UNIVERSITY OF CALIFORNIA, SAN DIEGO

Occlusion-Aware Hessians for Error Control in Irradiance Caching

A Thesis submitted in partial satisfaction of the requirements

for the degree Master of Science

in

Computer Science

by

Jorge Andres Schwarzhaupt

Committee in charge:

Henrik Wann Jensen, Chair
Sam Buss
Jurgen Schulze

2013

The Thesis of Jorge Andres Schwarzhaupt is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2013

DEDICATION

To my wife, Maureen.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank my friends and family for supporting me through the years. First, I would like to thank my wife Maureen, without whose encouragement and tenacious dedication to seeing me thrive, I might never have achieved what I have, both professionally and, especially, academically. I would also like to especially thank my parents Jorge and Oriana, who fostered in me an intense curiosity about the world and without whom I would not be here today.

I would also like to thank Henrik Wann Jensen, who kept encouraging me to dig deeper in order to understand the fundamentals of light transport, along with everyone at the UCSD Graphics Lab, especially Phi Hung Nguyen who helped me out tremendously. Thanks are also due to the other members of my thesis committee, Samuel Buss and Jurgen Schulze, for agreeing to take their valuable time to read and evaluate this thesis.

Portions of this thesis are based on a paper which I have co-authored with others. My contributions to this paper is listed below.

- Chapter 6 is based on the work that appears in the paper:
  - Jorge Schwarzhaupt, Wojciech Jarosz and Henrik Wann Jensen. "Practical Hessian-Based Error Control for Irradiance Caching." In *ACM Transactions on Graphics*, 31(6)1-10, 2012.

I was the primary investigator and author of this paper.

ABSTRACT OF THE THESIS

Occlusion-Aware Hessians for Error Control in Irradiance Caching

by

Jorge Andres Schwarzhaupt

Master of Science in Computer Science

University of California, San Diego, 2013

Henrik Wann Jensen, Chair

This thesis introduces a new error metric for irradiance caching that significantly outperforms the classic Split-Sphere heuristic. Our new error metric builds on recent work using second order gradients (Hessians) as a principled error bound for the irradiance. We add occlusion information to the Hessian computation, which greatly improves the accuracy of the Hessian in complex scenes, and this makes it possible for the first time to use a radiometric error metric for irradiance caching. We enhance the metric making it based on the relative error in the irradiance as well as robust in the presence of black occluders. The resulting error metric is efficient to compute, numerically robust, supports elliptical error bounds and arbitrary hemispherical sample distributions, and unlike the Split-Sphere heuristic it is not necessary to arbitrarily clamp the computed error thresholds. Our results demonstrate that the new error metric outperforms existing error metrics based on the Split-Sphere model and occlusion-unaware Hessians.

Chapter 1

Introduction

Starting with cave paintings dating back over 30,000 years in the past, mankind has been fascinated by the pursuit of realistic depictions of the world that surrounds us. Artists strive to interpret our world and reproduce it through an intensely personal point of view, while scientists rigorously study the innumerable phenomena that give rise to what we perceive as reality. As a tool, the computer seems especially suited to bridge these disciplines, giving rise to the field of computer graphics.

One particularly important branch of computer graphics is physically based rendering, which attempts to synthesize images by simulating the physical processes that give rise to the various visual phenomena that are visible in real life. Of these processes, none are more fundamental than the simulation of light transport, or the way in which light interacts with objects and the environment and travels from one point to another. Features in a scene will only be visible in an image if there is some measurable amount of light that reaches a suitable sensor, such as a CCD in a digital camera.

A related goal in computer graphics is photorealistic rendering, which attempts to produce imagery that is indistinguishable from photographs without necessarily requiring the rigorous simulation of the real-life physical phenomena that give rise to them. In this case, approximations to those phenomena are acceptable as long as the *visual* result is of sufficient quality to achieve the stated goal.

Importantly, both disciplines are concerned with the generation of *realistic* images, and in many cases the advances made in one fuel further investigation in the other. Applications of realistic image synthesis are pervasive in contemporary society, ranging from product renderings for advertising, to special effects in movies and television, to architectural visualization for real estate projects. It is indeed hard nowadays to browse a magazine or watch television without being witness to realistic computer generated imagery.

Being the most fundamental building block for realistic imagery, light transport has been researched since the earliest days of computer graphics, and algorithms for its efficient simulation have been key in the advancement of the field. With increased computational power being made available through the years, new algorithms that more rigorously simulate the light transport equations have been proposed and implemented, resulting in ever higher realism being achieved through computer graphics.

Global illumination is a term used to refer to those algorithms that attempt to fully simulate the effects of light bouncing more than once in a scene, as opposed to only visualizing the direct effect of a light source on the various surfaces in a scene. When light hits a surface, it doesn't merely reflect towards a suitable sensor that detects its effects; it also bounces and affects other surfaces in the scene, giving rise to indirect illumination effects such as color bleeding. The realism added to images by using a global illumination algorithm can be profound.

It is possible to divide the illumination in a scene into various conceptually separate sub-components: direct lighting is that light which leaves an emitting source and directly hits some surface in the scene, after which it reflects and lands on the imaging sensor; diffuse indirect lighting occurs when light strikes a diffuse surface, after which it is reflected simultaneously in all directions and hits other objects, one or more times, before landing on the camera; specular indirect illumination, or caustics, are the result of light from an emitter reflecting off a specular surface such as a mirror, or refracting through a specular object, such as a glass, producing focused indirect lighting effects; a final sub-component is that light which leaves an emitter and immediately hits the imaging sensor. In this thesis, we focus our attention on the diffuse indirect illumination component.

The diffuse component of the indirect illumination in a scene can be very computationally expensive to compute. In order to compute the indirect light reflected from a point in the scene towards the imaging sensor, we must account for light *arriving* at the point from all directions across the hemisphere over the point. Light arriving directly from a light source is discarded, since it should be taken into account by a direct illumination algorithm; only light that has bounced off at least one surface before arriving at the point where we're measuring is important. It should be apparent that this leads to a recursive algorithm, since to measure the light arriving from a certain direction we will need to compute the light *leaving* the surface that is visible in that direction. In reality, light can bounce infinitely many times before reaching any single point in the scene, which would make solving the diffuse indirect illumination for a scene intractable. However, part of the light is absorbed when it hits a surface, so that its intensity at each subsequent bounce is progressively diminished. This means that it is possible to compute a very close approximation of the true indirect illumination by only computing a finite number of bounces, though statistical methods exist that can make computing an unbounded number of bounces possible.

One of the most widely-used algorithms for computing the diffuse indirect illumination is Irradiance Caching [WRC88]. Irradiance is a radiometric quantity describing the incident light hitting a surface. This algorithm is born from the observation that indirect illumination varies slowly across diffuse surfaces, such that it is possible to compute the actual irradiance only sparsely across the scene; a diffuse material reflects light uniformly in all directions, meaning that the light reflected from the surface is invariant to the direction from which the surface is observed. This also means that, if the hemispheres over two different points are sufficiently similar, the irradiance computed at one point can be re-used at the other, trading a small amount of error in the resulting approximation for a large reduction in computational expense. In this algorithm, illumination at points where direct irradiance computation is not attempted is then approximated by interpolating the irradiance cached at those locations where it was. While in reality many surfaces break the diffuse material assumption, in practice the algorithm works surprisingly well in most real-world cases. Fundamental to the algorithm, then, is answering the question of *where* in the scene to directly compute the irradiance, such that interpolation at other locations in the scene will produce the least possible error in the approximation.

Ward et al. [WRC88] proposed the Split-Sphere model as a way of deriving an upper bound on the expected error of approximating the irradiance at a point where it was not computed by interpolating the irradiance computed at other locations. The split-sphere is an idealized representation of the environment that produces the largest possible change in the irradiance as the point where it will be approximated is translated, assuming that there are no strong indirect illumination sources. Ward's intuition for making this assumption was that strong indirect illumination would *not* be taken into account by the algorithm and would instead be directly sampled. The so-called split-sphere heuristic has been used, unchanged, as the method for error control for Irradiance Caching ever since its introduction.

Unfortunately, the assumptions that the split-sphere heuristic makes are broken, in practice, by even trivially simple scenes. Not only are strong indirect diffuse illumination effects surprisingly common, but the threshold for a radiometric feature in the scene to cause the split-sphere error bound to be too loose is also not very high. This has led to the introduction in the literature of an array of methods that attempt to fix these shortcomings while stopping short of actually modifying the split-sphere heuristic. This means that a state of the art implementation of Irradiance Caching must not only support the split-sphere heuristic itself, but must also implement most or all of these fixes in order to produce acceptable quality imagery.

Furthermore, as stated before, the split-sphere is only an idealized model and not the actual radiometric configuration of any given scene that is to be rendered. This leads the split-sphere heuristic to define its error measure in a purely geometric fashion, in effect discarding the effects of the *actual* radiometry of the scene, leading to cases where cache records are wastefully computed at radiometrically unimportant locations in the scene.

Jarosz et al. [JSKJ12] proposed the first practical replacement for the split-sphere heuristic in the context of a complete development of 2D light transport, by making use of the second derivative of irradiance - the irradiance Hessian - to derive a principled method for error control in the irradiance caching algorithm. While their results are promising, they stopped short of fully exploring the merits of their new method. Further to this, the irradiance Hessian they derived does not take inter-object occlusions into account, so that an important class of radiometric phenomena, indirect shadows, have no effect on their error control method.

In this thesis, we augment the work of Jarosz et al. by deriving an occlusion-aware irradiance Hessian, and extensively validate its use in a new method for controlling the interpolation error in Irradiance Caching. Our new method has the advantage over the classical split-sphere heuristic of decreased interpolation error for a given number of cache points, as well as only having a single user-facing parameter, leading to much simpler and intuitive control over the resulting image quality. Moreover, our new method is able to take advantage of most of the corrective fixes that are routinely applied to the split-sphere heuristic, leading to even lower error in the resulting images; some of the fixes produce no noticeable image quality improvement and can thus be discarded, leading to simpler and faster implementation.

1.1 Summary of Original Contributions

The contributions of this thesis are related to the computation of an occlusion-aware irradiance Hessian in arbitrary scenes, and the utilization of such a Hessian to control the approximation error in the context of the irradiance caching algorithm.

1.1.1 Occlusion-Aware Irradiance Hessian

Our first contribution is the analysis and derivation of the Hessian of the light vector field due to partially occluded polyhedral light sources. We show that this derivation can be successfully used to analytically compute the irradiance Hessian for an arbitrary polygonal 3D scene, and then analyze the reasons that render it impractical for general application to error control in the irradiance caching algorithm.

1.1.2 Practical Hessian-based Error Control for Irradiance Caching

Our second and major contribution is the derivation of a practical method for computing an occlusion-aware irradiance Hessian in 3D scenes of arbitrary complexity and construction. We show how this method can be practically leveraged in the context of Irradiance Caching to produce higher-quality images than have so far been possible using this algorithm, and compare the results to those obtained by the venerable split-sphere heuristic.

1.2 Organization of the Thesis

The thesis is divided into seven chapters. In the remaining chapters, we first focus on background material and previous work, leading to our own contributions. Chapter 2 describes the fundamentals of light transport and serves as a basis for the remainder of the thesis. Chapter 3 describes the irradiance caching algorithm in detail. Chapter 4 describes the split-sphere model and heuristic, shows its limitations and shortcomings, and describes the extensions that have been proposed to deal with them. Chapter 5 briefly describes the previous work on leveraging irradiance Hessians for error control in irradiance caching. In chapter 6, we first describe our derivation of an analytical irradiance Hessian for partially occluded polyhedral sources, and show why it is impractical for use within the irradiance caching algorithm. We then present our practical method for computing an occlusion-aware irradiance Hessian and show how it can be used to produce high-quality renderings using irradiance caching. Finally, we present our conclusions in chapter 7.

Chapter 2

Fundamentals of Light Transport

The goal of computer graphics rendering algorithms is to synthesize images that provide a representation of a virtual scene. In the specific case of *realistic* rendering, this goal is made more precise, and entails the production of imagery that accurately portray the real-world appearance of such a scene. Global illumination algorithms have become an integral part of this discipline, and strive to achieve an accurate simulation of the physical behavior of light that is emitted from light sources and then interacts with the various objects in the scene, the virtual camera sensor being (a very important) one of them.

In this chapter, we first explore the necessary background and develop the terminology necessary to formulate the problem that these algorithms must solve. Sections 2.1 and 2.2 provide this background, and in section 2.3 we explain the expressions that govern the behavior of light as it interacts with surfaces in the scene. Sections 2.4 and 2.5 describe two of the main techniques that have been developed to simulate these effects.

2.1 Models of Light

The dual nature of light gives rise to a variety of phenomena that computer graphics algorithms can attempt to simulate. Several models have been proposed that are able to capture progressively larger subsets of these phenomena, and that are thus also progressively complex. These are, in increasing order of complexity:

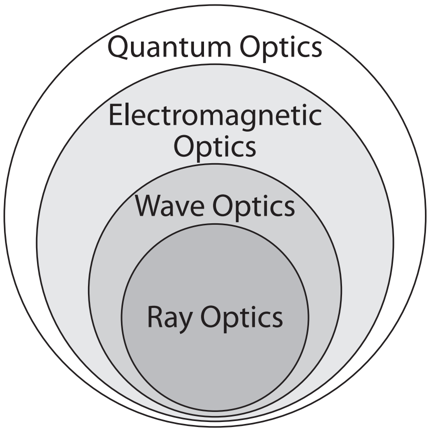


Figure 2.1: Different models for describing the properties of light, where each successive model is able to account for further optical phenomena. As for most of computer graphics, we will restrict ourselves to the ray optics model throughout this thesis.

ray optics, wave optics, electromagnetic optics and quantum optics (see Figure 2.1). Of these models, ray optics - also called geometric optics - is the one most commonly used in computer graphics applications.

This model makes several simplifying assumptions about the behavior of light that preclude several interesting light effects from being simulated; light can only be emitted, reflected and transmitted. This means that phenomena that depend on the wave nature of light (such as diffraction and interference), on the electromagnetic nature of light (such as polarization and dispersion) or the quantum mechanics effects of light (such as phosphorescence and fluorescence) cannot be easily reproduced. However, the visibility of these effects on many real-world scenes is generally quite limited, such that the limitations of the model rarely limit the realism of the images that can be produced. Throughout this thesis we will use the ray optics model exclusively.

2.2 Radiometry

Radiometry is the study of the physical measurement of electromagnetic radiation, of which an important component is visible light. Like any other rigorous field of study, it defines a common terminology for the quantities and units involved, most of which is used as well in global illumination algorithms. While in reality all of these radiometric quantities depend on the wavelength of light, we omit this dependence due to our use of the ray optics model.

2.2.1 Radiometric Quantities

Flux.

This is the fundamental quantity in radiometry. Flux, denoted ϕ, expresses the total energy that flows across a surface per unit time - for instance, we use flux to describe the power of light bulbs. Flux is expressed in terms of watts $[W = J \cdot s^{-1}]$.

Irradiance.

Commonly referred to using the letter E, irradiance describes the amount of power incident on a surface, per unit surface area. It is expressed in units of $[W \cdot m^{-2}]$:

$$E(\mathbf{x}) = \frac{d\phi}{dA}. \tag{2.1}$$

Irradiance is only meaningful with respect to a specific surface position $\mathbf{x}$, with surface normal $\mathbf{n}$. The converse quantity, measuring the flux *leaving* a surface, is called *radiant exitance* (M) or *radiosity* (B).

Radiance.

Radiance is defined as flux per unit projected area, per unit solid angle, and is measured in units of $[W \cdot sr \cdot m^{-2}]$. Intuitively, radiance expresses how much power arrives from a differential solid angle $d\vec{\omega}$ at a hypothetical differential area $dA^{\perp}$ that is perpendicular to that direction. It is a five-dimensional function that varies with position $\mathbf{x}$ and direction vector $d\vec{\omega}$ that can be expressed as

$$L(\mathbf{x}, \vec{\omega}) = \frac{d^2\phi}{d\vec{\omega} dA^{\perp}}. \tag{2.2}$$

Radiance is probably the most important quantity in global illumination algorithms because it captures the perceived brightness of objects, or their "ap-

Table 2.1: The fundamental radiometric quantities, with their symbol and units.

Symbol	Units	Description
ϕ	W	Flux or power
E	$W \cdot m^{-2}$	Irradiance
M	$W \cdot m^{-2}$	Radiant exitance
B	$W \cdot m^{-2}$	Radiosity
L	$W \cdot m^{-2} \cdot sr^{-1}$	Radiance

pearance". In practice, we are usually interested in measuring radiance at an actual surface as opposed to the hypothetical differential surface perpendicular to the direction of light flow. In this case, equation (2.2) becomes

$$L(\mathbf{x}, \vec{\omega}) = \frac{d^2\phi}{d\vec{\omega} dA(\mathbf{n} \cdot \vec{\omega})} \tag{2.3}$$

where the *foreshortening* term, $(\mathbf{n} \cdot \vec{\omega})$, takes into account the spreading of light at glancing angles.

We summarize the radiometric quantities, their notation and units in Table 2.1.

2.2.2 Radiometric Relationships

Given the importance of radiance in global illumination algorithms, it is usually useful to express the other radiometric quantities described in the previous subsection in terms of radiance. It is possible to invert equation (2.2) to express flux in terms of radiance, by integrating both sides over the hemisphere of directions Ω and area A, arriving at:

$$\phi = \int_A \int_\Omega L(\mathbf{x} \rightarrow \vec{\omega})(\mathbf{n} \cdot \vec{\omega}) d\vec{\omega} dA. \tag{2.4}$$

This can be combined with equation (2.1) in order to express irradiance and radiosity in terms of radiance:

$$E(\mathbf{x}) = \int_\Omega L(\mathbf{x} \leftarrow \vec{\omega})(\mathbf{n} \cdot \vec{\omega}) d\vec{\omega}, \tag{2.5}$$

$$M(\mathbf{x}) = B(\mathbf{x}) = \int_\Omega L(\mathbf{x} \rightarrow \vec{\omega})(\mathbf{n} \cdot \vec{\omega}) d\vec{\omega}. \tag{2.6}$$

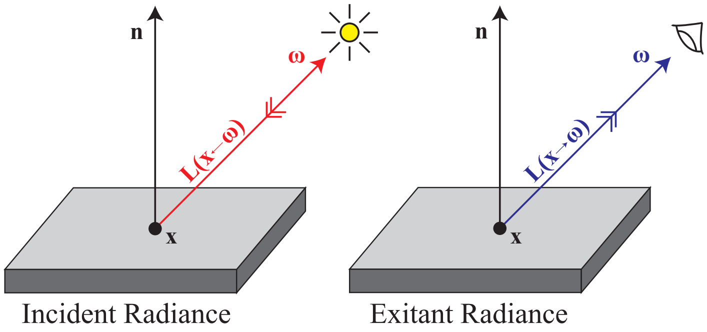


Figure 2.2: We can distinguish between two distinct types of radiance at a point $\mathbf{x}$. $L(\mathbf{x}\!\leftarrow\!\vec{\omega})$ represents radiance coming towards $\mathbf{x}$ from direction $\vec{\omega}$ (left), while $L(\mathbf{x}\!\rightarrow\!\vec{\omega})$ represents radiance leaving $\mathbf{x}$ in the direction $\vec{\omega}$ (right).

2.2.3 Incident and Exitant Radiance Functions

Throughout this thesis we will use the convention that $L(\mathbf{x} \rightarrow \vec{\omega})$ denotes radiance *leaving* $\mathbf{x}$ in direction $\vec{\omega}$ while $L(\mathbf{x} \leftarrow \vec{\omega})$ signifies radiance *arriving* at $\mathbf{x}$ from direction $\vec{\omega}$. In both cases, the direction vector $\vec{\omega}$ points *away* from the surface, as illustrated in Figure 2.2. This distinction is important because the two quantities measure fundamentally different sets of photons, such that in general

$$L(\mathbf{x}\!\rightarrow\!\vec{\omega}) \neq L(\mathbf{x}\!\leftarrow\!\vec{\omega}). \tag{2.7}$$

Photons propagate freely as long as they don't interact with their environment (as in a vacuum) or surfaces in the scene. This means that incident radiance at a point, from a direction $\vec{\omega}$, will continue on as exitant radiance in direction $-\vec{\omega}$

$$L(\mathbf{x}\!\leftarrow\!\vec{\omega}) = L(\mathbf{x}\!\rightarrow\!-\vec{\omega}). \tag{2.8}$$

This establishes the invariance of radiance along straight paths in a vacuum, a property due to the principle of conservation of energy, and allows us to define a simple relationship between the incident and exitant radiance at two distinct points. In order to do so, we first introduce the *visibility function* $v(\mathbf{x}, \vec{\omega}) = \mathbf{x}'$, where $\mathbf{x}'$ is the point on the surface closest to $\mathbf{x}$ in direction $\vec{\omega}$. The invariance property means that incident radiance at a point $\mathbf{x}$, from direction $\vec{\omega}$, is equal to the exitant radiance at $\mathbf{x}'$ in the direction $-\vec{\omega}$. This is illustrated in Figure 2.3 and can be expressed as:

$$L(\mathbf{x}\!\leftarrow\!\vec{\omega}) = L(\mathbf{x}'\!\rightarrow\!-\vec{\omega}). \tag{2.9}$$

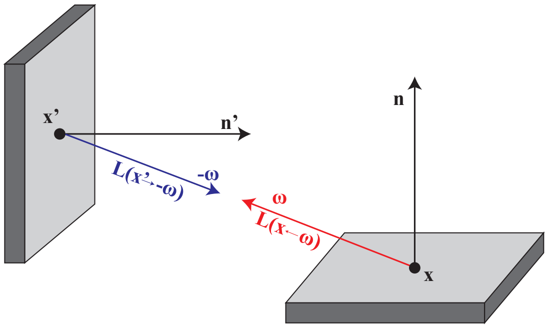


Figure 2.3: The incident radiance at **x** from direction $\vec{\omega}$ is equal to the exitant radiance from the closest visible surface in that same direction.

2.3 Light and Surfaces

Photons emitted into a scene interact with the different objects in it by being reflected or transmitted at the surface boundaries. In the remainder of this chapter we will explore these interactions.

2.3.1 The BRDF

Different materials, under the same lighting conditions, will *appear* different to an observer, due to the dissimilar interactions of the light with the materials. When light encounters an object in a scene, it will either be reflected, refracted or absorbed. Due to the ray optics model, we will assume that light incident at a surface exits at the same wavelength and at the same time, so that effects such as fluorescence and phosphorescence are ignored. We further make the simplifying assumption that light striking at a particular point in a surface will be reflected at the same location, precluding the simulation of subsurface scattering effects. In this case, the interaction between the light and the surface can be described by a reflectance function called the *bidirectional reflectance distribution function* of BRDF. The BRDF at a point **x** is defined to be the ratio of the differential radiance reflected in an exitant direction $\vec{\omega}'$, and the differential irradiance incident from a direction $\vec{\omega}$:

$$f_r(\mathbf{x}, \vec{\omega} \to \vec{\omega}') \equiv \frac{dL(\mathbf{x} \to \vec{\omega}')}{dE(\mathbf{x} \leftarrow \vec{\omega})} = \frac{dL(\mathbf{x} \to \vec{\omega}')}{L(\mathbf{x} \leftarrow \vec{\omega})(\mathbf{n} \cdot \vec{\omega}) d\vec{\omega}}. \tag{2.10}$$

where the last step follows from substituting equation (2.5) into the denominator.

Properties of the BRDF

1. **Range.** The BRDF can take any positive value.
2. **Domain.** At each point on a surface, the BRDF is a four-dimensional function; two dimensions specify the incoming direction and the other two specify the outgoing direction. Generally, the BRDF is anisotropic, meaning that the value of f_r will change depending on the rotation around the surface normal $\mathbf{n}$. However, for many real materials the BRDF does not depend on the orientation of the underlying surface.
3. **Reciprocity.** The value of f_r is invariant to an exchange of the incoming and outgoing directions. Mathematically, Helmholtz's reciprocity states that:
$$f_r(\mathbf{x}, \vec{\omega}, \vec{\omega}') = f(\mathbf{x}, \vec{\omega}', \vec{\omega}). \tag{2.11}$$
Therefore, the following notation is used to indicate that the directions can be interchanged:
$$f_r(\mathbf{x}, \vec{\omega} \leftrightarrow \vec{\omega}'). \tag{2.12}$$
4. **Relationship between incident and reflected radiance.** We can use the information in the BRDF to derive the relationship between incident and reflected radiance. Multiplying both sides of equation (2.10) by the denominator and integrating over the hemisphere we get:
$$L(\mathbf{x} \rightarrow \vec{\omega}') = \int_\Omega f_r(\mathbf{x}, \vec{\omega} \leftrightarrow \vec{\omega}') L(\mathbf{x} \leftarrow \vec{\omega})(\mathbf{n} \cdot \vec{\omega}) d\vec{\omega}. \tag{2.13}$$
This equation can be used to compute the reflected radiance at a point $\mathbf{x}$, and tells us that the radiance reflected off a surface can be computed by integrating the incident radiance arriving from the entire hemisphere of directions.
5. **Energy conservation.** Energy conservation requires that a surface not reflect more light than is incident upon it. This can be expressed as the following constraint:
$$\int_\Omega f_r(\mathbf{x}, \vec{\omega} \leftrightarrow \vec{\omega}')(\mathbf{n} \cdot \vec{\omega}) d\vec{\omega} \leq 1, \forall \vec{\omega}'. \tag{2.14}$$

2.3.2 The Rendering Equation

In order to produce images, computer graphics algorithms must determine the ”appearance” of all points that are visible, either directly or indirectly, in the final rendering. In the case of realistic image synthesis, this is usually accomplished by solving the *Rendering Equation*, also known as the *Light Transport Equation* [Kaj86].

Hemispherical Formulation

The rendering equation relates outgoing radiance at a point $\mathbf{x}$, in a certain direction $\vec{\omega}$, to the emitted and reflected radiance at that point. Since the reflected radiance depends upon all light impinging on that point at any given moment, and that light might in turn have been emitted or reflected from another point in the scene, this leads to a recursive formula:

$$\underbrace{L(\mathbf{x}\!\rightarrow\!\vec{\omega}')}_{outgoing} = \underbrace{L_e(\mathbf{x}\!\rightarrow\!\vec{\omega}')}_{emitted} + \underbrace{L_r(\mathbf{x}\!\rightarrow\!\vec{\omega}')}_{reflected}. \tag{2.15}$$

The local illumination model described by Equation (2.13) expresses the reflected radiance at a point on a surface. Substituting this into Equation (2.15) results in the hemispherical form of the rendering equation:

$$\underbrace{L(\mathbf{x}\!\rightarrow\!\vec{\omega}')}_{outgoing} = \underbrace{L_e(\mathbf{x}\!\rightarrow\!\vec{\omega}')}_{emitted} + \underbrace{\int_\Omega f_r(\mathbf{x}, \vec{\omega}\!\leftrightarrow\!\vec{\omega}') L(\mathbf{x}\!\leftarrow\!\vec{\omega})(\mathbf{n}\cdot\vec{\omega})d\vec{\omega}}_{reflected}. \tag{2.16}$$

Area Formulation

It is sometimes convenient to express the rendering equation not as an integration over the hemisphere of directions, but directly over points on the surface of scene geometry. This can be done using the change of variable relation

$$d\vec{\omega} = \frac{\mathbf{n}_\mathbf{y}\cdot -\vec{\omega}}{\|\mathbf{x}-\mathbf{y}\|^2} dA(\mathbf{y}), \tag{2.17}$$

where $\mathbf{y}$ is a point on a surface with normal $\mathbf{n}_\mathbf{y}$.

In contrast to the hemispherical formulation, where visibility is implicitly accounted for by using the ray casting function, in this case we must make the accounting explicit. We introduce the *visibility function* V:

$$\forall \mathbf{x},\mathbf{y} \in A : V(\mathbf{x}\leftrightarrow\mathbf{y}) = \begin{cases} 1 & \text{if } \mathbf{x} \text{ and } \mathbf{y} \text{ are mutually visible,} \\ 0 & \text{otherwise.} \end{cases} \tag{2.18}$$

and transform Equation (2.16) into an integration over surface area:

$$\begin{aligned} L(\mathbf{x}\rightarrow\vec{\omega}) &= L_e(\mathbf{x}\rightarrow\vec{\omega}) \\ &\quad + \int_A f_r(\mathbf{x},\mathbf{y}\leftrightarrow\mathbf{x})L(\mathbf{x}\leftarrow\mathbf{y})V(\mathbf{x}\leftrightarrow\mathbf{y})\frac{(\mathbf{n}\cdot\vec{\omega})(\mathbf{n_y}\cdot-\vec{\omega})}{\|\mathbf{x}-\mathbf{y}\|^2}dA(\mathbf{y}). \end{aligned} \tag{2.19}$$

We can simplify this expression by defining the *geometric coupling term*:

$$G(\mathbf{x}\leftrightarrow\mathbf{y}) = \frac{(\mathbf{n}\cdot\vec{\omega})(\mathbf{n_y}\cdot-\vec{\omega})}{\|\mathbf{x}-\mathbf{y}\|^2}, \tag{2.20}$$

resulting in

$$\begin{aligned} L(\mathbf{x}\rightarrow\vec{\omega}) &= L_e(\mathbf{x}\rightarrow\vec{\omega}) \\ &\quad + \int_A f_r(\mathbf{x},\mathbf{x}\leftrightarrow\mathbf{y})L(\mathbf{x}\leftarrow\mathbf{y})V(\mathbf{x}\leftrightarrow\mathbf{y})G(\mathbf{x}\leftrightarrow\mathbf{y})dA(\mathbf{y}). \end{aligned} \tag{2.21}$$

2.4 Methods for Solving the Rendering Equation

The rendering equation is far too complex to solve analytically in the general case, and very costly to compute even in simplified settings. This has lead to its approximation using numerical integration methods. Many algorithms have been proposed in the literature to fully or partially tackle this problem, which can generally be classified into two categories: Monte Carlo ray tracing methods and finite element methods.

2.4.1 Monte Carlo Ray Tracing Methods

In Monte Carlo methods, random samples of a function to be integrated are used to approximate the integral. The first use of these methods can be traced back to work on the atomic bomb during the second world war, and they have continued to be used in the field of neutron transport ever since [MU49]. The development of Monte Carlo methods to computer graphics was done independently, starting with their introduction by Appel [App68], who suggested the use of ray-casting as a method to generate images by intersecting the scene geometry with a single ray for each pixel in the final image. This approach was augmented by Whitted [Whi80] who proposed the recursive evaluation of the illumination in a scene by tracing specular reflection and refraction rays, as well as shadow rays for direct diffuse illumination. Furthermore, he also suggested randomly perturbing the surface normal in order to simulate rough specular reflections.

Complex surface and light interactions are naturally supported by ray tracing methods and, when combined with Monte Carlo techniques, are able to simulate a wide variety of complex effects such as depth-of-field, motion blur, glossy reflections and global illumination [CPC84, Coo86, Kaj86]. Monte Carlo ray tracing methods, such as path tracing or bidirectional path tracing [Kaj86, VG94] are able to simulate practically all effects that are possible under the ray optics model. Unfortunately, Monte Carlo techniques suffer from variance that manifests itself as high-frequency noise in the generated images. Though there has been ample research into ways to minimize this noise [Shi91, VG95], rendering with these methods still typically requires tracing hundreds of rays at each pixel to produce an acceptable rendering.

2.4.2 Finite Element Methods

Finite element methods have typically been collectively called *radiosity methods* in computer graphics, and where originally adapted from the radiative heat transfer literature. Radiosity methods compute the indirect illumination in a scene by discretizing the scene geometry into a set of small patches, which form the basis over which the global illumination computation is performed. This computation is solved by expressing the rendering equation as a system of linear equations describing the exchange of light energy between patches in the scene.

The initial work in radiosity was heavily restrictive of the complexity of the scenes for which it could compute solutions, working only for scenes with large area light sources and Lambertian surfaces - for which the BRDF is a simple constant [GTGB84, CG85]. These restrictions lead to an algorithm that is simply to implement and for which the resulting solutions are *view-independent* - allowing for a single solution to be reused when rendering multiple views of a single scene. Due to this property, radiosity techniques were well-suited for use when rendering walk-through animations of simple diffuse scenes, though the computation of these solutions requires significant computation time and have very high storage requirements. This led to further research that attempted to address these drawbacks, by adding support for more complex reflections [ICG86, SAWG91, CSSD96], view-dependent computation [SAS92] and hierarchical computation [HSA91]. High-frequency lighting effects can be supported through the use of discontinuity meshing, which can resolve sharp discontinuities in the lighting - such as hard shadow edges - by carefully subdividing the tessellation of the solution. All these additions greatly increase the computational complexity of the radiosity algorithm. Radiosity is also ill-suited for scenes with very complex geometry, since the cost of solving the lighting equations is directly related to the geometric complexity of the scene.

One of the main advantages of radiosity techniques over unbiased Monte Carlo ray tracing methods is that, for a certain simulation quality, the computation time is independent of the resolution of the final image. Computational effort can be efficiently reused over large regions of the rendered image, in contrast to unbiased ray tracing which require independent evaluation at each pixel. *Biased* Monte Carlo ray tracing relaxes this requirement and has made some of the most efficient rendering algorithms possible. Irradiance caching [WRC88], which forms the basis for this thesis, is one such method.

Chapter 3

Irradiance Caching

Since its introduction by Ward in 1988 [WRC88], irradiance caching has become one of the most widely-used techniques used to compute global illumination. In this chapter we first provide a complete description of the algorithm as originally presented by Ward, including derivations needed to fully understand the approach. We discuss the strengths that have led to its widespread use, and then analyze some of the shortcomings that motivate the main contributions of this thesis. We then explore various extensions that have been proposed in the literature to deal with them.

3.1 Algorithm Overview

Ray tracing algorithms compute a solution to the rendering equation by recursion (see Figure 3.1). At each level, shading is computed by (1) intersecting the ray with the scene geometry, (2) computing the direct contributions from light sources, (3) computing specular reflection and refraction and (4) computing diffuse contributions from reflecting surfaces. The complexity of this calculation is closely related to the cost of step (1) and the times it must be executed given the recursion implicit in steps 2 through 4. The contributions from steps 2 and 3 generally subtend small solid angles, leading to their effective integration using a limited number of rays. The diffuse contribution of step 4, in contrast, requires integration over the entire visible hemisphere. Approximating this integral therefore requires

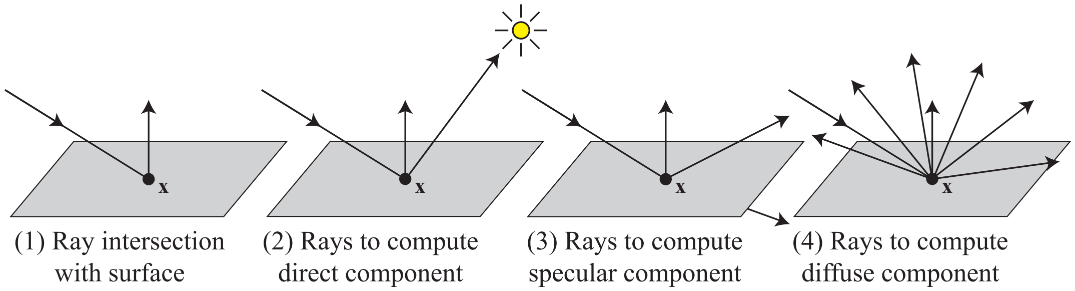

x
x
x
x
(1) Ray intersection with surface
(2) Rays to compute direct component
(3) Rays to compute specular component
(4) Rays to compute diffuse component

Figure 3.1: The four steps of ray tracing.

hundreds or even thousands of rays.

Ward et al. made the critical observation that, given the separation of the lighting integral into the disjoint components subsumed by steps 2 through 4, the diffuse indirect illumination tends to change slowly across Lambertian surfaces - direct lighting and specular reflection/refraction tend to be high frequency, but these components are taken into account by steps 2 and 3. Irradiance caching exploits this property: for the sake of efficiency, the costly indirect illumination integration is performed only at sparse locations in the scene. The results of this integration are stored in a cache and, where possible, reused through interpolation. We illustrate this approach in Figure 3.2 (right).

Irradiance caching thus marries some of the most beneficial properties of radiosity methods with the generality of Monte Carlo ray tracing approaches. Since the computational cost of interpolating irradiance from the cache is minuscule

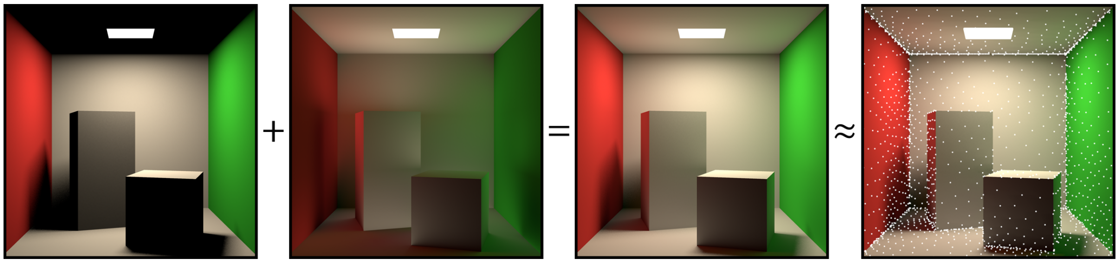


Figure 3.2: In irradiance caching, incident lighting is decomposed into a *direct* component (far left) and an indirect component (center left). The total lighting in the scene can be obtained by summing these two components (center right). Irradiance caching exploits the slow-varying nature of indirect illumination on lambertian surfaces, only computing this value at sparse locations (far right, cache records shown as white dots) and interpolating between these records whenever possible.

compared to directly sampling the irradiance - by performing the full integration - the cost to render images of a scene is independent of the resolution of those images. Whereas radiosity methods store their information in the scene's geometry mesh, the irradiance cache is stored in a separate data structure, decoupling it from any specific geometric representation and granting the algorithm far wider applicability. Finally, irradiance caching only computes the indirect illumination for points in the scene that are visible to the camera, gaining further efficiency when compared to radiosity approaches.

When using irradiance caching, the indirect illumination is computed as follows: If there is at least one cache record stored in the vicinity of **x**, then interpolate the irradiance from the stored value(s); otherwise, compute and store a new irradiance value at **x**. The algorithm runs lazily and on-demand, in a greedy fashion. In the following sections we will look at the algorithm in more detail, exploring how the irradiance values are computed, how the decision is made on whether to interpolate from the cache or to compute a new irradiance sample, and how the interpolation is made.

3.2 Irradiance Computation

This section presents the method by which a high quality approximation of the irradiance is computed. This method is invoked when querying the cache results in no useable cache records.

The irradiance at a point **x** can be expressed in terms of incident radiance using Equation 2.5:

$$E(\mathbf{x}) = \int_{\Omega} L(\mathbf{x}\leftarrow\vec{\omega})(\mathbf{n}\cdot\vec{\omega})d\vec{\omega} \tag{3.1}$$

We can estimate this integral using ray tracing by evaluating the following Monte Carlo estimator:

$$E(\mathbf{x}) \approx \frac{1}{N}\sum_{i=0}^{N-1}\frac{L(\mathbf{x}\leftarrow\vec{\omega})(\mathbf{n}\cdot\vec{\omega})}{pdf(\vec{\omega})}. \tag{3.2}$$

In order to reduce the variance in the estimate, Ward et al. used a cosine-weighted sample distribution with $pdf(\mathbf{x}) = (\mathbf{n}\cdot\vec{\omega})/\pi$, allowing the foreshortening term in the estimator to cancel out. Furthermore, the sample distribution is stratified by converting to polar coordinates, such that the estimator becomes:

$$E(\mathbf{x}) \approx \frac{\pi}{MN} \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} L(\mathbf{x}, \theta_j, \phi_k), \tag{3.3}$$

where

$$\theta_j = \sin^{-1}\left(\sqrt{\frac{j+\xi_1}{M}}\right), \tag{3.4}$$

$$\phi_k = 2\pi \frac{i+\xi_2}{N}, \tag{3.5}$$

and $(\xi_1, \xi_2) \in [0,1)^2$ are uniformly distributed random numbers. The angles θ_j and ϕ_k are expressed relative to the local coordinate frame at $\mathbf{x}$. As stated above, the algorithm uses a total of $M \times N$ samples to evaluate the irradiance at a point. In their original formulation, Ward et al. set $N = 2M$.

3.3 Irradiance Interpolation

In general, the method presented in the previous section is very costly to compute if we're interested in a high-quality approximation of the irradiance, since hundreds or even thousands of sample rays may be required. Irradiance caching therefore attempts to amortize this cost by only incurring it at as few locations of the scene as possible, and using interpolation or extrapolation everywhere else. Where the rate of change of the irradiance is low, only a sparse set of cache records should be computed; in parts of the scene where irradiance varies rapidly, a denser set of records should be created to enable an accurate approximation via interpolation. Estimating the rate of change of the irradiance is therefore of utmost importance for the performance of the algorithm.

Ward et al. developed a heuristic, described in the following section, that estimates an upper bound on the rate of change of irradiance for scenes assuming there are no concentrated sources of diffuse illumination. The reason behind this is that any concentrated light sources should be directly sampled - step 2 in section 3.1. If we denote this upper bound $\epsilon_i(\mathbf{x}, \mathbf{n})$, then its inverse is used as a weight for

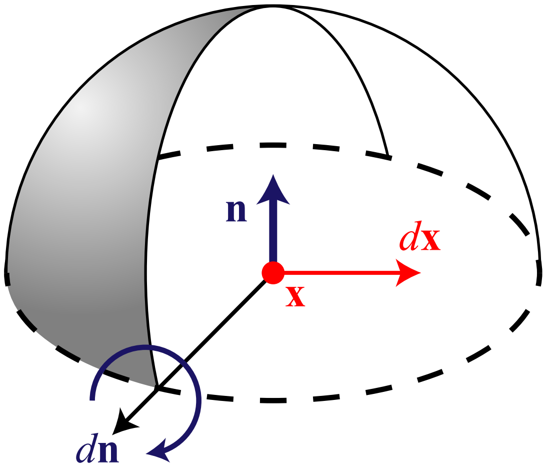


Figure 3.3: The "Split-Sphere" model. A surface is surrounded by a hypothetical sphere environment, which is half bright and half dark. The change in irradiance with respect to differential rotation of the normal **n** and translation along the surface is calculated and used as an estimate for the maximum change in irradiance in any possible scene.

interpolation between nearby cache records:

$$w_i(\mathbf{x}, \mathbf{n}) = \frac{1}{\epsilon_i(\mathbf{x}, \mathbf{n})} \tag{3.6}$$

Whenever a new shading location is reached during rendering, the cache is queried for any records for which the weight is higher than a user-specified threshold. If any exist, then these are used to compute a weighted average of the irradiance. If no records satisfy this criteria, then a new one is generated by directly computing the irradiance using Equation 3.3.

3.3.1 The Split-Sphere Model

In order to estimate an upper bound on the change in irradiance in a scene, $\epsilon_t(\mathbf{x}, \mathbf{n})$, Ward et al. introduced the "split-sphere" heuristic. In this simple model, a surface element is located at the center of a hypothetical sphere (see Figure 3.3). Half of the sphere is bright while the other half is dark, and the surface element's normal points towards the dividing line between both halves. This *hypothetical* environment has the largest possible irradiance gradient for a scene without concentrated sources of illumination.

Within this environment, irradiance is a function of **x**, the position, and **n**, the orientation, of the surface element. Therefore, a conservative upper bound on the change of irradiance is given by the first-order Taylor expansion of irradiance at $\mathbf{x}$, and we have

$$\epsilon_i(\mathbf{x}, \mathbf{n}) = E(\mathbf{x}_i) \left(\frac{4}{\pi} \frac{\|\mathbf{x} - \mathbf{x}_i\|}{R(\mathbf{x}_i)} + \sqrt{1 - (\mathbf{n} \cdot \mathbf{n}_i)} \right), \tag{3.7}$$

where:

$$\begin{aligned} \mathbf{x}_i &= \text{irradiance sample location} \\ \mathbf{n}_i &= \text{surface normal at } \mathbf{x}_i \\ E(\mathbf{x}_i) &= \text{irradiance at } \mathbf{x}_i \\ R(\mathbf{x}_i) &= \text{harmonic mean distance to surfaces from } \mathbf{x}_i. \end{aligned} \tag{3.8}$$

$R(\mathbf{x}_i)$ represents the average distance to surfaces that are visible from the sampling point, and corresponds to the radius of the hypothetical "split-sphere". It can be computed during rendering by measuring the length of the rays cast from that location during computation of Equation 3.3 as:

$$R(\mathbf{x}_i) = \frac{MN}{\sum_{j=0}^{M-1} \sum_{k=0}^{N-1} \frac{1}{r_{j,k}}}, \tag{3.9}$$

where $r_{j,k}$ is the sample distance corresponding to the $(j,k)^{\text{th}}$ ray.

As mentioned in the previous section, the weight for each irradiance cache record is given by the inverse of this estimated error bound. The estimated irradiance at a new location $\mathbf{x}$ is then the weighted average:

$$E(\mathbf{x}, \mathbf{n}) \approx \frac{\sum_{i \in S} w_i(\mathbf{x}, \mathbf{n}) E(\mathbf{x}_i)}{\sum_{i \in S} w_i(\mathbf{x}, \mathbf{n})}, \tag{3.10}$$

where:

$$\begin{aligned} S &= \left\{ i : w_i(\mathbf{x}, \mathbf{n}) > \frac{1}{a} \right\} \\ a &= \text{user set error threshold} \end{aligned} \tag{3.11}$$

Ward et al. further simplified the weight computation by removing the constant terms from Equation 3.7, since they are valid only for the hypothetical ”split-sphere” environment. The weight then becomes:

$$w_i(\mathbf{x}, \mathbf{n}) = \frac{1}{\frac{\|\mathbf{x}-\mathbf{x}_i\|}{R(\mathbf{x}_i)} + \sqrt{1 - \mathbf{n} \cdot \mathbf{n}_i}}. \tag{3.12}$$

This formulation of irradiance averaging easily supports the computation of the maximum distance from $\mathbf{x}_i$ at which the weight $w_i(\mathbf{x}, \mathbf{n})$ will still be over the $\frac{1}{a}$ error threshold, which is given simply by $aR(\mathbf{x}_i)$. This observation allowed Ward et al. to store the irradiance cache records in an octree, which enables efficient search for all records that will be valid for a certain shading location $\mathbf{x}$ in order to calculate Equation 3.10. It is straightforward to see that cache records at locations that are surrounded by ”nearby” geometry surfaces will have small valid radii, while records at locations that are far from any visible geometric features will have larger valid radii.

Weighting Function Variations

A few more notes should be given regarding the weighting function. As defined, it has the unfortunate property that it does not fall to zero at the cache record boundary, which Krivanek and Gautron [KG09] showed can lead to distracting artifacts in the final image, and proposed to fix this by subtracting $\frac{1}{a}$ from the cache weight, resulting in:

$$w_i^k(\mathbf{x}, \mathbf{n}) = \frac{1}{\frac{\|\mathbf{x}-\mathbf{x}_i\|}{R(\mathbf{x}_i)} + \sqrt{1 - \mathbf{n} \cdot \mathbf{n}_i}} - \frac{1}{a}. \tag{3.13}$$

When using this modified weight formula, cache records are used as long as $w_i^k(\mathbf{x}, \mathbf{n})$ is greater than zero.

Tabellion and Lamorlette [TL04] also proposed a variation to the weighting function:

$$w_i^t(\mathbf{x}, \mathbf{n}) = 1 - \epsilon_i(\mathbf{x}, \mathbf{n}), \tag{3.14}$$

where the weights also fall off to zero at the cache record boundaries (note that Tabellion and Lamorlette also extensively modified the error term, $\epsilon_i(\mathbf{x}, \mathbf{n})$, though

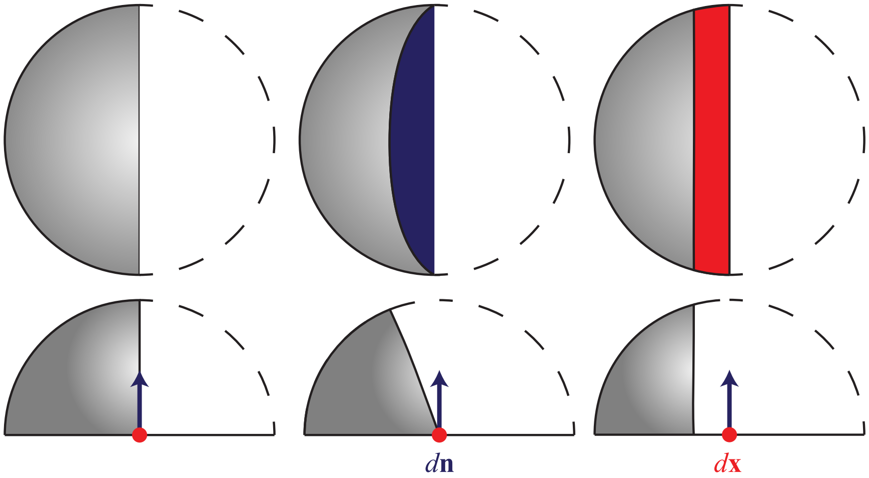


Figure 3.4: Top- and side-views of the "Split-Sphere" model, illustrating the method for computing the change in irradiance with rotation (purple) and translation (red).

we ignore this as it does not affect the weight computation). This formulation has the added benefit that $w_i^t(\mathbf{x}, \mathbf{n}) \in [0, 1]$, whereas the value for either Ward et al's or Krivanek and Gautron's weights is unbounded and goes to infinity close to the recorded cache record location, which can also lead to artifacts.

3.3.2 Derivation of the Model

We now present a more rigorous derivation of the irradiance gradient described in the previous section, and illustrate the process in Figure 3.4. The change in irradiance is dependent upon the translation of $\mathbf{x}$ and the rotation angle of the surface normal, θ. A conservative upper bound on this change is given by the first order Taylor expansion about $\mathbf{x}_i$:

$$\epsilon(\mathbf{x}, \theta) = E_i \left(\frac{\partial E}{\partial \mathbf{x}} (\mathbf{x} - \mathbf{x}_i) + \frac{\partial E}{\partial \theta} (\theta - \theta_i) \right). \tag{3.15}$$

Within the hypothetical "split-sphere" environment, the irradiance at a sample point will be proportional to the projected area of the bright half of the hemisphere. As $\mathbf{x}$ is translated, this area will incur change that can be approximated by the ratio of the added area (shown red in Figure 3.4.) to the old area:

$$\frac{\partial E}{\partial \mathbf{x}} \Delta \mathbf{x} \approx \frac{2R\Delta \mathbf{x}}{\frac{1}{2}\pi R^2} = \frac{4\Delta \mathbf{x}}{\pi R}. \tag{3.16}$$

where the numerator is an upper bound on the added area, and is a good approximation when $\Delta\mathbf{x}$ is small.

In the case of rotation of the surface normal, we use the same technique. In order to maximize the change in irradiance, the axis of rotation should be aligned with the line splitting the sphere into dark and bright halves. By looking at Figure 3.4, we can see that this change in orientation induces a change in the projected area of the bright half of the hemisphere, shaped as one half of an ellipse. The area of an ellipse is given by $\pi r_1 r_2$ and, in our case, $r_1 = R$ is simply the radius of the "split-sphere" while $r_2 = R \sin \Delta\theta$. We again take the ratio of the added area to the old area and have

$$\frac{\partial E}{\partial \theta} \Delta\theta \approx \frac{\frac{1}{2}\pi R^2 \sin \Delta\theta}{\frac{1}{2}\pi R^2} = \sin \Delta\theta. \tag{3.17}$$

We can now substitute these terms into Equation 3.15:

$$\epsilon(\mathbf{x}, \theta) = E_i \left(\frac{4}{\pi} \frac{|\mathbf{x} - \mathbf{x}_i|}{R} + \sin(\theta - \theta_i) \right). \tag{3.18}$$

This derivation has assumed a fixed, imaginary split-sphere environment that is a constant distance R away from $\mathbf{x}$. However, we need to apply this gradient to the actual scene that is sampled using Equation 3.3. Generalization of the above formula into vector-based quantities brings us to Equation 3.7.

3.3.3 Limitations of the Model

It can be seen by inspecting Equation 3.7 that the irradiance gradient implied by the "split-sphere" model is defined purely by the geometry in the scene - while there is an E_i term corresponding to the irradiance at the sample location, this acts only as a scaling factor. The model disregards the actual radiometric configuration of the scene. While this leads to a useful heuristic, it suffers from several limitations that make it harder to obtain high-quality results.

One of the main assumptions underlying the model is that there are no concentrated sources of indirect illumination. As stated previously, Ward et al. reasoned that any such sources should be taken into account as direct sources and, thus, not be a part of the indirect illumination at all. However, there are many examples in real scenes where strong sources of indirect illumination arise: the sun shining through an open door, for example. If it has a material with a high albedo, the surface that the sun shines on will become very bright - in effect, becoming an indirect diffuse light source. In this case, the irradiance gradient estimated using the "split-sphere" model will be too low, and the resulting cache will have too few samples in that area, leading to a sub-optimal result.

Another limitation arises from its geometric nature. Since the "split-sphere" model doesn't take into account the actual amounts of light that the various parts of the environment cast upon the sample location, but only the distances to them, the cache record distribution doesn't properly adapt to changes in the relative brightness of objects in the scene. For example, imagine a scene where one of the walls was painted perfectly black, such that its albedo is 0. If we render this scene using the "split-sphere" heuristic, then many irradiance cache samples will be placed along the floor, close to that wall, even though the wall induces no changes whatsoever to the irradiance at any point on the floor.

These limitations, precisely, give rise to the new method we propose in this thesis for bounding the change in the irradiance, and are analyzed further in Section 4.2.1.

3.4 First-Order Irradiance Gradients

As described thus far, the irradiance caching algorithm produces a piecewise-constant approximation of the irradiance on a surface, due to the extremely simple weighted average interpolation scheme. Ward and Heckbert [WH92] realized that it is possible to greatly improve the quality of this estimate by making better use of the information provided by the irradiance sampling process. While the "split-sphere" model provides only a directionless upper bound on the magnitude of the irradiance gradient, suitable for deciding the placement of cache records, it is also possible to compute the *actual* irradiance gradient from the information available in the sampled hemisphere. If both the irradiance and its gradient are available, it is possible to formulate a higher-order interpolation of the irradiance samples.

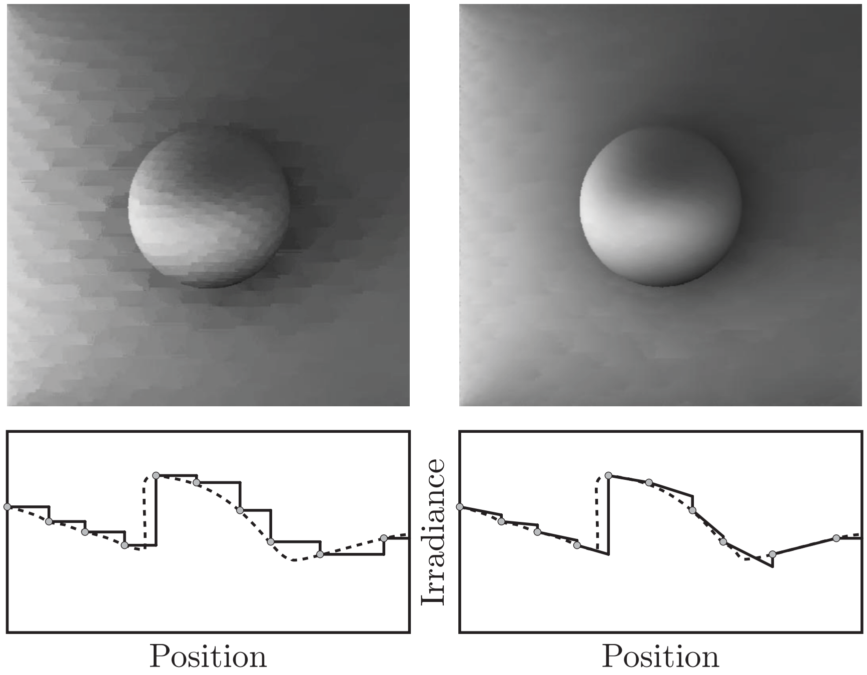


Figure 3.5: Simple scene rendered to show the advantage of using the irradiance gradients to improve the quality of the irradiance reconstruction. Images from Ward and Heckbert. [WH92].

Sampling the hemisphere through ray-tracing (Equation 3.2) provides us with not only the final approximation of the irradiance, but also tells us the direction, distance and brightness for each sample ray. Using this information, it is possible to compute an accurate approximation of the gradient to the sampled irradiance function. Since the irradiance is a five-dimensional function (three for position and two for direction), the irradiance gradient is a five-dimensional vector. However, it is more convenient to separate the gradient into its translational and rotational components, and compute both separately as two three-dimensional vector quantities, $\nabla_t E$ and $\nabla_r E$, respectively.

We extend Equation 3.10 to take advantage of this information, resulting in:

$$E(\mathbf{x}, \mathbf{n}) \approx \frac{\sum_{i \in S} w_i(\mathbf{x}, \mathbf{n})(E(\mathbf{x}_i) + (\mathbf{n}_i \times \mathbf{n}) \cdot \nabla_r E(\mathbf{x}_i) + (\mathbf{x} - \mathbf{x}_i) \cdot \nabla_t E(\mathbf{x}_i))}{\sum_{i \in S} w_i(\mathbf{x}, \mathbf{n})}. \tag{3.19}$$

This modification adds the contribution of the translational and rotational gradients as we move and / or rotate the surface away from a cache record's location, and can be interpreted as performing a piecewise-linear interpolation of the irradiance function (see Figure 3.5).

3.4.1 Derivation of the Rotational Gradient

Derivation of the rotational component of the irradiance gradient is simpler than that of the translational component, so we shall tackle it first. Recalling Equation 2.5 we have that the irradiance at $\mathbf{x}$ is equal to the integration over the visible hemisphere of the incoming radiance weighted by the cosine foreshortening term:

$$E(\mathbf{x}) = \int_{\Omega} L(\mathbf{x} \leftarrow \vec{\omega})(\mathbf{n} \cdot \vec{\omega}) d\vec{\omega}. \tag{3.20}$$

As we rotate the surface at $\mathbf{x}$, the incoming radiance $L(\mathbf{x} \leftarrow \vec{\omega})$ remains constant for any given direction $\vec{\omega}$, but the foreshortening term changes. Transforming Equation 2.5 to polar coordinates we have:

$$E(\mathbf{x}) = \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi/2} L(\mathbf{x}, \theta, \phi) \cos\theta d\theta d\phi, \tag{3.21}$$

such that the rotational gradient becomes:

$$\begin{aligned} \nabla_r E(\mathbf{x}) &= \nabla_r \left(\int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi/2} L(\mathbf{x}, \theta, \phi) \cos\theta d\theta d\phi \right) \\ &= \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi/2} L(\mathbf{x}, \theta, \phi) \nabla_r (\cos\theta) d\theta d\phi. \end{aligned} \tag{3.22}$$

Rotations of the surface that change ϕ have no effect. Only rotations that change θ affect the irradiance, and we have that $\frac{d}{d\theta}\cos\theta = -\sin\theta$. In order to compute the rotational gradient for the entire hemisphere, we integrate this differential change over the entire hemisphere, multiplied by the corresponding rotation axis $\hat{\mathbf{v}}$ (see Table 3.1):

$$\nabla_r E(\mathbf{x}) = -\int_{\phi=0}^{2\pi} \hat{\mathbf{v}} \int_{\theta=0}^{\pi/2} L(\mathbf{x}, \theta, \phi) \sin\theta d\theta d\phi. \tag{3.23}$$

The gradient resulting from Equation 3.23 can be approximated through Monte Carlo integration. In the irradiance caching algorithm the hemispherical

Table 3.1: Definitions used for the derivation of the translational irradiance gradient.

Symbol	Description
N	Number of hemisphere divisions in azimuthal angle ϕ
M	Number of hemisphere divisions in elevation angle θ
$\vec{\omega}_{j,k}$	Sample direction for the $(j,k)^{\text{th}}$ cell
$A_{j,k}$	Area of the $(j,k)^{\text{th}}$ cell, $(MN)^{-1}pdf(\vec{\omega}_{j,k})$
ϕ_k	Azimuthal angle at the center of cells $(\cdot,k)$
θ_j	Elevation angle at the center of cells $(j,\cdot)$
$(j_-,k),(j_+,k)$	Cell boundaries $(j \leftrightarrow j-1,k)$ and $(j \leftrightarrow j+1,k)$
$(j,k_-),(j,k_+)$	Cell boundaries $(j,k \leftrightarrow k-1)$ and $(j,k \leftrightarrow k+1)$
ϕ_{k_-}	Azimuthal angle at boundary $(\cdot,k_-)$
ϕ_{k_+}	Azimuthal angle at boundary $(\cdot,k_+)$
θ_{j_-}	Elevation angle at boundary $(j_-,\cdot)$
θ_{j_+}	Elevation angle at boundary $(j_+,\cdot)$
$\hat{\mathbf{u}}_k$	Tangent vector in the ϕ_k direction
$\hat{\mathbf{v}}_k$	Tangent vector in the $\phi_k + \frac{\pi}{2}$ direction
$\hat{\mathbf{v}}_{k_-}$	Tangent vector in the $\phi_{k_-} + \frac{\pi}{2}$ direction

samples are constructed using a cosine-weighted distribution, and thus we have:

$$\begin{aligned}\nabla_r E(\mathbf{x}) &\approx -\frac{\pi}{MN}\sum_{k=0}^{N-1}\left\{\hat{\mathbf{v}}\sum_{j=0}^{M-1} L(\mathbf{x},\theta_j,\phi_k)\frac{\sin\theta_j}{\cos\theta_j}\right\} \\ &= -\frac{\pi}{MN}\sum_{k=0}^{N-1}\left\{\hat{\mathbf{v}}\sum_{j=0}^{M-1} L(\mathbf{x},\theta_j,\phi_k)\tan\theta_j\right\}. \qquad (3.24)\end{aligned}$$

3.4.2 Derivation of the Translational Gradient

Derivation of the translational gradient is more involved than that for rotation, because we must now take possible changes in visibility into account: as we translate around $\mathbf{x}$, the surfaces visible to us will change. Ward and Heckbert's original derivation of the translation gradient looked at the change in area of the projected hemispherical cells as the sampling point is translated, deriving it from the independent motions of its walls along two almost-perpendicular vec-

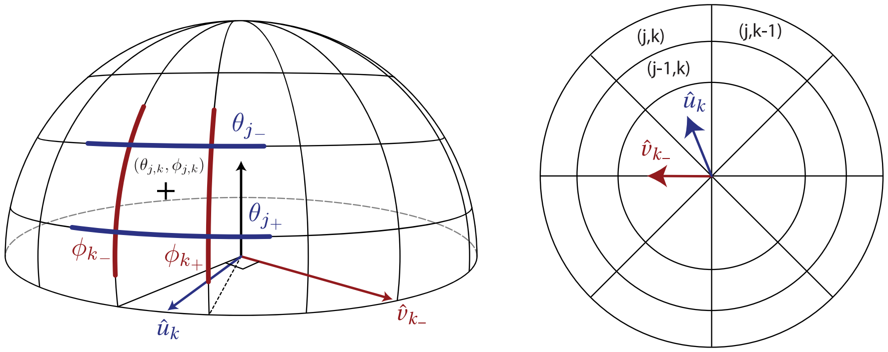


Figure 3.6: The stratified hemisphere used by Ward and Heckbert [WH92] in deriving the irradiance gradients. Figure from Jarosz [Jar08].

tors. Krivanek et al. [KGBP05] and then Jarosz [JSKJ12] presented closely related derivations that instead track these motions in the un-projected cells, and it is this approach that we present here. Notation for the following discussion is shown in Table 3.1 and illustrated in Figure 3.6.

We can think of the irradiance computation of Equation 3.2 as a weighted sum of the incoming radiance, where the contribution of each sample is the product of the radiance through $L(\mathbf{x} \leftarrow \vec{\omega}_{j,k})$, the solid angle of the corresponding hemispherical cell $A_{j,k}$, and the foreshortening term $\mathbf{n} \cdot \vec{\omega}_{j,k}$:

$$E(\mathbf{x}) \approx \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} A_{j,k} L(\mathbf{x} \leftarrow \vec{\omega}_{j,k})(\mathbf{n} \cdot \vec{\omega}_{j,k}). \tag{3.25}$$

$A_{j,k}$ is simply the integral over the bounds of each cell:

$$\begin{aligned} A_{j,k} &= \int_{\phi_{k_-}}^{\phi_{k_+}} \int_{\theta_{j_-}}^{\theta_{j_+}} \sin\theta d\theta d\phi \\ &= (\cos\theta_{j_-} - \cos\theta_{j_+})(\phi_{k_+} - \phi_{k_-}). \end{aligned} \tag{3.26}$$

We can obtain the translational gradient by differentiation of Equation 3.25:

$$\nabla_t E(\mathbf{x}) \approx \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} \nabla_t (A_{j,k} L_{j,k} \cos\theta_j).$$

Since translation of the projection center does not affect the incoming radiance from any cell and, furthermore, maintains the cosine term constant, we can simplify the expression to:

$$\nabla_t E(\mathbf{x}) \approx \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} \nabla_t (A_{j,k}) L_{j,k} \cos\theta_j. \tag{3.27}$$

We are only interested in the irradiance gradient along a surface - the irradiance function is discontinuous if one looks at translation in the normal direction. We can then reduce the change in cell area $A_{j,k}$ to computing the differential change in the cell walls with respect to translation along the base plane. For each cell, we express the gradient with respect to directional derivatives of the four neighboring cell walls:

$$\nabla_t A_{j,k} = \hat{\mathbf{u}}_k \cdot \nabla_{\hat{\mathbf{u}}_k} A_{j_-,k} - \hat{\mathbf{u}}_k \cdot \nabla_{\hat{\mathbf{u}}_k} A_{j_+,k} + \hat{\mathbf{v}}_{k_-} \cdot \nabla_{\hat{\mathbf{v}}_{k_-}} A_{j,k_-} - \hat{\mathbf{v}}_{k_+} \cdot \nabla_{\hat{\mathbf{v}}_{k_+}} A_{j,k_+}, \tag{3.28}$$

where $\nabla_{\hat{\mathbf{u}}_k} A_{j_-,k}$ is the scalar directional derivative of $A_{j_-,k}$ due to movement of boundary (j_-, k) in the $\hat{\mathbf{u}}_k$ direction, and the other quantities are similarly defined (see Table 3.1). Each of these derivatives can be interpreted as the product of the length of the respective cell wall with its rate of motion due to translation in the perpendicular direction.

While the gradient in Equation 3.27 is written in term of the translation gradients of the cells, it is more convenient to express it as a summation over all cell boundaries instead, given that neighboring cells share these. The movement of one of the cell walls will induce a larger contribution from a given cell, while reciprocally diminishing the contribution from the neighbor sharing that wall. Because of this, we consider the difference in incoming radiance between the two adjacent cells in computing the effect on the overall irradiance:

$$\begin{aligned} \nabla_t E(\mathbf{x}) \approx \sum_{k=0}^{N-1} & \left(\hat{\mathbf{u}}_k \sum_{j=1}^{M-1} \nabla_{\hat{\mathbf{u}}_k} A_{j_-,k} (L_{j,k} - L_{j-1,k}) \cos\theta_{j_-} \right) \\ & + \left(\hat{\mathbf{v}}_{k_-} \sum_{j=0}^{M-1} \nabla_{\hat{\mathbf{v}}_{k_-}} A_{j,k} (L_{j,k} - L_{j,k-1}) \cos\theta_j \right). \end{aligned} \tag{3.29}$$

We derive the rate of motion of each cell type in the following sections.

Translation Along $\hat{\mathbf{u}}$

Translation of cell wall (j_-, k) along $\hat{\mathbf{u}}_k$ induces a change in cell area given by:

$$\begin{aligned}\nabla_{\hat{\mathbf{u}}_k} A_{j_-,k} &= \nabla_{\hat{\mathbf{u}}_k} \theta_{j_-} \cdot \frac{\partial A_{j,k}}{\partial \theta_{j_-}}, \\ &= \nabla_{\hat{\mathbf{u}}_k} \theta_{j_-} \cdot (\phi_{k_+} - \phi_{k_-}) \sin \theta_{j_-}, \qquad (3.30)\end{aligned}$$

where $(\phi_{k_+} - \phi_{k_-}) \sin \theta_{j_-}$ arises from the differentiation of $A_{j,k}$ as defined in Equation 3.26. We now only need to determine the change in the elevation angle θ_{j_-}, for which we turn to the canonical hemispherical parametrization:

$$x = r \cos \phi \sin \theta, \qquad (3.31)$$

$$y = r \sin \phi \sin \theta, \qquad (3.32)$$

$$z = r \cos \theta, \qquad (3.33)$$

$$r = \sqrt{x^2 + y^2 + z^2}, \qquad (3.34)$$

$$\phi = \arctan\left(\frac{y}{x}\right), \qquad (3.35)$$

$$\theta = \arccos\left(\frac{z}{r}\right), \qquad (3.36)$$

and determine the change in θ as we translate along x:

$$\begin{aligned}\frac{d\theta}{dx} &= \frac{zx}{r^3\sqrt{1 - \frac{z^2}{r^2}}}, && \text{by differentiating Equation 3.36,} \\ &= \frac{\cos\theta x}{r^2\sqrt{1-\cos^2\theta}}, && \text{since } \frac{z}{r} = \cos\theta, \\ &= \frac{\cos\theta x}{r^2 \sin\theta}, && \text{because } \sin^2\theta + \cos^2\theta = 1, \\ &= \frac{\cos\theta\cos\phi\sin\theta}{r\sin\theta}, && \text{since } \frac{x}{r} = \cos\phi\sin\theta, \\ &= \frac{\cos\theta}{r}, && \text{because at } y = 0,\ \cos\phi = 1. \qquad (3.37)\end{aligned}$$

In order to apply this derivation to the hemispherical samples, we need to determine r. Ward and Heckbert [WH92] made the observation that, as the sample location moves, the rate of motion of the boundary between two neighboring cells depends upon the *closer* of the two samples - this observation takes the changes in occlusion that happen with translation into account. We therefore have:

$$\nabla_{\hat{\mathbf{u}}_k} A_{j_-,k} = \frac{(\phi_{k_+} - \phi_{k_-}) \sin\theta_{j_-} \cos\theta_{j_-}}{min(r_{j,k}, r_{j-1,k})}. \tag{3.38}$$

Translation Along $\hat{\mathbf{v}}$

In the case of cell walls of the form (j, k_-), we consider motion along $\hat{\mathbf{v}}_{k_-}$, where the change in area is given by:

$$\begin{aligned} \nabla_{\hat{\mathbf{v}}_{k_-}} A_{j,k_-} &= \nabla_{\hat{\mathbf{v}}_{k_-}} \phi_{k_-} \cdot \frac{\partial A_{j,k}}{\partial \phi_{k_-}}, \\ &= \nabla_{\hat{\mathbf{v}}_{k_-}} \phi_{k_-} \cdot (\cos\theta_{j_+} - \cos\theta_{j_-}). \end{aligned} \tag{3.39}$$

We again turn to the canonical parametrization in order to compute $\nabla_{\hat{\mathbf{v}}_{k_-}} \phi_{k_-}$:

$$\begin{aligned} \frac{d\phi}{dy} &= \frac{x}{x^2 + y^2}, && \text{by differentiating Equation 3.35,} \\ &= \frac{1}{x}, && \text{since we evaluate at } y = 0, \\ &= \frac{1}{r\cos\phi\sin\theta}, && \text{from Equation 3.31,} \\ &= \frac{1}{r\sin\theta}, && \text{because } \cos\phi = 1 \text{ at } y = 0. \end{aligned} \tag{3.40}$$

Again, we use the minimum of the neighboring cell distances for r:

$$\nabla_{\hat{\mathbf{v}}_{k_-}} A_{j,k_-} = \frac{(\cos\theta_{j_+} - \cos\theta_{j_-})}{\sin\theta_j min(r_{j,k}, r_{j-1,k})}. \tag{3.41}$$

Bringing everything together

We can now use the values derived for the cell wall differentials from Equations 3.38 and 3.41 in Equation 3.29 to get the final translation gradient:

$$\begin{aligned} \nabla_t E(\mathbf{x}) \approx \sum_{k=0}^{N-1} &\left(\hat{\mathbf{u}}_k \sum_{j=1}^{M-1} \frac{(\phi_{k_+} - \phi_{k_-}) \sin\theta_{j_-} \cos\theta_{j_-}}{min(r_{j,k}, r_{j-1,k})} (L_{j,k} - L_{j-1,k}) \cos\theta_{j_-} \right) \\ &+ \left(\hat{\mathbf{v}}_{k_-} \sum_{j=0}^{M-1} \frac{(\cos\theta_{j_+} - \cos\theta_{j_-})}{\sin\theta_j min(r_{j,k}, r_{j-1,k})} (L_{j,k} - L_{j,k-1}) \cos\theta_j \right). \end{aligned} \tag{3.42}$$

3.5 Tricks and Extensions

Several extensions to the original irradiance caching algorithm have been proposed in the literature, that attempt to work around some of the limitations of the split-sphere heuristic and improve the quality of the images rendered using the method. There is also a multitude of tricks that implementers have come up with to address these issues. We briefly summarize some of the most relevant ones in this section.

3.5.1 Overture Pass

The original irradiance caching algorithm, as described by Ward et al. [WRC88] and in Section 3.1, can result in distracting artifacts in the final rendered image. In many areas, there might only be a single suitable irradiance cache record from which the irradiance at a new location is extrapolated, resulting in a piecewise-constant approximation of the true irradiance. Before the introduction of first-order gradients by Ward and Heckbert [WH92], the use of an overture pass was a necessity for rendering pleasing images using irradiance caching.

When using an overture pass, the final image is computed in a multi-pass fashion: during the first pass, the irradiance cache is filled by rendering the entire image without actually computing the actual pixel values, which are computed during a second rendering pass. This approach ensures that the irradiance at all points in the image is always reconstructed by *interpolation* of at least two irradiance cache records, greatly reducing the appearance of distracting artifacts. The greatest drawback of this extension to the algorithm is the requirement to render the image in multiple passes, which Ward and Heckbert [WH92] acknowledge as one of the motivations behind their work on first-order irradiance gradients, aimed precisely at improving the quality of a single-pass irradiance reconstruction.

While originally used to reduce the artifacts from the piecewise-constant approximation of irradiance produced by the classical algorithm, this technique is still used in most modern implementations to improve the quality of the resulting images. Part of the motivation for this thesis is to improve the algorithm such that it can be used as a robust single-pass method.

3.5.2 Gradient-Based Record Density Control

This small extension was introduced by Ward and Heckbert [WH92] along with the use of the first-order gradients to improve the interpolation quality of the algorithm. Their observation was that, since the *actual* irradiance gradient could be computed from the sampling information, it was possible to use it instead of the split-sphere bound in guiding the irradiance cache record density. However, some locations in the scene might have a very small gradient even if nearby geometric features might lead to sudden changes in illumination, leading to artifacts in the image if only the actual gradient is used. They therefore proposed to use the *minimum* of both gradients - generally bounding the cache record radii using the gradient estimated using the split-sphere model, but switching to the *true* gradient approximation for regions of the image where the split-sphere might underestimate it.

While Ward and Heckbert fail to mention it, their observation and decision to use the first-order irradiance gradient as a bound for the irradiance cache record radii points to a fundamental deficiency in the split-sphere model: its hypothetical basis precludes it from taking the *actual* illumination in a scene into account. It should be noted that, as derived, the split-sphere model was defined to be a *conservative* bound to the possible change in irradiance with rotation and translation. As we have explained, though, this only holds if there are no concentrated sources of illumination in the environment. In reality this is seldom the case, and there usually are at least some such sources in most scenes.

The improvement afforded by this extension can be great in areas where the split-sphere heuristic under-estimates the true first-order gradient, but is unfortunately not enough to produce high-quality cache record distributions: in many areas the true gradient is too small to effectively control the record density, and extensive radius clamping is often still necessary to render pleasing images with irradiance caching. This observation is one of the strongest motivations for the work presented in this thesis.

3.5.3 Hierarchical Refinement

Irradiance caching uses a ”lazy evaluation” technique. As rendering progresses to a new pixel in the image, the cache is checked for records that are valid at the new location and, if none are found, the irradiance is computed using Equation 3.3 and a new record is stored; if suitable records are found, irradiance is interpolated from them. This leads to a dependence between the cache records computed for a scene and the order in which the pixels are visited - usually, a scanline order. Furthermore, it creates a problem in that pixels that have already been evaluated might fall into the validity region of newly created cache records; since those pixels will never be re-evaluated, unless an overture pass is used, the contribution of the new records is never taken into account, resulting many times in distracting image artifacts.

Hierarchical refinement is a technique that tries to deal with this situation. For a chosen value n, the renderer first processes only one out of every 2^n pixels of a scanline and only one out of every 2^n scanlines. A subsequent pass renders one out of every 2^{n-1} pixels, for one out of every 2^{n-1} scanlines, effectively quadrupling the number of image locations rendered, while skipping the pixels that have already been processed. This continues until all pixels in the image have been processed.

This process generally results in better irradiance cache record distributions and less evident image artifacts. The problem of some pixels not receiving the contribution from suitable records that were computed later in the process still remains, but is greatly ameliorated by the way in which the cache is progressively refined. Most implementations of irradiance caching combine hierarchical refinement with an overture pass, greatly improving the resulting image quality. Even so, this technique does nothing to address the more fundamental problems with the split-sphere heuristic, and extensive cache record radius clamping is often still required to produce acceptable images.

3.5.4 Neighbor Clamping

Whenever a new cache record needs to be computed, the irradiance at a scene location is computed using Equation 3.3. Because the sample rays used do not cover all directions in the hemisphere, small geometric features in the scene can be missed entirely, leading to a mean-distance to surfaces $R(\mathbf{x}_i)$ that is too large, producing a low local record density. If the missed feature is a strong source of indirect illumination, this can lead to visible image artifacts. Furthermore, the stochastic nature of the hemisphere sampling means that a nearby record might not miss the relevant geometric feature, exacerbating these artifacts.

One solution to this problem was proposed by Tabellion and Lamorlette [TL04], who use the minimum distance to surfaces, instead of the harmonic-mean, for $R(\mathbf{x}_i)$. However, this can lead to an overzealous algorithm that places too much importance on even minuscule features and generally results in caches that are denser than required.

Instead, neighbor clamping was proposed by Krivanek et al. [KBPZ06]. Two observations motivate this technique: first, as was mentioned earlier, that the stochastic nature of hemisphere sampling means that small features missed at any given cache record location will probably be "seen" by a neighboring record; second, that basic geometric coherence of the scene dictates that the radius of any given cache record should differ from that of its neighbors by no more than the distance between them - essentially satisfying the triangle inequality. While this second observation is not exactly true unless the minimum distance to surfaces is used for $R(\mathbf{x}_i)$, the resulting algorithm works surprisingly well even when using the harmonic-mean.

Technically, the algorithm works as follows: when a new record i is being added to the cache, all nearby records j are located and the value of $R(\mathbf{x}_i)$ is clamped to $R(\mathbf{x}_i) = min\{R(\mathbf{x}_i), R(\mathbf{x}_j) + \|\mathbf{x}_i - \mathbf{x}_j\|\}$. After this, the nearby records' $R(\mathbf{x}_j)$ value is similarly clamped using the new $R(\mathbf{x}_i)$ value.

Because neighbor clamping progressively modifies the irradiance cache, using an overture pass is necessary. The final image is only rendered once the entire cache has been computed.

3.5.5 Adaptive Caching

This technique was introduced by Krivanek et al. [KBPZ06] in the context of the *Radiance Caching* rendering algorithm, but is just as applicable to irradiance caching, and attempts to further improve the *perceived* quality of images rendered using it for global illumination computation.

The goal of adaptive caching is to eliminate artifacts that arise due to interpolation of the irradiance between neighboring cache records, which appear as discontinuities at the boundary of the area of influence of two or more records. This area is defined, for a record i, as $\{\mathbf{x} : w_i(\mathbf{x}, \mathbf{n}) \geq \frac{1}{a}\}$ where a is the user-defined error-threshold parameter. The idea is to modulate this parameter on a per-record basis, such that visible discontinuities due to interpolation are eliminated. A discontinuity is reported if two records have a visibly different outgoing radiance contribution anywhere in the overlap of their influence areas; wherever such discontinuities are discovered, one of the records' a parameter is reduced, effectively reducing that record's influence area. As with neighbor clamping, an overture pass is required to use this technique, so that the irradiance cache has fully converged before the actual pixel values are computed. We direct the reader to the relevant paper for an in-depth description of the algorithm, which is beyond the scope of this thesis.

While this technique can produce very good results, it fails to address any of the fundamental shortcomings of the split-sphere model, which is still the model used to derive the cache record weights. Furthermore, its implementation is fairly involved, especially in order to achieve sufficient performance - the method requires extensive re-visiting of image locations and proper tracking of each cache record's area of influence, which can easily require either excessive computation time or storage (or both) if the implementer is not careful.

Chapter 4

The Occlusion-Agnostic Irradiance Hessian

Though more than 20 years have passed since its inception, and many papers and techniques have been published to improve the irradiance caching algorithm, the venerable "Split-Sphere" heuristic has been used without major modification as the error metric to define, during rendering, whether to compute and add a new record or use those already in the cache. Minor changes have been proposed: Tabellion and Lamorlette [TL04] used the minimum instead of the harmonic-mean distance as the "split-sphere" radius; Krivanek et al. [KBPZ06] proposed several multi-pass techniques to improve upon the results obtained by using the split-sphere heuristic.

Jarosz et al. [JSKJ12] proposed, for the first time, to do away with the split-sphere heuristic entirely. Following a complete derivation of light transport in two dimensions, they observed that it is possible to follow a more principled approach in bounding the error induced by a first-order Taylor expansion approximation of the irradiance in a scene. Instead of deriving a loose error bound based on the hypothetical "split-sphere" environment, they derived the second-order translation gradient of irradiance - the irradiance Hessian - and showed how it can be used to guide the local record density of an irradiance cache, but stopped short of describing a practical replacement for the split-sphere heuristic. Their derivation of the irradiance Hessian fails to take object inter-occlusions into account - an interesting feature in many scenes, leading to important illumination phenomena. The rotational component of the gradient is also not taken into account. Their work serves as a base on which we build our new, practical irradiance cache error control method, and so we present in this chapter the most relevant derivations in that work.

4.1 Derivation of the Occlusion-Agnostic Translation Hessian

Jarosz et al. derived the translation component of the irradiance first- and second-order gradients by explicitly differentiating the *surface-area* formulation of the irradiance equation. We can transform Equation 2.5 into this form following the same steps we did for the rendering equation in Section 2.3.2:

$$\begin{aligned} E(\mathbf{x}) &= \int_{\Omega} L(\mathbf{x} \leftarrow \vec{\omega})(\mathbf{n} \cdot \vec{\omega}) d\vec{\omega}, \\ &= \int_{A} L(\mathbf{x} \leftarrow \mathbf{y}) V(\mathbf{x} \leftrightarrow \mathbf{y}) G(\mathbf{x} \leftrightarrow \mathbf{y}) dA(\mathbf{y}), \end{aligned} \tag{4.1}$$

where we have again used the geometric coupling term, $G(\mathbf{x} \leftrightarrow \mathbf{y})$, to transform the integration over the hemisphere of directions into an integration over points on the surfaces of scene geometry, and the visibility function $V(\mathbf{x} \leftrightarrow \mathbf{y})$ to explicitly account for the inter-visibility of points in the scene.

Differentiating this equation once, applying the product rule and omitting obvious parameters, we arrive at the following definition of the irradiance gradient:

$$\begin{aligned} \nabla_{\mathbf{x}} E(\mathbf{x}) &= \nabla_{\mathbf{x}} \int_{A} L(\mathbf{x} \leftarrow \mathbf{y}) V(\mathbf{x} \leftrightarrow \mathbf{y}) G(\mathbf{x} \leftrightarrow \mathbf{y}) dA(\mathbf{y}), \\ &= \int_{A} \nabla_{\mathbf{x}} L V G + L \nabla_{\mathbf{x}} V G + L V \nabla_{\mathbf{x}} G dA(\mathbf{y}). \end{aligned} \tag{4.2}$$

In the general case, each term in this equation would need to be solved to obtain a full irradiance gradient. However, if we restrict ourselves to diffuse scenes, then the exitant radiance from $\mathbf{y}$, $L(\mathbf{x} \leftarrow \mathbf{y})$, is constant, and thus its gradient is zero. Furthermore, differentiation of the binary visibility function is tricky; if we assume

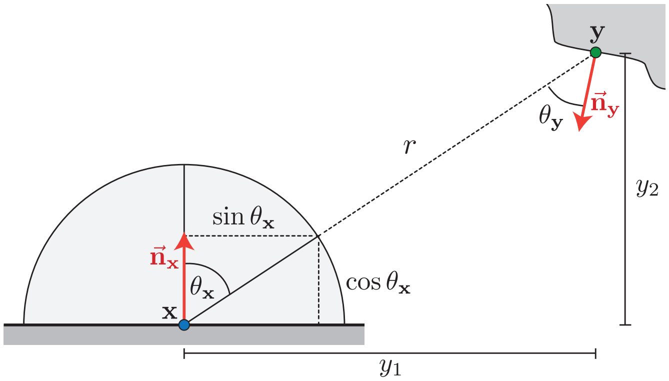


Figure 4.1: The quantities and geometric relationships used in deriving the translational derivative of the geometric term. Figure from Jarosz et al. [JSKJ12].

that there are no changes in visibility with translation, then this term of the equation is also zero. This leaves us with a much simpler equation to solve:

$$\nabla_\mathbf{x} E(\mathbf{x}) \approx \int_A L(\mathbf{x}\leftarrow\mathbf{y})V(\mathbf{x}\leftrightarrow\mathbf{y})\nabla_\mathbf{x} G(\mathbf{x}\leftrightarrow\mathbf{y})dA(\mathbf{y}), \tag{4.3}$$

where we only require to solve the gradient of the geometric term, $\nabla_\mathbf{x} G(\mathbf{x}\leftrightarrow\mathbf{y})$. Applying the product rule to that differentiation yields:

$$\nabla_\mathbf{x} G(\mathbf{x}\leftrightarrow\mathbf{y}) = (\mathbf{n_x}\cdot\vec{\omega})(\mathbf{n_y}\cdot-\vec{\omega})\nabla_\mathbf{x}\frac{1}{r^2} + \frac{\mathbf{n_x}\cdot\vec{\omega}}{r^2}\nabla_\mathbf{x}(\mathbf{n_y}\cdot-\vec{\omega}) + \frac{\mathbf{n_y}\cdot-\vec{\omega}}{r^2}\nabla_\mathbf{x}(\mathbf{n_x}\cdot\vec{\omega}), \tag{4.4}$$

where $r = \|\mathbf{x}-\mathbf{y}\|$ and we assume without loss of generality that $\mathbf{x}$ is the origin and $\mathbf{n_x} = \left(\begin{smallmatrix}0\\0\\1\end{smallmatrix}\right)$, as illustrated in Figure 4.1. The gradients of the individual terms are:

$$\nabla_\mathbf{x}\mathbf{y} = \nabla_\mathbf{x}\begin{pmatrix}y_1\\y_2\\y_3\end{pmatrix} = \begin{pmatrix}-1\\-1\\-1\end{pmatrix}, \tag{4.5}$$

$$\nabla_\mathbf{x}\frac{1}{r^n} = -\frac{n}{r^{n+1}}\nabla_\mathbf{x} r = \frac{n}{r^{n+2}}\mathbf{y}, \tag{4.6}$$

$$\nabla_\mathbf{x}\mathbf{n_x}\cdot\vec{\omega} = \nabla_\mathbf{x}\frac{\mathbf{n_x}\cdot\mathbf{y}}{r} = \frac{1}{r}\nabla_\mathbf{x}(\mathbf{n_x}\cdot\mathbf{y}) + (\mathbf{n_x}\cdot\mathbf{y})\nabla_\mathbf{x}\frac{1}{r} = -\frac{\mathbf{n_x}}{r} + \frac{\mathbf{n_x}\cdot\vec{\omega}}{r^2}\mathbf{y}, \tag{4.7}$$

$$\nabla_\mathbf{x}\mathbf{n_y}\cdot-\vec{\omega} = -\nabla_\mathbf{x}\frac{\mathbf{n_y}\cdot\mathbf{y}}{r} = \frac{\mathbf{n_y}}{r} + \frac{\mathbf{n_y}\cdot-\vec{\omega}}{r^2}\mathbf{y}. \tag{4.8}$$

Inserting these terms into Equation 4.4 gives us the following formula for the gradient of the geometric term:

$$\nabla_\mathbf{x} G(\mathbf{x}\leftrightarrow\mathbf{y}) = 4\frac{\cos\theta_\mathbf{x}\cos\theta_\mathbf{y}}{r^4}\mathbf{y} - \frac{\cos\theta_\mathbf{y}}{r^3}\mathbf{n_x} + \frac{\cos\theta_\mathbf{x}}{r^3}\mathbf{n_y}, \tag{4.9}$$

where we have substituted the dot products in the original formula for the corresponding cosines, for clarity of exposition. Inserting this expression into Equation 4.3 gives us a formula for an occlusion-agnostic irradiance gradient.

It is furthermore possible to differentiate Equation 4.3 in order to define the Hessian of irradiance, keeping the same assumptions that were used to arrive at that formulation for the gradient:

$$\mathbf{H_x}E(\mathbf{x}) \approx \int_A L(\mathbf{x}\!\leftarrow\!\mathbf{y})V(\mathbf{x}\!\leftrightarrow\!\mathbf{y})\mathbf{H_x}G(\mathbf{x}\!\leftrightarrow\!\mathbf{y})dA(\mathbf{y}). \tag{4.10}$$

We only need to differentiate the gradient of the geometric term, which we do by independently differentiating the terms in Equation 4.9:

$$\begin{aligned}
\mathbf{J_x}\left(\frac{\cos\theta_\mathbf{x}\cos\theta_\mathbf{y}}{r^4}\mathbf{y}\right) &= (\cos\theta_\mathbf{x}\cos\theta_\mathbf{y}\mathbf{y})\nabla_\mathbf{x}^\intercal\frac{1}{r^4} + \left(\frac{\cos\theta_\mathbf{x}}{r^4}\mathbf{y}\right)\nabla_\mathbf{x}^\intercal\cos\theta_\mathbf{y} \\
&\quad + \left(\frac{\cos\theta_\mathbf{y}}{r^4}\mathbf{y}\right)\nabla_\mathbf{x}^\intercal\cos\theta_\mathbf{x} + \left(\frac{\cos\theta_\mathbf{x}\cos\theta_\mathbf{y}}{r^4}\right)\mathbf{J}_\mathbf{x}^\intercal(\mathbf{y}) \\
&= 6\frac{\cos\theta_\mathbf{x}\cos\theta_\mathbf{y}}{r^6}\mathbf{y}\mathbf{y}^\intercal - \frac{\cos\theta_\mathbf{y}}{r^5}\mathbf{y}\mathbf{n}_\mathbf{x}^\intercal \\
&\quad + \frac{\cos\theta_\mathbf{x}}{r^5}\mathbf{y}\mathbf{n}_\mathbf{y}^\intercal - \frac{\cos\theta_\mathbf{x}\cos\theta_\mathbf{y}}{r^4}\mathbf{I}_3, \qquad (4.11) \\
\mathbf{J_x}\left(\frac{\cos\theta_\mathbf{y}}{r^3}\mathbf{n_x}\right) &= 4\frac{\cos\theta_\mathbf{y}}{r^5}\mathbf{n_x}\mathbf{y}^\intercal + \frac{1}{r^4}\mathbf{n_x}\mathbf{n}_\mathbf{y}^\intercal, \qquad (4.12) \\
\mathbf{J_x}\left(\frac{\cos\theta_\mathbf{x}}{r^3}\mathbf{n_y}\right) &= 4\frac{\cos\theta_\mathbf{x}}{r^5}\mathbf{n_y}\mathbf{y}^\intercal - \frac{1}{r^4}\mathbf{n_y}\mathbf{n}_\mathbf{x}^\intercal, \qquad (4.13)
\end{aligned}$$

where $\mathbf{J_x}$ denotes the Jacobian operator taken with respect to $\mathbf{x}$, and $\mathbf{I}_3$ is the 3×3 identity matrix. Combining these terms according to Equation 4.9 results in:

$$\begin{aligned}
\mathbf{H_x}G(\mathbf{x}\!\leftrightarrow\!\mathbf{y}) &= 24\frac{\cos\theta_\mathbf{x}\cos\theta_\mathbf{y}}{r^6}(\mathbf{y}\mathbf{y}^\intercal) - 4\frac{\cos\theta_\mathbf{x}\cos\theta_\mathbf{y}}{r^4}\mathbf{I}_3 \\
&\quad - \frac{1}{r^4}(\mathbf{n_x}\mathbf{n}_\mathbf{y}^\intercal + \mathbf{n_y}\mathbf{n}_\mathbf{x}^\intercal) - 4\frac{\cos\theta_\mathbf{y}}{r^5}(\mathbf{n_x}\mathbf{y}^\intercal + \mathbf{y}\mathbf{n}_\mathbf{x}^\intercal) \\
&\quad + 4\frac{\cos\theta_\mathbf{x}}{r^5}(\mathbf{n_y}\mathbf{y}^\intercal + \mathbf{y}\mathbf{n}_\mathbf{y}^\intercal). \qquad (4.14)
\end{aligned}$$

Finally, combining Equations 4.23 & 4.14 results in the formula for the occlusion-agnostic irradiance Hessian. One important feature of Equations 4.3 & 4.23 is that they can be solved using Monte Carlo integration for arbitrary sample distributions.

It is important to note that Jarosz et al. [JSKJ12] also derived an *occlusion-aware* irradiance gradient, the derivation of which was already shown in Section 3.4.2. Unfortunately, they found that extending that derivation to a Hessian is infeasible. It makes the simplification of operating on the strata boundary edges and only considering motion perpendicular to each edge which, in effect, makes the implicit assumption that, after translation of the center of projection, the strata remain rectangles in the spherical parametrization. However, in reality, the edges of each stratum might warp into an arbitrary quadrilateral. While this approximation is reasonable in the case of the gradient and, in effect, produces very good results practically in that case, it becomes increasingly problematic for higher-order differentiation, and error overwhelms the calculations.

4.2 Application to Irradiance Caching

As we saw in Section 3.3.3, one of the primary contributors to error in irradiance caching is the suboptimal placement of cache records when the algorithm is guided by the split-sphere heuristic. With the formulas for the irradiance Hessian in hand, we can investigate an alternative to that heuristic that is greatly superior.

Our goal is to use each cache record as far from its location as possible, while ensuring that the resulting extrapolation error is below some user-defined threshold. We first define the total error, ϵ^t, of a cache record as the integrated absolute difference between the true irradiance and the extrapolated irradiance:

$$\epsilon^t = \iint_A |E(\mathbf{x}_i + \mathbf{x}) - E'(\mathbf{x}_i + \mathbf{x})| d\mathbf{x}, \tag{4.15}$$

where A is the area of support of the cache record, and $E'(\mathbf{x}_i + \mathbf{x}) = E(\mathbf{x}_i) + \nabla_{\mathbf{x}} E(\mathbf{x}_i) \cdot \mathbf{x}$ is the first-order Taylor expansion of the irradiance around the i^{th} cache record. Ideally, this equation should be solved as-is, but that would require knowledge of the true irradiance $E(\mathbf{x}_i + \mathbf{x})$ - which is precisely the value we are attempting to approximate. Instead, we can use a second-order Taylor expansion of the irradiance as an oracle for this ground-truth. We approximate $E(\mathbf{x}_i + \mathbf{x}) \approx E(\mathbf{x}_i) + \nabla_{\mathbf{x}} E(\mathbf{x}_i) \cdot \mathbf{x} + \frac{1}{2}\mathbf{x}^{\intercal}\mathbf{H}_{\mathbf{x}} E(\mathbf{x}_i)\mathbf{x}$, which simplifies Equation 4.15 tremendously:

$$\epsilon^t \approx \frac{1}{2} \iint_A |\mathbf{x}^{\intercal}\mathbf{H}_{\mathbf{x}} E(\mathbf{x}_i)\mathbf{x}| d\mathbf{x}. \tag{4.16}$$

Here we assume that $\mathbf{x}$ is a 2D point on a *surface*, and that the Hessian is a 2×2 matrix defining the second derivatives of irradiance on the tangent space of such surface. This can be obtained by projecting the 3×3 Hessian matrix from Equation 4.23 onto the tangent plane. To do so, we define $\mathbf{a} = \mathbf{H}_{\mathbf{x}} E(\mathbf{x}_i)\mathbf{u}_1$ and $\mathbf{b} = \mathbf{H}_{\mathbf{x}} E(\mathbf{x}_i)\mathbf{u}_2$ and the tangential Hessian becomes:

$$\mathbf{H}_{\mathbf{x}}^{2\times 2} = \begin{bmatrix} \mathbf{u}_1 \cdot \mathbf{a} & \mathbf{u}_1 \cdot \mathbf{b} \\ \mathbf{u}_2 \cdot \mathbf{a} & \mathbf{u}_2 \cdot \mathbf{b} \end{bmatrix} \tag{4.17}$$

where $\mathbf{u}_1$ and $\mathbf{u}_2$ are any two orthonormal vectors in the tangent space of the surface.

The Hessian inside Equation 4.16 defines a quadric surface. In order to define the radius for the cache record, we conservatively bound the Hessian using the maximum curvature of the surface, which is given by λ_1, the maximum eigenvalue of $\mathbf{H}_{\mathbf{x}}^{2\times 2} E(\mathbf{x}_i)$:

$$\begin{aligned} \epsilon^t &\leq \frac{1}{2} \iint_A |\lambda_1 \mathbf{x}^2| d\mathbf{x}, \\ &\leq \pi \int_0^{R_i} |\lambda_1| r^2 r dr \\ &\leq \frac{\pi}{4} \lambda_1 R_i^4, \end{aligned} \tag{4.18}$$

where the last r term in the integral arises from the change of variables to polar coordinates. Inverting this equation and solving for R_i results in a formula for the radius of cache record i:

$$R_i = \sqrt[4]{\frac{4\epsilon^t}{\pi \lambda_1}}, \tag{4.19}$$

that induces a total integrated error below ϵ^t in the extrapolated irradiance. Jarosz et al. termed this the "Radiometric Hessian" error measure.

4.2.1 Analysis

Jarosz et al. performed a cursory evaluation of the performance of the "Radiometric Hessian" error metric, and showed that it significantly outperforms the split-sphere heuristic in a few simple but representative test cases, reducing the RMS error of the reconstructed irradiance by as much as 10x, as can be seen in

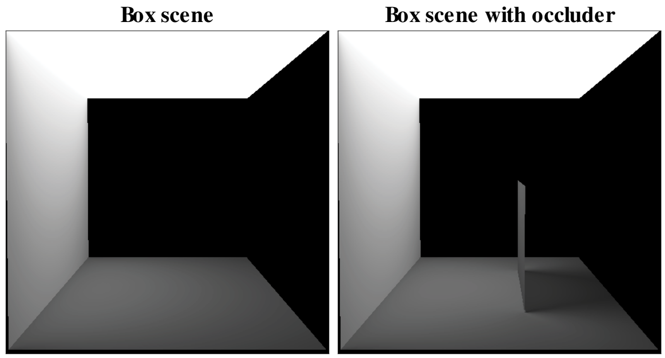


Figure 4.2: Scene configuration for the relevant comparison renderings. Figure from Jarosz et al. [JSKJ12].

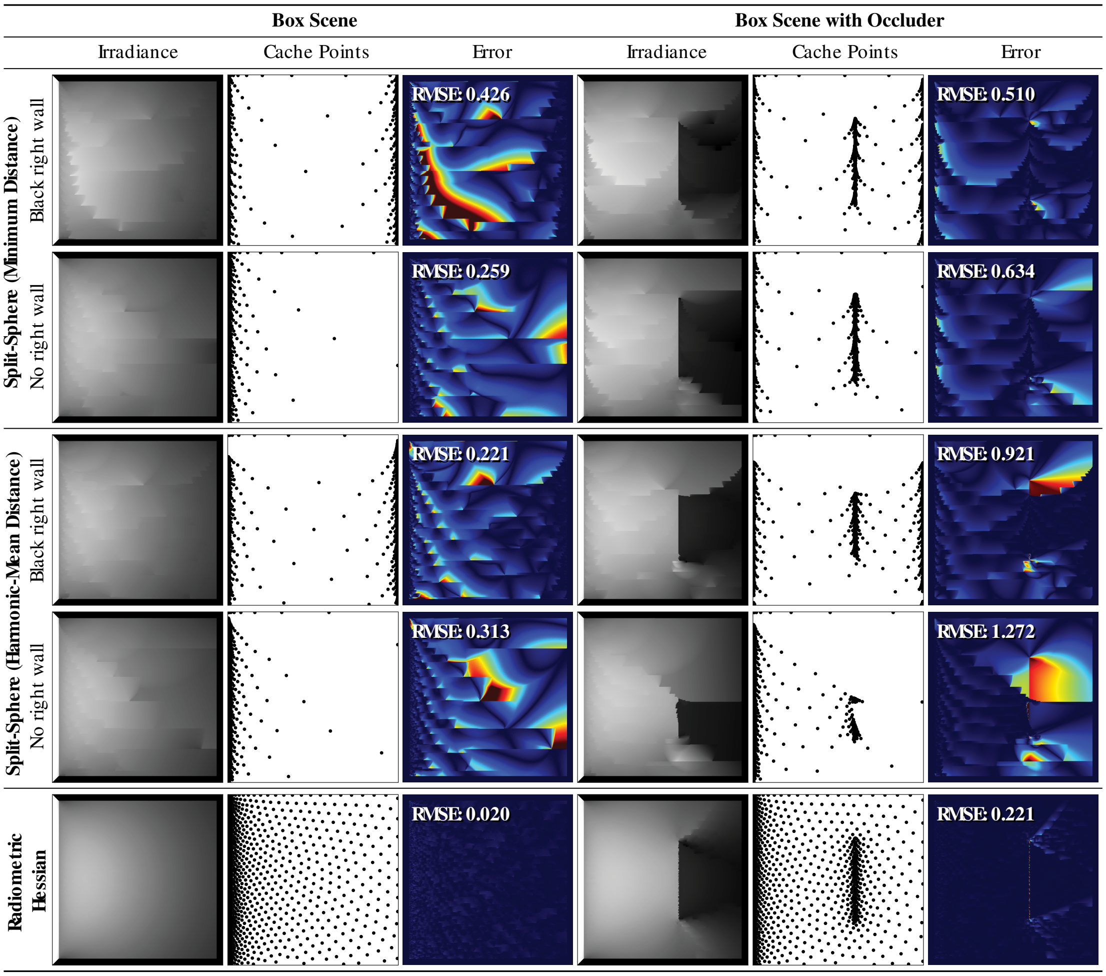


Figure 4.3: Comparison of the results produced by irradiance caching using both the split-sphere heuristic, as well as the radiometric hessian, as error-control methods. Figure from Jarosz et al. [JSKJ12].

Figure 4.3 (Figure 4.2 shows the scene configurations). Irradiance is interpolated in a similar way as introduced by Ward and Heckbert [WH92]:

$$E(\mathbf{x}) \approx \frac{\sum_{i \in S} w_i(\mathbf{x})(E(\mathbf{x}_i) + (\mathbf{x} - \mathbf{x}_i) \cdot \nabla_t E(\mathbf{x}_i))}{\sum_{i \in S} w_i(\mathbf{x})}.$$

We should note that Jarosz et al. did not indicate the weighting function they used for the cache records during interpolation ($w_i(\mathbf{x})$ in the Equation), and they did not consider the rotational component of the irradiance gradient at all. These two factors preclude the implementation of their approach, as-is, in a practical, general rendering algorithm. Since their analysis was performed on such simple scenes, the rotational component was not a requirement in order to produce their results. With respect to the weights, they used a modified version of the Tabellion and Lamorlette weighting function [1] (described in Section 3.3.1):

$$w_i(\mathbf{x}) = 1 - \frac{\|\mathbf{x}_i - \mathbf{x}\|}{R_i}, \tag{4.20}$$

where R_i is the cache record radius derived using the methods of the previous section.

The test cases illustrate several of the shortcomings detailed in Section 3.3.3 for the split-sphere heuristic. Cache records aggressively concentrate around edges and corners, which is wasteful and leads to high error in areas of the scene that remain under-sampled. It is very difficult to improve the record distribution produced by the split-sphere without relying on additional heuristics, such as minimum and maximum radius clamping, Krivanek et al.'s neighbor-clamping, or adaptive caching, which increase the number of parameters that need to be finely tuned and balanced. In contrast, the Hessian-based approach obtains a high-quality record distribution without relying on additional corrective measures.

Another fundamental deficiency of the split-sphere is visible in Figure 4.3: radiometrically equivalent scenes can produce vastly different record distributions and error profiles. The second row for both methods shows the result of removing the black right wall and revealing the environment. Even though this results in a radiometrically identical image, the record distribution for the split-sphere changes

[1] This was discussed with the authors of that paper.

dramatically; the Hessian-based approach, on the other hand, produces the same record distribution for both cases.

4.2.2 Anisotropic Cache Records

As previously mentioned, the Hessian in Equation 4.16 describes a quadric surface, giving us a natural way to obtain anisotropic cache records. Instead of bounding the Hessian using only its maximum eigenvalue λ_1, we can work in the coordinate system defined by *both* principal curvatures, an orthonormal basis for which are the two eigenvectors $\mathbf{v}_1$ and $\mathbf{v}_2$ with the corresponding curvatures defined by the eigenvalues, and express the integrated error in the support of the cache record as:

$$\epsilon^t \leq \frac{1}{2} \iint_A \left(|\lambda_1| x^2 + |\lambda_2| y^2 \right) dy dx. \tag{4.21}$$

The shape that minimizes the integrated error while maximizing the support area of the cache record is an ellipse with axes $\mathbf{v}_1$ and $\mathbf{v}_2$. We again invert this equation, providing us with the radii along these axes:

$$\left(R_i^{\lambda_1}, R_i^{\lambda_2} \right) = \sqrt[4]{\frac{4\epsilon^t}{\pi}} \left(\sqrt[4]{\frac{1}{\lambda_1}}, \sqrt[4]{\frac{1}{\lambda_2}} \right). \tag{4.22}$$

Using elliptical records further reduces the error in the image by allowing cache records to be used farther along the direction in which irradiance is varying slowly; this in turn allows samples to pack more densely across directions of higher irradiance variation, reducing the overall error in the irradiance reconstruction.

While the idea of using anisotropic cache records is not a new one, and was suggested by Herzog et al. [HMS09], their approach relied on simply reducing the cache record radius along the gradient direction to produce elliptical support areas. This unfortunately means that the eccentricity of the cache records must be set by the user, whereas using the Hessian deduces the eccentricity automatically in a principled fashion by minimizing error.

4.3 Limitations

The most severe limitation on the method described by Jarosz et al. is that it does not provide a complete solution to replace the split-sphere heuristic in an irradiance caching implementation, since it foregoes analysis of the rotational component.

Furthermore, their formula for computing the error in the irradiance approximation results in an absolute measure of error, which is impractical because it means that the resulting error is dependent upon the absolute values of the irradiance in the scene. In the context of rendering, this means that two scenes that vary only in the power of the light sources will produce different cache record distributions, even though renderings of both scenes would be equivalent to a scaling factor.

These and other practical issues of implementation will be discussed in Chapter 6.

4.3.1 The Geometric Hessian

One practical problem that any radiometrically-dependent method for error control must deal with, and that was specifically addressed by Jarosz et al., arises in situations where points in the scene receive extremely little or no irradiance. In this case, the irradiance Hessian is either ill- or un-defined, which can result in large errors in the rendered image, as shown in Figure 4.4.

In order to deal with this problem, we first note that the irradiance Hessian eigenvalues are directly proportional to the incoming radiance over the hemisphere. It is therefore possible to conservatively bound the error induced by the second-order Taylor approximation by replacing the incident radiance $L(\mathbf{x}\!\leftarrow\!\mathbf{y})$ in Equation 4.23 with the maximum radiance $L^{\max}$ of any surface in the scene:

$$\mathbf{H}_{\mathbf{x}}E^{\max} \approx \int_A L^{\max}\mathbf{H}_{\mathbf{x}}G(\mathbf{x}\!\leftrightarrow\!\mathbf{y})dA(\mathbf{y}). \tag{4.23}$$

The maximum Hessian of irradiance now reduces to the integral of the Hessian of the geometric coupling term: $\mathbf{H}_{\mathbf{x}}E^{\max} \propto \int \mathbf{H}_{\mathbf{x}}G(\mathbf{x} \leftrightarrow \mathbf{y})$, and $L^{\max}$

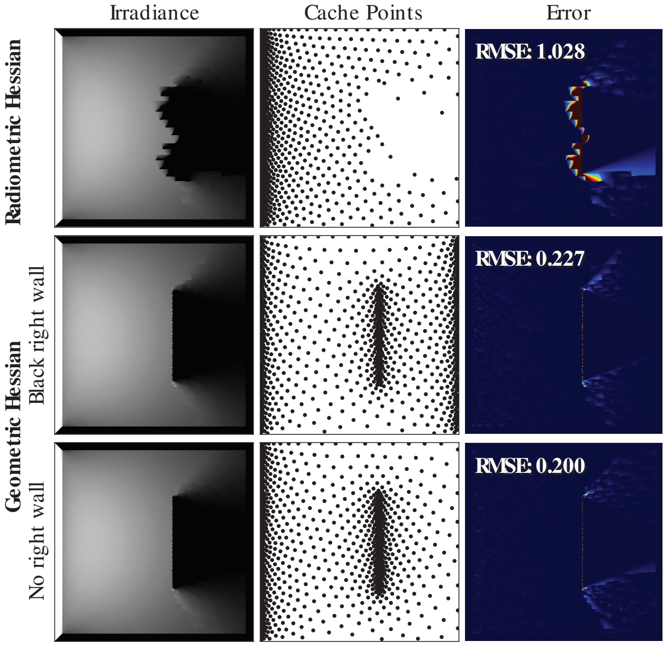


Figure 4.4: Rendering illustrating a failure case for the purely radiometric hessian error-control method, which fails when the occluder is nearly black. In this case, the geometric hessian is able to robustly resolve this issue. Figure from Jarosz et al. [JSKJ12].

simply acts as a constant scaling factor. This allows us to fold it into the user parameter ϵ^t and replace the actual maximum irradiance, which is difficult to estimate, with a canonical value $L^{\max} = 1$.

Inserting this new Hessian bound in place of $\mathbf{H}_\mathbf{x}E(\mathbf{x})$ in Equation 4.16 results in a conservative error bound, which can be solved in the same way as Equation 4.19. Jarosz et al. termed this the *Geometric Hessian*.

Figure 4.4 examines the performance of the geometric Hessian in comparison to the radiometric Hessian for the failure case, and shows that the new formulation robustly deals with the issue of extremely low or null irradiance.

One issue with the upper-bound on the Hessian is that it is a purely geometric quantity, which means that changes in the scene geometry will produce different cache point distributions even for radiometrically identical scenes. This is unfortunate, because it means that there are situations where a radiometric approach might produce more optimal cache record distributions. Even so, the geometric Hessian retains the error reduction properties of the radiometric Hessian while being more robust to arbitrary lighting configurations, making it a more practical candidate for replacing the venerable Split-Sphere heuristic.

Chapter 5

The Occlusion-Aware Irradiance Hessian

While the ideas presented by Jarosz et al. [JSKJ12] and discussed in the previous section produce compelling results when applied to irradiance caching, they still result in sub-optimal cache record distributions due to the use of an irradiance Hessian that does not take object inter-occlusions into account.

Jarosz et al. derived their 3D irradiance Hessian by first analyzing the behavior of irradiance in a 2D setting, leading to an occlusion-aware derivation of a 2D irradiance Hessian. In this 2D analysis, it was observed that including occlusion information in the Hessian leads to important accuracy improvements in the irradiance reconstruction when compared to an occlusion-agnostic Hessian.

These findings motivate the current chapter, where we present a full derivation of an occlusion-aware irradiance Hessian for polyhedral light sources. We begin by going over the work of Arvo [Arv94], who derived the first derivative of the *light field* due to diffuse polygonal emitters (the *irradiance Jacobian*), and then extend that derivation to the second derivative. This results in a rank-3 tensor, which we project to a 3x3 irradiance Hessian. We then discuss how we might apply this new Hessian derivation to irradiance caching.

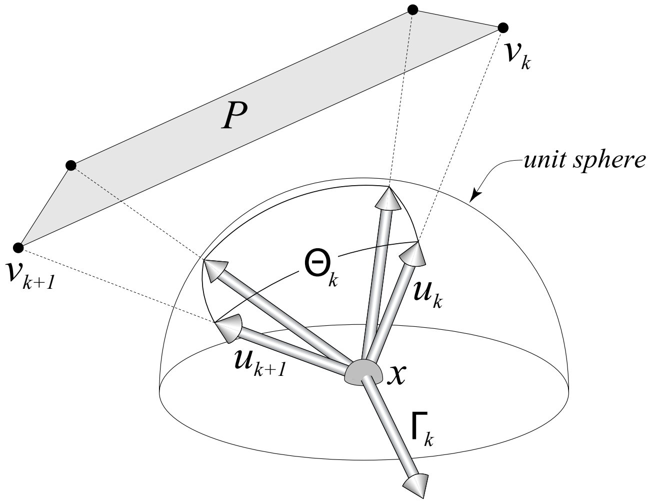


Figure 5.1: The notation and geometric relationships used. Figure from Arvo [Arv94].

5.1 The Irradiance Jacobian for Polyhedral Sources

Principal to this section is a vector field $\Phi : \mathbb{R}^3 \to \mathbb{R}^3$ known as the *light field*. In terms of radiance, the light field is given by

$$\Phi(\mathbf{x}) \equiv \int_{S^2} L(\mathbf{x} \to \vec{\omega}) \vec{\omega} d\vec{\omega}, \tag{5.1}$$

where S^2 is the unit sphere in $\mathbb{R}^3$. In simple terms, $\Phi(\mathbf{x})$ is the integral over all unit vectors through $\mathbf{x}$ weighted by the radiance in each direction. At all points in space it defines a vector quantity termed the *vector irradiance*. On surfaces, it is simple to relate the vector irradiance to irradiance if we recall Equation (2.5):

$$E(\mathbf{x}) = \int_{\Omega} L(\mathbf{x} \leftarrow \vec{\omega})(\mathbf{n} \cdot \vec{\omega}) d\vec{\omega}, \tag{5.2}$$

from which it follows that $E(\mathbf{x})$ and $\Phi(\mathbf{x})$ are related by

$$E(\mathbf{x}) = -\Phi(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}), \tag{5.3}$$

since on surfaces only the visible hemisphere carries radiance (such that radiance in any direction below the horizon is zero).

For a diffuse polyhedral source with constant emission M and vertices $v_1, v_2, \ldots, v_n$, the light field $\Phi(\mathbf{x})$ can be expressed analytically:

$$\Phi(\mathbf{x}) = \frac{M}{2\pi} \sum_{i=1}^{n} \Theta_i(\mathbf{x}) \Gamma_i(\mathbf{x}) \tag{5.4}$$

where $\Theta_1, \ldots, \Theta_n$ are the angles subtended by the edges of the occluder as seen from $\mathbf{x}$ and the vectors $\Gamma_1, \ldots, \Gamma_n$ are unit normals of the polygonal cone with cross section P and apex $\mathbf{x}$ (see Figure 5.1). For any $1 \leq k \leq n$ we have:

$$\Theta_k(\mathbf{x}) = \cos^{-1}\left(\frac{v_k - \mathbf{x}}{\|v_k - \mathbf{x}\|} \cdot \frac{v_{k+1} - \mathbf{x}}{\|v_{k+1} - \mathbf{x}\|} \right), \tag{5.5}$$

and

$$\Gamma_k(\mathbf{x}) = \frac{(v_k - \mathbf{x}) \times (v_{k+1} - \mathbf{x})}{\|(v_k - \mathbf{x}) \times (v_{k+1} - \mathbf{x})\|}, \tag{5.6}$$

where $v_{n+1} \equiv v_1$. When $M = 1$, the corresponding expression $-\Phi(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x})$ is the form factor between a differential patch at $\mathbf{x}$ and the polygonal patch P. Importantly, because the light field is a true vector field, the vector irradiance due to multiple sources may be obtained by summing the contributions from each source individually.

The derivative DF of a differentiable vector function $F : \mathbb{R}^3 \to \mathbb{R}^3$ is represented by a 3×3 Jacobian matrix. We shall denote the Jacobian matrix for the vector irradiance at $\mathbf{x}$ as $\mathbf{J}_{\mathbf{x}}(\Phi)$, that is,

$$\mathbf{J}_{\mathbf{x}}(\Phi) \equiv D\Phi(\mathbf{x}) = \left[\frac{\partial \Phi_i(\mathbf{x})}{\partial \mathbf{x}_j} \right]. \tag{5.7}$$

The obvious approach to obtaining this matrix is to differentiate Equation (5.4) with respect to $\mathbf{x}$. This result holds for unoccluded polygonal sources.

To derive an expression that applies for partially occluded sources, we express $\Phi(\mathbf{x})$ in terms of *vertex vectors*, which correspond to vertices of the spherical projection of the polygon, as depicted in Figure 5.1. These vectors point toward vertices of two distinct types, depending on how they arise. *Intrinsic* vertices exist either on the source or the blocker geometry explicitly, while *apparent* vertices arise when the edge of a blocker, as seen from $\mathbf{x}$, crosses the edge of the source or of another blocker (see Figure 5.2). We will express $\mathbf{J}_{\mathbf{x}}(\Phi)$ in terms of derivatives

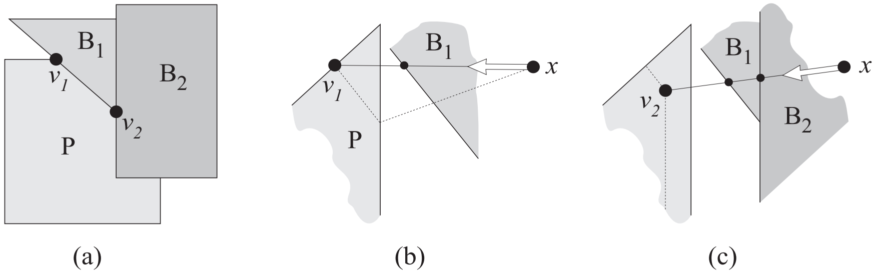


Figure 5.2: (a) The view from $\mathbf{x}$ of the two types of "apparent" vertices. (b) The vertex vector for v_1 resulting from a blocker edge and a source edge. (c) The vertex vector for v_2 resulting from two blocker edges. Figure from Arvo [Arv94].

of these vectors. The derivative of a vertex vector is a 3×3 matrix, which we term the *vertex Jacobian*.

We let $v'_1, v'_2, \ldots, v'_m$ be the vertices of P', the light source P after clipping away portions that are occluded with respect to point $\mathbf{x}$. We then define the vertex vectors $\mathbf{u}_1(\mathbf{x}), \mathbf{u}_2(\mathbf{x}), \ldots, \mathbf{u}_m(\mathbf{x})$ by

$$\mathbf{u}_k(\mathbf{x}) = \frac{v'_k - \mathbf{x}}{\|v'_k - \mathbf{x}\|}. \tag{5.8}$$

We define $\mathbf{w}_1(\mathbf{x}), \ldots, \mathbf{w}_m(\mathbf{x})$ to be the cross products

$$\mathbf{w}_k(\mathbf{x}) = \mathbf{u}_k(\mathbf{x}) \times \mathbf{u}_{k+1}(\mathbf{x}). \tag{5.9}$$

From this point on we assume that $\mathbf{u}_k$ and $\mathbf{w}_k$ are functions of position and thus omit the explicit dependence on $\mathbf{x}$. Expressing Θ_k and Γ_k in terms of $\mathbf{w}_k$ we have

$$\Theta_k = \sin^{-1}\|\mathbf{w}_k\| \tag{5.10}$$

and

$$\Gamma_k = \frac{w_k}{\|\mathbf{w}_k\|} \tag{5.11}$$

In order to compute $\mathbf{J}(\Phi)$ we first consider the k^{th} term of the summation. Differentiating, we have

$$\mathbf{J}(\Theta_k\Gamma_k) = \Gamma_k\nabla\Theta_k + \Theta_k\mathbf{J}(\Gamma_k), \tag{5.12}$$

where $\Gamma_k\nabla\Theta_k$ is the outer product of the vector Γ_k and the gradient $\nabla\Theta_k$. We now compute $\nabla\Theta_k$ and $\mathbf{J}(\Gamma_k)$. For brevity, we denote the vertex vectors $\mathbf{u}_k$ and $\mathbf{u}_{k+1}$

by $\mathbf{a}$ and $\mathbf{b}$ respectively, and the cross product $\mathbf{a} \times \mathbf{b}$ by $\mathbf{w}$. Then the gradient of Θ_k with respect to $\mathbf{x}$ is

$$\begin{aligned}\nabla\Theta_k &= \nabla \sin^{-1}\|\mathbf{w}\| \\ &= \frac{1}{\sqrt{1-\mathbf{w}^\mathsf{T}\mathbf{w}}}\left(\frac{\mathbf{w}^\mathsf{T}}{\|\mathbf{w}\|}\right)\mathbf{J}(\mathbf{w}) \\ &= \left(\frac{\mathbf{w}^\mathsf{T}}{\mathbf{a}^\mathsf{T}\mathbf{b}}\right)\frac{\mathbf{J}(\mathbf{w})}{\|\mathbf{w}\|} \end{aligned} \tag{5.13}$$

and in the case of Γ_k

$$\begin{aligned}\mathbf{J}(\Gamma_k) &= D\left(\frac{\mathbf{w}}{\|\mathbf{w}\|}\right) \\ &= \frac{\mathbf{J}(\mathbf{w})}{\|\mathbf{w}\|} - \frac{\mathbf{w}\mathbf{w}^\mathsf{T}}{\|\mathbf{w}\|^3}\mathbf{J}(\mathbf{w}) \\ &= \left(\mathbf{I} - \frac{\mathbf{w}\mathbf{w}^\mathsf{T}}{\mathbf{w}^\mathsf{T}\mathbf{w}}\right)\frac{\mathbf{J}(\mathbf{w})}{\|\mathbf{w}\|}. \end{aligned} \tag{5.14}$$

From the previous equations we obtain an expression for $\mathbf{J}(\Theta_k\Gamma_k)$ in terms of $\mathbf{J}(\mathbf{w})$ and the vertex vectors $\mathbf{a}$ and $\mathbf{b}$:

$$\mathbf{J}(\Theta_k\Gamma_k) = \left[\frac{\mathbf{w}}{\|\mathbf{w}\|}\left(\frac{\mathbf{w}^\mathsf{T}}{\mathbf{a}^\mathsf{T}\mathbf{b}}\right) + \sin^{-1}\|\mathbf{w}\|\left(\mathbf{I} - \frac{\mathbf{w}\mathbf{w}^\mathsf{T}}{\mathbf{w}^\mathsf{T}\mathbf{w}}\right)\right]\frac{\mathbf{J}(\mathbf{w})}{\|\mathbf{w}\|}$$

The above expression may be written compactly as

$$\mathbf{J}(\Theta_k\Gamma_k) = \mathbf{E}(\mathbf{a},\mathbf{b})\mathbf{J}(\mathbf{a}\times\mathbf{b}), \tag{5.15}$$

where the function $\mathbf{E}$ is the *edge matrix* defined by

$$\mathbf{E}(\mathbf{a},\mathbf{b}) \equiv \left(\frac{1}{\mathbf{a}^\mathsf{T}\mathbf{b}}\right)\frac{\mathbf{w}\mathbf{w}^\mathsf{T}}{\mathbf{w}^\mathsf{T}\mathbf{w}} + \frac{\cos^{-1}\mathbf{a}^\mathsf{T}\mathbf{b}}{\|\mathbf{w}\|}\left(\mathbf{I} - \frac{\mathbf{w}\mathbf{w}^\mathsf{T}}{\mathbf{w}^\mathsf{T}\mathbf{w}}\right). \tag{5.16}$$

To simplify the Jacobian of $\mathbf{a}\times\mathbf{b}$, we define another matrix-valued function $\mathbf{Q}$ by

$$\mathbf{Q}(\mathbf{p}) \equiv \begin{bmatrix} 0 & -p_z & p_y \\ p_z & 0 & -p_x \\ -p_y & p_x & 0 \end{bmatrix}. \tag{5.17}$$

Then for any pair of vectors $\mathbf{p}$ and $\mathbf{q}$, we have $\mathbf{p}\times\mathbf{q} = \mathbf{Q}(\mathbf{p})\mathbf{q}$. Writing the cross product as a matrix multiplication leads to a convenient expression for the Jacobian matrix of $F \times G$, where F and G are vector fields in $\mathbb{R}^3$. Thus,

$$\mathbf{J}(F\times G) = \mathbf{Q}(F)\mathbf{J}(G) - \mathbf{Q}(G)\mathbf{J}(F). \tag{5.18}$$

We thus arrive at an expression for the irradiance Jacobian due to the visible portion of polygonal source P:

$$\mathbf{J}(\Phi) = \frac{M}{2\pi} \sum_{i=1}^{m} \mathbf{E}(\mathbf{u}_i, \mathbf{u}_{i+1})[\mathbf{Q}(\mathbf{u}_i)\mathbf{J}(\mathbf{u}_{i+1}) - \mathbf{Q}(\mathbf{u}_{i+1})\mathbf{J}(\mathbf{u}_i)]. \tag{5.19}$$

This expression can be simplified further by collecting the factor of each $\mathbf{J}(\mathbf{u}_i)$ into a single matrix. We therefore define the *corner matrix* $\mathbf{C}$ to be the matrix-valued function

$$\mathbf{C}(\mathbf{a}, \mathbf{b}, \mathbf{c}) \equiv \mathbf{E}(\mathbf{a}, \mathbf{b})\mathbf{Q}(\mathbf{a}) - \mathbf{E}(\mathbf{b}, \mathbf{c})\mathbf{Q}(\mathbf{c}). \tag{5.20}$$

Then the final expression for the irradiance Jacobian can be written as the sum over all vertex Jacobians transformed by corner matrices:

$$\mathbf{J}(\Phi) = \frac{M}{2\pi} \sum_{i=1}^{m} \mathbf{C}(\mathbf{u}_{i-1}, \mathbf{u}_i, \mathbf{u}_{i+1})\mathbf{J}(\mathbf{u}_i), \tag{5.21}$$

where $\mathbf{u}_0 \equiv \mathbf{u}_m$ and $\mathbf{u}_{m+1} \equiv \mathbf{u}_1$.

In the case of the vertex Jacobians, we shall take a small detour from Arvo's derivation in order to produce an expression of the Jacobian that is both slightly more general as well as more amenable to extension to a second order. For this, we need to re-define the vertex vectors $\mathbf{u}$ so that they explicitly depend upon the polygonal edges that cause them to arise. We let $\mathbf{u}_i$ be the vertex vector associated with the i^{th} vertex on the emitter, whether it corresponds to the case of an apparent vertex or not. Let e_i and e_{i+1} be the polygon edges whose intersection, from the point of view of $\mathbf{x}$ and in clockwise order, defines the vertex. These edges may belong to the emitter (in the case of intrinsic vertices), to blockers, or to both (in the case of apparent vertices). Furthermore, let O_i and P_i be any two points on the i^{th} edge. We can now define the *edge vectors* $\mathbf{m}_i$ as:

$$\mathbf{m}_i = (O_i - \mathbf{x}) \times (P_i - O_i) \tag{5.22}$$

and we also define, for convenience, another vector as:

$$\mathbf{d}_i = \mathbf{m}_i \times \mathbf{m}_{i+i} \tag{5.23}$$

which, in turn, allow us to write a new expression for the vertex vectors:

$$\mathbf{u}_i = \frac{\mathbf{d}_i}{\|\mathbf{d}_i\|} \tag{5.24}$$

We are now ready to find the vertex Jacobian, which we do by simply differentiating the expression for the vertex vectors. We have:

$$\begin{aligned}\mathbf{J}(\mathbf{u}) &= \frac{\|\mathbf{d}\|\mathbf{J}(\mathbf{d}) - \mathbf{d}\nabla^{\intercal}(\|\mathbf{d}\|)}{\|\mathbf{d}\|^2} \\ &= \frac{1}{\|\mathbf{d}\|}\mathbf{J}(\mathbf{d}) - \frac{\mathbf{d}\mathbf{d}^{\intercal}}{\|\mathbf{d}\|}\mathbf{J}(\mathbf{d}) \\ &= \left(\mathbf{I}_3 - \mathbf{d}\mathbf{d}^{\intercal}\right)\frac{\mathbf{J}(\mathbf{d})}{\|\mathbf{d}\|}. \end{aligned} \tag{5.25}$$

In the case of $\mathbf{J}(\mathbf{d})$ we have the Jacobian of a cross product, so we make use of the $\mathbf{Q}$ function as before:

$$\mathbf{J}(\mathbf{d}) = \mathbf{Q}(\mathbf{m}_i)\mathbf{J}(\mathbf{m}_{i+1}) - \mathbf{Q}(\mathbf{m}_{i+1})\mathbf{J}(\mathbf{m}_i) \tag{5.26}$$

and the same happens with $\mathbf{J}(\mathbf{m}_i)$:

$$\begin{aligned}\mathbf{J}(\mathbf{m}_i) &= \mathbf{Q}(O_i - \mathbf{x})\mathbf{J}(P_i - O_i) - \mathbf{Q}(P_i - O_i)\mathbf{J}(O_i - \mathbf{x}) \\ &= \mathbf{Q}(P_i - O_i). \end{aligned} \tag{5.27}$$

5.2 Derivation of the Irradiance Hessian

We now move on to the vector irradiance Hessian. We repeat the expression for the Jacobian:

$$\mathbf{J}(\Phi) = \frac{M}{2\pi}\sum_{i=1}^{m}\mathbf{C}(\mathbf{u}_{i-1}, \mathbf{u}_i, \mathbf{u}_{i+1})\mathbf{J}(\mathbf{u}_i).$$

We only need to derive the expression for the derivative of the k^{th} element in the summation, given that differentiation is a linear operation. We see that this is

$$\begin{aligned}\mathbf{H}(\Theta_k\Gamma_k) &= D\left[C(\mathbf{u}_{k-1}, \mathbf{u}_k, \mathbf{u}_{k+1})\mathbf{J}(\mathbf{u}_k)\right] \\ &= C'(\mathbf{u}_{k-1}, \mathbf{u}_k, \mathbf{u}_{k+1})\mathbf{J}(\mathbf{u}_k) + C(\mathbf{u}_{k-1}, \mathbf{u}_k, \mathbf{u}_{k+1})\mathbf{H}(\mathbf{u}_k) \end{aligned} \tag{5.28}$$

We first dive into the derivative of $C(\mathbf{u}_{k-1}, \mathbf{u}_k, \mathbf{u}_{k+1})$. We remember that

$$\mathbf{C}(\mathbf{a}, \mathbf{b}, \mathbf{c}) \equiv \mathbf{E}(\mathbf{a}, \mathbf{b})\mathbf{Q}(\mathbf{a}) - \mathbf{E}(\mathbf{b}, \mathbf{c})\mathbf{Q}(\mathbf{c}).$$

and switch from the explicit subscripts on $\mathbf{u}$ to $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$ for compaction, so

$$D\left[C(\mathbf{a},\mathbf{b},\mathbf{c})\right] = D\left[\mathbf{E}(\mathbf{a},\mathbf{b})\mathbf{Q}(\mathbf{a}) - \mathbf{E}(\mathbf{b},\mathbf{c})\mathbf{Q}(\mathbf{c})\right] \tag{5.29}$$

Again due to linearity of the differentiation operator, we need only focus on one of the products, where we have

$$D\left[\mathbf{E}(\mathbf{a},\mathbf{b})\mathbf{Q}(\mathbf{a})\right] = D\left[\mathbf{E}(\mathbf{a},\mathbf{b})\right]\mathbf{Q}(\mathbf{a}) + \mathbf{E}(\mathbf{a},\mathbf{b})D\left[\mathbf{Q}(\mathbf{a})\right]. \tag{5.30}$$

We now begin by differentiating $\mathbf{Q}(\mathbf{a})$. Remembering that $\mathbf{a}$ is a function of $\mathbf{x}$, and that we are differentiating with respect to $\mathbf{x}$, we have that

$$\begin{aligned}\frac{d}{d\mathbf{x}}\mathbf{Q}(\mathbf{a}) &= \frac{d\mathbf{Q}(\mathbf{a})}{d\mathbf{a}}\frac{d\mathbf{a}}{d\mathbf{x}} \\ &= \frac{d\mathbf{Q}(\mathbf{a})}{d\mathbf{a}} \cdot \mathbf{J}(\mathbf{a}) \end{aligned}\tag{5.31}$$

where $\cdot$ represents the tensor inner product, given that $\frac{d\mathbf{Q}(\mathbf{a})}{d\mathbf{a}}$ is a rank-3 tensor, and $\mathbf{J}(\mathbf{a})$ was given in Equation 5.25. Given the form of $\mathbf{Q}(\mathbf{a})$ (see Equation 5.17) differentiation is fairly simple. In order to avoid the space usage of displaying rank-3 tensors, we switch to index notation here and where applicable in the future:

$$\frac{\partial\mathbf{Q}(\mathbf{a})}{\partial\mathbf{a}}_{ijk} = \frac{\partial\mathbf{Q}(\mathbf{a})_{ij}}{\partial\mathbf{a}_k}. \tag{5.32}$$

We now turn our attention to $d\left[\mathbf{E}(\mathbf{a},\mathbf{b})\right]$. Returning to the definition of $\mathbf{w} \equiv \mathbf{a} \times \mathbf{b}$, we have

$$D\left[\mathbf{E}(\mathbf{a},\mathbf{b})\right] = D\left[\frac{\mathbf{w}\mathbf{w}^\intercal}{\mathbf{a}^\intercal\mathbf{b}\mathbf{w}^\intercal\mathbf{w}} + \frac{\cos^{-1}\mathbf{a}^\intercal\mathbf{b}}{\|\mathbf{w}\|}\left(\mathbf{I} - \frac{\mathbf{w}\mathbf{w}^\intercal}{\mathbf{w}^\intercal\mathbf{w}}\right)\right] \tag{5.33}$$

and again we'll look at the derivatives of each element separately. First,

$$\begin{aligned} D\left[\frac{\mathbf{w}\mathbf{w}^\intercal}{\mathbf{a}^\intercal\mathbf{b}\mathbf{w}^\intercal\mathbf{w}}\right] &= D\left[\frac{\mathbf{w}\mathbf{w}^\intercal}{\|\mathbf{w}\|^2\mathbf{a}^\intercal\mathbf{b}}\right] \\ &= \frac{(\|\mathbf{w}\|^2\mathbf{a}^\intercal\mathbf{b})d(\mathbf{w}\mathbf{w}^\intercal) - \nabla(\|\mathbf{w}\|^2\mathbf{a}^\intercal\mathbf{b}) \otimes \mathbf{w}\mathbf{w}^\intercal}{\|\mathbf{w}\|^4(\mathbf{a}^\intercal\mathbf{b})^2} \end{aligned}\tag{5.34}$$

where $\nabla(\|\mathbf{w}\|^2\mathbf{a}^\intercal\mathbf{b}) \otimes \mathbf{w}\mathbf{w}^\intercal$ is an outer product. We have

$$\nabla(\|\mathbf{w}\|^2\mathbf{a}^\intercal\mathbf{b}) = \nabla(\|\mathbf{w}\|^2)\mathbf{a}^\intercal\mathbf{b} + \|\mathbf{w}\|^2\nabla(\mathbf{a}^\intercal\mathbf{b}) \tag{5.35}$$

$$\nabla(\|\mathbf{w}\|^2) = 2\mathbf{w}^\intercal\mathbf{J}(\mathbf{w}) \tag{5.36}$$

$$\begin{aligned}\nabla(\mathbf{a}^\intercal\mathbf{b}) &= \nabla(\sqrt{1 - \|\mathbf{w}\|^2}) \\ &= \frac{-\|\mathbf{w}\|^2}{\sqrt{1 - \|\mathbf{w}\|^2}}\mathbf{w}^\intercal\mathbf{J}(\mathbf{w}) \end{aligned}\tag{5.37}$$

and $\mathbf{J}(\mathbf{w}) = \mathbf{Q}(\mathbf{a})\mathbf{J}(\mathbf{b}) - \mathbf{Q}(\mathbf{b})\mathbf{J}(\mathbf{a})$, as we have seen before.

In the case of $d(\mathbf{w}\mathbf{w}^\intercal)$, we notice that this corresponds to a rank-1 tensor outer product, and differentiate accordingly, resulting in the following expression:

$$\begin{aligned} d(\mathbf{w}\mathbf{w}^\intercal) &= d(\mathbf{w} \otimes \mathbf{w}) \\ &= (\mathbf{J}(\mathbf{w}) \otimes \mathbf{w})^{[i,k,j]} + \mathbf{w} \otimes \mathbf{J}(\mathbf{w}) \end{aligned} \tag{5.38}$$

where $(\mathbf{J}(\mathbf{w}) \otimes \mathbf{w})^{[i,k,j]}$ indicates a generalized transposition in which elements of the tensor with index $[i, j, k]$ become the elements with index $[i, k, j]$. The motivation for this transposition is straightforward if we look at the expressions in index notation. We have that:

$$\begin{aligned} (\mathbf{w}\mathbf{w}^\intercal)_{i,j} &= (\mathbf{w} \otimes \mathbf{w})_{i,j} \\ &= \mathbf{w}_i \mathbf{w}_j \end{aligned} \tag{5.39}$$

and so

$$\frac{\partial (\mathbf{w} \otimes \mathbf{w})_{i,j}}{\partial \mathbf{x}_k} = \frac{\partial \mathbf{w}_i}{\partial \mathbf{x}_k} \mathbf{w}_j + \mathbf{w}_i \frac{\partial \mathbf{w}_j}{\partial \mathbf{x}_k} \tag{5.40}$$

which corresponds to the previous expression (Equation 5.38).

For the second element of $D\left(\mathbf{E}(\mathbf{a}, \mathbf{b})\right)$ we have

$$\nabla \left[\frac{\cos^{-1} \mathbf{a}^\intercal \mathbf{b}}{\|\mathbf{w}\|} \right] \otimes \left(\mathbf{I} - \frac{\mathbf{w}\mathbf{w}^\intercal}{\mathbf{w}^\intercal \mathbf{w}} \right) + \frac{\cos^{-1} \mathbf{a}^\intercal \mathbf{b}}{\|\mathbf{w}\|} d \left(\mathbf{I} - \frac{\mathbf{w}\mathbf{w}^\intercal}{\mathbf{w}^\intercal \mathbf{w}} \right) \tag{5.41}$$

$$\begin{aligned} \nabla \left[\frac{\cos^{-1} \mathbf{a}^\intercal \mathbf{b}}{\|\mathbf{w}\|} \right] &= \nabla \left(\frac{\sin^{-1} \|\mathbf{w}\|}{\|\mathbf{w}\|} \right) \\ &= \frac{\|\mathbf{w}\| \nabla \sin^{-1} \|\mathbf{w}\| - \nabla \|\mathbf{w}\| \sin^{-1} \|\mathbf{w}\|}{\|\mathbf{w}\|^2} \\ &= \frac{\|\mathbf{w}\| \left(\frac{\mathbf{w}^\intercal}{\mathbf{a}^\intercal \mathbf{b}} \right) \frac{\mathbf{J}(\mathbf{w})}{\|\mathbf{w}\|} - \frac{\mathbf{w}^\intercal}{\|\mathbf{w}\|} \mathbf{J}(\mathbf{w}) \sin^{-1} \|\mathbf{w}\|}{\|\mathbf{w}\|^2} \\ &= \left(\frac{1}{\mathbf{a}^\intercal \mathbf{b} \mathbf{w}^\intercal \mathbf{w}} - \frac{\sin^{-1} \|\mathbf{w}\|}{\|\mathbf{w}\|^3} \right) \mathbf{w}^\intercal \mathbf{J}(\mathbf{w}) \\ &= \left(\frac{1}{\mathbf{a}^\intercal \mathbf{b} \mathbf{w}^\intercal \mathbf{w}} - \frac{\cos^{-1} \mathbf{a}^\intercal \mathbf{b}}{\|\mathbf{w}\|^3} \right) \mathbf{w}^\intercal \mathbf{J}(\mathbf{w}) \end{aligned} \tag{5.42}$$

and finally,

$$\begin{aligned} d\left(\mathbf{I} - \frac{\mathbf{w}\mathbf{w}^\intercal}{\mathbf{w}^\intercal\mathbf{w}}\right) &= -d\left(\frac{\mathbf{w}\mathbf{w}^\intercal}{\mathbf{w}^\intercal\mathbf{w}}\right) \\ &= \frac{(\nabla\|\mathbf{w}\|^2) \otimes (\mathbf{w}\mathbf{w}^\intercal) - \|\mathbf{w}\|^2 d(\mathbf{w}\mathbf{w}^\intercal)}{\|\mathbf{w}\|^4} \\ &= \frac{(2\mathbf{w}^\intercal\mathbf{J}(\mathbf{w})) \otimes (\mathbf{w}\mathbf{w}^\intercal) - (\mathbf{w}^\intercal\mathbf{w})d(\mathbf{w}\mathbf{w}^\intercal)}{(\mathbf{w}^\intercal\mathbf{w})^2} \end{aligned} \tag{5.43}$$

We now only need to derive the vertex Hessian in order to complete our derivation of the vector irradiance Hessian. This simply means differentiating the expression for the vertex Jacobian once more. We remember that:

$$\mathbf{J}(\mathbf{u}) = (\mathbf{I}_3 - \mathbf{d}\mathbf{d}^\intercal)\frac{\mathbf{J}(\mathbf{d})}{\|\mathbf{d}\|}$$

(see Equation 5.25) and have that

$$\mathbf{H}(\mathbf{u}) = \frac{\|\mathbf{d}\| d\left[(\mathbf{I}_3 - \mathbf{u}\mathbf{u}^\intercal)\mathbf{J}(\mathbf{d})\right] - \nabla(\|\mathbf{d}\|) \otimes \left[(\mathbf{I}_3 - \mathbf{u}\mathbf{u}^\intercal)\mathbf{J}(\mathbf{d})\right]}{\|\mathbf{d}\|^2} \tag{5.44}$$

and we tackle each component of this expression separately to aid comprehension:

$$\begin{aligned} d\left[(\mathbf{I}_3 - \mathbf{u}\mathbf{u}^\intercal)\mathbf{J}(\mathbf{d})\right] &= d(\mathbf{I}_3 - \mathbf{u}\mathbf{u}^\intercal) \cdot \mathbf{J}(\mathbf{d}) + (\mathbf{I}_3 - \mathbf{u}\mathbf{u}^\intercal) \cdot \mathbf{H}(\mathbf{d}) && (5.45) \\ d(\mathbf{I}_3 - \mathbf{u}\mathbf{u}^\intercal) &= -d(\mathbf{u}\mathbf{u}^\intercal) \\ &= -(\mathbf{J}(\mathbf{u}) \otimes \mathbf{u})^{[i,k,j]} - \mathbf{u} \otimes \mathbf{J}(\mathbf{u})(\text{See Equation 5.38}) && (5.46) \\ \mathbf{H}(\mathbf{d}) &= d\left[\mathbf{Q}(\mathbf{m}_i)\mathbf{J}(\mathbf{m}_{i+1}) - \mathbf{Q}(\mathbf{m}_{i+1})\mathbf{J}(\mathbf{m}_i)\right] && (5.47) \\ d\left[\mathbf{Q}(\mathbf{m}_i)\mathbf{J}(\mathbf{m}_{i+1})\right] &= d\mathbf{Q}(\mathbf{m}_i) \cdot \mathbf{J}(\mathbf{m}_{i+1}) + \mathbf{Q}(\mathbf{m}_i)\mathbf{H}(\mathbf{m}_{i+1}) && (5.48) \\ \mathbf{H}(\mathbf{m}_i) &= d\left[\mathbf{Q}(P_i - O_i)\right] = 0 && (5.49) \\ d(\|\mathbf{d}\|) &= \frac{\mathbf{d}}{\|\mathbf{d}\|}\mathbf{J}(\mathbf{d}) = \mathbf{u}^\intercal\mathbf{J}(\mathbf{d}). && (5.50) \end{aligned}$$

5.3 Evaluation

In order to evaluate our new analytic formulation, we implemented it in a small C++ path-tracer and used it to evaluate the irradiance Hessian within a simple scene, which features a lambertian ground plane, a diffuse planar light source directly over it, and two planar occluders at varying distances between the

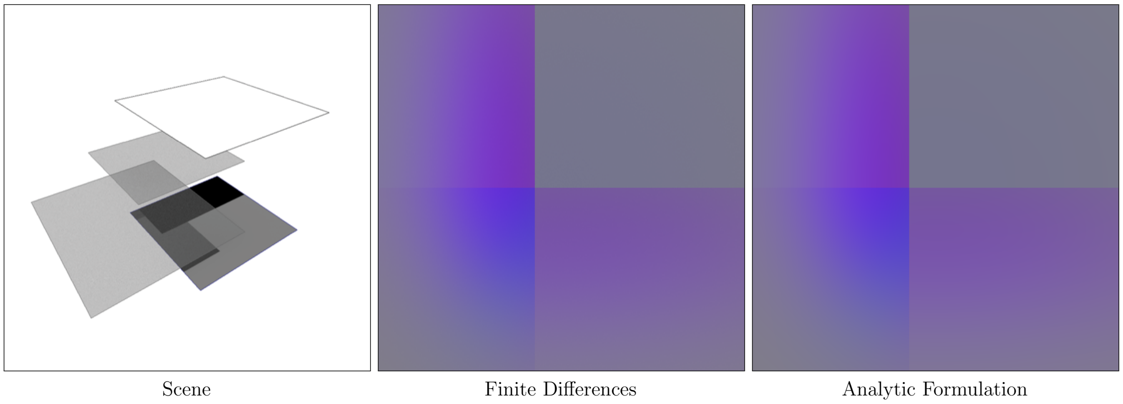


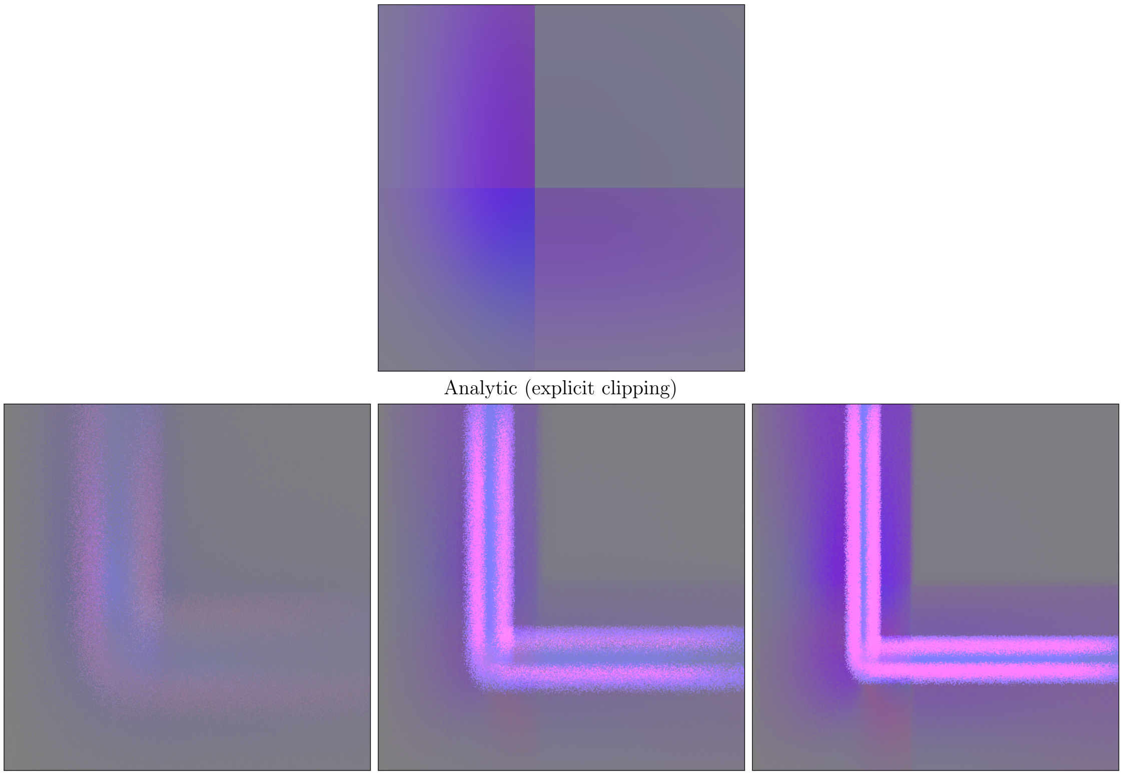
Figure 5.3: Visualization of the irradiance Hessian eigenvalues, after projection onto the surface. The scene is shown on the left: a diffuse plane is illuminated by a large diffuse source, which is occluded by two rectangular planes at varying distances. The result of our new analytic formula closely matches that of a finite-differences computation.

ground plane and light source (see Figure 5.3, left). The simple nature of the scene also allows us to compute the irradiance analytically at any point on the ground plane, which enables us to efficiently and accurately compute an accurate irradiance Hessian using finite differences.

Figure 5.3 compares the results of our new analytic formulation to those obtained using a finite-differences method. Since the full irradiance Hessian is a rank-3 tensor, we ease visualization by performing two transformations: first, the full Hessian is projected onto the ground plane, giving us a 3x3 Hessian matrix. Secondly, even though the ground plane exists within a 3D world, it is only a 2D surface, and we can further project the Hessian onto it, resulting in a 2x2 Hessian matrix. We then visualize the two eigenvalues for this 2x2 matrix. As can be seen in the Figure, the results obtained using our analytic formulation perfectly match the reference.

5.4 Limitations

We can now put together a complete expression for the occlusion-aware vector irradiance Hessian, using the expressions derived in the previous section.



Analytic (explicit clipping)

Stratified - 100 samples

Stratified - 1156 samples

Stratified - 4356 samples

Figure 5.4: Visualization of the irradiance Hessian eigenvalues, after projection onto the surface. Even for this simple scene, our proposed geometric reconstruction from the hemispherical samples results in an unusable Hessian estimate.

Unfortunately, there are several practical limitations that prevent doing so within the context of the irradiance caching algorithm.

First and foremost, the computation required is quite intense - we need to solve for the full rank-3 tensor Hessian, even though we are only interested in the Hessian matrix describing the second derivative of irradiance on a surface.

Secondly, while the Hessian works well in a general case, it is not easy to apply it within the context of irradiance caching. Our idea in deriving the Hessian was to apply it to the polygons implicit in a stratified hemisphere sampling pattern, but this does not work well in practice (see Figure 5.4). This corresponds to a pathological case where the irradiance derivatives are always undefined (the derivatives are undefined whenever three or more polygon edges coincide at a single "apparent vertex" from the perspective of the shade point [Arv94].)

Therefore, though deriving the vector irradiance Hessian proved an interesting and enlightening exercise, we decided not to use it in our work. We present it here as it might still prove useful in other contexts.

Chapter 6

A New Hessian-Based Error Control Method

In this chapter we develop a novel, practical error heuristic for irradiance caching based on irradiance Hessians. Inspired by the preliminary results shown by Jarosz et al. and discussed in Chapter 4, we extend and improve the approach and perform the necessary evaluation to turn this idea into a practical error heuristic for general, complex scenes. The core of our error heuristic is a new formulation of irradiance Hessians which accounts for occlusion changes. This leads to higher-quality images compared to previously proposed techniques (see Figure 6.1), especially in scenes with complex indirect illumination containing occlussion and indirect penumbra as we will show in our results. Furthermore, our approach naturally supports elliptical cache records by exploiting the anisotropic error information contained within the Hessian. We also show that, compared to modifications of the Split-Sphere heuristic, our approach produces substantially better cache point distributions without relying on radius clamping and excessive parameter tuning. Finally, our formulation is not bound to any particular stratification of the hemisphere, which we show by replacing the traditional polar coordinates stratification for a concentric-mapped approach [SC97].

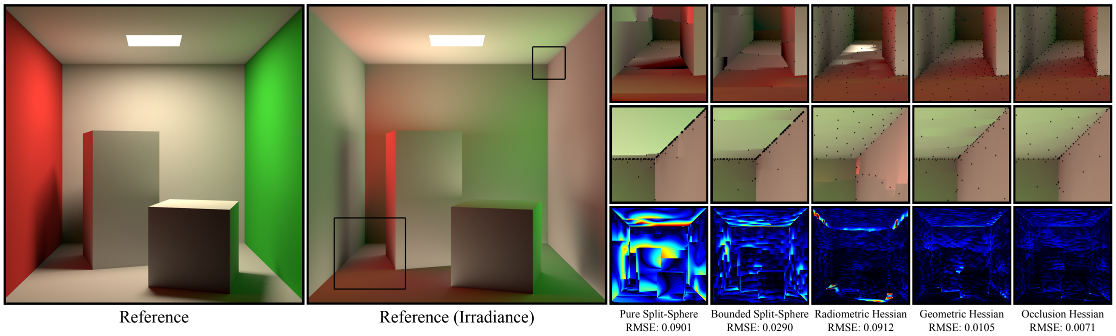


Figure 6.1: The Cornell box scene demonstrates how the Occlusion Hessian significantly outperforms both the Pure and the Bounded Split-Sphere (clamped to the gradient and 150px max spacing) for irradiance caching. It also performs significantly better than the occlusion-unaware Hessian error metrics by Jarosz et al. [JSKJ12].

Approach & Contributions.

In this chapter, we follow the same basic recipe as suggested by Jarosz et al. [JSKJ12] and outlined in Chapter 4, but propose a number of fundamental improvements that turn it into a practical, well-tested error control for irradiance caching in complex scenes:

- We derive an occlusion-aware Hessian for the error term, accounting for irradiance changes in scenes with occlusions.
- We also automatically obtain an occlusion-aware gradient which (unlike previous formulations) can easily be applied to general hemispherical sample distributions.
- We derive a more perceptually-motivated error criterion based on relative instead of absolute error, further reducing artifacts.
- We show how to make a radiometric (and not geometric) error metric robust, resulting in the first practical error metric for irradiance caching that is radiometrically meaningful.
- In contrast to Jarosz et al. [JSKJ12], we address how to take rotations of the surface normals into account, resulting in a complete and practical error-control formulation.
- We validate our proposed metric in complex scenes and compare to previous approaches, including the Split-Sphere with radius and gradient clamping, pro-

gressive refinement, and neighbor clamping [KBPZ06].

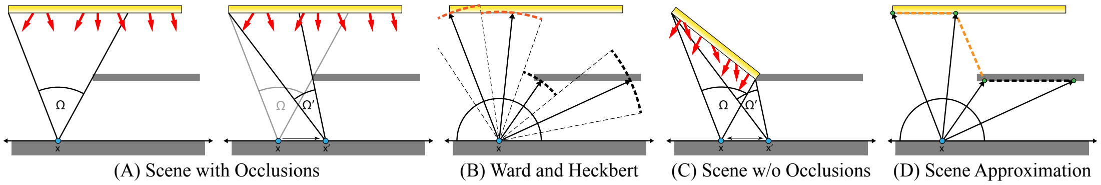
6.1 Overview

Jarosz et al. [JSKJ12] derived irradiance caching in a 2D setting and proposed the Hessian-based strategy for controlling error in irradiance caching. They derived a Hessian for irradiance assuming no occlusions in the scene. The Hessian based error metric is promising, but it becomes inaccurate in the presence of occlusions and consequently Jarosz et al. presented a Geometric Hessian that ignored the actual radiance values and instead provided a smooth cache distribution throughout the scene. The Geometric Hessian performs well in most scenes, but like the Split-Sphere heuristic it does not adapt in scenes where indirect illumination varies significantly. Consider a floor next to a wall that is either dark or bright. In the case of the dark wall, the sampling density on the floor can be quite low. In the case of the bright wall the floor needs more samples to account for the rapid local changes in the irradiance. The Geometric Hessian (as well as the Split-Sphere heuristic) will use the exact same sampling density in both cases. Having a radiometric error metric would enable the necessary variation in the sampling, and it would also enable a more precise control of the actual error in the extrapolated irradiance, which would lead to a higher quality irradiance estimation throughout the scene.

6.2 A Practical Occlusion-Aware Hessian

The core to our approach is an irradiance gradient and Hessian that accurately considers scenes with occlusions, like the one shown in Figure 6.2 (A). Ward et al. [WH92] previously derived an accurate occlusion-aware irradiance gradient by interpreting the hemispherical samples as a coarse sampling of the surrounding geometry (Figure 6.2 (B)) and considering the change of strata areas due to occlusions as $\mathbf{x}$ is translated.

A natural next step would be to consider this geometric approximation



(A) Scene with Occlusions (B) Ward and Heckbert (C) Scene w/o Occlusions (D) Scene Approximation

Figure 6.2: The scene depicted in (C) has the same irradiance first and second derivative as the one in (A), but there are no occlusions. (B) shows Ward and Heckbert's [WH92] interpretation of the stratified sample information used to derive the irradiance gradient, while (D) shows the interpretation we perform in order to approximate (C).

of the environment, and compute the Hessian of the contribution of each strata. Jarosz et al. [JSKJ12] considered this while ignoring occlusions between strata. This makes the computation far simpler, but, unfortunately, ignoring occlusion derivatives is known to produce suboptimal results in irradiance caching [KGBP05].

This problem is in fact the same as computing the occlusion-aware form-factor Hessian of each polygonal stratum, which has been derived previous in the context of radiosity [Hol96, HS98]. Unfortunately, the discontinuous geometric approximation in Figure 6.2 (B) corresponds to a pathological case for these methods where the irradiance derivatives are always undefined[1]. Hence, these previous approaches do not produce usable results when applied to Ward et al.'s stratified environment.

Geometric Interpretation of Hemispherical Samples.

To address this problem, we take a different approach. Our key insight is that we can convert a scene with occlusions (Figure 6.2 (A)) into a scene which is radiometrically equivalent at the shade point **x**, but which contains no occlusions (Figure 6.2 (C)). Remarkably, this modified scene has the same irradiance, irradiance gradient, and irradiance Hessian at **x** as the original scene.

Equivalence arises because irradiance at point **x** is directly proportional to the solid angle subtended by the uniform diffuse emitter, as seen from **x**. The gradient and Hessian are, then, proportional to the change in solid angle as **x** is

[1] The derivatives are undefined whenever three or more polygon edges coincide at a single "apparent vertex" from the perspective of the shade point [Arv94].

translated. In the case of Figure 6.2 (C), the emitter subtends the same solid angles Ω and Ω', as it does in Figure 6.2 (A); hence, the irradiance, gradient and Hessian must also be the same. Of course for very large displacements this is not necessarily true, but we are only concerned with this equivalence at a differential scale.

Given this insight about equivalence, our procedure becomes clear. We interpret the hemispherical samples and construct a continuous, piecewise-linear approximation (Figure 6.2 (D)) of the surrounding scene geometry, which contains no occlusions from the point of view of $\mathbf{x}$. We can then compute an irradiance gradient and Hessian of this triangulated environment which accounts for occlusions occurring in the physical scene, while not explicitly considering occlusions in our calculations.

Assuming we have a triangulated representation of the hemispherical environment, the irradiance at $\mathbf{x}$ defined in Equation (3.3) can be re-expressed in terms of this approximate geometry as:

$$E(\mathbf{x}, \mathbf{n}) \approx \sum_{j=1}^{M} L_{\triangle_j} \, F_{\triangle_j}(\mathbf{x}), \tag{6.1}$$

where M is the number of triangles in the tessellated hemisphere, and $L_{\triangle}$ and $F_{\triangle}(\mathbf{x})$ are the observed radiance and form-factor of triangle $\triangle$ at $\mathbf{x}$.

The gradient and Hessian of Equation (6.1) can be readily computed by summing the gradients and Hessians of the triangle form-factors:

$$\nabla_{\mathbf{x}} E \approx \nabla_{\mathbf{x}} \left(\sum_{j=1}^{M} L_{\triangle_j} F_{\triangle_j}(\mathbf{x}) \right) = \sum_{j=1}^{M} L_{\triangle_j} \nabla_{\mathbf{x}} F_{\triangle_j}(\mathbf{x}), \tag{6.2}$$

and

$$\mathbf{H}_{\mathbf{x}} E \approx \mathbf{H}_{\mathbf{x}} \left(\sum_{j=1}^{M} L_{\triangle_j} F_{\triangle_j}(\mathbf{x}) \right) = \sum_{j=1}^{M} L_{\triangle_j} \mathbf{H}_{\mathbf{x}} F_{\triangle_j}(\mathbf{x}). \tag{6.3}$$

Luckily, since our interpretation of the stratified samples induces no occlusion changes with translation, we can compute these form-factor derivatives independently for each triangle while ignoring occlusions.

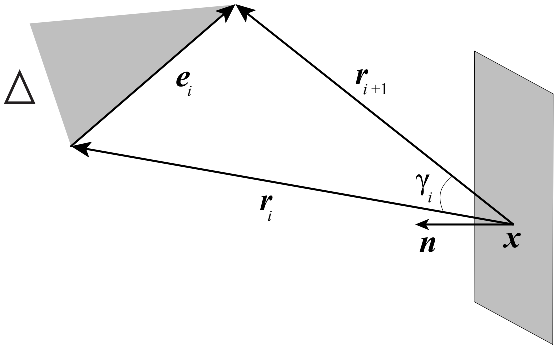


Figure 6.3: Illustration of the notation used in the form-factor, its gradient and Hessian.

Form Factor, Gradient, and Hessian

The formula for the point-to-triangle form-factor $F_\triangle(\mathbf{x})$ is given by [HS98]:

$$F_\triangle(\mathbf{x}) = \frac{1}{2\pi}\mathbf{n} \cdot \sum_i^3 \Gamma_i \tag{6.4}$$

where i indexes the three vertices, $\mathbf{r}_i$, expressed as vectors from $\mathbf{x}$, and $\Gamma_i = \mathbf{r}_i \times \mathbf{r}_{i+1}$ and has norm γ_i (see Figure 6.3).

Holzschuch and Sillion derived the following formulas for the gradient and Hessian of the form-factor:

$$\begin{aligned} \nabla_{\mathbf{x}} F_\triangle(\mathbf{x}) = &- \frac{-1}{2\pi} \sum_i \mathbf{n} \times \mathbf{e}_i I_1 \\ &+ 2\mathbf{n} \cdot (\mathbf{r}_i \times \mathbf{r}_{i+1})(\mathbf{r}_i I_2 + \mathbf{e}_i J_2) \end{aligned} \tag{6.5}$$

and

$$\begin{aligned} \mathbf{H}_{\mathbf{x}} F_\triangle(\mathbf{x}) = &- \frac{1}{\pi} \sum_i Q(\mathbf{n} \times \mathbf{e}_i, \mathbf{r}_i I_2 + \mathbf{e}_i J_2) \\ &- \mathbf{n} \cdot (\mathbf{r}_i \times \mathbf{e}_i) I_2 \mathbf{I}_{3\times3} \\ &+ 2\mathbf{n} \cdot (\mathbf{r}_i \times \mathbf{e}_i)(Q(\mathbf{r}_i, \mathbf{r}_i) I_3 \\ &+ Q(\mathbf{e}_i, \mathbf{e}_i) K_3 + 2 J_3 Q(\mathbf{r}_i, \mathbf{e}_i)), \end{aligned} \tag{6.6}$$

where:

$$\begin{aligned}
I_1 &= \frac{\gamma_i}{\|\mathbf{e}_i \times \mathbf{r}_i\|}, \\
I_2 &= \frac{1}{2\|\mathbf{e}_i \times \mathbf{r}_i\|^2} \left(\frac{\mathbf{e}_i \cdot \mathbf{r}_{i+1}}{\|\mathbf{r}_{i+1}\|^2} - \frac{\mathbf{e}_i \cdot \mathbf{r}_i}{\|\mathbf{r}_i\|^2} + \|\mathbf{e}_i\|^2 I_1 \right), \\
I_3 &= \frac{1}{4} \frac{1}{\|\mathbf{e}_i \times \mathbf{r}_i\|^2} \left(\frac{\mathbf{e}_i \cdot \mathbf{r}_{i+1}}{\|\mathbf{r}_{i+1}\|^4} - \frac{\mathbf{e}_i \cdot \mathbf{r}_i}{\|\mathbf{r}_i\|^4} + 3\|\mathbf{e}_i\|^2 I_2 \right), \\
J_2 &= \frac{1}{2\|\mathbf{e}_i\|^2} \left(\frac{1}{\|\mathbf{r}_i\|^2} - \frac{1}{\|\mathbf{r}_{i+1}\|^2} \right) - \frac{\mathbf{e}_i \cdot \mathbf{r}_i}{\|\mathbf{e}_i\|^2} I_2, \\
J_3 &= \frac{1}{4\|\mathbf{e}_i\|^2} \left(\frac{1}{\|\mathbf{r}_i\|^4} - \frac{1}{\|\mathbf{r}_{i+1}\|^4} \right) - \frac{\mathbf{e}_i \cdot \mathbf{r}_i}{\|\mathbf{e}_i\|^2} I_3, \\
K_3 &= \frac{1}{\|\mathbf{e}_i\|^2} \left(I_2 - \|\mathbf{r}_i\|^2 I_3 - 2(\mathbf{r}_i \cdot \mathbf{e}_i) J_3 \right), \text{ and} \\
Q(\mathbf{a}, \mathbf{b}) &= \mathbf{a}\mathbf{b}^\intercal + \mathbf{b}\mathbf{a}^\intercal.
\end{aligned}$$

Note that **a** and **b** are column vectors, such that $Q(\mathbf{a}, \mathbf{b})$ results in a 3x3 matrix.

Arvo [Arv94] presented a formula for the irradiance Jacobian due to a diffusely emitting triangle, which results in an alternate, but equivalent, expression for the form-factor gradient. This is discussed in section 5.1

The Hessian in Equation (6.3) is fully 3D, whereas we are interested only in the Hessian across the surface on which it was computed. To project the 3×3 Hessian matrix onto the surface, we use the same method outlined in section 4.2.

Constructing the Triangulated Environment.

To construct our triangulated environment, we consider the hit distance/position for each sample ray we trace. We then go over all the samples and connect neighbors into triangles, defining a triangular mesh that covers the entire hemisphere. Though any triangulation algorithm could be used, we use the connectivity information implicit to the stratification to make this step efficient (see Figure 6.5). The mesh, along with the incoming radiance information, becomes a 3-dimensional snapshot of the scene, as it is seen from **x**. Figure 6.4 illustrates this for a shade point in the Cornell box scene.

The most important features visible in Figure 6.4 are the large triangles that connect the surface of the occluder geometry to that of the wall. It is these

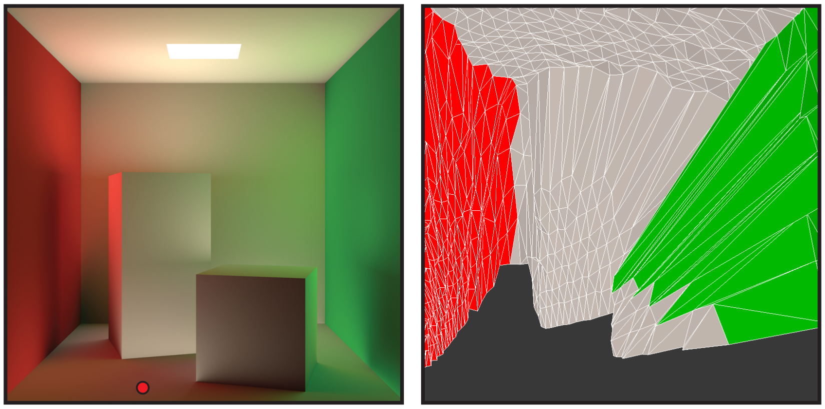


Figure 6.4: Example of our mesh-based geometry approximation using the stratified sample data. The left image shows the Cornell Box with a red dot indicating the visualized location in the scene. The right image shows the triangle mesh approximation after applying our method, in this case using a total of 4096 gather rays. Note that rays that hit nothing are not shown.

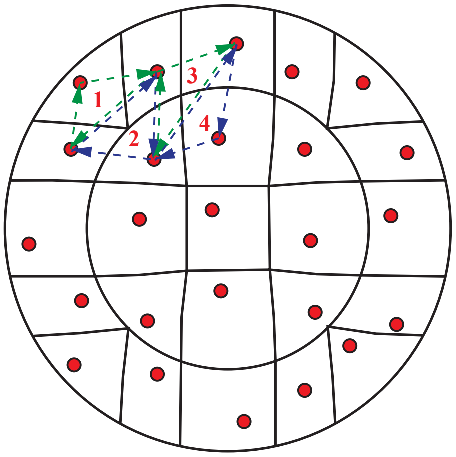


Figure 6.5: The implicit connectivity information inherent in the stratified sampling allows efficient triangulation of the environment approximation. Final gather rays are shown as red points, and the implicit triangulation is shown in green.

polygons that encode the occlusion information of the scene from the point of view of the sampling location $\mathbf{x}$. While these triangles have a large geometric extent, they subtend a similar solid angle as other triangles, and thus their contribution to irradiance is not disproportionate. As $\mathbf{x}$ is translated, however, the solid angle of these large triangles will grow (or shrink) much faster than for other triangles, leading to a large gradient and Hessian due to occlusion changes.

To evaluate the irradiance gradient and Hessian, we need to define the incoming radiance $L_{\triangle}$ due to each of the triangles in the mesh. In 2D, as shown in Figure 6.2(d), this corresponds to the incoming radiance stored at the vertex that is farthest away from the shading location $\mathbf{x}$. The same idea applies in 3D, though, we choose the farthest of the three vertices of each triangle (instead of two as in the 2D case). Intuitively, vertices that are farthest from $\mathbf{x}$ define the color of objects that become disoccluded during translation of $\mathbf{x}$. Ward and Heckbert [WH92] used a similar idea to estimate the differential change in occlusion between strata when computing the gradient. As shown in Figure 6.6, this heuristic results in a good approximation of the true first and second derivatives of the irradiance for scenes with significant occlusions. Not surprisingly, the previous occlusion-less Hessian derived by Jarosz et al. [JSKJ12] does not match the ground truth.

Our formulation allows us to implicitly account for occlusion changes in both the gradient and Hessian without resorting to more complex form factor computations which account for occlusions explicitly, and while avoiding the patho-

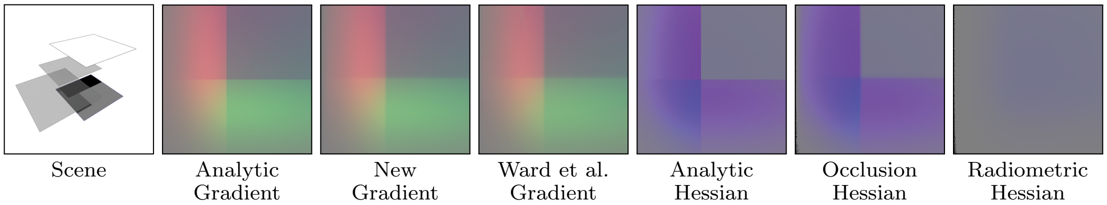


Figure 6.6: The left figure shows the scene configuration: a bright area emitter was placed over a large plane, with two occluders located between the light and the plane at different distances. The three middle-left figures show a visualization of the first derivative of the irradiance across the bottom plane, using the mapping $(dE/dx, dE/dy) \rightarrow (r, g)$, while the three right figures show the eigenvalues of the irradiance Hessian using the mapping $(\lambda_1, \lambda_2, \lambda_3) \rightarrow$ (r,g,b). The numerical results use 4K sample rays per pixel.

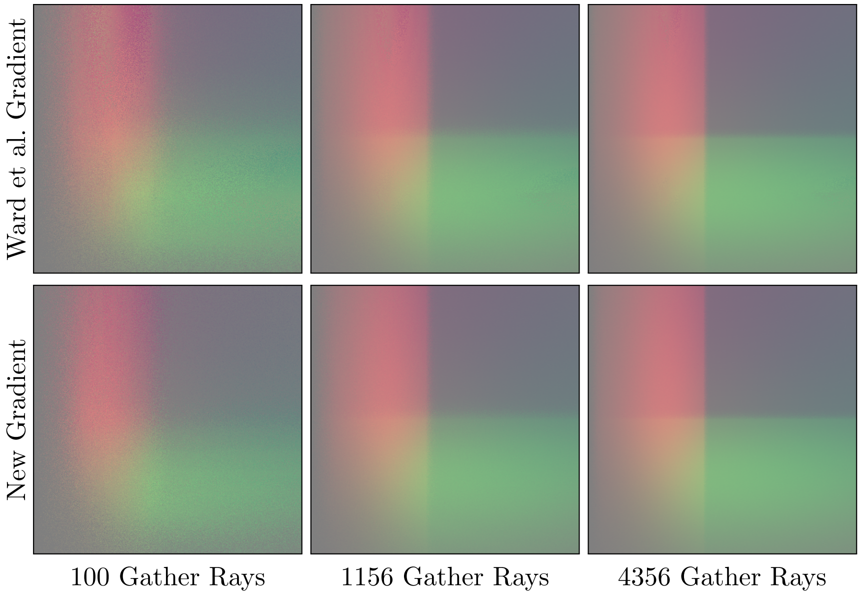


Figure 6.7: Quality comparison between the classic Ward et al. irradiance gradient and our new formulation. While the gradient approximations differ slightly – especially at lower gather ray counts the Ward et al. gradient suffers from stronger artifacts – the qualitative results are similar.

logical cases that preclude their use in irradiance caching. Furthermore, our formulation has practical benefits over the the standard approach proposed by Ward because it allows us to use arbitrary hemispherical distributions without re-deriving the method. Though we tried the standard longitude-latitude stratification, we use Shirley and Chiu's [SC97] concentric mapping in our implementation because it produces more regular sampling patterns, leading to better results.

6.3 Application to Irradiance Caching

To compute the anisotropic validity region for the cache records, we could now use Equations (4.21) and (4.22), while using our improved occlusion-aware Hessian. However, Equation (4.21) is suboptimal since it defines the error with respect to the absolute variation in irradiance. We would instead like to define a relative error term. This has a number of advantages: 1) it more strongly relates to our visual system's response to contrast instead of absolute changes in intensity and 2) it makes the error independent of the absolute scale of the scene and absolute intensity of the light sources used, making its parameters more compatible across scenes. To accomplish this, we modify the error formula by dividing the Hessian by the indirect irradiance computed at the sample location:

$$\epsilon_i^r \approx \frac{1}{2}\iint_A \frac{|\Delta\mathbf{x}^{\intercal}\,\mathbf{H}_{\mathbf{x}}^{2\times 2}(E_i)\,\Delta\mathbf{x}|}{E_i}\,d\Delta\mathbf{x} \tag{6.7}$$

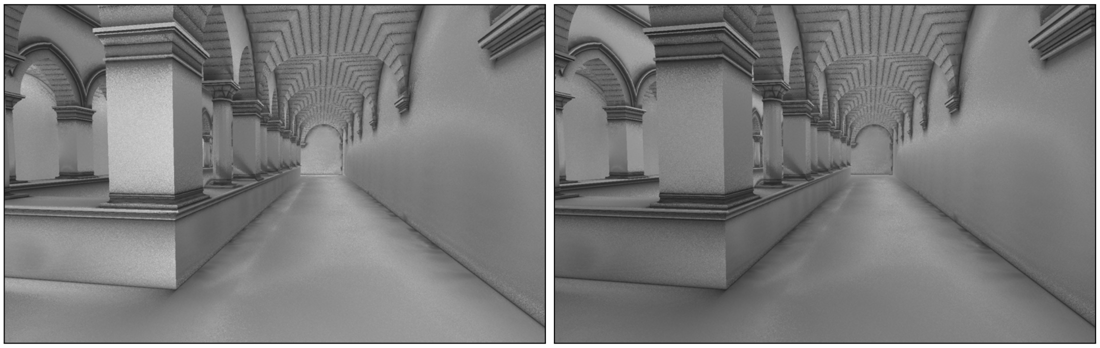
$$= \frac{1}{2}\iint_A \left(\frac{|\lambda_1|}{E_i}x^2 + \frac{|\lambda_2|}{E_i}y^2\right)\,dy\,dx. \tag{6.8}$$

This leads to the following anisotropic cache record radius equation:

$$\left(R_i^{\lambda_1},\, R_i^{\lambda_2}\right) = \sqrt[4]{\frac{4\epsilon^r E_i}{\pi}}\left(\sqrt[4]{\frac{1}{\lambda_1}},\, \sqrt[4]{\frac{1}{\lambda_2}}\,\right), \tag{6.9}$$

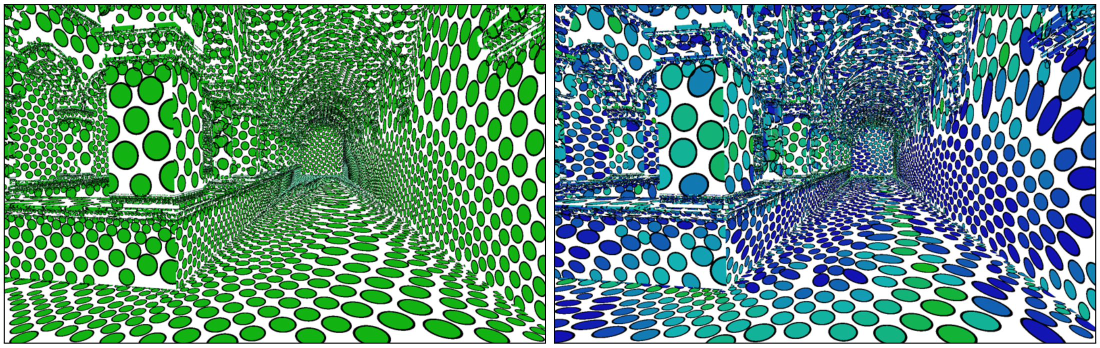
where ϵ^r becomes the primary (relative) error control parameter in our algorithm. The primary differences compared to Equation (4.22) are that we use our new occlusion-aware Hessian, and that the cache point size is scaled by the fourth-root of the indirect irradiance (more absolute error can be tolerated in bright regions of the scene than in dark regions).

In Figure 6.8 we compare the radius derived using a relative error versus an absolute error in the Sponza scene. The walls in the inner part of the Sponza courtyard have much brighter indirect irradiance, which allows for larger radii when accounting for relative error. For an absolute error, doubling the intensity of the lighting would modify the sample distribution (forcing the user to manipulate the error threshold to obtain the same image), whereas this would have no effect on our relative metric. In Figure 6.9 we additionally visualize the effect of allowing anisotropic cache records compared to forcing isotropic records (by using the minimum of the two radii from Equation 6.9). Anisotropic records adapt their eccentricity to the local irradiance curvature, allowing for fewer cache records (16.5K vs. 20.5K) for the same error threshold. In practice, we clamp the major axis of the elliptical cache records at twice the length of the minor axis, to prevent artifacts that can arise in cases where the irradiance Hessian is locally very small in one direction.



Occlusion Hessian (Absolute) Occlusion Hessian (Relative)

Figure 6.8: Visualization of the radii estimated for the Occlusion Hessian using both absolute and relative measures for the total error. The relative measure allows larger radii in bright regions where absolute differences in irradiance have a smaller impact on the perceived error.



Isotropic Occlusion Hessian Anisotropic Occlusion Hessian

Figure 6.9: Comparison of isotropic to anisotropic cache records for the relative Occlusion Hessian metric with the same threshold. The eccentricity of the cache records is visualized as the filled-in color, with green representing isotropic records and dark blue representing maximum anisotropy. Note that we have clamped the major axis at twice the length of the minor axis.

Pathological Cases.

One of the fundamental problems with any radiometric approach is that pathological cases (where all or most of the gather rays return black) can result in undefined or infinite radii. When computing full global illumination with many bounces, this is rarely a problem; however, when computing only a single bounce of indirect, this problem occurs more often. Jarosz et al. [JSKJ12] noted this issue, which forced them to revert to a completely geometric approach (the Geometric Hessian). To retain the benefits of a radiometric approach, we instead add 1% of the indirect irradiance E_i to all the triangle radiances $L_{\triangle_j}$ before computing the Hessian in Equation (6.3). In the case where *all* gather rays return black, we set all triangle radiances $L_{\triangle_j} = 1$. This has the effect of switching to an occlusion-aware geometric Hessian only for this special case. Note that we only add 1% during the Hessian computation and do not modify the irradiance stored in the cache.

Evaluation.

In Figure 6.10 we show the effect of incorporating these improvements (moving from an absolute to a relative error, and adding 1% of the irradiance) when applied to both the occlusion-less Radiometric Hessian, and our Occlusion Hessian. These two changes (moving left to right) improve both methods, but the most striking improvement comes from accounting for occlusions in the Hessian itself (bottom row vs. top row).

To validate that our radius computation is robust to low gather ray counts, in Figure 6.11 we compare the cache point distribution for the Cornell box scene using 256 gather rays and 4096 gather rays. Even with 16 times less gather rays, the cache point distribution remains qualitatively the same.

In Figure 6.12 we evaluate the Radiometric Hessian and Occlusion Hessian on a simple scene with an indirect occluder. The Occlusion Hessian successfully concentrates samples in regions of rapid irradiance change due to occlusions (the indirect penumbra regions).

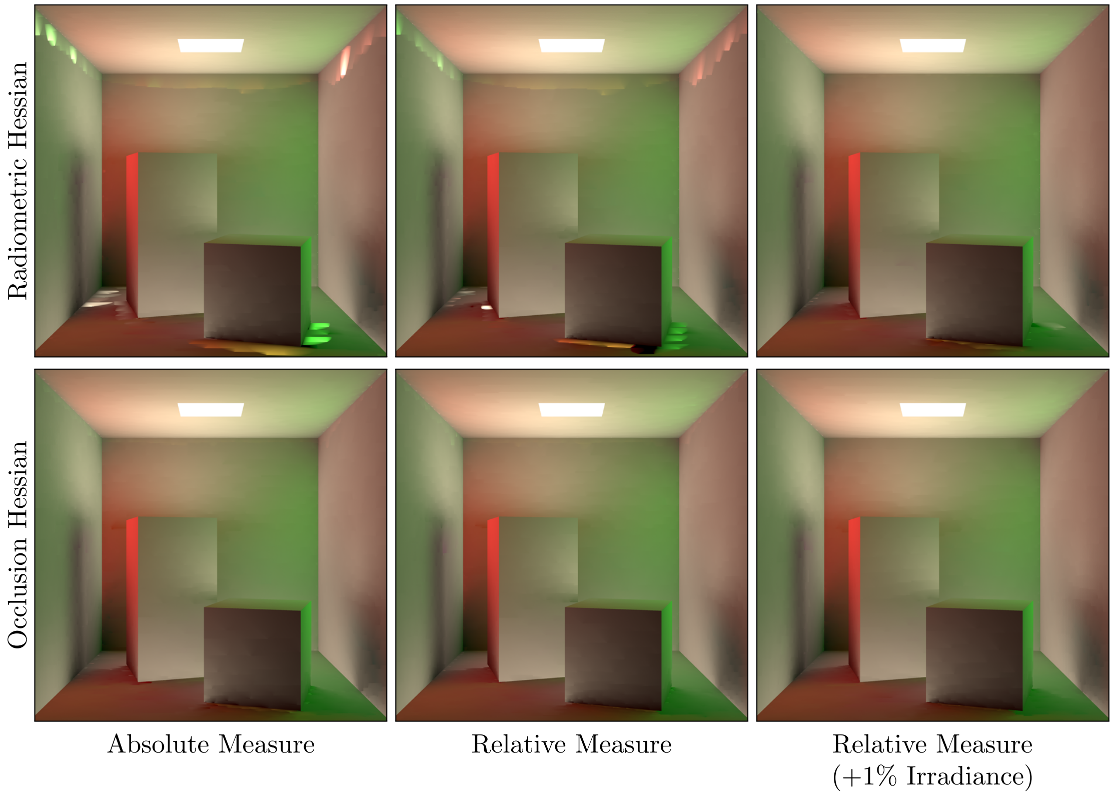
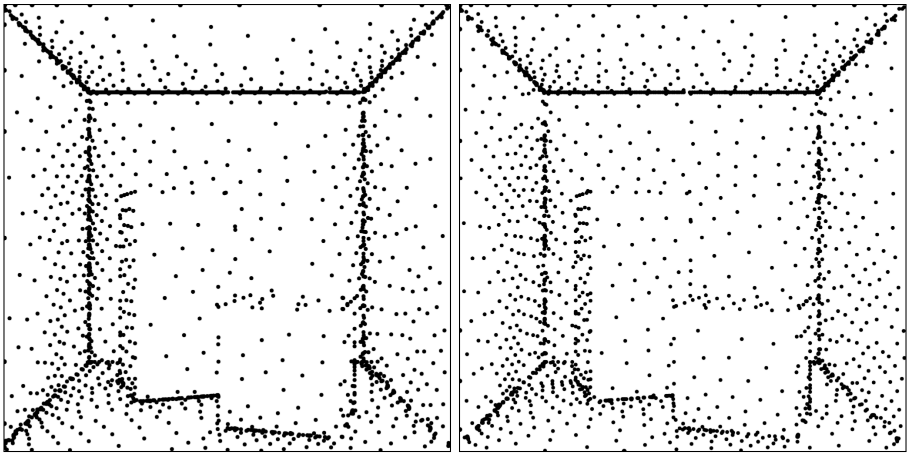


Figure 6.10: Comparison of the Radiometric and Occlusion Hessian methods and our improvements for relative error and robustness. The first column shows the result when using an absolute error, while the second column uses a relative measure. The third column adds 1% of the total irradiance E_i to all the triangle radiances $L_{\triangle_j}$ prior to computing the Hessian. While this improves the result for the Radiometric Hessian method, it still retains a few distracting artifacts, that are not present for the Occlusion Hessian.



256 Gather Rays

4096 Gather Rays

Figure 6.11: Our method is robust to low gather ray counts, producing nearly identical cache point distributions with 256 (left) as with 4096 (right) gather rays per cache record.

6.3.1 The Rotational Component

To account for changes in irradiance due to rotation, we could follow a similar procedure and derive a rotational irradiance Hessian by using rotational form-factor derivatives in Equations (6.2) and (6.3). Since the rotational derivatives only account for the change in the cosine foreshortening term as the surface normal is rotated, and no occlusion changes occur, the computation simplifies to the first and second derivatives of a cosine. Unfortunately, this approach does not work well. The problem is that occlusions in fact do occur: for any significant rotation, the change in the cosine factor is insignificant compared to the impact of regions from the lower hemisphere becoming disoccluded and contributing to the irradiance. In effect, rotational derivates in the upper hemisphere ignore all occlusions (the occlusions along the hemispherical boundary), and this has by far the biggest impact on the change in irradiance. Unfortunately, since we can only reliably sample the upper hemisphere, we have little hope of detecting what is under the upper hemisphere to accurately predict the change in irradiance.

In the face of these challenges, we opt for a simple but intuitive strategy as originally proposed by Tabellion and Lamorlette [TL04]. We enforce a maximum deviation angle $\Delta\mathbf{n}_{\max}$ that is allowed during extrapolation: no cache records are allowed to extrapolate beyond an e.g. 10° deviation in surface normal. We set the default value to $\Delta\mathbf{n}_{\max} = 0.2$ radians, but allow the user to modify this if needed.

6.3.2 Irradiance Storage and Interpolation

With each cache point i we store: the location and normal $(\mathbf{x}_i, \mathbf{n}_i)$, the irradiance (E_i), irradiance gradients $(\nabla_{\mathbf{x}} E_i,\ \nabla_{\mathbf{n}} E_i)$, the two anisotropic radii $(R_i^{\lambda_1}$, $R_i^{\lambda_2})$, and the corresponding eigenvectors $(\mathbf{v}_i^{\lambda_1}, \mathbf{v}_i^{\lambda_2})$.

Note that computing the eigenvalue decomposition for $\mathbf{H}_{\mathbf{x}}^{2\times 2}$ in Equation (6.8) results in eigenvectors in the tangent-space defined by vectors $\mathbf{u}_1$ and $\mathbf{u}_2$. To make these usable during interpolation, we transform each tangent-space eigenvector $(\mathbf{v}_i')$ into world-space coordinates $(\mathbf{v}_i = [\mathbf{u}_1 \cdot \mathbf{v}_i', \mathbf{u}_2 \cdot \mathbf{v}_i']^{\intercal})$ before storing it in the cache.

To interpolate cache records, we use the same first-order extrapolation strategy as proposed by Ward and Heckbert [WH92], but with a modified weighting function:

$$E(\mathbf{x}, \mathbf{n}) \approx \frac{\sum_{i\in S} \mathrm{w}_i(\mathbf{x})(E_i + \nabla_{\mathbf{x}} E_i \cdot \Delta\mathbf{x}_i + \nabla_{\mathbf{n}} E_i \cdot \Delta\mathbf{n}_i)}{\sum \mathrm{w}_i(\mathbf{x})} \tag{6.10}$$

where $\Delta\mathbf{x}_i = \mathbf{x} - \mathbf{x}_i$ and $\Delta\mathbf{n}_i = \mathbf{n} \times \mathbf{n}_i$ are the translational and rotational deviation of the shading location $\mathbf{x}$ with normal $\mathbf{n}$ to the cache point.

The set S includes all cache records whose valid regions (Equation (6.9)) overlap $\mathbf{x}$ and whose normal deviation is less than $\Delta\mathbf{n}_{\max}$. To weight the cache records, we propose to use a kernel weighting in both translation and orientation that falls off to zero at the boundaries:

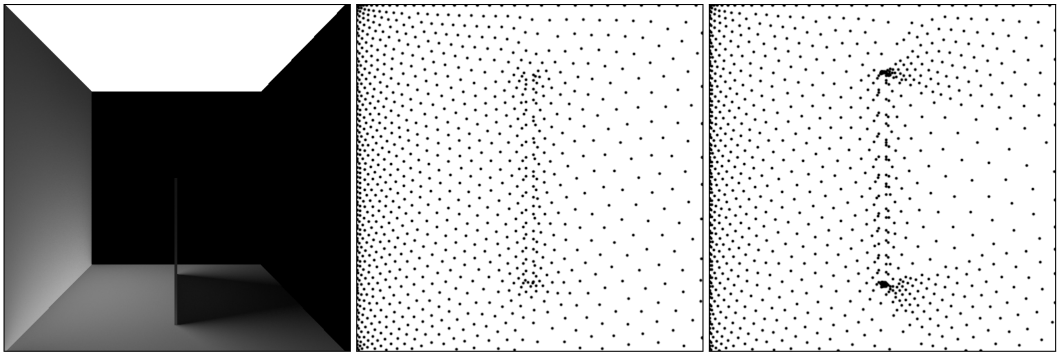
$$\mathrm{w}_i(\mathbf{x}) = k(1 - t_{\mathbf{x}}, 0, 1) \times k(t_{\mathbf{n}}, \cos\Delta\mathbf{n}_{\max}, 1), \tag{6.11}$$

where k is a simple tent filter, $k(t, t_{\min}, t_{\max}) = \frac{t - t_{\min}}{t_{\max} - t_{\min}}$, and

$$t_{\mathbf{x}} = \sqrt{\left[\frac{\Delta\mathbf{x}_i \cdot \mathbf{v}_i^{\lambda_1}}{R_i^{\lambda_1}}\right]^2 + \left[\frac{\Delta\mathbf{x}_i \cdot \mathbf{v}_i^{\lambda_2}}{R_i^{\lambda_2}}\right]^2 + \left[\frac{\Delta\mathbf{x}_i \cdot \mathbf{n}_i}{R_i^{\lambda_1}}\right]^2}, \tag{6.12}$$

$$t_{\mathbf{n}} = (\mathbf{n} \cdot \mathbf{n}_i), \tag{6.13}$$

are the translational and rotational distances. We use the cache record only if $\mathrm{w}_i(\mathbf{x})$ is greater than zero. Note that k could easily be replaced with a higher-order smooth kernel that falls off to zero, but we did not find this to provide a significant benefit.



Scene

Radiometric Hessian

Occlusion Hessian

Figure 6.12: A simple scene with indirect occlusions (left) where we visualize the emitted radiance on the emitter above and the indirect illumination on the other surfaces. The Radiometric Hessian (middle) ignores occlusions, producing a relatively uniform distribution whereas our Occlusion Hessian method (right) successfully adapts to the irradiance change near the penumbra region.

6.4 Results

We implemented the new error metric by modifying the native Irradiance Cache implementation in version 2 of PBRT [PH10]. All results were rendered using 4 samples per pixel on a PC with a 2.66Ghz Intel Core i7-920 CPU. Our implementation of the Split-Sphere heuristic follows Ward's [WRC88] original formulation and uses the harmonic mean distance. For the Bounded Split Sphere we clamp this radius to a maximum pixel size and to the computed irradiance gradient (increasing the sample density in areas where the first-order gradient magnitude exceeds the Split-Sphere prediction). All methods use 4096 rays per irradiance sample, and we enforce a minimum cache point radius equivalent to the projected size of 1 pixel at the sample location, except where noted. For the Hessian based

Table 6.1: Rendering times for the scenes and methods presented.

	Cornell Box	**Sponza**	**San Miguel**
Method	Figure 6.1	Figure 6.16	Figure 6.18
Split-Sphere	01:02.3	53:08.4	07:15:22.4
Radiometric Hessian	01:06.5	n/a	n/a
Geometric Hessian	01:04.7	n/a	n/a
Occlusion Hessian	01:16.5	54:35.6	07:17:43.7

methods there is no upper bound for the cache point radii. All Hessian based results use the default value $\Delta\mathbf{n}_{\max} = 0.2$ radians as the maximum normal deviation angle. We report the render times in Table 6.1.

Figure 6.1 shows a comparison of the Occlusion Hessian error control method against the Split-Sphere Heuristic as well as the Geometric and Radiometric Hessian metrics defined by Jarosz et al. [JSKJ12]. All images were rendered at a resolution of 1800x1800 pixels with a single bounce of indirect illumination. The error thresholds were adjusted for each method to enable the use of 1700 +/- 2% cache records. With just 1700 cache records the Split-Sphere method shows severe interpolation artifacts. Even with a maximum radius of 150px (a lower threshold is not possible with 1700 cache records) the Bounded Split-Sphere method shows significant errors throughout the rendered image. The Radiometric Hessian works well in lit regions, but has artifacts in shadowed regions. The Geometric Hessian works well in most of the scene, but shows interpolation artifacts on the walls and the ceiling. The Occlusion Hessian produces the best overall result, resolving the detail in the shadow while maintaining a sufficient density of samples on the walls and ceiling to reconstruct the irradiance. For the simple geometry in this scene the Occlusion Hessian added roughly 20% overhead to the computation time. For our more complex scenes this overhead becomes negligible as the time is dominated by ray tracing.

Figure 6.13 compares the behavior of our Occlusion Hessian method to the Bounded Split-Sphere heuristic with increasing cache point densities. All images used 4096 gather rays, and the Split-Sphere used a maximum radius of 150px (a tighter bound was not possible for very limited cache point counts). Note how the Occlusion Hessian produces a high-quality result even with 500 cache points, while the Split-Sphere suffers from large artifacts even with 1000 cache points.

In Figure 6.14 we extended the Split-Sphere with progressive refinement, neighbor clamping, and an overture pass [KBPZ06]. Rendering of the enhanced Split-Sphere image took 123.4s while the Occlusion Hessian rendering, with the overture pass, took 122.6s. The quality obtained by the Occlusion Hessian is visibly better, even though the improvement for the Split-Sphere is significant.

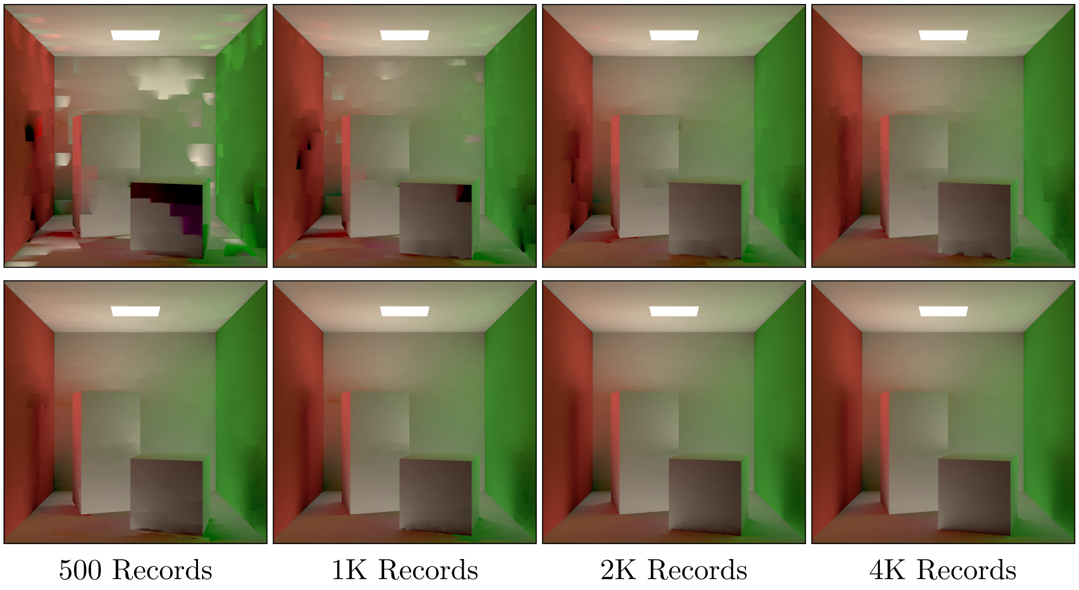


Figure 6.13: Comparison of the convergence behavior for the Bounded Split-Sphere (top) and our Occlusion Hessian (bottom).

Note that the overture pass has less impact on the Occlusion Hessian as the error metric more closely follows the actual error in the irradiance.

The Geometric Hessian is generally robust in most scenes, but since it is a purely geometric metric, like the Split-Sphere it does not adapt to changes in the illumination. In Figure 6.15 the Cornell box has been changed such that the backwall is completely black. This adds contrast to the indirect illumination, which the Occlusion Hessian is able to detect and consequently it produces a higher quality output with fewer artifacts than the Geometric Hessian.

Figure 6.16 shows close-ups of Sponza rendered using the Occlusion Hessian and the Split Sphere heuristic (both unbounded and with a maximum radius of 20 pixels). For this scene we used a photon map to add multiple bounces of indirect illumination. The irradiance cache is used to gather the irradiance at the first diffuse surface seen by the eye, and we used 32K +/- 1% cache records for the full images. The resulting images show how the Split-Sphere heuristic has interpolation artifacts in the shadows on the columns, while the Occlusion Hessian is much better at reconstructing the lighting details. Figure 6.17 shows a direct visualization of the cache point radii for the Sponza scene with the various methods. Note that we only visualize the smaller of the two radii for the anisotropic methods. The Split-

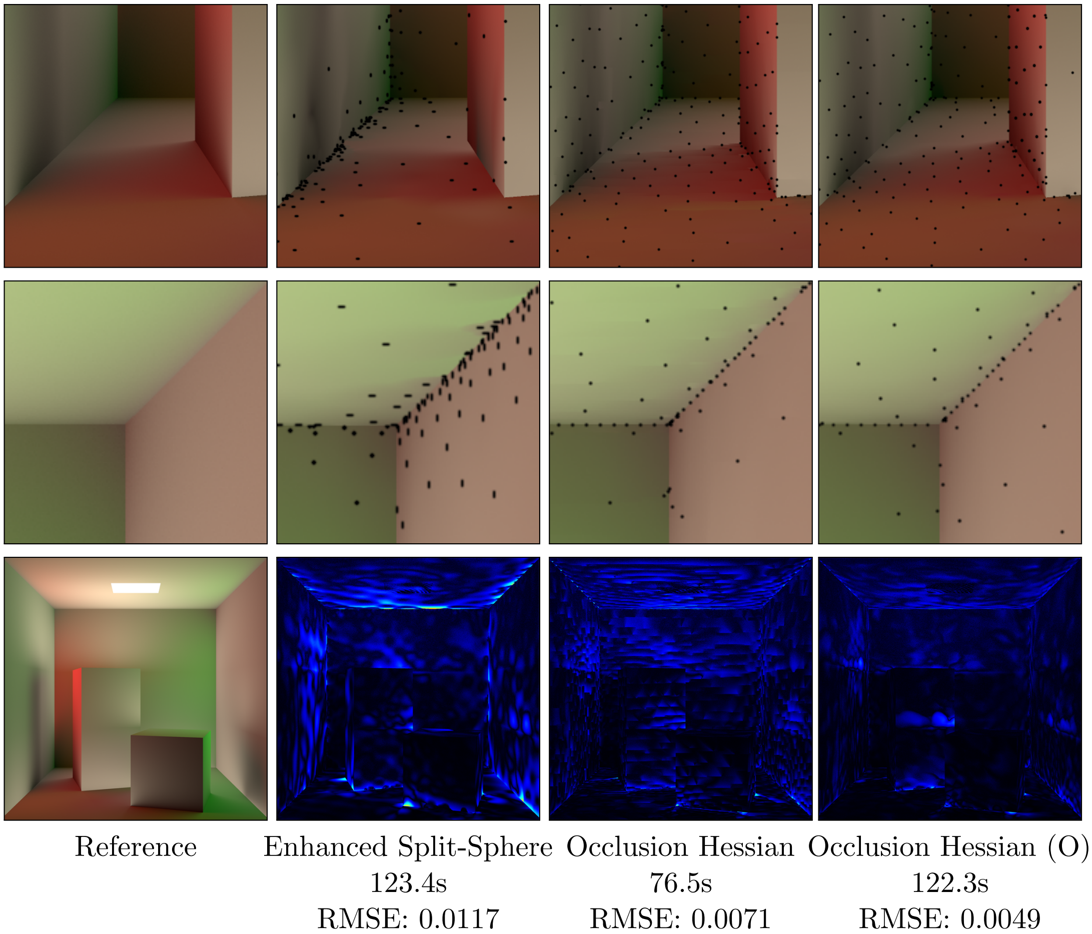


Figure 6.14: We show the result of adding progressive refinement, neighbor clamping and an overture pass [KG09] to the Split-Sphere, and compare to our Occlusion Hessian method without and with an overture pass. While these additions significantly improve the Split-Sphere result, render time is increased, and it still does not match the quality achieved by the Occlusion Hessian without an overture pass.

Sphere images show that additionally bounding the cache point radii by the true first-order gradient leads to smaller radii in areas of high irradiance variance, such as the indirect penumbras visible at the base of the columns. This behavior is, as expected, mirrored by the Occlusion Hessian, which defines the radii in proportion to the rate of change of irradiance. However, the contrast is much higher for the Split-Sphere images - the radii variations are much larger - which explains the worse results obtained from this metric (e.g. sampling the scene outside of the penumbra regions leads to cache points with very large radii and the penumbra regions become severely under-sampled).

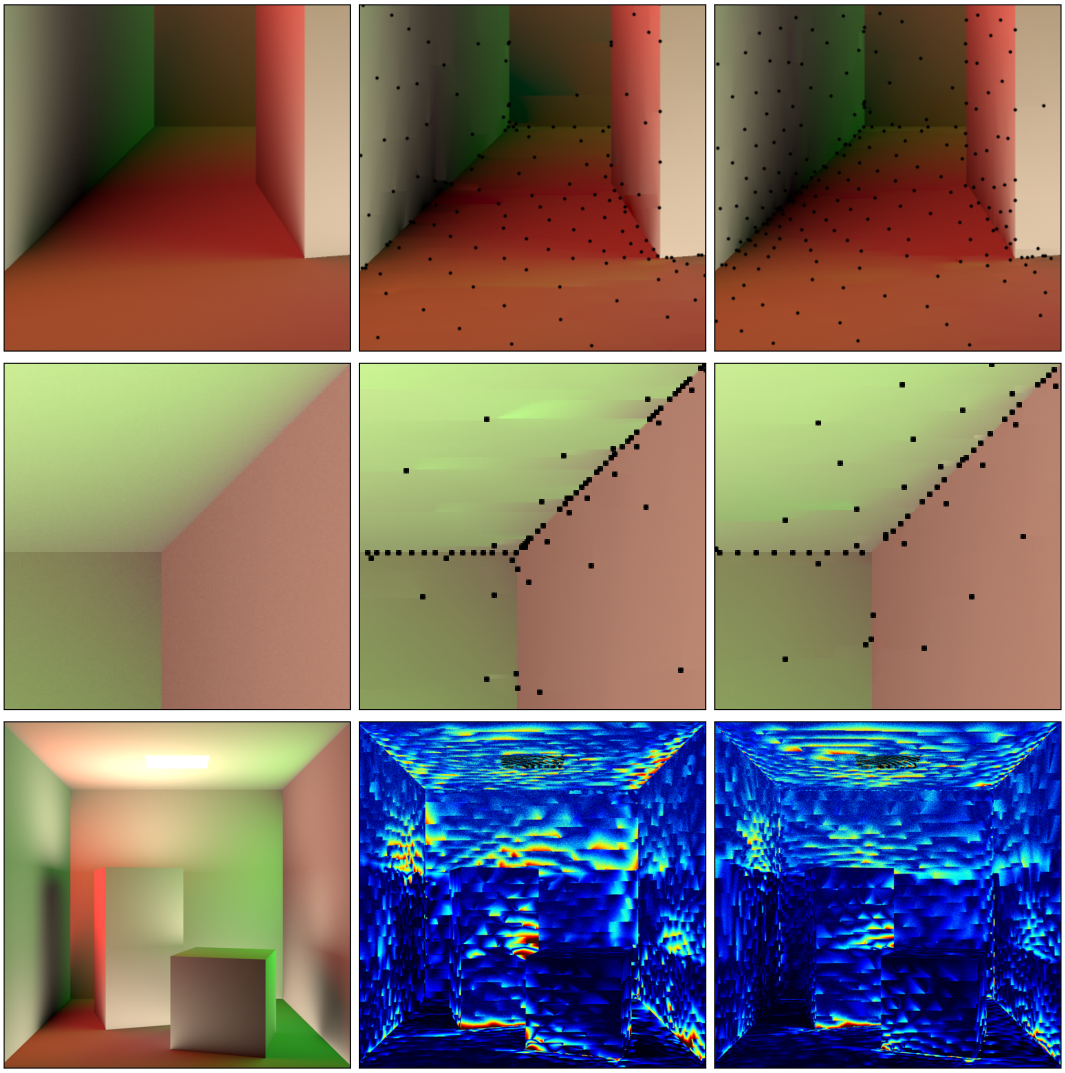
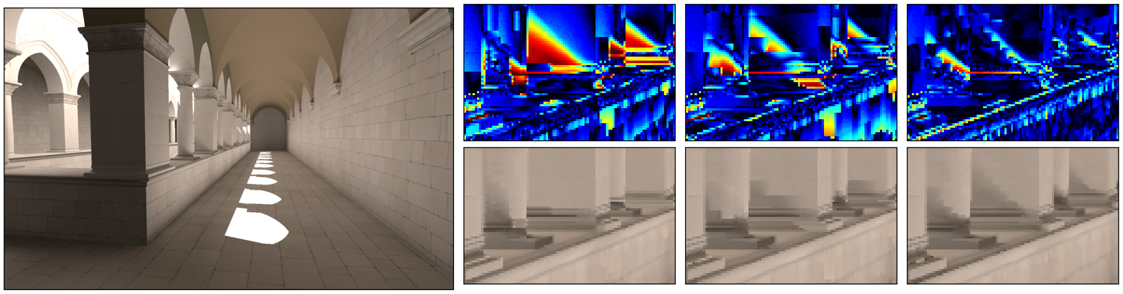


Figure 6.15: Irradiance rendering of the Cornell Box after changing the back wall so it has an albedo of 0. The Geometric Hessian does not take the radiometry of the scene into account, so its cache point distribution is the same as for the standard Cornell Box, which is sub-optimal. Our new Occlusion Hessian adapts its distribution to match the radiometry in this new configuration, producing a higher quality result. Note that we have scaled the difference image color range to ease comparison.



Reference | Pure Split-Sphere | Bounded Split-Sphere | Occlusion Hessian

Figure 6.16: In Sponza the Occlusion Hessian is superior to the Split-Sphere in both defining shadow details and eliminating interpolation artifacts across surfaces. The shadows at the base on the column are captured by the Occlusion Hessian, while the Split-Sphere heuristic shows severe interpolation artifacts, even when bounded to 20px and clamped by the gradient.

Figures 6.18 and 6.19 show interior and exterior views of the PBRT San Miguel scene. The methods were tuned so that 50K cache points (+/- 1%) would be produced. A photon map was used for the secondary light bounces, with 1 million photons stored. Our Occlusion Hessian approach was able to properly define all indirect shadows in the image, and exhibits fewer interpolation artifacts in areas of the scene where the irradiance varies slowly. Rendering times for both the Bounded Split-Sphere and the Occlusion Hessian were within 0.5% of each other.

6.5 Conclusion

In this chapter we have described a new method for controlling the approximation error in irradiance caching. The new method is based on computing the second derivative of irradiance - the translation Hessian - while taking object inter-occlusions into account. A second-order Taylor expansion of irradiance is used as an oracle to the true irradiance, in order to estimate the error induced by first-order Taylor expansion during rendering. The Hessian also naturally supports anisotropic cache points. As a result, record density closely matches the rate of change of irradiance in the scene, leading to reduced error in the rendered images when compared to all previous approaches. Furthermore, our method does

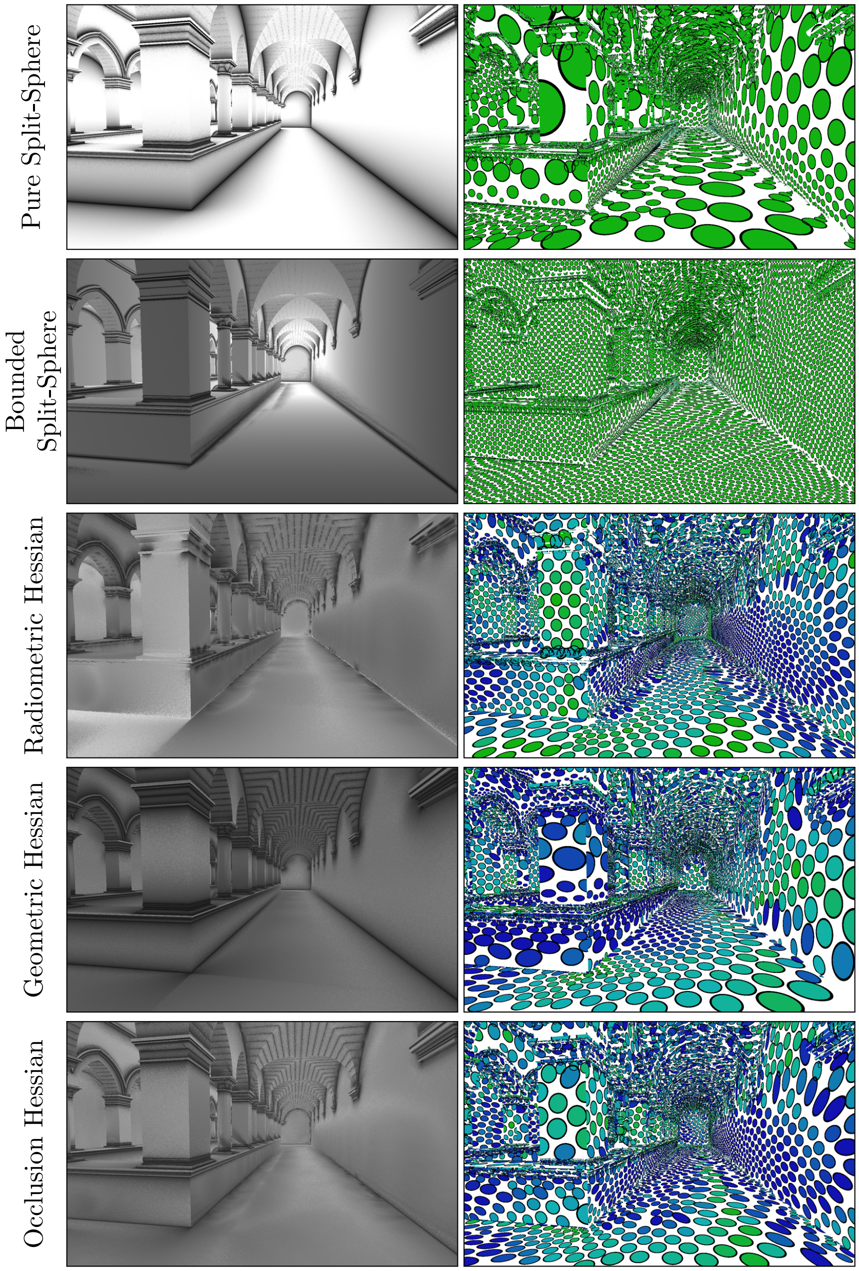


Figure 6.17: Direct visualization of the radii (left) and footprints (right) of the cache records for various error metrics. The Pure Split-Sphere, Radiometric and Geometric Hessians are all unable to detect the indirect shadows on the columns, while the Bounded Split-Sphere (clamped to 20px and the gradient) as well as the Occlusion Hessian correctly predict smaller radii in these rapidly changing regions. Unfortunately, the contrast in the Split-Sphere is too high, leading to lost detail due to the greedy nature of the irradiance caching algorithm.

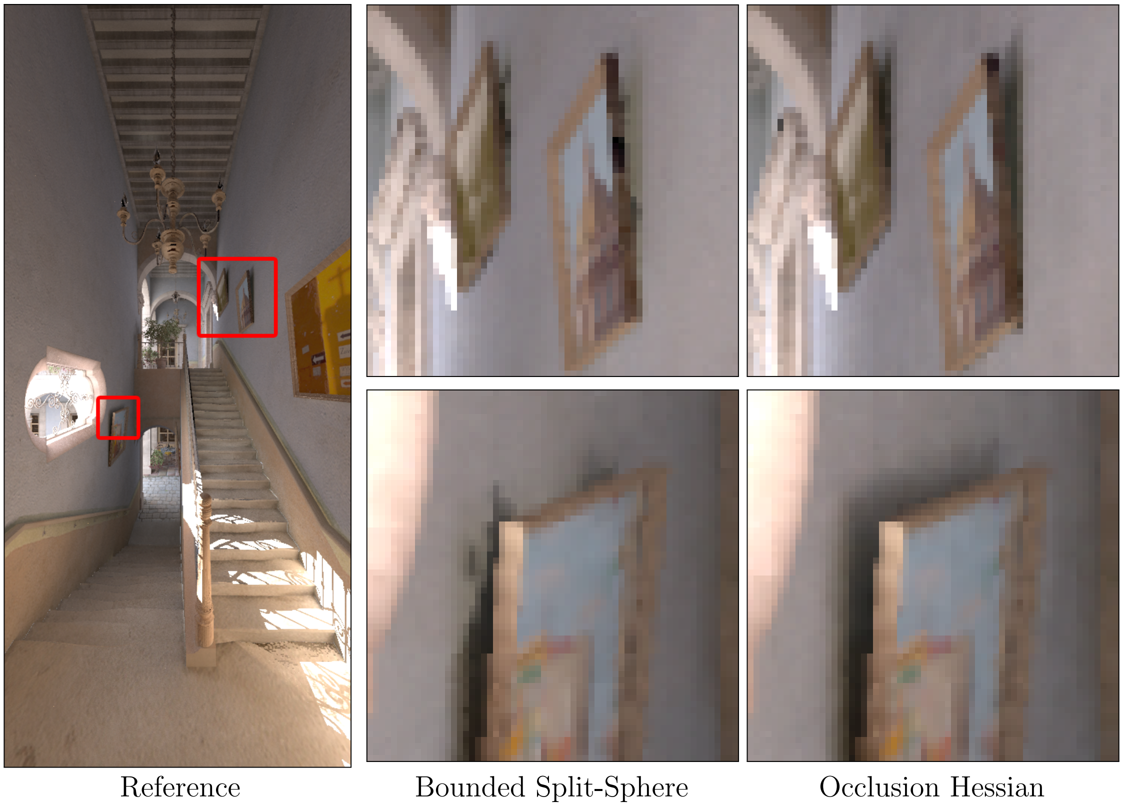


Figure 6.18: An interior of San Miguel with strong indirect illumination. The close-ups show how the Occlusion Hessian is able to resolve the shadows behind the paintings, while the Split-Sphere heuristic (clamped to 20px and the gradient) largely misses these shadows.

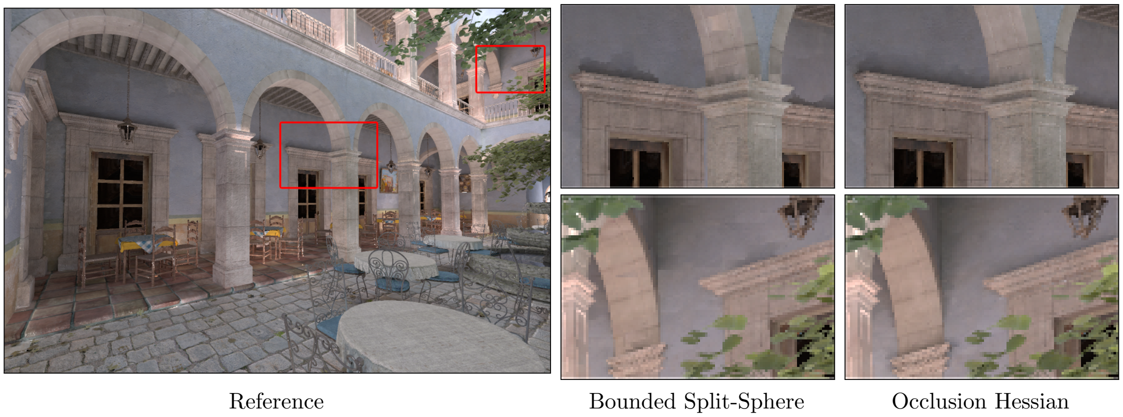


Figure 6.19: An exterior of San Miguel with strong indirect illumination (only the irradiance is shown). The close-ups show how the Occlusion Hessian is able to resolve the shadows above the door frames, while the Split-Sphere heuristic (clamped to 20px and the gradient) results in broken, unrealistic shadows.

not rely heavily on user-defined minimum and maximum cache point radii, leading to more intuitive user control. The robustness of our new Occlusion Hessian method also minimizes the need to use other corrective techniques, such as progressive refinement, neighbor clamping, or an overture pass, resulting in a simpler implementation.

6.6 Acknowledgements

The material in this chapter is, in part, a reproduction of the material published in Jorge Schwarzhaupt, Wojciech Jarosz and Henrik Wann Jensen. “Practical Hessian-Based Error Control for Irradiance Caching.” In *ACM Transactions on Graphics*, 31(6)1-10, 2012. The thesis author was the primary investigator and author of this paper.

Chapter 7

Conclusions

The goal of this thesis was the improvement of the irradiance caching algorithm. To reach this goal we have derived a new method for controlling the approximation error inherent to the algorithm, which is based on a new formulation of the irradiance Hessian that takes object inter-occlusions into account. We estimate the error induced by a first-order Taylor expansion by turning to a second-order Taylor expansion as an oracle of the true irradiance. Our Hessian also naturally supports anisotropic cache points, so that record density closely matches the rate of change of irradiance in the scene, leading to reduced error in the rendered images when compared to all previous approaches.

The results in the preceding chapter clearly show that we have produced a method that significantly improves upon the state of the art regarding irradiance caching. While the original algorithm was derived in a very principled manner, it was subsequently improved upon using much more ad-hoc methods that attempted to ameliorate specific perceived short-comings, leading to a state where implementing a good irradiance caching algorithm required referencing many sources and understanding how to control the parameters of however many of these fixes one wished to support. One of our primary goals in this work was to derive a principled and robust error metric that would render the original algorithm usable in real-world, modern scenes, without needing to rely on (too many) extra parameters, and we feel we have achieved this.

The work we performed in deriving our final occlusion-aware method, where we perform a transformation of the three-dimensional scene so it has no occlusions, shows very clearly how it is sometimes required to think outside the box in order to come to a good, practical solution to a problem. In our case, we wanted to find a way to accurately and - just as importantly - quickly compute the second derivative of irradiance in scenes with occlusions; it is only too ironic that the ultimate answer to this required us to strip occlusions away from our input scenes! Given the simplicity of our approach and the generality it gains when using a sampling-based approach - since we don't need to find the *true* transformation but only the one for our sampled representation of the scene - we are quite hopeful that it will find use beyond that which we have given it.

We have also contributed a full derivation of the second derivative of the light-field (vector irradiance) in the presence of occlusions, which though ultimately unsuitable for our purposes might yet prove useful in some other context, and further research as to the applicability of the derivation in the area of light transport would be interesting. Indeed, the mathematical framework exists already in order to extend these formulas to a third derivative of Irradiance. While it would be algebraically complex - it would, after all, be a rank-4 tensor - it could be used, for instance, to bound the error arising from a second-order Taylor expansion of irradiance. Jarosz et al. [JSKJ12] performed a cursory evaluation of using a second-order Taylor expansion for approximating the irradiance in a scene and found that it held promise, but was hard to control - a good error bound becomes much more important than for a first-order expansion because the rate of change of the interpolation can vary much more wildly for the second-order case.

Bibliography

[App68] Arthur Appel. Some techniques for shading machine renderings of solids. In *Proceedings of the April 30–May 2, 1968, spring joint computer conference*, AFIPS '68 (Spring), pages 37–45, New York, NY, USA, 1968. ACM.

[Arv94] James Arvo. The irradiance jacobian for partially occluded polyhedral sources. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, SIGGRAPH '94, pages 343–350, New York, NY, USA, 1994. ACM.

[CG85] Michael F. Cohen and Donald P. Greenberg. The hemi-cube: a radiosity solution for complex environments. *SIGGRAPH Comput. Graph.*, 19(3):31–40, July 1985.

[Coo86] Robert L. Cook. Stochastic sampling in computer graphics. *ACM Trans. Graph.*, 5(1):51–72, January 1986.

[CPC84] Robert L. Cook, Thomas Porter, and Loren Carpenter. Distributed ray tracing. *SIGGRAPH Comput. Graph.*, 18(3):137–145, January 1984.

[CSSD96] Per H. Christensen, Eric J. Stollnitz, David H. Salesin, and Tony D. DeRose. Global illumination of glossy environments using wavelets and importance. *ACM Trans. Graph.*, 15(1):37–71, January 1996.

[GTGB84] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. Modeling the interaction of light between diffuse surfaces. *SIGGRAPH Comput. Graph.*, 18(3):213–222, January 1984.

[HMS09] Robert Herzog, Karol Myszkowski, and Hans-Peter Seidel. Anisotropic radiance-cache splatting for efficiently computing high-quality global illumination with lightcuts. In Marc Stamminger and Philip Dutré, editors, *Computer Graphics Forum (Proc. Eurographics)*, volume 28(2), pages 259–268, München, Germany, 2009. Wiley-Blackwell.

[Hol96] Nicolas Holzschuch. *Le Contrôle de l'Erreur dans la Méthode de Radiosité Hiérarchique*. PhD thesis, Université Joseph Fourier (Grenoble I), Mar 1996.

[HS98] Nicolas Holzschuch and François X. Sillion. An exhaustive error-bounding algorithm for hierarchical radiosity. *Computer Graphics Forum*, 17(4):197–218, 1998.

[HSA91] Pat Hanrahan, David Salzman, and Larry Aupperle. A rapid hierarchical radiosity algorithm. *SIGGRAPH Comput. Graph.*, 25(4):197–206, July 1991.

[ICG86] David S. Immel, Michael F. Cohen, and Donald P. Greenberg. A radiosity method for non-diffuse environments. *SIGGRAPH Comput. Graph.*, 20(4):133–142, August 1986.

[Jar08] Wojciech Jarosz. *Efficient Monte Carlo Methods for Light Transport in Scattering Media*. PhD thesis, UC San Diego, September 2008.

[JSKJ12] Wojciech Jarosz, Volker Schönefeld, Leif Kobbelt, and Henrik Wann Jensen. Theory, analysis and applications of 2D global illumination. *ACM Transactions on Graphics (to be Presented at ACM SIGGRAPH 2012)*, 31, 2012. accepted for publication.

[Kaj86] James T. Kajiya. The rendering equation. In *Computer Graphics (Proceedings of SIGGRAPH 86)*, pages 143–150, August 1986.

[KBPZ06] Jaroslav Křivánek, Kadi Bouatouch, Sumanta Pattanaik, and Jirí Zára. Making radiance and irradiance caching practical: Adaptive caching and neighbor clamping. In *Rendering Techniques 2006 (Proceedings of the Eurographics Symposium on Rendering)*, pages 127–138, June 2006.

[KG09] Jaroslav Křivánek and Pascal Gautron. *Practical Global Illumination with Irradiance Caching.* Synthesis lectures in computer graphics and animation. Morgan & Claypool, 2009.

[KGBP05] Jaroslav Křivánek, Pascal Gautron, Kadi Bouatouch, and Sumanta Pattanaik. Improved radiance gradient computation. In *SCCG '05: Proceedings of the 21th spring conference on Computer graphics*, pages 155–159, New York, NY, USA, 2005. ACM Press.

[MU49] Nicholas Metropolis and Stanislaw Ulam. The monte carlo method. *Journal of the American Statistical Association*, 44(247):335–341, 1949.

[PH10] Matt Pharr and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation.* Morgan Kaufmann, 2nd edition, 2010.

[SAS92] Brian E. Smits, James R. Arvo, and David H. Salesin. An importance-driven radiosity algorithm. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '92, pages 273–282, New York, NY, USA, 1992. ACM.

[SAWG91] Françis X. Sillion, James R. Arvo, Stephen H. Westin, and Donald P. Greenberg. A global illumination solution for general reflectance distributions. In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '91, pages 187–196, New York, NY, USA, 1991. ACM.

[SC97] Peter Shirley and Kenneth Chiu. A low distortion map between disk and square. *Journal of Graphics, GPU, and Game tools*, 2(3):45–52, 1997.

[Shi91] Peter Shirley. Discrepancy as a quality measure for sample distributions. In *In Eurographics '91*, pages 183–194. Elsevier Science Publishers, 1991.

[TL04] Eric Tabellion and Arnauld Lamorlette. An approximate global illumination system for computer generated films. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2004)*, 23(3):469–476, August 2004.

[VG94] Eric Veach and Leonidas Guibas. Bidirectional estimators for light transport. In *Photorealistic Rendering Techniques (Proceedings of the Eurographics Workshop on Rendering)*, pages 147–162, June 1994.

[VG95] Eric Veach and Leonidas J. Guibas. Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '95, pages 419–428, New York, NY, USA, 1995. ACM.

[WH92] Gregory J. Ward and Paul S. Heckbert. Irradiance gradients, 1992.

[Whi80] Turner Whitted. An improved illumination model for shaded display. *Commun. ACM*, 23(6):343–349, June 1980.

[WRC88] Gregory J. Ward, Francis M. Rubinstein, and Robert D. Clear. A ray tracing solution for diffuse interreflection. *SIGGRAPH Comput. Graph.*, 22:85–92, June 1988.