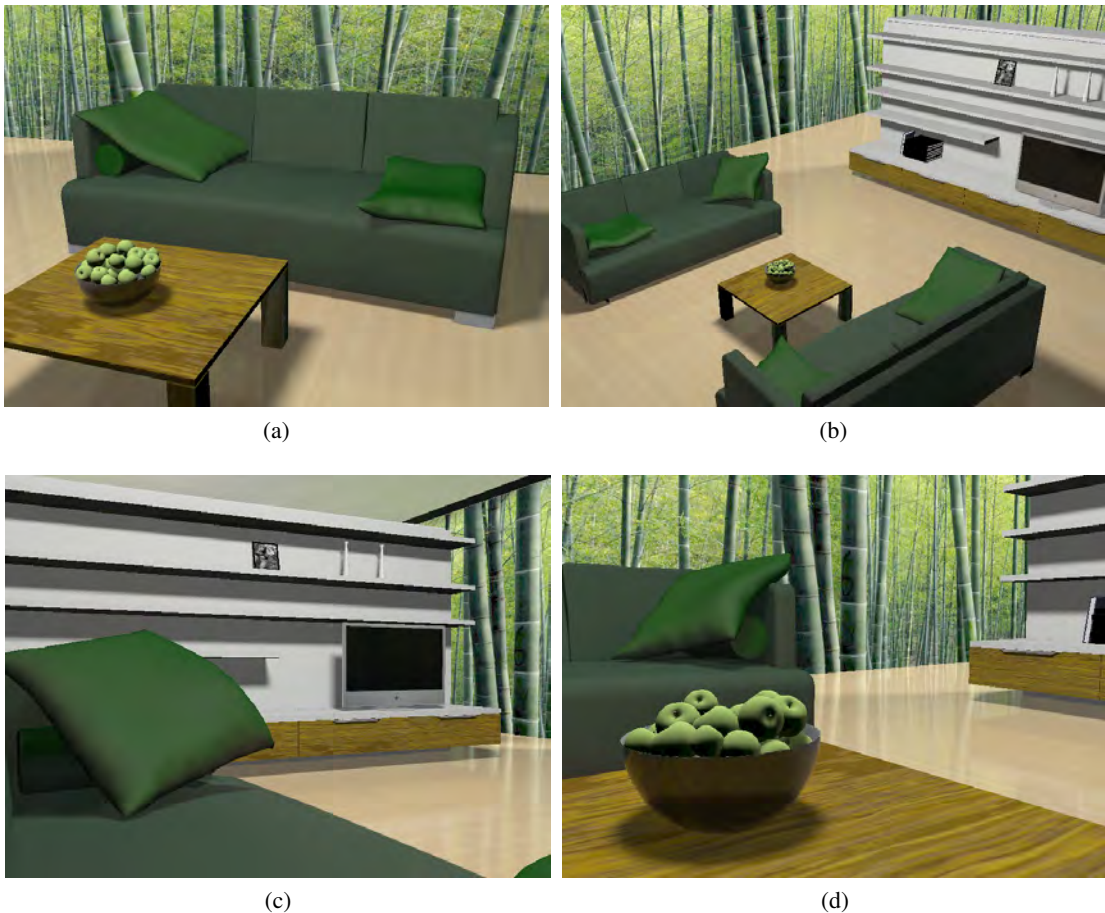


Figure 4.47: Exemplary renderings of an apartment using the proposed algorithms, part 1/3



(a)



(b)



(c)



(d)

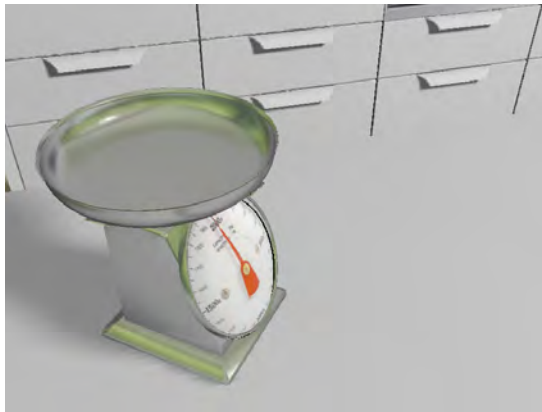
Figure 4.48: Exemplary renderings of an apartment using the proposed algorithms, part 2/3



(a)



(b)



(c)



(d)

Figure 4.49: Exemplary renderings of an apartment using the proposed algorithms, part 3/3

Conclusion

In this thesis an algorithm was presented that achieves the rendering of view-dependent materials in real-time. Scene points are rendered by approximatively integrating the incident light according to the local material properties.

Scattering properties of materials (BSDFs) are represented using a parametric model based on the Kurt-Szirmay-Kalos-Krivanek BRDF [Kurt et al. 2010], which is extended in this thesis to additionally allow representing transmission distributions. Due to its large number of parameters, the model's expressivity makes it particularly suited for fitting to measured material data.

Besides the material representation, the algorithm also requires a representation of the incident light distribution at each rendered scene point. As the actual immediate application of the method is rendering *static* scenes for architectural indoor walkthroughs, the incident light representation does not change over time. Thus, multiple bounces of light propagation are pre-computed in an advance step and stored as diffuse irradiance light maps. From this precomputed illumination, an environment map of incident radiance is calculated and stored for each scene object. Then, the environment map is used as an approximative incident light representation across all points of that object.

An optional intersection step of secondary rays with bounding planes of the scene is proposed to estimate the actual reflected geometry, thus obtaining a better approximation of the actual incident radiance from the environment map. This estimation works well when the reflected scene objects do not differ significantly from the intersected bounding planes, which is likely in indoor scenes, as planes are common in architecture. The environment maps are then reused as approximative omnidirectional representations of incident radiance across all scene points of an object.

Large planar reflectors such as floors or walls are one of the most problematic cases for environment map shading, as the incident radiance varies significantly across such objects. Thus a different representation of incident light is used for rendering large planar objects: each frame, the scene is rendered into a mirror buffer as seen from a mirrored point of view. This buffer then contains the correct incident radiance of each point in the plane, however only for the

perfect mirror direction. For rather sharp specular reflection it is likely that *neighboring pixels* in the buffer contain exitant radiances that are similar to the incident radiances from *neighboring directions of incidence*. Thus, the buffer allows approximative incident radiance queries also for slightly different directions than the correctly captured perfect mirror directions.

With these representations at hand, the specular part of the outgoing radiance is computed at each pixel by integrating the incident light according to the surface's material properties. The diffuse part of the outgoing radiance is directly derived from the precomputed irradiance information. The specular integral is approximated during walkthroughs by sampling the incident light representation and weighting the samples according to the material. The employed angular sampling pattern is deterministic and exactly the same at each surface point. As proposed by Colbert et al. [Colbert et al. 2007], hardware filtering of the incident light representation is employed to counteract aliasing artifacts induced by the low sampling (5-20 samples per pixel). All samples are exclusively taken within the specular lobe, which introduces bias but benefits performance, as the lobe is the most relevant part of the specular integral's domain. As the samples are placed symmetrically and equidistantly within the lobe, it is covered uniformly even when taking very few samples. Due to this strategic placement, the required number of samples to obtain a smooth, uniform lobe coverage is lower than for existing (quasi-)random sampling methods. The integration bias caused by ignoring incident radiance from the border of the lobe and beyond is negligible next to the employed simplifications of the incident lighting.

Incident radiance samples are filtered using MIP-maps of the incident light representations, such that the union of samples densely covers the angular domain of the lobe. MIP-mapping is a cheaper alternative to taking more samples, unfortunately causing visible artifacts at stronger filtering levels. The glossier the lobe, the larger the integration domain, and thus, the stronger the required filtering level. Increasing the number of samples reduces the required filtering, visually improving the result but also degrading the performance. Thus, very glossy materials are one of the main limitations of the proposed methods, as the numbers of samples required for a smooth, pleasing result are not feasible in real-time.

In the proposed planar reflection rendering approach, filtering limits are not the worst issue caused by very glossy lobes. The assumption that neighboring pixels of the mirror buffer contain valid incident radiances from similar directions is increasingly likely to be violated farther off the perfect mirror direction. Thus, the maximum gloss limit is even smaller for the planar rendering approach than for the environment map sampling method.

Both rendering methods turned out to produce best results for very specular to moderately glossy materials, as very few samples then suffice to capture the involved lobe adequately, without the need of strong filtering. Specular metals, plastics or dielectrics can thus be rendered very fast using 3-7 samples while contributing significantly to the visual quality of the scene.

The correctness of the rendered scene illumination can be arbitrarily high during the off-line precomputation phase, qualifying the overall diffuse light distribution for prototyping purposes. However, the view-dependent overlay produced by the proposed method can never be physically correct as it involves substantial approximations in representing the incident light at each scene point, and only approximates the specular integral during walkthroughs. It nevertheless enriches the visual quality of the scene, producing an appearance that humans would qualify as more realistic than a purely diffuse rendering.

The performance of the method is superior to existing (quasi-)random importance sampling methods, as the proposed sampling scheme achieves a more uniform coverage of the specular lobe using fewer samples. The gained performance benefits interactivity or could alternatively be spent on other tasks at run-time.

On the whole, the method achieves a convincing appearance of important view-dependent material classes in real-time, thus improving scene immersion and enhancing the visual quality of interactive scene walkthroughs.

List of Figures

1.1	From measurement to rendering	1
1.2	Teaser: scenes rendered with the proposed algorithm	2
1.3	Teaser: visually enriched rendering due to view-dependent materials	4
1.4	Teaser: intersecting secondary rays with bounding planes	7
2.1	Spherical coordinates and solid angles [Szirmay-Kalos 2008]	10
2.2	Plot and rendering of a diffuse BRDF	14
2.3	Plot and rendering of specular BRDFs	15
2.4	Varying Fresnel reflectance of some materials [Akenine-Möller et al. 2008]	16
2.5	Rendering of materials featuring Fresnel effects	16
2.6	Measured scattering distribution of 'scattergloss aluminum'	18
2.7	Random vs. quasi-random numbers [Szirmay-Kalos 2008])	20
3.1	Reconstructing nonmeasured probabilities by linear interpolation	27
3.2	Reconstructing nonmeasured probabilities by halfway-vector reconstruction	28
3.3	Shadowing/masking effects of microfacets [Blinn 1977]	31
3.4	Renderings with different values of the α -parameter	31
3.5	Transmission geometry of the BTDF model	32
3.6	Evaluating the transmission lobe of the BSDF model	33
3.7	Renderings of transmissive materials (glass) with increasing degree of roughness	34
3.8	Hierarchical brute-force model fitting	36
3.9	2D-slices through the 5D scalar field of fitting errors	37
3.10	Renderings of fitted models for three test datasets	38
3.11	Transfer probability measurements and model fits of the 'scattergloss aluminum' dataset	39
3.12	Transfer probability plots of varying θ_i in the plane of incidence for the 'scattergloss aluminum' dataset	40
3.13	Rendering results of the 'scattergloss aluminum' models using 1 or 2 lobes	41
3.14	Transfer probability measurements and model fits of the white foil dataset	42
3.15	Transfer probability plots of varying θ_i in the plane of incidence for the white foil dataset	43
3.16	Rendering results of the white foil models using 1 or 2 lobes	43
3.17	Transfer probability measurements and model fits of the 'Miro 5 aluminum' dataset	44

3.18	Transfer probability plots of varying θ_i in the plane of incidence for the 'Miro 5 aluminum' dataset	45
3.19	Rendering results of the 'Miro 5 aluminum' models using 1 or 2 lobes	46
4.1	Algorithm overview	56
4.2	Diffuse illumination rendered using light maps	58
4.3	Correct vs. approximated representation of incident lighting	60
4.4	Standard cubic environment map look-up	61
4.5	Bounding plane intersection environment map look-up	62
4.6	Bounding plane intersection artifact	63
4.7	Illustrations of uniform sampling and importance sampling	65
4.8	Uniform vs. importance sampling renderings, taken from Pharr and Humphreys [Pharr and Humphreys 2004]	65
4.9	Illustration of θ_h, ϕ_h	66
4.10	Noise in (Quasi-)Monte-Carlo importance sampling	68
4.11	Random vs. deterministic sampling renderings [Colbert et al. 2007]	69
4.12	Equally spaced samples vs. Quasi-Monte-Carlo placement	71
4.13	Plot of the non-linear importance transformation for θ_h	72
4.14	Linear approximation of the importance transformation for θ_h	73
4.15	Distortion effect of the transformation formula of ϕ_h	74
4.16	Uniformity-preserving transformation formulae	74
4.17	Equally spaced samples for isotropic and anisotropic reflection	75
4.18	Mapping structure of the modified transformation formulae	76
4.19	Coverage of the lobe using MIP-mapped samples	78
4.20	Deriving the MIP-map level for a sample	78
4.21	Filtering results	80
4.22	Illustration of the importance pdf p and the clamped pdf p_{PT}	81
4.23	Visible horizon lines due to the correlated sampling pattern	83
4.24	Material renderings produced by the regular sampling approach	84
4.25	Illustration of the error when a single CEM is reused for large objects	86
4.26	Illustration of the mirror rendering	86
4.27	The mirror-buffer captures incident light from a single direction	87
4.28	Obtaining the mirror-buffer look-up position for a directional sample	88
4.29	Tangent plane assumption for radiance samples	89
4.30	Tangent plane discontinuity illustration	90
4.31	Rendering with vs. without the additional plane intersection step	92
4.32	Quasi-Monte-Carlo renderings of planar reflectors	92
4.33	Single-ring vs. multiple-rings sampling pattern illustration	94
4.34	Rendering using the single-ring vs. multiple-rings sampling pattern	95
4.35	Results of the planar reflection rendering	95
4.36	Performance benchmark scene of the cube map approach	98
4.37	Average performance (fps) of the cube map sampling approach	98
4.38	Performance evaluation of the cube map sampling algorithm	98

4.39	Curved surface rendering results (metals)	99
4.40	Curved surface rendering results (insulators)	100
4.41	Curved surface rendering results (transparent)	101
4.42	Close-up rendering results	102
4.43	Average performance (fps) of the planar sampling approach	103
4.44	Performance evaluation of the planar mirror buffer sampling algorithm	103
4.45	Planar surface rendering results	104
4.46	Comparison of the real-time sampling and converged random importance sampling of a very blurry planar reflector	105
4.47	Exemplary renderings of an apartment using the proposed algorithms, part 1/3 . . .	108
4.48	Exemplary renderings of an apartment using the proposed algorithms, part 2/3 . . .	109
4.49	Exemplary renderings of an apartment using the proposed algorithms, part 3/3 . . .	110

List of Tables

3.1	Fitting errors of the Alanod 2000GS scattergloss aluminum dataset	39
3.2	Fitting errors of the white foil dataset	43
3.3	Fitting errors of the Miro 5 brushed aluminum dataset	46
4.1	Average performance (fps) of the cube map sampling approach	98
4.2	Average performance (fps) of the planar sampling approach	103

Bibliography

- [Akenine-Möller et al. 2008] Akenine-Möller, T., Haines, E., and Hoffman, N. 2008. Real-Time Rendering 3rd Edition. published by A. K. Peters, Ltd. ISBN: 987-1-56881-424-7, 1045 pages, Natick, MA, USA
- [Arvo and Kirk 1990] Arvo, J., Kirk, D. 1990. Particle Transport and Image Synthesis. In *Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 1990)* ACM, New York, NY, USA, pages 63 - 66
- [Bastos et al. 1999] Bastos, R., Hoff, K., Wynn, W. Lastra, A. 1999. Increased Photorealism for Interactive Architectural Walkthroughs. In *Proceedings of the 1999 Symposium on Interactive 3D Graphics* ACM, New York, NY, USA, pages 183 - 190
- [Blinn and Newell 1976] Blinn, J. F., and Newell, M. E. 1976. Texture and Reflection in Computer-Generated Images. In *Proceedings of SIGGRAPH 1976* ACM, New York, NY, USA, page 266
- [Blinn 1977] Blinn, J. F. 1977. Models of Light Reflection for Computer-Synthesized Pictures. In *Proceedings of SIGGRAPH 1977* Vol. 11, Nr. 2, pages 192-198
- [Cabral et al. 1987] Cabral, B., Nelson, M., Springmeyer, R. 1987. Bidirectional Reflection Functions from Surface Bump Maps. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 1987)* ACM, New York, NY, USA, pages 273-281, August 1987
- [Cohen et al. 1993] Cohen, M. F., Wallace, J., Hanrahan, P. 1993. Radiosity and Realistic Image Synthesis. Academic Press Professional Inc. San Diego, CA, USA, 1993
- [Colbert et al. 2007] Colbert M., Krivanek J. 2007. Real-time Shading with Filtered Importance Sampling. In *ACM SIGGRAPH 2007 sketches (SIGGRAPH '07)*, ACM, New York, NY, USA, Article 71
- [Cook and Torrance 1981] Cook, R. L. and Torrance, K. E. 1981. A Reflectance Model for Computer Graphics. In *Proceedings of the 8th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 1981)* ACM, New York, NY, USA, pages 307-316

- [Crow 1984] Crow, F. C. 1984. Summed-area Tables for Texture Mapping. In *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 1984)* ACM, New York, NY, USA, pages 207-212
- [Diefenbach et al. 1997] Diefenbach P. J., Badler N. I. 1997. Multi-pass Pipeline Rendering: Realism for Dynamic Environments. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, Providence, Rhode Island, USA, pages 59-70
- [Fuhrmann et al. 2004] Fuhrmann A. L., Tobler R. F., Maierhofer S. 2004. Real-time Glossy Reflections on Planar Surfaces. In *Proceedings of the 3rd International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa (AFRIGRAPH '04)*, ACM, New York, NY, USA, pages 87-91
- [Goral et al. 1984] Goral, C. M., Torrance, K. E., Greenberg, D. P., Battaile, B. 1984 Modeling the Interaction of Light between Diffuse Surfaces. In *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1984* ACM, New York, NY, USA, pages 213-222
- [Green 2003] Green, R. 2003 Spherical Harmonic Lighting: The Gritty Details. In *Archives of the Game Developers Conference* March 2003
- [Greene 1986] Greene, N. 1986. Environment Mapping and other Applications of World Projections. In *IEEE Computer Graphics and Applications*, Vol. 6, Nr. 11, Los Alamitos, CA, USA, pages 21-29
- [He et al. 1991] He X. D., Torrance K. E., Sillion F. X., Greenberg D. P. 1991. A Comprehensive Physical Model for Light Reflection. In *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 1991)* pages 175 - 186
- [Heckbert 1991] Heckbert, P. S. 1991. Simulating Global Illumination Using Adaptive Meshing. Ph.D. Thesis at the University of Berkeley, CA 94720 UCB/CSD 91/636, CS Division, U.C. Berkeley, May 1991
- [Heidrich and Seidel 1999] Heidrich, W., Seidel, H. P. 1999. Realistic Hardware-Accelerated Shading and Lighting. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 1999)* ACM, New York, NY, USA, pages 171-178, August 1999
- [Jensen 1996] Jensen, H. 1996. Global Illumination using Photon Maps. In *Proceedings of the Eurographics Workshop on Rendering Techniques '96* pages 21-30, December 1996, Porto, Portugal
- [Kajiya 1986] Kajiya, J. T. 1986. The Rendering Equation. in *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1986* Vol. 20, pages 143-150, New York, USA, August 1986

- [Kautz and McCool 1999] Kautz, J., and McCool, M. 1999. Interactive Rendering with Arbitrary BRDFs using Separable Approximations. In *Proceedings of the 10th Eurographics Workshop on Rendering* pages 281-292
- [Kautz and McCool 2000] Kautz, J., and McCool, M. 2000. Approximation of Glossy Reflection with Prefiltered Environment Maps. In *Graphics Interface* pages 119-126, May 2000
- [Kautz et al. 2000] Kautz, J., Vasquez, P. P., Heidrich, W., Seidel, H. P. 2000. A Unified Approach to Prefiltered Environment Maps. In *Proceedings of the 11th Eurographics Workshop on Rendering* pages 185-196, June 2000
- [Kilgard 1999] Kilgard, M. J. 1999. Improving Shadows and Reflections via the Stencil Buffer. Technical Report, Nvidia Corp. 1999
- [Koenderink and van Doorn, 1998] Koenderink, J., van Doorn, A. 1998. Phenomenological Description of Bidirectional Surface Reflection. In *Journal of the Optical Society of America A* Vol. 15, Nr. 11, pages 2903-2912, 1998
- [Kurt and Edwards, 2009] Kurt, M., Edwards, D. 2009. A Survey of BRDF Models for Computer Graphics. In *ACM SIGGRAPH Computer Graphics, Building Bridges - Science, the Arts and Technology Homepage Archive* Vol. 43, Issue 2, Article Nr. 4, ACM New York, NY, USA, May 2009
- [Kurt et al. 2010] Kurt, M., Szirmay-Kalos L., Krivanek, J. 2010. An Anisotropic BRDF Model for Fitting and Monte-Carlo Rendering. In *ACM SIGGRAPH Computer Graphics*, Vol. 44, Nr. 1, Article 3 (February 2010), pages 1-15
- [Lafortune et al. 1997] Lafortune, E. P. F., Foo, S. C., Torrance, K. E., Greenberg, D. P. 1997. Non-linear Approximation of Reflectance Functions. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1997* Los Angeles, CA, USA, pages 117-126
- [Lambert 1760] Lambert, J. H. 1760. *Photometria sive de mensura de gratibus luminis, colorum umbrae*. Eberhard Klett
- [Lawrence et al. 2006] Lawrence, J., Ben-Artzi, A., DeCoro, C., Matusik, W., Pfister, H., Ramamoorthi, R. and Rusinkiewicz, S. 2006. Inverse Shade Trees for Non-Parametric Material Representation and Editing. In *ACM Transactions on Graphics (Proc. SIGGRAPH)* Vol. 25, Nr. 3, July 2006, Boston, MA, USA, pages 735-745
- [Lawrence et al. 2004] Lawrence, J., Rusinkiewicz, S. and Ramamoorthi, R. 2004. Efficient BRDF Importance Sampling using a Factored Representation. In *ACM Transactions on Graphics (Proc. SIGGRAPH)* Vol. 23, Issue 3, August 2004, New York, NY, USA, pages 494-503

- [Matusik et al. 2003] Matusik, W., Pfister, H., Brand, M., McMillan, L. 2003. A Data-Driven Reflectance Model. In *ACM Transactions on Graphics (Proc. SIGGRAPH)* Vol. 22, Nr. 3, July 2003, New York, NY, USA, pages 759-769
- [McReynolds et al. 2000] McReynolds, T., Blythe, D., Grantham, B., Nelson, S. 2000. Advanced Graphics Programming Techniques using OpenGL. In *SIGGRAPH 2000 Course Notes, course 32*
- [Miller and Hoffman 1984] Miller, G., Hoffman, C. 1984. Illumination and Reflection Maps: Simulated Objects in Simulated Real Environments. *SIGGRAPH 1984 Advanced Computer Graphics Seminar Notes* 1984
- [Neumann et al. 1999] Neumann, L., Neumann, A., Szirmay-Kalos, L. 1999. Compact Metallic Reflectance Models. In *Computer Graphics Forum* Vol. 18, Issue 3, 1999, pages 161-172
- [Ng et al. 2003] Ng, R., Ramamoorthi, R., Hanrahan, P. 2003. All-Frequency Shadows using Non-Linear Wavelet Lighting Approximation. In *ACM Transactions on Graphics - Proceedings of ACM SIGGRAPH 2003* Vol. 22, Issue 3, July 2003, pages 376 - 381, ACM New York, NY, USA
- [Ng et al. 2004] Ng, R., Ramamoorthi, R., Hanrahan, P. 2004. Triple Product Wavelet Integrals for All-Frequency Relighting. In *ACM Transactions on Graphics - Proceedings of ACM SIGGRAPH 2004* Vol. 23, Issue 3, August 2004, pages 477 - 487, ACM New York, NY, USA
- [Ngan et al. 2005] Ngan, A., Durand, F., Matusik, W. 2005. Experimental Analysis of BRDF Models. In *ACM Proceedings of the Eurographics Symposium on Rendering 2005* pages 117-226
- [Nicodemus 1970] Nicodemus, F. E. 1970. Reflectance Nomenclature and Directional Reflectance and Emissivity. In *Applied Optics* Vol. 9, Issue 6, pages 1474-1475
- [Nielsen and Christensen 2002] Nielsen, K. H., Christensen, N. J. 2002. Real-Time Recursive Specular Reflections on Planar and Curved Surfaces using Graphics Hardware. In *Proceedings of the 10th International Conference in Central Europe on Computer Graphics, Visualisation and Computer Vision 2002, WSCG Short Papers* pages 91-98, 2002
- [Pharr and Humphreys 2004] Pharr, M. and Humphreys, G. 2004. Physically Based Rendering: From Theory to Implementation. Morgan Kaufmann Publishers Inc, San Francisco, CA, USA
- [Phong 1975] Phong, B. T. 1975. Illumination for Computer-Generated Pictures. In *Communications of the ACM* Vol. 18, Issue 6, June 1975, New York, NY, USA, pages 311-317
- [Popescu et al. 2006] Popescu, V., Mei, C., Dauble, J., Sacks, E. 2006. Reflected-Scene Importers for Realistic Reflections at Interactive Rates. In *Computer Graphics Forum* Vol. 25, Issue 3, pages 313-323

- [Ramamoorthi 2009] Ramamoorthi, R. 2009. Precomputation-based Rendering. In *Foundations and Trends in Computer Graphics and Vision* Vol. 3, Issue 4, April 2009, pages 281-369
- [Rusinkiewicz 1997] Rusinkiewicz, S. 1997. A Survey of BRDF Representation in Computer Graphics. CS348, Winter 1997 <http://www.cs.princeton.edu/~smr/cs348c-97/surveypaper.html>
- [Rusinkiewicz 1998] Rusinkiewicz, S. 1998. A New Change of Variables for Efficient BRDF Representation. In *Proceedings of the Eurographics Workshop on Rendering 1998*, Springer Wien, New York, NV, pages 11-22
- [Ruszczynski 2006] Ruszczynski, A. 2006. Non-linear Optimization. Princeton University Press 41 William Street, Princeton, New Jersey, USA, ISBN-13: 978-0-691-11915-1
- [Sari 2008] Sari, M. 2008. Photorealistic Rendering Of Measured BRDF Data. Master Thesis at the Institute of Computer Graphics and Algorithms, Vienna University of Technology. A-1040 Vienna, Austria
- [Schlick 1994] Schlick, C. 1994. An Inexpensive BRDF Model for Physically Based Rendering. In *Computer Graphics Forum* Vol. 13 Nr. 3, pages 233-246
- [Schroeder and Sweldens 1995] Schroeder, P., Sweldens, W. 1995. Spherical Wavelets: Efficiently Representing Functions on the Sphere. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 1995)* ACM. New York, NY, USA, pages 161-172
- [Sloan et al. 2002] Sloan, P. P., Kautz, J., Snyder, J. 2002. Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques* ACM. New York, NY, USA, pages 527-53
- [Suykens et al. 2003] Suykens, F., vom Berge, K., Lagae, A., Dutre, P. 2003. Interactive Rendering with Bidirectional Texture Functions. In *Proceedings of the Eurographics 2003 Computer Graphics Forum* Vol. 22, Nr. 3, June 2003
- [Szirmay-Kalos et al. 2005] Szirmay-Kalos, L., Aszodi, B., Lazanyi, I. and Premecz, M. 2005. Approximate Ray-Tracing on the GPU with Distance Impostors. In *Proceedings of the Eurographics 2005 Computer Graphics Forum* Vol. 24, Nr. 3, Dublin, Ireland pages 171-176
- [Szirmay-Kalos 2008] Szirmay-Kalos, L. 2008. Monte-Carlo Methods in Global Illumination - Photo-Realistic Rendering with Randomization. VDM Verlag, Saarbrücken, Germany
- [Torrance and Sparrow 1967] Torrance, K. E., Sparrow, E. M. 1967. Theory for Off-Specular Reflection from Roughened Surfaces. In *Journal of the Optical Society of America* 1967 Vol. 57, Nr. 9, pages 1105 - 1114

- [Veach 1997] Veach, E. 1997. Robust Monte-Carlo Methods for Light Transport Simulation. Ph.D. Thesis at the University of Stanford Department of Computer Science, University of Stanford, CA 94305, USA, December 1997
- [Ward 1992] Ward, G. 1992. Measuring and Modeling Anisotropic Reflection. In *Proceedings of the 19th annual conference on Computer Graphics and interactive techniques* ACM, New York, NY, USA, pages 265-272
- [Walter 2005] Walter, B., 2005. Notes on the Ward BRDF. *Technical Report PCG-05-06* Program of Computer Graphics, Cornell University
- [Walter et al. 2007] Walter, B., Marschner, S. R., Li, H. and Torrance, K. E. 2007. Micro-facet Models for Refraction through Rough Surfaces. In *Proceedings of the Eurographics Symposium on Rendering* June 2007, Grenoble, France, pages 195-206
- [Westin et al. 1992] Westin, S., Arvo, J., and Torrance, K. E. 1992. Predicting Reflectance Functions from Complex Surfaces. In *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1992* Chicago, IL, USA, pages 255-264
- [Williams 1983] Williams, L. 1983. Pyramidal Parametrics. In *Proceedings of the 10th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 1983)* Vol. 17, Nr. 3, pages 1-11, New York, NY, USA, July 1983