Theory, Analysis and Applications of 2D Global Illumination

WOJCIECH JAROSZ
Disney Research Zürich and University of California, San Diego,
VOLKER SCHÖNEFELD
Limbic Software, RWTH Aachen University, and University of California, San Diego,
LEIF KOBBELT
RWTH Aachen University,
and
HENRIK WANN JENSEN
University of California, San Diego

We investigate global illumination in 2D and show how this simplified problem domain leads to practical insights for 3D rendering.

We first derive a full theory of 2D light transport by introducing 2D analogues to radiometric quantities such as flux and radiance, and deriving a 2D rendering equation. We use our theory to show how to implement algorithms such as Monte Carlo ray tracing, path tracing, irradiance caching, and photon mapping in 2D, and demonstrate that these algorithms can be analyzed more easily in this domain while still providing insights for 3D rendering.

We apply our theory to develop several practical improvements to the irradiance caching algorithm. We perform a full second-order analysis of diffuse indirect illumination, first in 2D, and then in 3D by deriving the irradiance Hessian, and show how this leads to increased accuracy and performance for irradiance caching. We propose second-order Taylor expansion from cache points, which results in more accurate irradiance reconstruction. We also introduce a novel error metric to guide cache point placement by analyzing the error produced by irradiance caching. Our error metric naturally supports anisotropic reconstruction and, in our preliminary study, resulted in an order of magnitude less error than the "split-sphere" heuristic when using the same number of cache points.

Categories and Subject Descriptors: I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism—*Color, shading, shadowing, and texture*; I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism—*Raytracing*

General Terms: Theory, Algorithms, Performance

Additional Key Words and Phrases: global illumination, rendering, gradient, irradiance caching, irradiance gradients, Hessian, Taylor series

1. INTRODUCTION

Simulating the propagation of light within a virtual scene is a challenging problem. This transport is described by the *rendering equation* [Kajiya 1986], which states that the lighting arriving at any point on a surface in a scene depends on all other points in the scene. This definition results in an infinitely recursive and infinitely dimensional equation for computing global illumination.

While there are many approaches for solving the rendering equation numerically, each one has strengths and weaknesses. Unfortunately, theoretical and empirical analysis of these approaches can be very difficult, due to the mathematical and computational complexity inherent with the rendering equation in general and with any method in particular. Furthermore, this combined complexity makes it difficult for people to visualize, reason about, and grasp the underlying concepts that are present in the equations — making it more difficult to discover high-level structures in the system and to exploit them in solutions. In this article we present a theoretical framework to make this analysis and development more feasible.

A common recipe for dealing with such highly complex problems is to formulate a similar, but simplified problem which is easier to analyze, perform the necessary derivations or analysis, and then try to adapt the results to the original problem. We take this approach. We derive a full theory of light transport in 2D, examine this simplified formulation, and apply the resulting analysis to various practical situations in 3D light transport algorithms. A major benefit of this approach is that while the key problem remains intact, the underlying concepts and mathematical equations are greatly simplified; thus, they are easier to analyze and can be explored more efficiently.

The use of two-dimensional light transport is not new. Simplifying light transport to two-dimensions has been used by several previous researchers [Heckbert 1992; Durand et al. 2005; Ramamoorthi et al. 2007] to allow for easier discussion and derivation, before applying these concepts to their higher-dimensional counterparts. Typically these analyses have been restricted to a specific subset of light transport. In this paper we take this concept further and examine global illumination by formulating the full rendering equation in two-dimensions.

1.1 Contributions

The primary contribution of this article is a detailed model of two-dimensional light transport. This model offers a powerful **theoretical framework** which allows for simpler investigation of complex global illumination concepts. This form of conceptual "dimensionality reduction" provides a number of tangible benefits over dealing with the full rendering equation.

Firstly, **visualizing** various global illumination concepts becomes much easier. For instance, in two-dimensional light transport, the irradiance falling on an object is a 1D function which can be visualized directly as a 1D plot on each surface. This type of visualization

DOI 10.1145/——-.——-
http://doi.acm.org/10.1145/——-.——-

becomes increasingly difficult in 3D. Moreover, in a 2D world, any higher-order derivative of the irradiance is a 1D function as well, and hence can be visualized in the same manner.

As an added benefit, **computation time** is decreased drastically when dealing with 2D light transport and its derivative, which makes **prototyping** and **experimenting** easier. Our framework highlights the fact that when moving from 3D to 2D, radiance reduces from a 5D function to just a 3D function. This makes brute-force reference solutions, which are often tedious or impractical to compute in the full 3D global illumination context, feasible to compute during the course of experimentation.

Deriving complex global illumination concepts is much easier with two-dimensional light transport due to the simplified expressions involved. We show how these concepts are simpler to derive in the low-dimensional setting and show how to subsequently generalize them to the full-dimensional problem. As a concrete example, we derive higher-order derivatives of global illumination in 2D, then generalize these second-order derivatives to 3D **irradiance Hessians**. We demonstrate how these Hessians can be applied to the irradiance caching algorithm for improved results. This preliminary investigation using our 2D to 3D framework is our other major contribution.

Lastly, the two-dimensional global illumination framework described in this paper also has potential applications in **teaching**. Learning and understanding the principles of global illumination can be a difficult task for students. While the key ideas, such as shooting rays, can usually be understood quickly, many of the concepts can initially seem highly abstract and unintuitive. In the two-dimensional setting, many of these concepts are simplified, allowing for easier explanation, and more direct illustration, which makes subsequently learning the three-dimensional counterparts easier. A simple example of this is the solid angle, which turns into a plane angle – a concept students are already familiar with. Missing this understanding can lead to a lack of higher-level intuition about the concepts, and even worse, can sometimes lead to false intuitions.

2. PREVIOUS WORK

Several previous researchers have considered the simplified 2D domain for some subset of light transport. Our work also extends existing work in global illumination and illumination gradients.

2.1 2D Light Transport

One of the first detailed descriptions of a 2D world in modern literature can be found in a 1884 novella by Abbott, *Flatland: A romance of many dimensions* [Abbott 1884]. This text coined the term *flatland*, which was later adopted by researchers to describe a 2D world. However, Abbott's fictional world is not physically based and can be considered *embedded* in a 3D world.

Edelsbrunner et al. [1983] were among the first graphics researchers to utilize a hypothetical 2D world to analyze and solve 3D problems. This geometric investigation of hidden surface removal was later revised and extended by Pocchiola [1990]. Heckbert [1992] was the first of a number of researchers [Gortler et al. 1993; Orti et al. 1996; Durand et al. 1996] to analyze the radiosity algorithm in 2D. He derived a specialized 2D version of radiosity and used his analysis to derive a new and improved 3D variant.

In more recent work, Durand et al. [2005] used 2D analysis to develop a model of light transport in frequency space. They model each operation of light transport, including emission, transmission and reflection, as operators on the radiance field. These operators can then be expressed and efficiently approximated in frequency space. After the 2D analysis, they applied the technique to the 3D light transport problem.

Recently, Ramamoorthi et al. [2007] published a first-order analysis of direct lighting and derived a net visibility gradient for scenes with distant environment lighting. Again, they first performed a 2D analysis to gain valuable insights and then applied these to the full 3D problem. We derive a general framework for 2D global illumination and use this to perform a *second*-order analysis of lighting, accounting for indirect illumination and local light sources.

Although some work has been done in the field of 2D light transport, a formal introduction and proper definition of the terms is missing. We provide a *theoretical framework* for 2D light transport, which we derive from the ground-up using a 2D analogy to radiometry. We furthermore build on this foundation to obtain a 2D counterpart to the *rendering equation* [1986], which allows us to easily formulate 2D variants of existing 3D physically-based rendering algorithms (such as Monte Carlo ray tracing, path tracing [Kajiya 1986], photon mapping [Jensen 2001], and irradiance caching [Ward et al. 1988]). We believe our framework could be useful for analysing other complex global illumination algorithms, such as Metropolis light transport [Veach and Guibas 1997], in a simplified setting.

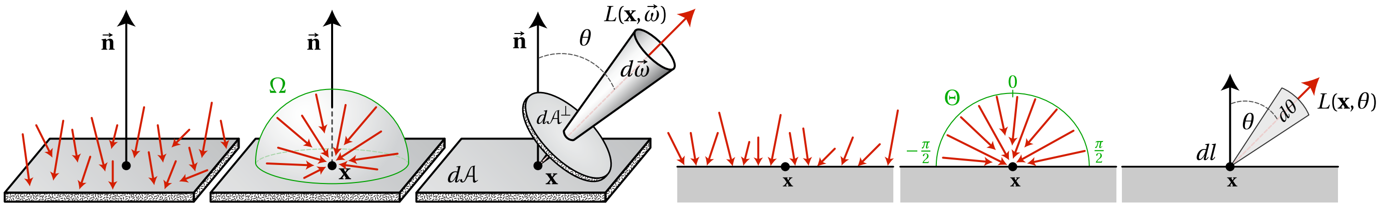
2.2 Illumination Gradients and Error Control

As an application to our 2D model, we demonstrate how to extend the irradiance caching algorithm [Ward et al. 1988; Ward and Heckbert 1992] to higher order derivatives.

Computing accurate illumination gradients is useful in a number of computer graphics problems. Arvo [1994] computed the irradiance Jacobian due to a polyhedral light source and used the results in applications ranging from computing isolux contours to finding extrema of the irradiance field. Holzschuch et al. [1995; 1996; 1998] derived a form factor gradient and Hessian for error estimation in the radiosity algorithm.

Ward et al. [1988] introduced *irradiance caching*, which accelerates the computation of diffuse indirect illumination through sparse sampling and interpolation. To determine the sample density they derived a loose upper-bound on interpolation error by considering a first-order Taylor expansion of irradiance within a hypothetical "split-sphere" environment. Ward and Heckbert [1992] later derived a method to accurately estimate the true gradient of the indirect irradiance within Lambertian scenes and used this to obtain higher-order interpolation between cache points, but still used the split-sphere heuristic to determine sample density. Tabellion and Lamorlette [2004] applied irradiance caching in movie production and introduced minor modifications to the split-sphere model. Annen et al. [2004] used spherical harmonics to cache the spherical radiance distribution at sparse locations in the scene, and derived a gradient of the spherical harmonic coefficients to facilitate higher quality interpolation. Křivánek et al. [2005] used this same concept to generalize the irradiance caching algorithm to glossy surfaces and subsequently generalized Ward and Heckbert's irradiance gradient formulation to a radiance gradient which accounts for occlusions [Křivánek et al. 2005]. Jarosz et al. [2008; 2008] generalized the irradiance gradient computation to consider the effects of participating media.

Nearly all of these methods use the split-sphere heuristic in largely unmodified form to determine the placement of cache points over the scene. This is a critical component of the algorithm which significantly influences efficiency and quality, but unfortunately suffers from a number of problems in practice. Most implementations impose additional constraints (such as minimum and maximum radii in world-space and screen space [Larson and Shakespeare



(a) 3D Flux (b) 3D Irradiance (c) 3D Radiance (d) 2D Flux (e) 2D Irradiance (f) 2D Radiance

Fig. 1: Comparison of radiometric quantities in 3D (left) and 2D (right). Flux measures the amount of light that hits a finite surface area (3D) or arc-length (2D) from all directions, irradiance integrates the light arriving at a single point over the whole hemisphere (3D) or hemicircle (2D), and radiance expresses the amount of light arriving at or leaving a single point from a differential solid (3D) or plane angle (2D).

1998]) which must be carefully tuned to obtain satisfactory results. Křivánek et al. [2006] addressed some of these practical considerations by iteratively enforcing the triangle inequality on the radii to mitigate the deficiencies in the original split-sphere approach. In Sections 6 and 8 we take the first significant departure from this line of work by leveraging a second-order analysis of indirect illumination. We use our derivations to obtain an entirely new error metric for irradiance caching which reduces error by more than an order of magnitude (without relying excessively on additional constraints such as radius clamping), and which naturally supports anisotropic cache points. We also show how to use our analysis to perform second-order extrapolation in irradiance caching. Similar investigations have been performed for error estimation and error control in radiosity [Shirley 1991; Arvo et al. 1994; Holzschuch and Sillion 1998].

2.3 Overview

In the remainder of this paper we first define our 2D world and derive a theory of light transport in 2D (Section 3). This builds upon 2D counterparts to common radiometric terms and results in the derivation of a 2D reflection and rendering equation. Based on this framework, we discuss in Section 4 how to implement a variety of global illumination algorithms in 2D and show examples of how this simplified domain allows for more intuitive analysis. In Section 5 we perform a first- and second-order analysis of global illumination in 2D. As a concrete use-case, we discuss several ways to apply these derivatives to irradiance caching in 2D (Section 6). In Section 7 we extend our second-order analysis of global illumination to 3D and demonstrate (Section 7) how this can be used to improve irradiance caching in a number of practical ways, including higher-order extrapolation, an improved error metric, and anisotropic reconstruction.

3. A THEORY OF 2D LIGHT TRANSPORT

To accurately model the physical properties of the world to closely match their 3D counterparts, we need to first define the 2D world. All objects in the 2D world are at most two dimensional shapes with 1D boundaries which we will call *curves*. Curves are the 2D equivalent of surfaces. Consequently, whereas 3D surfaces have *surface area*, the curve is measured in *arc length*.

In previous work there are two fundamentally different approaches to define the 2D world:

—**The Embedded Model** describes, similarly to Abbot's fictional *Flatland*, a planar slice embedded in a 3D world.

—**The Intrinsic Model** describes the 2D world as a self-contained domain that is not embedded in a higher-dimensional space.

The embedded model can be regarded as a straightforward restriction of the 3D model to 2D. This requires only trivial modifications to existing algorithms and facilitates simpler visualization (considering only a particular 2D slice of the full 3D problem). Unfortunately, this model does not necessarily reduce the complexity of the underlying rendering algorithm (e.g. light transport still occurs in a full 3D scene, but is simply visualized in a 2D slice). A more pressing issue is that a simple 2D slice of a 3D physical world does not result in an entirely consistent 2D formulation. For light transport, one concern is what happens to light that leaves the planar slice. Is it allowed to leave? If so, this violates the conservation of energy principle of the original 3D model, invalidating intuitive properties of many familiar radiometric concepts.

In the intrinsic model, on the other hand, all light is generated, reflected and absorbed inside the world so energy is conserved. This results in a more self-consistent theory and simplified problem complexity. For these reasons, we use the intrinsic model throughout this article.

Assumptions. To develop rendering algorithms and an analysis framework, we need to first precisely define the behavior of light in a 2D world. We assume geometric optics and a model of light analogous to that described by Dutré et al. [2006]. We assume that light travels in straight lines within a vacuum, ignoring participating media.

The short wavelength limit of Maxwell's equations, resulting in the Eikonal equation, is the foundation of geometric optics. The most fundamental treatment of 2D radiometric quantities might therefore start directly from these equations. We however, take a more pragmatic approach common in computer graphics. For simplicity, we use the didactic principle of "counting" or measuring photons to build up radiometric definitions and proceed partially by analogy to familiar 3D counterparts. Though this is far from a rigorous derivation, it is sufficient for most applications in computer graphics. By doing this, we presuppose that the two-dimensional theory behaves, apart from geometric modifications, analogously to the three-dimensional one. This approach is beneficial in our context, but may not be the only (or best) approach in other application domains. More rigorous investigations may be of interest when e.g. wave effects need to be considered.

In the remainder of this section we will derive the fundamental radiometric quantities of the 2D world and compare them to their standard 3D counterparts. In doing so, it is useful to take note of the *units* of the corresponding quantities as well as the *dimensionality* of the functions they define. We illustrate the most important quantities in Figure 1 and aggregate all the quantities in Table I.

Table I. : Radiometric quantities in 2D and 3D

		3D		2D	
Quantity	**Symbol**	**Expression**	**Units**	**Expression**	**Units**
Spectral Radiant Energy	Q_λ	$Q_\lambda = n_\lambda \frac{hc}{\lambda}$	J	$Q_\lambda = n_\lambda \frac{hc}{\lambda}$	J
Radiant Energy	Q	$Q = \int_0^\infty Q_\lambda \, d\lambda$	J	$Q = \int_0^\infty Q_\lambda \, d\lambda$	J
Flux	Φ	$\Phi_{3D}(A) = \frac{dQ(A)}{dt}$	$\mathrm{W} = \mathrm{J} \cdot \mathrm{s}^{-1}$	$\Phi_{2D}(\mathfrak{L}) = \frac{dQ(\mathfrak{L})}{dt}$	$\mathrm{W} = \mathrm{J} \cdot \mathrm{s}^{-1}$
Irradiance	E	$E_{3D}(\mathbf{x}) = \frac{d\Phi_{3D}(A)}{da(\mathbf{x})}$	$\mathrm{W} \cdot \mathrm{m}^{-2}$	$E_{2D}(\mathbf{x}) = \frac{d\Phi_{2D}(\mathfrak{L})}{dl(\mathbf{x})}$	$\mathrm{W} \cdot \mathrm{m}^{-1}$
Radiosity	B	$B_{3D}(\mathbf{x}) = \frac{d\Phi_{3D}(A)}{da(\mathbf{x})}$	$\mathrm{W} \cdot \mathrm{m}^{-2}$	$B_{2D}(\mathbf{x}) = \frac{d\Phi_{2D}(\mathfrak{L})}{dl(\mathbf{x})}$	$\mathrm{W} \cdot \mathrm{m}^{-1}$
Radiance	L	$L_{3D}(\mathbf{x}, \vec{\omega}) = \frac{d^2\Phi_{3D}(A)}{(\vec{\mathbf{n}} \cdot \vec{\omega})^+ \, d\vec{\omega} \, da(\mathbf{x})}$	$\mathrm{W} \cdot \mathrm{m}^{-2} \cdot \mathrm{sr}^{-1}$	$L_{2D}(\mathbf{x}, \theta) = \frac{d^2\Phi_{2D}(\mathfrak{L})}{\cos\theta \, d\theta \, dl(\mathbf{x})}$	$\mathrm{W} \cdot \mathrm{m}^{-1} \cdot \mathrm{rad}^{-1}$

3.1 2D Radiometry

The Photon. We assume the basic building block of lighting is the photon. Each photon has a particular wavelength, λ, and a corresponding electromagnetic energy:

$$e_\lambda = \frac{hc}{\lambda}, \tag{1}$$

where $h \approx 6.626068 \times 10^{-34}$ J · s is Planck's constant, and c is the speed of light. We assume the speed of light in 2D is the same as in a 3D world ($c = 299,792,458$ m/s in a vacuum); hence, the energy of a single photon is expressed in units of joules [J] in either domain.

Radiometry is the field that studies the measurement of electromagnetic radiation. Conceptually, all the remaining radiometric quantities are derived off of the basic principle of "counting" or measuring photons.

Spectral Radiant Energy. A collection of n_λ photons of the same wavelength λ carry a combined *spectral radiant energy* which is the sum of their individual energies:

$$Q_\lambda = n_\lambda \, e_\lambda = n_\lambda \, \frac{hc}{\lambda}. \tag{2}$$

Spectral radiant energy always refers to the energy of a particular wavelength of light.

Radiant Energy. The radiometric quantities we deal with in graphics are spectral in nature; however, for convenience, we often opt to drop the dependence on wavelength for simplicity. To accomplish this, we can introduce the *radiant energy*, which is the total energy associated with a collection of photons of all wavelengths. This is computed by integrating the spectral radiant energy over all possible wavelengths:

$$Q = \int_0^\infty Q_\lambda \, d\lambda. \tag{3}$$

We base the remaining radiometric quantities off of radiant energy. Strictly speaking, Equation (3) changes the radiometric units, but in graphics we typically drop the dependence on wavelength as a notational convenience and ignore this technicality. Alternatively, each of the subsequent quantities can be interpreted as being independently defined for each wavelength of light.

Radiant Power/Flux. *Radiant power* or *flux* is the differential radiant energy per unit time:

$$\Phi = \frac{dQ}{dt}. \tag{4}$$

In terms of units, flux is identical in the 2D and 3D domains and is expressed in watts $[\mathrm{W} = \mathrm{J} \cdot \mathrm{s}^{-1}]$. However, the concept of flux has a slightly different *meaning* in these two domains.

In 3D, flux is the time rate of energy flowing from/into/through some *surface* A:

$$\Phi_{3D}(A) = \frac{dQ(A)}{dt}. \tag{5}$$

Effectively, it is a measure of the number of photons arriving at or leaving a surface per unit time (see Figure 1a). For example, we can refer to the total power incident on a wall or table, or that an area light source emits 50 watts of radiant power. Hence, to talk about flux in 3D we always need to refer to a surface. This means that, in 3D, flux is a function defined over surfaces, A.

In 2D, flux is also a measure of the number of photons flowing from/into/through an object, but since objects in 2D are defined using curves, flux is defined over some *curve* $\mathfrak{L}$ (see Figure 1d):

$$\Phi_{2D}(\mathfrak{L}) = \frac{dQ(\mathfrak{L})}{dt}. \tag{6}$$

Hence, in a 2D environment we can refer to the total flux incident upon a particular curve, or the total flux incident upon all curves in a scene. This subtle distinction implies that the *dimensionality* of these two *functions* is different, even though the units of the *value* of the functions are the same.

Irradiance and Radiosity. In 3D, the irradiance is the incident flux density per unit *surface area*:

$$E_{3D}(\mathbf{x}) = \frac{d\Phi_{3D}(A)}{da(\mathbf{x})}. \tag{7}$$

This is a measure of the number of photons that arrive at a differential surface area, per unit time (see Figure 1b), and has units of $[\mathrm{W} \cdot \mathrm{m}^{-2}]$.

The corresponding quantity in a 2D world is the incident flux density per unit *arc length*. Irradiance in 2D measures the flux *arriving* at a differential length dl around a point $\mathbf{x}$ on a curve $\mathfrak{L}$:

$$E_{2D}(\mathbf{x}) = \frac{d\Phi_{2D}(\mathfrak{L})}{dl(\mathbf{x})}. \tag{8}$$

Therefore, it is measured in watts per unit arc length $[\mathrm{W}\cdot\mathrm{m}^{-1}]$. Conceptually, irradiance in 2D is a measure of the number of photons arriving at a infinitesimal arc length on a curve, per unit time (see Figure 1e).

Due to the spatial dependence of the differential in the definition of irradiance (*da* or *dl*), it has different *units* in the 2D and 3D domains. This change in units is a subtle difference, but has a large effect on algorithms such as density estimation in photon mapping.

Irradiance is a measure of incident light. The corresponding exitant measure of flux density is called the *radiant exitance*, or *radiosity*, *B*. In 2D these express the flux *leaving* a differential area *dl* around a point **x** on a curve. They have identical units as irradiance, but simply refer to the energy *leaving* instead of *arriving* at a point.

Radiance. In 3D, radiance represents the differential flux at a point **x** on a surface, per differential solid angle $d\vec{\omega}$, per differential area perpendicular to direction $\vec{\omega}$, $da^{\perp}$:

$$L_{3D}(\mathbf{x},\vec{\omega}) = \frac{d^2\Phi_{3D}(A)}{d\vec{\omega}\, da^{\perp}(\mathbf{x},\vec{\omega})}. \tag{9}$$

Radiance in a 3D scene has units of $[\mathrm{W}\cdot\mathrm{sr}^{-1}\cdot\mathrm{m}^{-2}]$ and measures the number of photons hitting/leaving a differential area on a surface along a differential cone of directions perpendicular to that surface, per unit time (see Figure 1c).

The radiometric counterpart for 2D scenes represents the differential flux at a point **x** on a curve, per differential angle $d\theta$, per differential arc length perpendicular to θ, $dl^{\perp}$. It has units of watts per meter per radian $[\mathrm{W}\cdot\mathrm{rad}^{-1}\cdot\mathrm{m}^{-1}]$ and can be expressed as:

$$L_{2D}(\mathbf{x},\theta) = \frac{d^2\Phi_{2D}(\mathfrak{L})}{d\theta\, dl^{\perp}(\mathbf{x},\theta)}. \tag{10}$$

Radiance in 2D measures the number of photons that arrive/leave from a differential set of directions $d\theta$ onto a hypothetical differential arc-length perpendicular to θ (see Figure 1f).

Foreshortening. In both 2D and 3D, radiance measures the photons that arrive onto a *hypothetical* arc length or surface area *perpendicular* to a chosen direction. In practice, however, we are often interested in measuring photons hitting an *actual* curve or surface. To measure radiance on an arbitrary curve or surface from an arbitrary direction, we need to consider the spreading of light when measuring at different incident angles. This is accomplished by introducing a foreshortening term, which involves a cosine in both 2D and 3D:

$$da^{\perp}(\mathbf{x},\vec{\omega}) = (\vec{\omega}\cdot\vec{\mathbf{n}})^{+}\, da(\mathbf{x}), \tag{11}$$

$$dl^{\perp}(\mathbf{x},\theta) = \cos\theta\, dl(\mathbf{x}). \tag{12}$$

In 3D, light will spread proportional to the cosine of the angle between the incident direction and the surface normal. We express this using the standard dot product clamped to the upper hemisphere, which we denote $(\vec{\omega}\cdot\vec{\mathbf{n}})^{+}$. In 2D, the spreading is also proportional to the cosine to the surface normal. In our notation we define $\theta \in [-\pi/2, \pi/2]$, so $\theta = 0$ perpendicular to the local curve. Hence, we do not need an explicit concept of a surface normal, and can express foreshortening simply as $\cos\theta$. Inserting Equations (12) and (11) into Equations (10) and (9) results in the complete definition of radiance at an actual curve or surface.

Radiometric Integral Relationships. Radiance is arguably the most important radiometric quantity for rendering. It is most directly related to the observed color of an object and is the quantity that needs to be computed for each pixel in a rendered image in 3D. More importantly, radiance is the quantity propagated along rays. Since it is so instrumental in rendering, it is useful to express the other radiometric quantities in terms of radiance.

Equations (9) and (10) express radiance in terms of flux, but it is also possible to invert this relationship by integrating both sides, arriving at the following integral identities:

$$\Phi_{3D}(A) = \int_A \int_\Omega L_{3D}(\mathbf{x},\vec{\omega})\,(\vec{\mathbf{n}}\cdot\vec{\omega})^{+}\, d\vec{\omega}\, da(\mathbf{x}), \tag{13}$$

$$\Phi_{2D}(\mathfrak{L}) = \int_{\mathfrak{L}} \int_\Theta L_{2D}(\mathbf{x},\theta)\,\cos\theta\, d\vec{\omega}\, dl(\mathbf{x}), \tag{14}$$

where $\mathfrak{L}$ denotes the domain of arc lengths, Ω denotes the hemisphere of directions, and Θ denotes the hemicircle of angles.

We can similarly express irradiance in terms of radiance:

$$E_{3D}(\mathbf{x}) = \int_\Omega L_{3D}(\mathbf{x}\leftarrow\vec{\omega})\,(\vec{\mathbf{n}}\cdot\vec{\omega})^{+}\, d\vec{\omega}, \tag{15}$$

$$E_{2D}(\mathbf{x}) = \int_\Theta L_{2D}(\mathbf{x}\leftarrow\theta)\,\cos\theta\, d\theta, \tag{16}$$

where the arrow indicates the radiance arrives at **x**. Radiosity can be defined analogously using exitant radiance.

Discussion. It is useful to discuss the differences between radiance in 2D and 3D. Firstly, radiance has different physical units in these two domains. This small difference in units has large implications. For instance, teaching students about 2D radiance is simpler, because it does not involve the concept of solid angles, and a plane angle can be visualized more easily. However, examining Equations (9) and (10) reveals another, more important difference. Radiance in 2D is only a **three-dimensional function** (two for position **x** and one for angle θ) instead of a **five-dimensional function** for the 3D world (three for position **x** and two for direction $\vec{\omega}$). By reducing the domain dimensionality by one, we reduce the parameter space of the radiance function by two dimensions. This distinction has a large practical implication as it makes computing, visualizing, and analyzing radiance in a 2D world much more feasible, both computationally and conceptually. In certain cases, this dimensionality reduction can even lead to closed-form analytical solutions, like form factor computation, which are much more difficult to express and compute in the 3D world.

3.2 Light Interaction

In the following section we will explore the interaction of light with surfaces in 3D and the corresponding concept in a 2D world.

The BRDF. When light encounters objects in the scene, it interacts with the surfaces by being scattered or absorbed. If we make the simplifying assumption that photons striking an object will scatter at the same location, then the interaction between the light and a surface (in 3D) or a curve (in 2D) can be described using a function called the *bidirectional reflectance distribution function*. The BRDF expresses the relationship between differential irradiance and differential reflected radiance at a point **x**. In 3D and 2D this is expressed as:

$$f_{3D}(\mathbf{x},\vec{\omega}\to\vec{\omega}') \equiv \frac{dL^r_{3D}(\mathbf{x}\to\vec{\omega}')}{dE_{3D}(\mathbf{x}\leftarrow\vec{\omega})} = \frac{dL^r_{3D}(\mathbf{x}\to\vec{\omega}')}{L_{3D}(\mathbf{x}\leftarrow\vec{\omega})\,(\vec{\mathbf{n}}\cdot\vec{\omega})^{+}\, d\vec{\omega}}, \tag{17}$$

$$f_{2D}(\mathbf{x},\theta\to\theta') \equiv \frac{dL^r_{2D}(\mathbf{x}\to\theta')}{dE_{2D}(\mathbf{x}\leftarrow\theta)} = \frac{dL^r_{2D}(\mathbf{x}\to\theta')}{L_{2D}(\mathbf{x}\leftarrow\theta)\,\cos\theta\, d\theta}. \tag{18}$$

The last step results from substituting Equations (15) and (16) into the denominators.

Using this analogous definition for the BRDF within the intrinsic model, the formulation is significantly simplified while, at the same time, key properties such as reciprocity, range, energy conservation and the relation between incident and reflected radiance are maintained and closely match those of the 3D model:

(1) **Domain.** For a particular point $\mathbf{x}$ in 3D, f_{3D} is a four-dimensional function (two to specify the incoming direction, and two to specify the outgoing direction). In the 2D world, f_{2D} is a two-dimensional function (one dimension each for the incoming and outgoing angle). Furthermore, if the BRDF is allowed to vary spatially over objects, this leads to an additional two dimensions in the 3D world and an additional one dimension in the 2D world. Hence, in the most general form, f_{3D} is six-dimensional, and f_{2D} is only three-dimensional.

(2) **Range.** The BRDF can take on any non-negative value in both the 2D and 3D worlds.

(3) **Reciprocity.** The value of the BRDF remains unchanged if the incident and outgoing directions are swapped. Mathematically, Helmholtz's law of reciprocity states that:

$$f_{3D}(\mathbf{x},\vec{\omega}'\to\vec{\omega}) = f_{3D}(\mathbf{x},\vec{\omega}\to\vec{\omega}'), \tag{19}$$

$$f_{2D}(\mathbf{x},\theta'\to\theta) = f_{2D}(\mathbf{x},\theta\to\theta'). \tag{20}$$

Because of this property we denote the BRDF as $f_{3D}(\mathbf{x},\vec{\omega}'\leftrightarrow\vec{\omega})$ and $f_{2D}(\mathbf{x},\theta'\leftrightarrow\theta)$ to indicate that the directions can be interchanged.

From a practical point of view, reciprocity means that reflection is invariant to the direction of light flow, i.e., the reflected radiance remains unchanged if the light and observer are swapped. This property is essential for reproducing many global illumination algorithms in 2D since it allows light to be traced either in the forward or backward direction.

(4) **Energy conservation and normalization.** Due to energy conservation, a point cannot reflect more light than it receives. This can be expressed as the following constraint:

$$\int_{\Omega} f_{3D}(\mathbf{x},\vec{\omega}\leftrightarrow\vec{\omega}')\,(\vec{\mathbf{n}}\cdot\vec{\omega})^{+}\,d\vec{\omega} \le 1, \quad \forall\vec{\omega}', \tag{21}$$

$$\int_{\Theta} f_{2D}(\mathbf{x},\theta\leftrightarrow\theta')\cos\theta\,d\theta \le 1, \quad \forall\theta'. \tag{22}$$

Though these constraints are seemingly identical, the integral of the cosine term in 2D and 3D is different. This means that the normalization constants in commonly used BRDFs will change. For example, the 3D diffuse BRDF is $\frac{\rho}{\pi}$ while the 2D counterpart is $\frac{\rho}{2}$ where ρ is the albedo, or fraction of reflected light.

(5) **Specular interactions.** If the surface is completely smooth then the BRDF contains a singularity and the scattering of light is defined by the law of reflection, refraction, and the Fresnel equations. Since in 3D these equations already operate within the plane defined by the incident direction and surface normal, they remain unchanged in the 2D world.

(6) **Relationship between incident and reflected radiance.** The information in the BRDF can be used to derive the relationship between incident and reflected radiance. By multiplying both sides of Equations (17) and (18) by the denominators and integrating over all directions or angles we can derive an expression for computing the reflected radiance at a point $\mathbf{x}$:

$$L^r_{3D}(\mathbf{x}\to\vec{\omega}') = \int_{\Omega} f_{3D}(\mathbf{x},\vec{\omega}\leftrightarrow\vec{\omega}')\,L_{3D}(\mathbf{x}\leftarrow\vec{\omega})\,(\vec{\mathbf{n}}\cdot\vec{\omega})^{+}\,d\vec{\omega}, \tag{23}$$

$$L^r_{2D}(\mathbf{x}\to\theta') = \int_{\Theta} f_{2D}(\mathbf{x},\theta\leftrightarrow\theta')\,L_{2D}(\mathbf{x}\leftarrow\theta)\,\cos\theta\,d\theta. \tag{24}$$

In essence, the reflected radiance off of a surface or curve can be computed by integrating all the incident radiance arriving over the hemisphere of directions or hemicircle of angles. This expression describes the local behavior of light as it interacts with surfaces and is know as the *local illumination* model.

The Rendering Equation. Whereas the BRDF describes the local illumination model, for global illumination the key integral identity is the rendering equation. It describes how incident and emitted radiance are combined into exitant radiance:

$$\underbrace{L_{2D}(\mathbf{x}\to\theta')}_{\text{outgoing}} = \underbrace{L^e_{2D}(\mathbf{x}\to\theta')}_{\text{emitted}} + \underbrace{L^r_{2D}(\mathbf{x}\to\theta')}_{\text{reflected}}. \tag{25}$$

The rendering equation is the equation that is solved, or approximated, by all global illumination algorithms.

There are two commonly used and equivalent formulations of the reflected radiance term L^r_{2D} that we have adopted in the 2D model:

- **The Hemicircular Formulation** integrates the incident radiance over the hemicircle above the point and is given in Equation (24).
- **The Arc-Length Formulation** demonstrates the global nature of the rendering equation. Instead of integrating over the hemicircle of directions, we can instead integrate over every point $\mathbf{y}$ on all curves $\mathfrak{L}$ in the scene. To do this, we need the geometric coupling term:

$$G_{2D}(\mathbf{x}\leftrightarrow\mathbf{y}) = \cos\theta_{\mathbf{x}}\frac{d\theta(\mathbf{x})}{dl(\mathbf{y})} = \frac{\cos\theta_{\mathbf{x}}\cos\theta_{\mathbf{y}}}{\|\mathbf{x}-\mathbf{y}\|}, \tag{26}$$

which converts differential angle at $\mathbf{x}$, $d\theta(\mathbf{x})$, to differential arc length at $\mathbf{y}$, $dl(\mathbf{y})$. By plugging into the hemicircular formulation above, we arrive at:

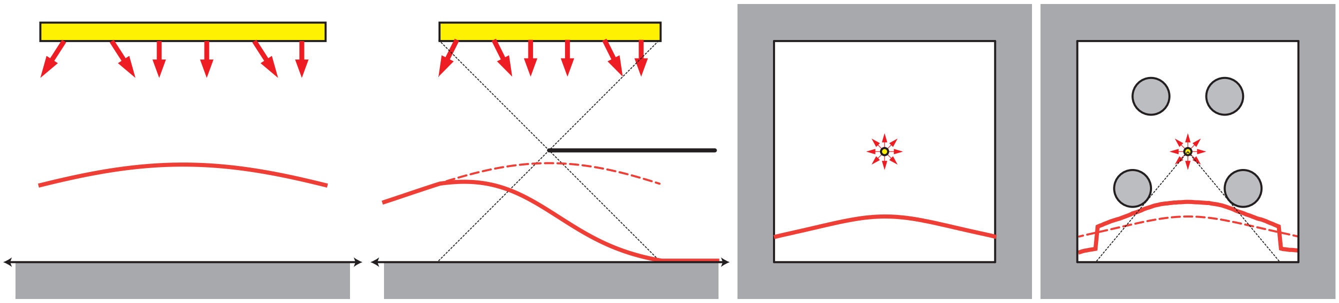
$$L^r_{2D}(\mathbf{x}\to\mathbf{e}) = \int_{\mathfrak{L}} f_{2D}(\mathbf{x},\mathbf{y}\leftrightarrow\mathbf{e})\,L_{2D}(\mathbf{x}\leftarrow\mathbf{y}) \\ V_{2D}(\mathbf{x}\leftrightarrow\mathbf{y})\,G_{2D}(\mathbf{x}\leftrightarrow\mathbf{y})\,dl(\mathbf{y}), \tag{27}$$

where we use the binary visibility function $V_{2D}(\mathbf{x}\leftrightarrow\mathbf{y})$ to exclude points $\mathbf{y}$ which are not directly visible from $\mathbf{x}$.

Discussion. An important difference between the rendering equation in 2D and 3D is that the geometry term in Equation (26) only involves inverse falloff, instead of the usual inverse-squared falloff in 3D. This implies that light has a different falloff profile in an intrinsic 2D world.

4. 2D RENDERING ALGORITHMS

The rendering equation is very costly to compute, and even in two dimensions it is far too complex to solve analytically in the general case. Many algorithms have been proposed in the literature to numerically approximate the rendering equation. In this section we review a small set of commonly used algorithms and explain the necessary changes to apply them to the 2D global illumination problem. We also demonstrate that well-known properties of these 3D algorithms can be easily visualized and analyzed in the simplified 2D domain.



(A) Direct (B) Direct + Occlusion (C) Indirect (D) Indirect + Occlusion

Fig. 2: Our 2D test scenes. We show the ground-truth irradiance in solid red and the un-occluded irradiance with the dotted line.

In 3D, the classical quantity of interest for rendering is a viewport on the image plane of a virtual camera, the 2D equivalent of which is a 1D image line segment. However, in 2D it becomes possible to directly visualize the structure of the *light field* across all objects in the scene. We do this by evaluating the radiance or irradiance field for every point on a curve.

4.1 Test Scenes

In order to evaluate and compare different rendering algorithms (and to perform our second-order analysis of global illumination in Section 5) we developed a few test scenes, each constructed to highlight specific lighting scenarios. We illustrate these test scenes, and plot the irradiance in Figure 2.

Scene (A) features direct lighting without any indirect lighting and occlusion. We have a diffuse line receiver at the bottom and a line light at the top emitting uniform radiance. In this simple scene, we can compute an analytic solution for the irradiance (as well as any order derivative) at any point on the receiving line.

Scene (B) highlights the effect of occlusion on the irradiance field. Compared to (A), an occluder is added, blocking part of the line light with respect to the receiving line. We can compute irradiance as well as all derivatives analytically for this scene as well.

Scene (C) involves direct and indirect lighting in a scene without occlusions. Light is emitted by a point light in the center and is reflected by the closed square "room" with diffuse albedo of $\rho = 0.2$. This scene has the interesting property of including indirect illumination, but no occlusions, resulting in a completely smooth irradiance field which can be differentiated many times. We compute reference solutions for this scene numerically.

Scene (D) includes indirect illumination and occlusions. The only difference from (C) is the addition of four circular occluders, which introduce discontinuities in the irradiance derivatives.

We use these four scenes to analyze a number of standard rendering algorithms which we adapt to the 2D domain. In all our results we plot the ground truth irradiance as red, and the first and second derivatives in green and blue. We typically visualize values computed using one of the rendering algorithms in black.

4.2 Stochastic Ray Tracing

The idea of stochastic ray tracing, as summarized by Dutré et al. [2006], is to use a Monte Carlo technique to evaluate the rendering equation. To compute the reflected radiance at a point $\mathbf{x}$ into direction ψ, the hemicircle above $\mathbf{x}$ is sampled by casting rays and recursively tracing them into the scene. The recursion is terminated by a criterion such as maximum recursion depth, or Russian roulette [Arvo and Kirk 1990]. Mathematically speaking, by applying Monte Carlo integration to Equation 25 we get:

$$L_{2D}(\mathbf{x}\to\psi) \approx L^e_{2D}(\mathbf{x}\to\psi) + \frac{1}{N}\sum_{i=1}^{N}\frac{f_{2D}(\mathbf{x},\theta_i\leftrightarrow\psi)\,L_{2D}(\mathbf{x}\leftarrow\theta_i)\cos\theta_i}{pdf(\theta_i)} \qquad (28)$$

where N is the number of samples and $pdf(\theta_i)$ is the probability density (PDF) of choosing sample θ_i. The PDF can be chosen similarly to the 3D PDFs, although care has to be taken to properly normalize it with respect to the change of domain. For example, the 3D uniform PDF over Ω is $\frac{1}{2\pi}$ whereas the 2D uniform PDF over Θ is $\frac{1}{\pi}$.

4.3 Path Tracing

Path tracing was introduced by Kajiya [1986] as a general solution to the rendering equation. Although it converges slowly, due to its simplicity and robustness it is a very widespread algorithm, especially for ground truth comparisons.

In the stochastic ray tracing algorithm the hemicircular integral is recursively evaluated, leading to a tree-like structure of ray tracing steps. The key idea in path tracing is to choose only one random sample on the hemicircle, effectively generating a path instead of a tree. To reduce variance, a number of these paths are evaluated for each pixel (or each primary shading location on a curve). Thus the radiance at $\mathbf{x}$ can be approximated by

$$L_{2D}(\mathbf{x}\to\mathbf{y}) \approx L^e_{2D}(\mathbf{x}\to\mathbf{y}) + \frac{1}{N}\sum_{i=1}^{N}\frac{L^i_{2D}}{pdf_i}, \qquad (29)$$

with

$$L^i_{2D} = \left(\prod_{j=1}^{M^i-1} f_r(\mathbf{x}^i_j,\mathbf{x}^i_{j+1}\leftrightarrow\mathbf{x}^i_{j-1})\cos\theta^i_j\right) L^e_{2D}(\mathbf{x}^i_{M^i}\to\mathbf{x}^i_{M^i-1}),$$

where N is the number of paths, M^i is the length of path i, $\mathbf{x}^i_j$ are the vertices of path i, with $\mathbf{x}^i_1 = \mathbf{x}$ and $\mathbf{x}^i_0 = \mathbf{y}$, θ^i_j is the angle between the normal at path vertex $\mathbf{x}^i_j$ and the incident direction $\mathbf{x}^i_{j+1} - \mathbf{x}^i_j$, and pdf_i is the probability density of choosing path i.

Path tracing avoids the exponential growth of naïve recursive Monte Carlo ray tracing and, when combined with an unbiased termination criterion such as Russian roulette, delivers an unbiased solution of the rendering equation.

In Figures 3 we show the results of using 2D path tracing to render test scenes A-D, demonstrating that the characteristics of the original 3D algorithm translate exactly to the 2D domain. Although path tracing is unbiased (the expected value is the correct solution)

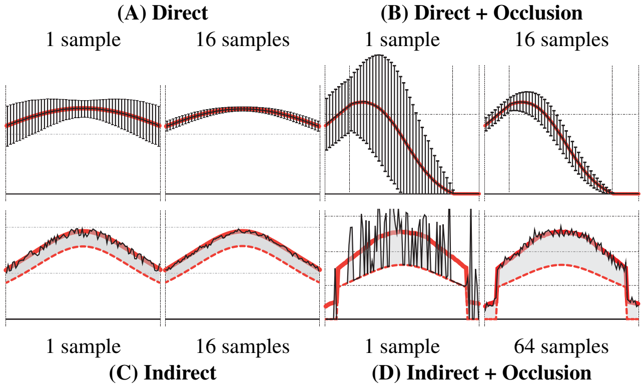


Fig. 3: Analysis of path tracing on the four test scenes from Figure 2. For the direct illumination scenes (top), we show the mean and standard deviation of path tracing in black. In the indirect illumination scenes (bottom), the dotted red line is the analytic direct illumination and the solid red line is the irradiance after adding the indirect component. We show the value computed using path tracing in black. The 2D algorithm retains the same characteristics as the original 3D algorithm, while being significantly more efficient. The expected value matches the analytic solution, and increasing the number of paths rapidly reduces the variance.

it suffers from noise. The variance diminishes, however, as the number of paths is increased, providing an increasingly accurate approximation to the ground truth.

4.4 Photon Mapping

Jensen [2001] introduced the two-pass photon mapping algorithm to accelerate the computation of global illumination, especially with regard to caustics. In this algorithm the first pass is a lighting simulation where virtual "photons" which carry packets of flux are emitted from light sources and cached in a global data structure called a photon map at scattering interactions with surfaces. This is followed by a rendering pass, which uses a form of Monte Carlo ray tracing, but where recursive evaluation of radiance is truncated by approximating indirect lighting using photon density estimation.

Special care must be taken to apply photon mapping to the 2D domain. Since photons are packets of flux, in 2D, computing the photon density gives us the flux per arc-length. Usually the set P of the k-nearest photons is used and the distance to the k^{th} photon determines the radius r of the density estimate. In order to estimate the radiance from this set, we consider the BRDF and normalize the total flux of all photons in P by the arc-length of the search radius. By assuming that the curve is locally flat, projecting the radius onto the curve results in an arc-length of $2r$. We can estimate the reflected radiance from the photon map as

$$L^r_{2D}(\mathbf{x}\to\theta) \approx \sum_{i=1}^{N} f_{2D}(\mathbf{x},\theta\leftrightarrow\theta_i)\frac{\Delta\Phi_i(\mathbf{x}\leftarrow\theta_i)}{2r} \qquad (30)$$

where $\Delta\Phi_i$ is the flux of photon i and θ_i is the incident angle of the photon.

In Figures 4, 5 and 6 we show the results of applying photon mapping to scenes A-D. The results again confirm that the overall properties of photon mapping (such as bias and noise) are preserved when moving to the 2D domain, allowing for analysis in a simplified setting. In Figure 4 (left) we show the characteristic over-estimation of radiance when using k-nearest neighbor density estimation as compared to a fixed radius (right). Figure 6 visualizes the overestimation of shadowed regions due to the well-known boundary bias. In Figure 5 we show how the low-frequency noise and bias introduced by photon mapping (left) is reduced by using final gathering with stochastic ray tracing (right). Our 2D global illumination framework facilitates detecting, visualizing and analyzing various sources of bias much more easily.

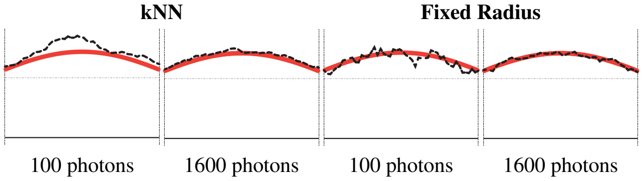


Fig. 4: The average of 100 independent runs of photon mapping (black) using kNN (left) and fixed-radius (right) density estimation applied to scene A from Figure 2. kNN density estimation with a constant kernel overestimates the true values (red), while using fixed-width density estimation reduces this problem. Both approaches are consistent and converge to the correct solution in the limit.

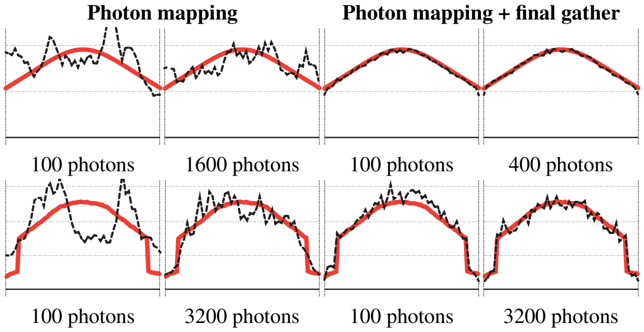


Fig. 5: Direct visualization of the photon map (left) and final gathering of the photon map using stochastic ray tracing (right) for scenes C and D in Figure 2. We visualize photon mapping in black and the ground truth in red. Final gathering reduces the low frequency noise introduced by photon mapping.

4.5 Irradiance Caching

Irradiance caching is a popular technique to accelerate the computation of indirect lighting on diffuse surfaces. It is often used as a final gather [Reichert 1992] pass in conjunction with another technique, such as photon mapping, to obtain high quality radiance estimates.

Though we are typically interested in computing radiance, in Lambertian scenes the BRDF is constant with respect to the incident

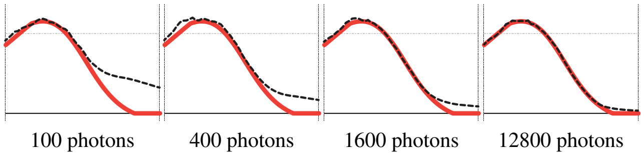


Fig. 6: The average of 100 independent runs of photon mapping (black) in scene B compared to ground truth (red). The shadowed region is overestimated due to the well-known boundary bias, which diminishes with more photons.

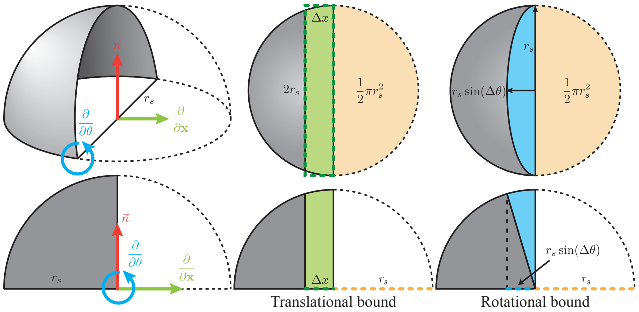


Fig. 7: The hypothetical "split-sphere" (top) and "split-circle" (bottom) environments for determining the valid radius of cache points in irradiance caching.

and reflected angles, so we can transform Equation (24) into

$$L^r_{2D}(\mathbf{x} \to \psi) = f_{2D}(\mathbf{x}) \int_\Theta L_{2D}(\mathbf{x} \leftarrow \theta) \cos\theta \, d\theta \tag{31}$$

$$= f_{2D}(\mathbf{x}) E_{2D}(\mathbf{x}). \tag{32}$$

This allows us to use the irradiance instead of the radiance to determine the reflected radiance.

Ward et al. [1988] observed that computing irradiance using Monte Carlo ray tracing is extremely costly, but the resulting illumination often varies slowly across surfaces. Irradiance caching exploits this property by computing irradiance sparsely across the scene and interpolating and extrapolating these values whenever possible.

To improve the quality of extrapolation, Ward and Heckbert [1992] proposed to compute and cache not just the irradiance E, but also its gradient $\nabla_\mathbf{x} E$, to perform a first-order Taylor expansion of around each cache point. The irradiance at a location $\mathbf{x}$ is approximated as:

$$E(\mathbf{x}) \approx \frac{\sum_{i \in S} w_i(\mathbf{x})(E_i + \nabla_\mathbf{x} E_i \Delta \mathbf{x}_i)}{\sum w_i(\mathbf{x})}, \tag{33}$$

where S is the set of cache points contributing to $\mathbf{x}$ and E_i and $\nabla_\mathbf{x} E_i$ are the irradiance and irradiance gradient of cache point i.

Ward et al. [1988] derived a loose upper bound on the expected change in irradiance using an imaginary "split-sphere" in 3D environments (Figure 7 top). They used the resulting "error" function to i) determine the maximum distance to extrapolate each cache point (determining the set S above), and used the reciprocal of this error function to ii) determine the weight w_i of nearby cache points for interpolation.

To apply irradiance caching in 2D, we consider an imaginary "split-circle" environment around the shading location (see Figure 7 bottom). The error function expresses the expected change in irradiance as we translate away from the cache point, Figure 7 (center), and as we rotate away from the surface normal at the cache point, (right). This can be expressed mathematically as:

$$\varepsilon_i(\mathbf{x}, \theta) = \frac{|\mathbf{x} - \mathbf{x}_i|}{r_i} + \sin(|\theta - \theta_i|). \tag{34}$$

where ε_i, $\mathbf{x}_i$, and θ_i are the error function, position, and surface normal of cache point i respectively and r_i is the radius of the imaginary "split-circle" environment. This radius is usually chosen to be either the minimum [Tabellion and Lamorlette 2004] or harmonic-mean [Ward et al. 1988] length of all rays cast over the hemicircle. The weighting function is based on the inverse of the error: $w_i(\mathbf{x}) = \varepsilon_i^{-1}(\mathbf{x})$.

Ward et al. [1988] derived this heuristic to obtain a conservative upper-bound on the error and it has been used (largely unmodified) in virtually all subsequent irradiance caching approaches [Ward and Heckbert 1992; Tabellion and Lamorlette 2004; Křivánek et al. 2005; Křivánek et al. 2005]. Unfortunately, this heuristic can lead to suboptimal cache placement even in trivial lighting scenarios, which leads to increased render times. In the next section we perform a second-order analysis of global illumination and then apply the results to derive a novel error metric which solves these problems.

5. SECOND-ORDER ANALYSIS OF 2D GLOBAL ILLUMINATION

In this section we derive 2D irradiance gradients and then extend this to irradiance Hessians to perform a second-order analysis of indirect illumination. Performing this analysis in 2D has two major advantages. Firstly, since a curve is a one dimensional subspace, all derivatives of the irradiance on the curve are scalar-valued functions. For 3D surfaces the first derivative is a two-dimensional gradient vector and the second-derivative results in a 2×2 Hessian matrix. This is a significant simplification that allows us to more easily investigate the effects and applicability of higher order derivatives. Secondly, because of the reduced dimensionality of the problem, we can compute numerical reference values in 2D that are sufficiently precise to allow comparison and verification. This is extremely difficult in 3D, as unbiased results are usually very noisy and performing finite differences on these values further exaggerates this noise.

We split the irradiance gradient into two parts, a rotational part and a translation part. Since it is more complex, we focus on the translational gradient in this paper. A similar analysis could be performed for the simpler case of the rotational gradient.

5.1 2D Irradiance Gradients

There are two popular methods for computing gradients of indirect irradiance in the literature. When applied to 2D, the first one is based on the arc-length formulation of the rendering equation [Křivánek et al. 2005; Annen et al. 2004]. The second one is based on Ward et al.'s [1992] original gradient formulation which has since been generalized [Křivánek et al. 2005; Jarosz et al. 2008]. We provide new derivations of these two approaches which improves upon previous results and we show how to extend the analysis to compute second-order derivatives. Without loss of generality, we assume a local coordinate frame, i.e. $\mathbf{x}$ is at the origin and the tangent coincides with the x-axis.

Arc-Length Formulation. The arc-length gradient formulation is based on applying the gradient operator directly to the arc-length form of the irradiance function. Omitting obvious parameters and subscripts and applying the product rule yields:

$$\nabla_\mathbf{x} E_{2D}(\mathbf{x}) = \nabla_\mathbf{x} \int_\mathfrak{L} L_{2D}(\mathbf{x} \leftarrow \mathbf{y}) \, V_{2D}(\mathbf{x}, \mathbf{y}) \, G_{2D}(\mathbf{x}, \mathbf{y}) \, dl(\mathbf{y}) \tag{35}$$

$$= \int_\mathfrak{L} \nabla_\mathbf{x} L V G + L \nabla_\mathbf{x} V G + L V \nabla_\mathbf{x} G \, dl(\mathbf{y}). \tag{36}$$

A similar decomposition was used previously [Jarosz et al. 2008; Jarosz et al. 2008].

In the most general case, we could solve for each term to obtain a full irradiance gradient. However, if we restrict ourselves to diffuse scenes, the radiance at $\mathbf{y}$, $L_{2D}(\mathbf{x} \leftarrow \mathbf{y})$, does not change as $\mathbf{x}$ is translated. Hence, the gradient of the incident radiance is

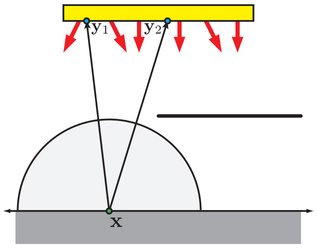


Fig. 8: The arc-length formulation of the translational gradient. If the scene is assumed to be Lambertian, after translation (right) the new projected radiance value is only changed due to the geometric relationship between **x** and **y**. This formulation does not account for visibility, so it fails when occluders are present.

$\nabla_{\mathbf{x}} L_{2D}(\mathbf{x} \leftarrow \mathbf{y}) = 0$. Furthermore, if we make the assumption (discussed in detail later) that the visibility is constant with respect to translation, i.e. $\nabla_{\mathbf{x}} V_{2D}(\mathbf{x},\mathbf{y}) = 0$, this simplifies to:

$$\nabla_{\mathbf{x}} E_{2D}(\mathbf{x}) = \int_{\mathfrak{L}} L(\mathbf{x} \leftarrow \mathbf{y})\, V_{2D}(\mathbf{x},\mathbf{y})\, \nabla_{\mathbf{x}} G_{2D}(\mathbf{x},\mathbf{y})\; dl(\mathbf{y}). \tag{37}$$

This depends on the gradient of the geometry term from Equation (26), which we show in Appendix A is:

$$\nabla_{\mathbf{x}} G_{2D} = 3\frac{\cos\theta_{\mathbf{x}}\cos\theta_{\mathbf{y}}}{r^3}\mathbf{y} - \frac{\cos\theta_{\mathbf{y}}}{r^2}\vec{\mathbf{n}}_{\mathbf{x}} + \frac{\cos\theta_{\mathbf{x}}}{r^2}\vec{\mathbf{n}}_{\mathbf{y}}. \tag{38}$$

Here $\vec{\mathbf{n}}_{\mathbf{x}}$ and $\vec{\mathbf{n}}_{\mathbf{y}}$ refer to the unit surface normal vectors at **x** and **y** respectively, $r = \|\mathbf{x}-\mathbf{y}\|$, and we assume without loss of generality that **x** is the origin. This formulation computes a 2D gradient vector. To get the tangential component, we can project the gradient onto the normalized tangent vector, or transform the input into tangent space and only compute the first component of the gradient. We omit these details here for simplicity.

We can evaluate the irradiance gradient numerically by approximating the integral using Monte Carlo integration:

$$\nabla_{\mathbf{x}} E_{2D}(\mathbf{x}) \approx \frac{1}{N}\sum_{i=1}^{N} \frac{L(\mathbf{x} \leftarrow \mathbf{y}_i)\, V_{2D}(\mathbf{x},\mathbf{y}_i)\, \nabla_{\mathbf{x}} G_{2D}(\mathbf{x},\mathbf{y}_i)}{pdf(\mathbf{y}_i)}. \tag{39}$$

We frequently use the hemispherical formulation of the rendering equation. In this case, the above formula needs to be converted by performing the differential change from Equation (26).

Figure 8 illustrates this formulation graphically. We started out with the arc-length formulation of the irradiance, which integrates over all curves. Upon translation of the reference point **x** we keep the points **y** on all other curves fixed, and predict how the geometric relationship will change.

An important limitation of this formulation is that changes in the binary visibility function are completely ignored. While this assumption allows us to greatly simplify the equation, it also introduces an error in the presence of occlusion. We illustrate this effect in Figure 8 (right) and discuss the resulting problems further in Section 6.1.

Stratified Formulation. Ward and Heckbert [1992] formulated the gradient directly over the stratified hemispherical sampling of the environment. Working directly on the stratified integration domain allowed them to consider the relationship between neighboring strata to account for occlusion changes. To achieve this in 2D, we discretize the hemicircle into arc sectors. These strata can be regarded as projections of the environment. When **x** is translated, each stratum will slide around on the hemicircle due to the change in projection. By considering how the boundary between two strata moves with respect to occlusions, we can approximate the effect of the visibility term on the irradiance gradient. This is also highly related to Arvo's [1994] derivation of an irradiance Jacobian by considering the movement of intrinsic and apparent polygon vertices projected onto the hemisphere. We illustrate this in Figure 9.

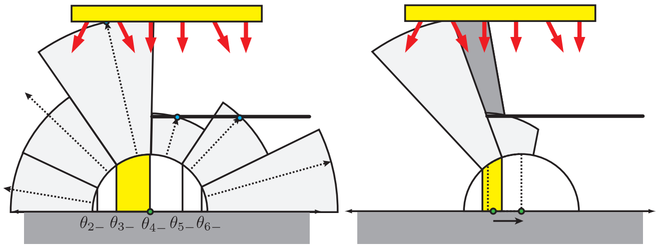


Fig. 9: The stratified formulation of the gradient tracks the boundaries between neighboring strata and can account for occlusion changes.

More precisely, the hemicircular formulation of the rendering equation is discretized into N sectors $\Theta_i = [\theta_{i_-} \ldots \theta_{i_+}]$ where $\bigcup_i \Theta_i = \Theta$:

$$E_{2D}(\mathbf{x}) = \int_{\Theta} L_{2D}(\mathbf{x} \leftarrow \theta)\, \cos\theta\; d\theta \tag{40}$$

$$= \sum_{i=1}^{N} \int_{\Theta_i} L_{2D}(\mathbf{x} \leftarrow \theta)\, \cos\theta\; d\theta. \tag{41}$$

Note that we do not impose any other restrictions on the sectors. This is more general than previous methods [Ward and Heckbert 1992; Křivánek et al. 2005], which derived the gradient only for specific probability distributions and induced stratification. We will see in Section 7.1 how this benefit extends to 3D.

For every stratum we pick a random sample in the domain of the stratum and cast a ray into the corresponding direction. This gives us an estimate of the incident radiance and hence an approximation of the mean radiance of the whole stratum. Assuming that the radiance is constant across the stratum, we can approximate the previous equation by pulling the radiance outside the integral

$$E_{2D}(\mathbf{x}) \approx \sum_{i=1}^{N} L_{2D}(\mathbf{x} \leftarrow \theta_i) \int_{\Theta_i} \cos\theta\; d\theta. \tag{42}$$

We can now solve the integral on the right side analytically to obtain

$$E_{2D}(\mathbf{x}) \approx \sum_{i=1}^{N} L_{2D}(\mathbf{x} \leftarrow \theta_i)\, \left(\sin\theta_{i_+} - \sin\theta_{i_-}\right), \tag{43}$$

where θ_{i_+} and θ_{i_-} are the boundaries of stratum Θ_i.

In other words, we are assuming the whole stratum has constant radiance, and we multiply it by the projected length to get the irradiance due to this stratum. If we define $L_i = L_{2D}(\mathbf{x} \leftarrow \theta_i)$ and assume that radiance is zero on the bottom hemicircle ($L_i = 0$ for $i < 1$ and $i > N$) we can rearrange the previous sum to

$$E_{2D}(\mathbf{x}) \approx \sum_{i=1}^{N+1} (L_{i-1} - L_i)\, \sin\theta_{i_-}, \tag{44}$$

where we use the fact that $\theta_{i_+} = \theta_{(i+1)_-}$. We can think of this as a summation of the differences in radiance across neighboring strata.

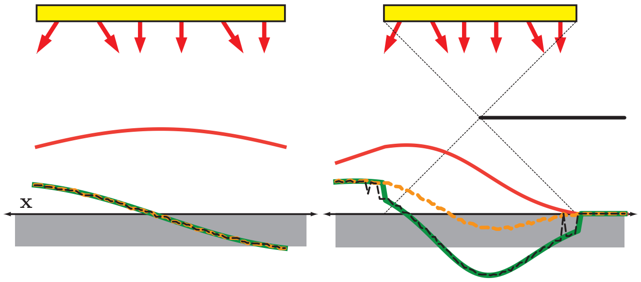


Fig. 10: We visualize the analytic irradiance (red) and gradient (green) and compare the gradient to the numerically computed result using 100 samples per point with the arc-length (orange) and stratified (black) formulations. The arc-length gradient ignores occlusion changes and therefore fails in the penumbra region (right).

Now we can apply the gradient operator, and by again assuming diffuse reflections we get

$$\nabla_{\mathbf{x}} E_{2D}(\mathbf{x}) \approx \sum_{i=1}^{N+1} (L_{i-1} - L_i)\, \nabla_{\mathbf{x}} \sin\theta_{i\_} \tag{45}$$

$$= \sum_{i=1}^{N+1} (L_i - L_{i-1})\, \frac{\cos^2\theta_{i\_}}{r_{i-}}. \tag{46}$$

Here r_{i-} is the distance of the point at the border between the two strata, corresponding to $\theta_{i\_}$. Ward and Heckbert [1992] noticed that because the stratum with the smaller r is closer to $\mathbf{x}$, it moves faster upon translation and occludes the other stratum, and hence defines the speed of the border. To leverage this fact we define $r_{i-} = \min(r_i, r_{i-1})$ and then get

$$\nabla_{\mathbf{x}} E_{2D}(\mathbf{x}) \approx \sum_{i=2}^{N} (L_i - L_{i-1})\, \frac{\cos^2\theta_{i\_}}{\min(r_i, r_{i-1})}. \tag{47}$$

Note that we have shrunk the domain of the summation, as the cosine will vanish for the first and last boundaries where $\theta_{i\_} = \frac{\pi}{2}$ and $\theta_{i\_} = -\frac{\pi}{2}$.

5.2 2D Irradiance Hessians

Arc-Length Formulation. We can easily differentiate the arc-length gradient in Equation (37) one more time to obtain the second derivative (which we denote $\mathbf{H}_{\mathbf{x}}$):

$$\mathbf{H}_{\mathbf{x}}(E_{2D}) \approx \int_{\mathfrak{L}} L(\mathbf{x} \leftarrow \mathbf{y})\, V_{2D}(\mathbf{x},\mathbf{y})\, \mathbf{H}_{\mathbf{x}}(G_{2D}(\mathbf{x},\mathbf{y}))\, dl(\mathbf{y}). \tag{48}$$

This reduces to computing the second derivative of the geometry term from Equation (26), which we show in Appendix B is:

$$\begin{aligned}\mathbf{H}_{\mathbf{x}}(G_{2D}(\mathbf{x},\mathbf{y})) = {} & 15\frac{\cos\theta_{\mathbf{x}}\cos\theta_{\mathbf{y}}}{r^5}(\mathbf{y}\mathbf{y}^{\mathsf{T}}) - 3\frac{\cos\theta_{\mathbf{x}}\cos\theta_{\mathbf{y}}}{r^3}\mathbf{I}_2 \\ & - \frac{1}{r^3}\left(\vec{\mathbf{n}}_{\mathbf{x}}\vec{\mathbf{n}}_{\mathbf{y}}^{\mathsf{T}} + \vec{\mathbf{n}}_{\mathbf{y}}\vec{\mathbf{n}}_{\mathbf{x}}^{\mathsf{T}}\right) - 3\frac{\cos\theta_{\mathbf{y}}}{r^4}\left(\vec{\mathbf{n}}_{\mathbf{x}}\mathbf{y}^{\mathsf{T}} + \mathbf{y}\vec{\mathbf{n}}_{\mathbf{x}}^{\mathsf{T}}\right) \\ & + 3\frac{\cos\theta_{\mathbf{x}}}{r^4}\left(\vec{\mathbf{n}}_{\mathbf{y}}\mathbf{y}^{\mathsf{T}} + \mathbf{y}\vec{\mathbf{n}}_{\mathbf{y}}^{\mathsf{T}}\right),\end{aligned} \tag{49}$$

where $\mathbf{I}_2$ is the 2×2 identity matrix, and terms of the form $\mathbf{y}\mathbf{y}^{\mathsf{T}}$ compute the outer product matrix. Again, this equation computes a 2×2 Hessian matrix which we can easily project to obtain the tangential second derivative.

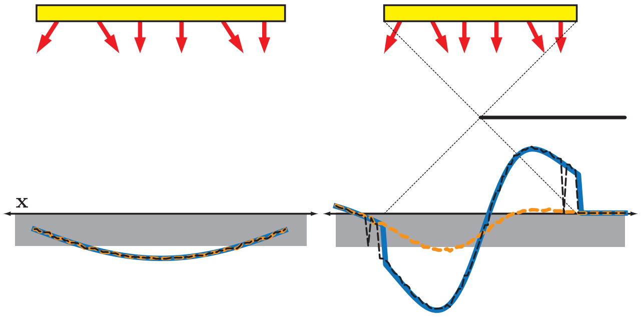


Fig. 11: We visualize the ground-truth irradiance Hessian (blue) and compare to the numerically computed result using 100 samples per point with the arc-length (orange) and stratified (black) Hessian formulations. The arc-length Hessian works well in the absence of occlusion, but fails in the penumbra region.

Stratified Formulation. To obtain the second derivative of the irradiance while taking into account occlusions, we consider the stratified gradient formulation in Equation (45). Obtaining the Hessian simply involves differentiating the $\sin\theta_{i\_}$ term one more time:

$$\mathbf{H}_{\mathbf{x}}(E_{2D})(\mathbf{x}) \approx \sum_{i=1}^{N+1} (L_i - L_{i-1})\, \mathbf{H}_{\mathbf{x}}(\sin\theta_{i\_}) \tag{50}$$

$$= \sum_{i=2}^{N} (L_i - L_{i-1})\, 3\frac{\cos\theta_{i\_}^2 \sin\theta_{i\_}}{\min(r_i^2, r_{i-1}^2)}. \tag{51}$$

6. ANALYSIS AND APPLICATIONS IN 2D

We now analyze the different formulations and discuss some practical applications of our second-order derivations.

6.1 Gradient Formulation Comparison

In Figure 10 we compare the results of the arc-length and stratified gradient formulations on scenes A and B from Figure 2. In the absence of occlusion (scene A, left), all assumptions we made for the arc-length gradient are met, since the visibility function is constant. We can see that both gradient formulations perform well and generate consistent results. We also investigate the effect of occlusion on the gradients, as depicted in Figure 10 (scene B, right). Here the two formulations produce matching results in the fully-lit or fully occluded regions, but, as expected, in the penumbra the arc-length formulation fails to capture the occlusion effects. As a result, the gradient magnitude is underestimated where occlusions happen. The stratified gradient formulation, on the other hand, performs very well and resembles the analytic solution closely. The large spikes near the left and right boundary of the penumbra are due to under-sampling the very small regions between the light and occluder as seen from the ground plane.

6.2 Hessian Formulation Comparison

In Figure 11 we perform a similar comparison for the Hessian. We plot the second derivative of irradiance for scenes A and B and confirm that the omission of occlusion similarly affects the irradiance Hessian.

6.3 Second-Order Extrapolation

Given the irradiance, as well as its first- and second-order derivatives, we can easily extend irradiance caching to perform a second-order

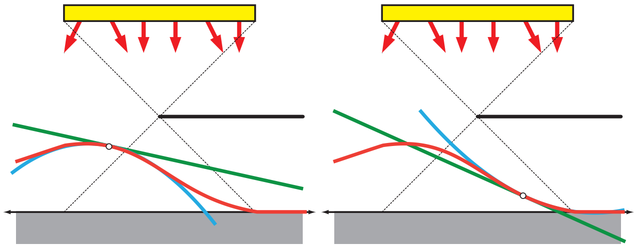


Fig. 12: Performing a first-order (green) and second-order (blue) Taylor extrapolation of the irradiance (red) around two cache points in scene A from Figure 2. The second-order expansion matches the irradiance more faithfully, but the maximum extrapolation distance needs to be set carefully to avoid over-shooting the irradiance.

Taylor expansion around cache points. This results in a simple generalization of Equation (33):

$$E(\mathbf{x}) \approx \frac{\sum_{i\in S} w_i(\mathbf{x})\left(E_i + \nabla_{\mathbf{x}} E_i \Delta\mathbf{x}_i + \frac{1}{2}\Delta\mathbf{x}_i \mathbf{H}_{\mathbf{x}}(E_i)\Delta\mathbf{x}_i\right)}{\sum_{i\in S} w_i(\mathbf{x})}. \tag{52}$$

In Figure 12 we compare the results of extrapolating two cache points using first-order and second-order Taylor expansion. The first-order approach (green) is used by the standard irradiance gradients technique and extrapolates cached values along a line. A second-order expansion (blue) approximates the irradiance locally as a parabola and more closely matches the ground truth solution. This approach shows promise, however, the figure also illustrates that the maximum radius needs to be chosen carefully to avoid over-shooting the ground-truth solution. In Section 8.1 we extend this technique to 3D.

6.4 Hessian Error-Metric for Irradiance Caching

We can also use our second-order analysis to derive a more principled error metric for irradiance caching. Our goal is to use the largest radius for each cache point such that the resulting error is bounded by some user-specified threshold. We define the **total error** ε^t of a cache point as the integrated absolute difference between the correct irradiance and the extrapolated irradiance[1]:

$$\varepsilon^t = \int_{-R_i}^{R_i} |E(\mathbf{x}_i + x) - E'(\mathbf{x}_i + x)|\, dx, \tag{53}$$

where $E'(\mathbf{x}_i + x) = E_i(\mathbf{x}_i) + \nabla_{\mathbf{x}} E_i(\mathbf{x}_i)\, x$ is the first-order Taylor expansion of the irradiance.

In the ideal setting, we should solve for the largest R_i such that the above equation is satisfied where E is the ground-truth radiance. This, however, requires knowledge of the true irradiance in the vicinity of the cache point, which is impractical. Instead, we can use the second-order Taylor expansion of the irradiance as an oracle for the ground-truth: $E(\mathbf{x}_i + x) \approx E_i(\mathbf{x}_i) + \nabla_{\mathbf{x}} E_i(\mathbf{x}_i)\, x + \frac{1}{2} x \mathbf{H}_{\mathbf{x}}(E_i(\mathbf{x}_i))\, x$. We illustrate this in Figure 13. The total error then simplifies to an expression which we can easily integrate:

$$\varepsilon^t \approx \hat{\varepsilon}^t = \frac{1}{2}\int_{-R_i}^{R_i} |x \mathbf{H}_{\mathbf{x}}(E_i)\, x|\, dx = \frac{1}{3} |h_{\mathbf{x}}(E_i)|\, R_i^3, \tag{54}$$

where we express the absolute value of the tangential second derivative as $|h_{\mathbf{x}}(E_i)|$.

Inverting this equation allows us to determine the maximum cache point radius R_i which induces an approximate total error $\hat{\varepsilon}^t$:

$$R_i = \sqrt[3]{\frac{3\hat{\varepsilon}^t}{|h_{\mathbf{x}}(E_i)|}}. \tag{55}$$

This provides us with a simple and principled way to enforce a user-specified error bound. Note that, although approximate, $\hat{\varepsilon}^t$ has physical meaning: it specifies an approximate bound for total integrated error of each cache point, and is influenced by the radiometric and geometric configuration of the scene.

In Figure 14 we compare this new error metric to previous split-sphere heuristics for a simple scene with a white wall, black wall, and no wall. We show the cache point distribution and the resulting RMS error across the receiving line. In all cases our new approach is able to obtain an equal or much lower error than previous methods. Furthermore, geometric methods based on the split-sphere heuristic can produce completely different cache point densities even in scenes that are radiometrically identical (middle and bottom). Our error heuristic based on the second-order Taylor expansion does not suffer from this problem, producing exactly the same cache distribution and error in such scenarios. We show a similar comparison for a scene containing indirect illumination and occlusion in Figure 15. Here we see that the Hessian-based error metric produces slightly improved results even when using the arc-length formulation which ignores visibility. Accounting for occlusions in the Hessian improves the RMS error even further.

7. SECOND-ORDER ANALYSIS OF 3D GLOBAL ILLUMINATION

As seen in the previous section, analyzing concepts in the 2D domain is both mathematically and conceptually easier. Aside from the value for teaching, this also leads to computational savings which allows more rapid experimentation and faster prototyping. The simplified 2D setting allowed us to gather a strong intuition about practical rendering problems. We saw that the Ward gradient simplifies to a well-founded, numerically correct, and easy-to-understand technique in the 2D domain, providing a better understanding of its behavior. This was all possible while providing a direct way to visualize the concepts we were exploring. We could quickly demonstrate how second-order derivatives might provide a benefit in rendering, and could easily visualize these concepts as simple curves on planar graphs. These concepts become increasingly difficult to illustrate in higher dimensions.

Equipped with these benefits we now build upon our 2D analysis and apply a similar second-order analysis to 3D global illumination. We first derive an irradiance gradient using a surface-area and stratified formulation, and then generalize the 2D surface-area Hessian to 3D.

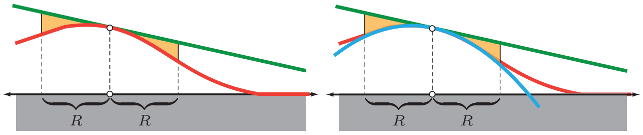


Fig. 13: In the ideal setting (left) we would compute the radius R by constraining the integrated difference (orange) between the first-order Taylor expansion (green) and the true irradiance (red). Since this requires knowledge of the true irradiance, we instead (right) use the second-order Taylor expansion (blue) as an oracle for the true irradiance.

[1] We also investigated other definitions of error such as average error, average squared error, and integrated squared difference, but found that minimizing total error produced results with the lowest RMS error across the image.

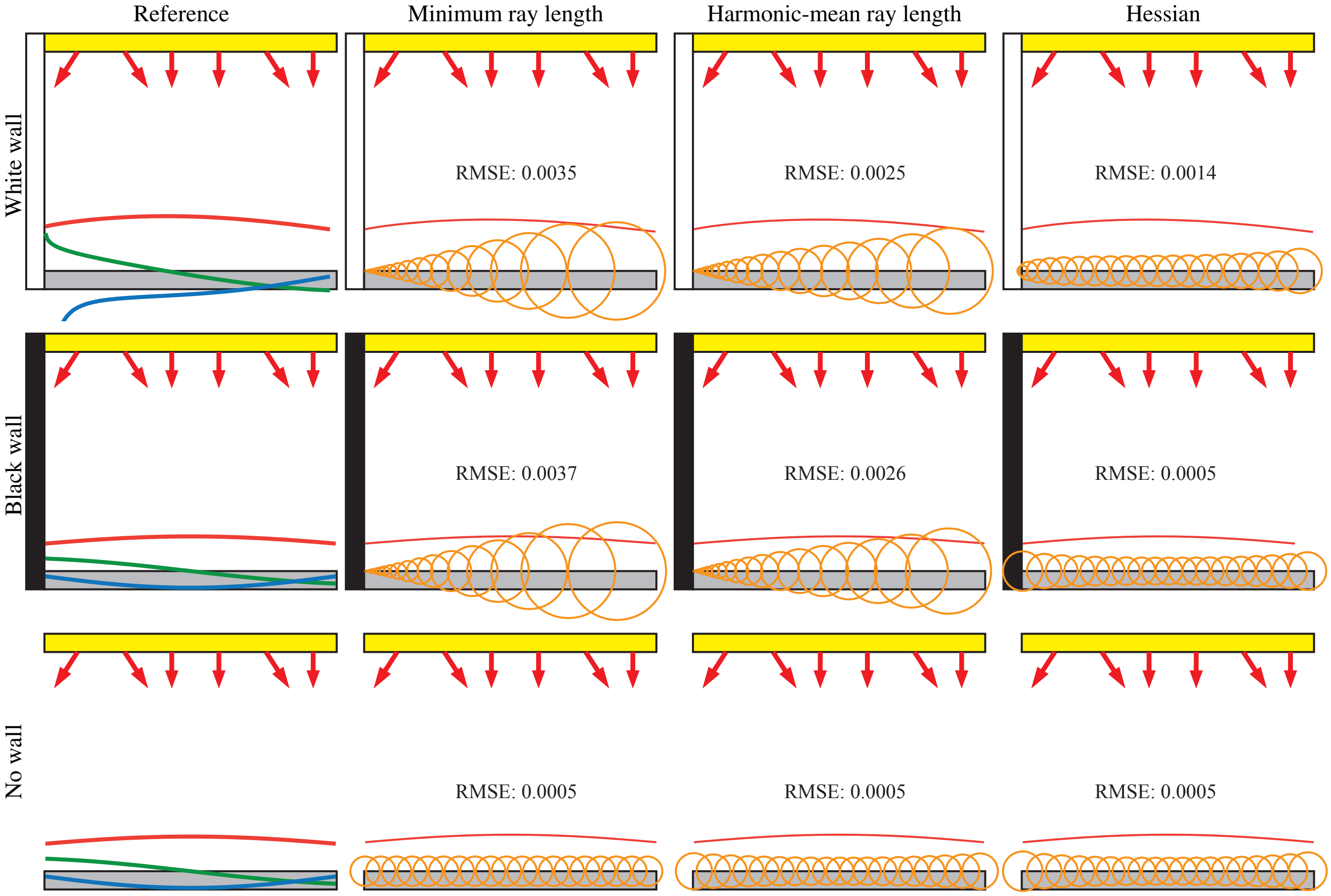


Fig. 14: Cache point distribution for a simple 2D test scene. The first scene (top) contains a light source, a left wall with an albedo of 1 and a receiving plane. The irradiance and its first two derivatives are plotted in red, green and blue, respectively. In the second scene (middle) the albedo of the left wall is changed to 0 and in the third scene (bottom) the wall is removed. The right columns show the cache point distributions for 20 cache points (orange circles) placed with the minimum ray length, harmonic-mean ray length and Hessian based methods, respectively. We also report the RMS error. Our Hessian-based approach significantly reduces error compared to the split-sphere methods and produces identical cache points in radiometrically equivalent scenes (bottom two rows).

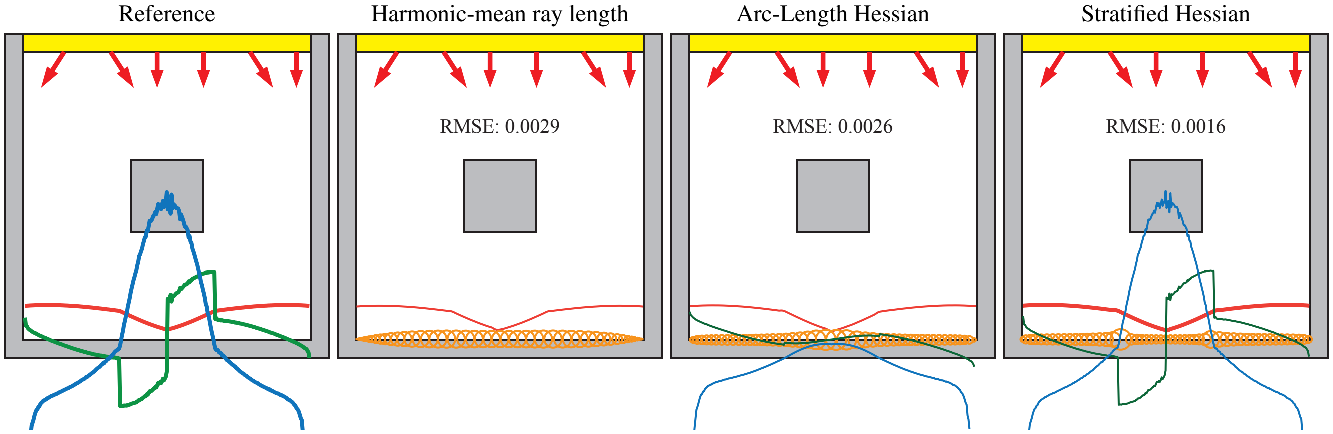


Fig. 15: Radius estimation in the presence of occlusion and indirect illumination. The scene contains an occluder block in the center of the scene. All surfaces have an albedo of 0.5. Even the arc-length Hessian (center right) which ignores occlusions provides a lower RMS error than the split-sphere heuristic (center left). The stratified Hessian provides further error reduction.

We will also see that analysis in the simplified domain can help determine the cause of inaccuracies or failures in 3D rendering techniques. Though correct in 2D, we will see that the stratified Ward gradient actually involves significant approximations in 3D and cannot be employed directly for Hessian computations due to the dominance of this approximation error for higher-order derivatives.

We apply our derivations to the irradiance caching algorithm for improved results and in light of these new insights suggest exciting new directions for future research.

7.1 Irradiance Gradients

Surface-Area Formulation. Because we derived the arc-length gradient in a vector formulation, deriving the 3D counterpart is simple. We can apply the gradient operator and use the same argumentation as before to get rid of the visibility and radiance gradients:

$$\begin{aligned}\nabla_{\mathbf{x}} E_{3D}(\mathbf{x}) &= \nabla_{\mathbf{x}} \int_a L_{3D}(\mathbf{x}\leftarrow\mathbf{y})\, V_{3D}(\mathbf{x},\mathbf{y})\, G_{3D}(\mathbf{x},\mathbf{y})\, da(\mathbf{y}) && (56)\\ &\approx \int_a L_{3D}(\mathbf{x}\leftarrow\mathbf{y})\, V_{3D}(\mathbf{x},\mathbf{y})\, \nabla_{\mathbf{x}} G_{3D}(\mathbf{x},\mathbf{y})\, da(\mathbf{y}). && (57)\end{aligned}$$

We show in Appendix C that the gradient of the geometry term is:

$$\nabla_{\mathbf{x}} G_{3D}(\mathbf{x},\mathbf{y}) = 4\frac{\cos\theta_{\mathbf{x}}\cos\theta_{\mathbf{y}}}{r^4}\mathbf{y} - \frac{\cos\theta_{\mathbf{y}}}{r^3}\vec{\mathbf{n}}_{\mathbf{x}} + \frac{\cos\theta_{\mathbf{x}}}{r^3}\vec{\mathbf{n}}_{\mathbf{y}}. \quad (58)$$

Again, we can evaluate this continuous gradient formulation numerically using Monte Carlo integration.

Stratified Gradient. The surface-area gradient formulation above does not account for occlusion changes. To incorporate this, we extend the stratified gradient formulation to 3D.

The key idea in this formulation is the same in 2D and 3D; however, now we need to track 2D strata of the hemisphere instead of arc sectors of a hemicircle. To define this mathematically, let $\Omega_{i,j} = [\phi_{i_-}..\phi_{i_+}] \times [\theta_{j_-}..\theta_{j_+}]$ with $(i,j) \in [1\ldots N] \times [1\ldots M]$, and $\phi_{(i-1)_+} = \phi_{i_-} < \phi_{i_+} = \phi_{(i+1)_-}$ and $\theta_{(j-1)_+} = \theta_{j_-} < \theta_{j_+} = \theta_{(j+1)_-}$. Again, we only require that all the strata are disjoint and $\cup_{i,j}\Omega_{i,j} = \Omega$. We illustrate such a spherical parametrization and discretization in Figure 16.

We can now rewrite the hemispherical irradiance integral equation as

$$E_{3D}(\mathbf{x}) = \sum_{i=1}^{N}\sum_{j=1}^{M} \int_{\Omega_{i,j}} L_{3D}(\mathbf{x} \leftarrow \vec{\omega}) \cos\theta_{\vec{\omega}}\, d\vec{\omega}. \quad (59)$$

Like in the 2D case, we assume that the radiance in every cell is constant, and can then separate the integral to get:

$$\begin{aligned}E_{3D}(\mathbf{x}) &\approx \sum_{i=1}^{N}\sum_{j=1}^{M} L_{i,j} \int_{\Omega_{i,j}} \cos\theta_{\vec{\omega}} d\vec{\omega} && (60)\\ &= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} L_{i,j}\,(\phi_{i_+} - \phi_{i_-})(\sin^2\theta_{j_+} - \sin^2\theta_{j_-}), && (61)\end{aligned}$$

and rearranging in terms of strata differences gives us:

$$\begin{aligned}&= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=2}^{M} (L_{i-1,j} - L_{i,j})\,(\phi_{i_+} - \phi_{i_-})\sin^2\theta_{j_-} && (62)\\ &+ \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} (L_{i,j} - L_{i,j-1})\,(\sin^2\theta_{j_+} - \sin^2\theta_{j_-})\,\phi_{i_-}.\end{aligned}$$

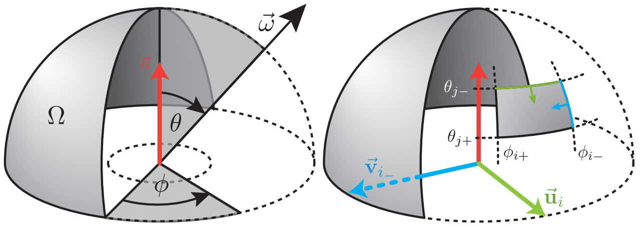


Fig. 16: The spherical parametrization of the hemisphere (left). The stratified gradient formulation considers the movement of the boundaries of each stratum (right).

To compute the gradient in 3D we need to consider the movement of each stratum over the hemisphere as we move the center of projection. This can be expressed by taking the gradient of Equation (62). Unfortunately doing this is complicated since we need to consider the arbitrary movement of a cell wall and the spherical angles at the strata boundaries are not independent. To make this practical, Ward et al. [1992] only considered movement of each cell boundary in one direction, perpendicular to the boundary (small green and blue arrows in Figure 16 (right)). This approximation restricts the movement of each stratum to be axis-aligned in the spherical parametrization. If we take a similar approach, we can compute the gradient as:

$$\begin{aligned}\nabla_{\mathbf{x}} E_{3D}(\mathbf{x}) &\approx \frac{1}{2}\sum_{i=1}^{N}\sum_{j=2}^{M}(L_{i-1,j} - L_{i,j})(\phi_{i_+} - \phi_{i_-})\nabla_{\mathbf{x}}\sin^2\theta_{j_-} && (63)\\ &+ \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}(L_{i,j} - L_{i,j-1})(\sin^2\theta_{j_+} - \sin^2\theta_{j_-})\nabla_{\mathbf{x}}\phi_{i_-}\\ &= \sum_{i=1}^{N}\vec{\mathbf{u}}_i\sum_{j=2}^{M}(L_{i-1,j} - L_{i,j})(\phi_{i_+} - \phi_{i_-})\frac{\sin\theta_{j_-}\cos^2\theta_{j_-}}{r_{i,j-}}\\ &+ \sum_{i=1}^{N}\vec{\mathbf{v}}_{i_-}\sum_{j=1}^{M}(L_{i,j} - L_{i,j-1})\frac{\sin^2\theta_{j_+} - \sin^2\theta_{j_-}}{r_{i-,j}\sin\theta_j}, && (64)\end{aligned}$$

where $\vec{\mathbf{u}}_i = [\cos\phi_i, \sin\phi_i]^\mathsf{T}$ and $\vec{\mathbf{v}}_{i_-} = \left[\sin\phi_{i_-}, -\cos\phi_{i_-}\right]^\mathsf{T}$ are unit vectors in the tangent plane as illustrated in Figure 16 (right). As in the 2D case we need to choose proper values for the radii $r_{i-,j}$ and $r_{i,j-}$ of the boundaries by using the minimum of the stratum radii as proposed by Ward and Heckbert [1992].

This formulation is different than previous stratified formulations in two important ways. Firstly, we can use an arbitrary stratification (that satisfies the above-stated conditions) and not just uniform or cosine-weighted stratifications. Secondly, previous approaches [Ward and Heckbert 1992; Křivánek et al. 2005; Jarosz et al. 2008] approximated the integrals in Equation (60) by point-sampling the cosine terms at the center of each stratum, whereas here we analytically integrate the cosine over each stratum.

7.2 3D Irradiance Hessian

Surface-Area Formulation. It is possible to obtain an irradiance Hessian by differentiating the surface-area gradient formulation. This involves differentiating the 3D geometric coupling term:

$$\mathbf{H}_{\mathbf{x}}(E_{3D}) \approx \int_a LV\,\mathbf{H}_{\mathbf{x}}(G_{3D})\, da(\mathbf{y}). \quad (65)$$

In Appendix D we show that the Hessian of the 3D geometry term is:

$$
\begin{aligned}
\mathbf{H}_{\mathbf{x}}(G_{3D}) = {} & 24\frac{\cos\theta_{\mathbf{x}}\cos\theta_{\mathbf{y}}}{r^6}(\mathbf{y}\mathbf{y}^{\mathsf{T}}) - 4\frac{\cos\theta_{\mathbf{x}}\cos\theta_{\mathbf{y}}}{r^4}\mathbf{I}_3 \\
& - \frac{1}{r^4}\left(\vec{\mathbf{n}}_{\mathbf{x}}\vec{\mathbf{n}}_{\mathbf{y}}^{\mathsf{T}} + \vec{\mathbf{n}}_{\mathbf{y}}\vec{\mathbf{n}}_{\mathbf{x}}^{\mathsf{T}}\right) - 4\frac{\cos\theta_{\mathbf{y}}}{r^5}\left(\vec{\mathbf{n}}_{\mathbf{x}}\mathbf{y}^{\mathsf{T}} + \mathbf{y}\vec{\mathbf{n}}_{\mathbf{x}}^{\mathsf{T}}\right) \\
& + 4\frac{\cos\theta_{\mathbf{x}}}{r^5}\left(\vec{\mathbf{n}}_{\mathbf{y}}\mathbf{y}^{\mathsf{T}} + \mathbf{y}\vec{\mathbf{n}}_{\mathbf{y}}^{\mathsf{T}}\right), \qquad (66)
\end{aligned}
$$

where $\mathbf{I}_3$ is the 3×3 identity matrix. This provides a full 3×3 Hessian matrix, which we can project down to the tangent components if needed.

Note that as before, the surface-area formulation ignores occlusion changes.

Stratified Formulation. The stratified formulation, however, tries to determine the change in projected area of each stratum by considering the rate of movement of the strata boundaries. This has the potential to account for occlusion changes. In 2D we extended this formulation to a Hessian where the boundaries between strata are points on a hemicircle. As we described in the previous section, in 3D this formulation makes a simplifying assumption by operating on boundary *edges* and only considering motion perpendicular to each edge. In effect it assumes that after translating the center of projection, the strata remain rectangles in the spherical parametrization. This is an approximation since, in reality, the edges of a stratum will not necessarily remain aligned with the lines of longitude and latitude but may warp to an arbitrary spherical quad. This approximation was reasonable in the case of the gradient, but unfortunately it becomes increasingly problematic when we consider higher-order derivatives. In our investigation we found that extending this formulation to a Hessian is not feasible since the error induced by this approximation overwhelms the calculations. In Section 9 we suggest more promising approaches to incorporate occlusion changes in the Hessian; however, in the following section, we show that even without the stratified formulation we can use the 3D Hessian to solve several practical problems in irradiance caching.

8. ANALYSIS AND APPLICATIONS IN 3D

We now apply our 3D analysis to obtain practical benefits within irradiance caching. We first show that a second-order Taylor expansion of the irradiance can provide error reduction in irradiance caching. We also show that we can obtain even better improvements by using the Hessian to derive the cache point radii. We derive a Hessian-based error metric for irradiance caching which is superior to the classical split-sphere heuristic, resulting in over an order of magnitude less error in the image for the same number of cache points. Figure 17 shows the test scenes we use to analyze our 3D implementation. The box scene consists of a floor, a left wall with albedo 0.5, a right wall with albedo 0, and a ceiling area light. The front and back walls are missing, and the environment is black. We also consider a variation of this scene with an occluder.

We restrict ourselves to these simple scenes to demonstrate the potential for irradiance Hessians in a controlled setting. In this case, tracing rays is extremely fast so computing the Hessian (unoptimized) results in roughly 50% more computation time. This number will decrease significantly even with moderately complex scenes, which we plan to investigate in future work.

8.1 Second-Order Extrapolation

As in 2D, we can easily extend irradiance caching to perform a second-order Taylor expansion at each cache point before interpolation by using Equation (52). In 3D a gradient expansion extrapolates each cache point as a linear plane and a Hessian expansion extrapolates along a quadric. In Figure 18 we compare the results of irradiance caching with constant, gradient, and Hessian extrapolation for the two scenes in Figure 17. We used the minimum distance split-sphere heuristic. Hessian extrapolation provides more than a 2× reduction in RMS error for the no occlusions scene (top) compared to gradient extrapolation. For the scene with the occluder (bottom), the Hessian still provides some benefit since it can more smoothly reconstruct the fully-illuminated regions, but, as expected, it does not improve the quality in the penumbra since the surface-area Hessian ignores visibility changes. This suggests that future work in deriving a visibility-aware Hessian could provide further benefits.

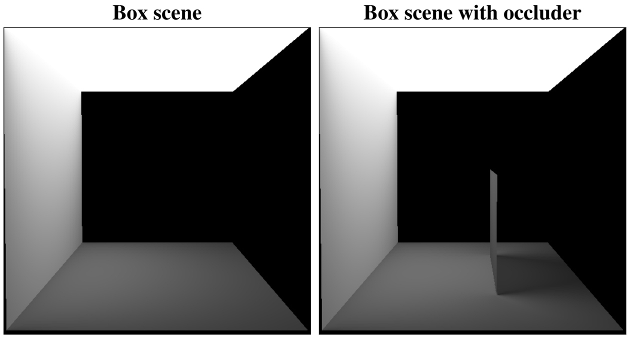


Fig. 17: Our 3D test scenes. A box (left) with a ceiling light, a ground plane, and grey left and black right walls; and (right) the same scene but with a diffuse occluder producing indirect contact shadows. We consider only the indirect irradiance on the floor.

Second-order extrapolation requires only slightly increased storage since the Hessian matrix can be compactly stored as a symmetric 3×3, or even a symmetric 2×2 matrix. In uncompressed greyscale form this is an increase from 8 floats to 11 floats per cache point.

8.2 Hessian Error Metric for Irradiance Caching

Though it is possible to obtain modest improvements in interpolation quality by using the Hessian for second-order expansion, one of the primary contributors to error in the irradiance caching algorithm is the suboptimal placement of cache points. As in 2D, we will derive a principled Hessian-based error metric for 3D irradiance caching, which we show can provide over an order of magnitude improvement in the RMS error. In addition to the benefits we already saw in 2D, we will also show that such a metric naturally allows for anisotropic cache points, which further improves rendering quality.

We again define the **total error** ε^t of a cache point as the integrated difference between the correct irradiance and the extrapolated irradiance over the support of the cache point:

$$
\varepsilon^t = \iint_A |E(\mathbf{x}_i + \mathbf{x}) - E'(\mathbf{x}_i + \mathbf{x})| \, d\mathbf{x}, \qquad (67)
$$

where A is the area of support of the cache point, and $E'(\mathbf{x}_i + \mathbf{x}) = E_i(\mathbf{x}_i) + \nabla_{\mathbf{x}} E_i(\mathbf{x}_i) \cdot \mathbf{x}$ is the first-order Taylor expansion of the irradiance. We again use the second-order Taylor expansion of the irradiance as an oracle for the ground-truth, which results in the following approximation for the total error:

$$
\varepsilon^t \approx \hat{\varepsilon}^t = \frac{1}{2} \iint_A |\mathbf{x}^t \, \mathbf{H}_{\mathbf{x}}(E_i) \, \mathbf{x}| \, d\mathbf{x}. \qquad (68)
$$

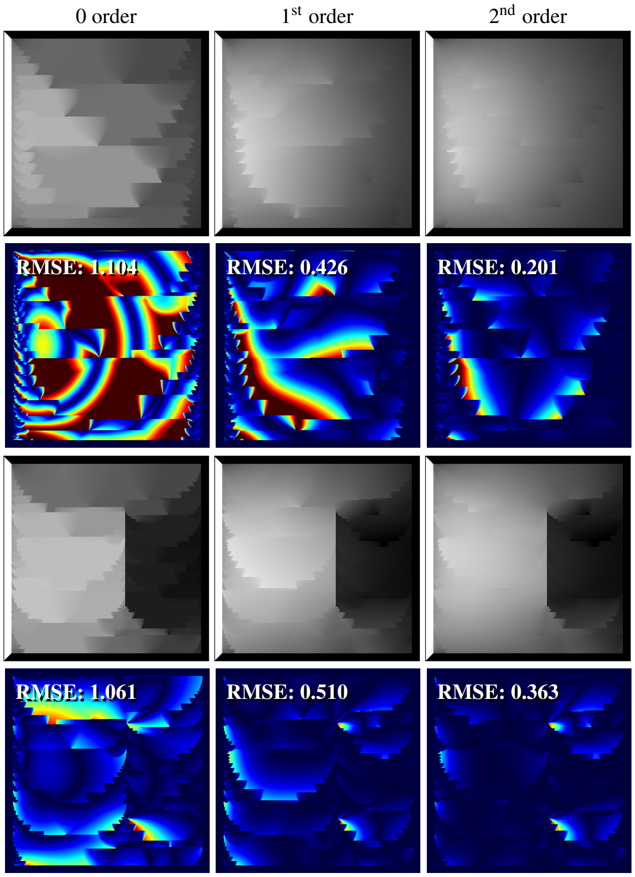


Fig. 18: Irradiance reconstruction improves (RMS error decreases) when using progressively higher order Taylor expansions (constant, gradient, and Hessian extrapolation). Hessian extrapolation provides more than a 2× reduction in error compared to gradient extrapolation in the no occlusion scene (top). The occlusion scene (bottom) also exhibits improvement; however, since the surface-area Hessian formulation ignores visibility changes, the improvement is less significant and primarily concentrated in fully-illuminated regions.

Here we assume that $\mathbf{x}$ is a 2D point on a surface and the irradiance Hessian is a 2×2 matrix defining the second derivatives in the tangent space of the surface. This can be obtained by projecting the full 3×3 Hessian matrix (Equation (66)) onto the tangent plane.

The Hessian inside the integral defines a quadric surface. To obtain a cache point radius, we conservatively bound this using the maximum curvature of the surface, which is the maximum absolute eigenvalue λ_1 of $\mathbf{H}_\mathbf{x}(E_e)$:

$$\hat{\varepsilon}^t \leq \pi \int_0^{R_i} |\lambda_1| r^2 r \, dr = \frac{\pi}{4} \lambda_1 R^4, \tag{69}$$

where the last r term in the integral is due to the change of variables into polar coordinates. Solving this equation for R_i gives us a formula for computing a cache point radius:

$$R_i = \sqrt[4]{\frac{4\hat{\varepsilon}^t}{\pi\lambda_1}}, \tag{70}$$

which induces a certain amount of error into the scene. We call this our radiometric Hessian metric. Note that in 3D the radius is proportional to the reciprocal fourth-root of the maximum Hessian eigenvalue, while in 2D the radius was proportional to the reciprocal cube-root of the second-derivative.

Discussion. In Figure 19 (left) we analyze the behavior of our error metric in the no occluder box scene. We show the results of both the minimum distance and harmonic-mean variants of the split-sphere heuristic and compare to our radiometric Hessian-based error metric. We use 30k gather rays and the same number of cache points (around 1000) for all methods, performing gradient extrapolation using the stratified gradient formulation.

Methods based on the split-sphere heuristic concentrate cache points very aggressively around edges and corners, which is wasteful and can lead to high error. Unfortunately it is difficult to produce an efficient distribution of cache points without relying excessively on additional heuristics, such as finely-tuned minimum and maximum radii. Though it may be possible to alleviate this problem to a degree by incorporating radius clamping, this is an inelegant way to fix a broken heuristic and requires the user to adjust three parameters instead of one. Our goal is to provide an automatic error metric which does not need to rely *excessively* on radius clamping. Another related problem with the split-sphere heuristic is that the distributions often have extreme variation in the radii over small distances, which can lead to large cache points "eating" away at the distribution within their support radius. Křivánek et al. [2006] addresses this by retroactively enforcing the triangle inequality on the split-sphere radii. Our Hessian-based radius calculation obtains a higher quality point distribution than the split-sphere without relying on these additional correction measures.

Another fundamental challenge of purely geometric approaches such as the split-sphere heuristic is that scenes which are identical radiometrically but different geometrically can produce vastly different cache point distributions. We demonstrate this by removing the right wall and exposing the black environment (second row of each method in Figure 19). Even though this produces a radiometrically identical image, the cache point distribution produced by the split-sphere is quite different in the two cases which results in vastly different errors.

As Figure 19 shows, our Hessian-based error metric does not suffer from these problems. It computes the radii using a principled estimation of error from a second-order Taylor expansion. Our Hessian-based approach produces cache point distributions which densely cover the surface and vary more gradually. This provides a significant improvement in rendering quality. Simply replacing the split-sphere heuristic with our radiometric Hessian metric reduces the error by a factor of 2× to 20×. Furthermore, since this approach is based on radiometric quantities (and not purely geometric relationships) it obtains identical cache point distributions regardless of whether the black wall is present.

In Figure 19 (right) we examine the behavior for the box scene when an occluder is placed within the box (Figure 17 right). The split-sphere methods perform largely the same and we see that our Hessian-based error metric again provides superior results.

In Figure 20 (top) we show a failure case for our radiometric Hessian where we have modified the occluder to have an albedo of 0.01. This scene points out a fundamental challenge with a purely radiometric error metric: if the irradiance is (close to) zero, then the Hessian is not well defined and will estimate an infinite (or very large) radius. Note that this is not due to omitting occlusions in the surface-area Hessian calculation, but is a fundamental property of a radiometric approach. This can be seen in the Radiometric Hessian cache point distribution for the occlusion scene. Though regions of zero indirect irradiance are not actually common in physical scenes,

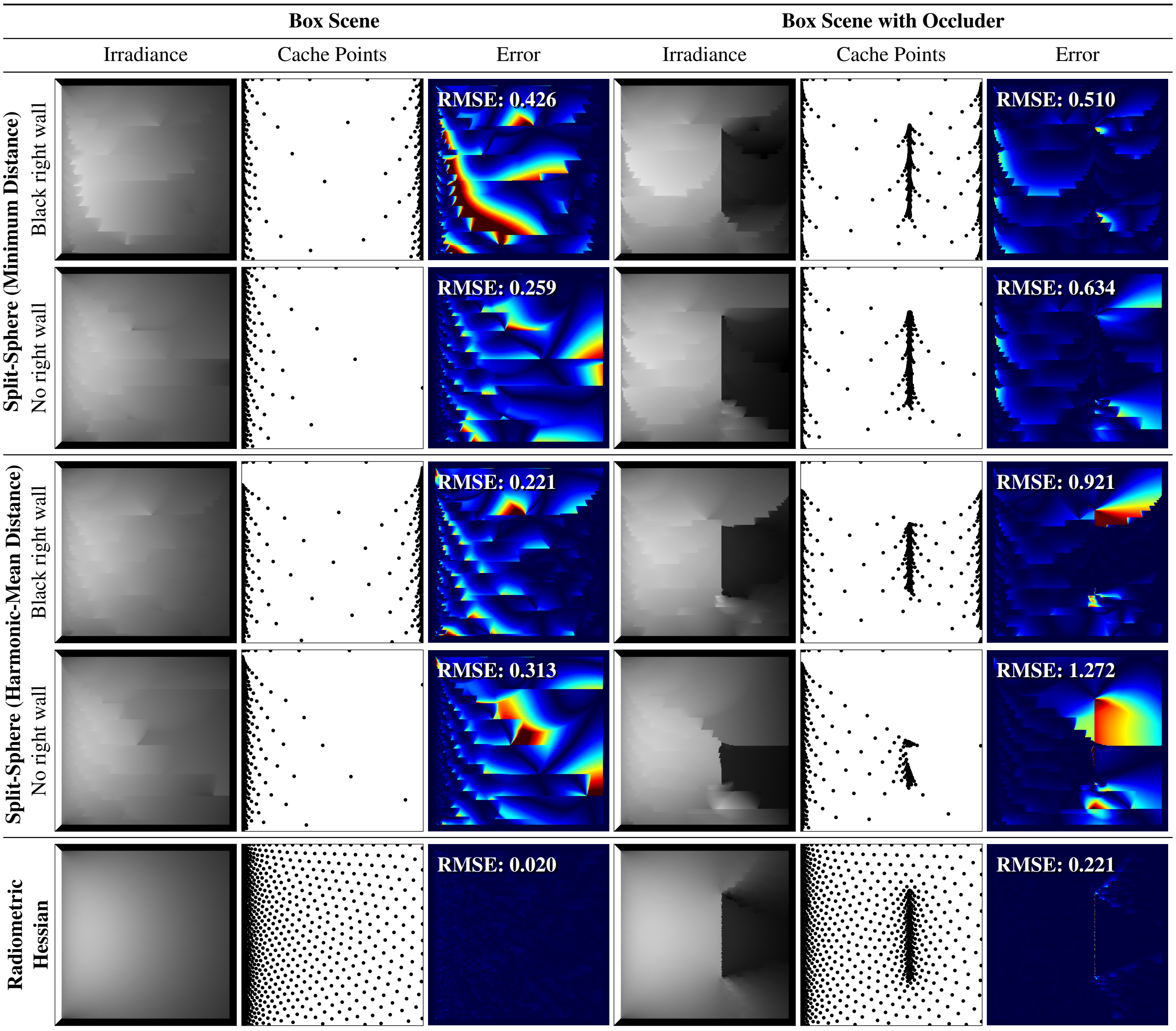


Fig. 19: We compare the split-sphere heuristic for computing cache point radii to our Hessian-based error metric for the simple 3D scenes in Figure 17. All methods use roughly 1000 cache points. We analyze the cache point distribution and error when viewing the floor from above. For the split-sphere methods we also show the results when the black wall on the right is removed. Our radiometric Hessian metric provides a 2× to 20× reduction in error compared to split-sphere heuristics with identical results regardless of whether the black wall is present.

synthetic scenes may often have such configurations by e.g. only simulating a small number of indirect bounces.

8.3 Geometric Hessian Error for Irradiance Caching

To address the issues with the radiometric Hessian and obtain a metric which is robust to arbitrary lighting configurations we consider a conservative bound on the radiometric Hessian error defined in Equation (68). We first note that the magnitude of the Hessian's eigenvalues are directly proportional to the radiance over the hemisphere, e.g. if the radiance is doubled then the eigenvectors also double. We could therefore conservatively bound the Hessian error by replacing the incident radiance L in Equation (65) with the maximum radiance $L_{\max}$ of any surface in the scene:

$$\mathbf{H}_{\mathbf{x}}(E_{3D}^{\max}) = \int_a L^{\max}\,\mathbf{H}_{\mathbf{x}}(G_{3D})\,da(\mathbf{y}). \tag{71}$$

With this change, the maximum Hessian of the irradiance reduces to the integral of the Hessian of the geometry term: $\mathbf{H}_{\mathbf{x}}(E_{3D}^{\max}) \propto \int \mathbf{H}_{\mathbf{x}}(G_{3D})$. Unfortunately, estimating $L^{\max}$ is difficult in practice. However, since it is a constant for the entire scene, changing it simply applies a different global scaling factor to all cache point radii. We fold this scaling factor into the user parameter ε^t, and can

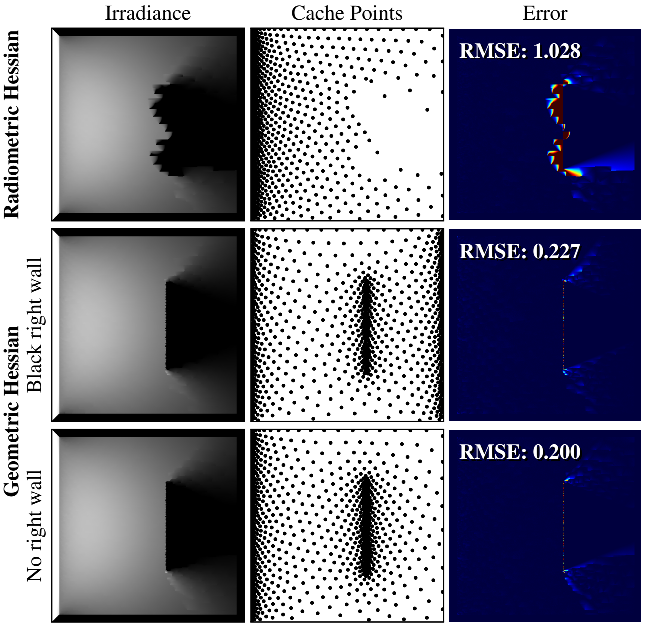


Fig. 20: A simple modification of Figure 19 where the occluder is nearly black. A purely radiometric error metric (top) can have difficulty robustly handling regions of (nearly) black irradiance. These metrics can predict unbounded cache point radii when the irradiance and all gradients are zero. Our solution using the Geometric Hessian (bottom) robustly resolves this issue, producing well-behaved cache point distributions.

therefore replace the maximum radiance with a canonical value[2] $L^{\max} = 1$.

Inserting this Hessian bound in place of $\mathbf{H}_{\mathbf{x}}(E_{3D})$ in Equation (68) gives us a conservative error bound, which we can solve in a similar fashion as Equation (70). We call this our geometric Hessian metric.

Discussion. The bottom two rows of Figure 20 examine the performance of the geometric Hessian metric in comparison to the radiometric Hessian for our failure case. Since the upper-bound on the Hessian is a purely geometric quantity, changes in the scene geometry will produce slightly different cache point distributions even in radiometrically identical scenes; however, our results indicate that the geometric Hessian preserves the error reduction properties of the radiometric Hessian while being more robust to arbitrary lighting configurations. We believe with more extensive investigation this error metric could prove quite practical as a replacement for the classical split-sphere heuristic.

Our goal is to achieve improved cache point placement without resorting to finely-tuned minimum and maximum radii. In practice, some form of conservative radius clamping should be used since under-sampling could always produce arbitrarily inaccurate radii. However, the error metric should ideally provide reasonable results without clamping. In Figure 21 we show that the geometric Hessian error metric is quite robust to under-sampling and noisy input without the need for clamping. Even when using only 16 gather rays over the hemisphere with noise artificially injected in the radiance, this metric is able to robustly produce high-quality distributions.

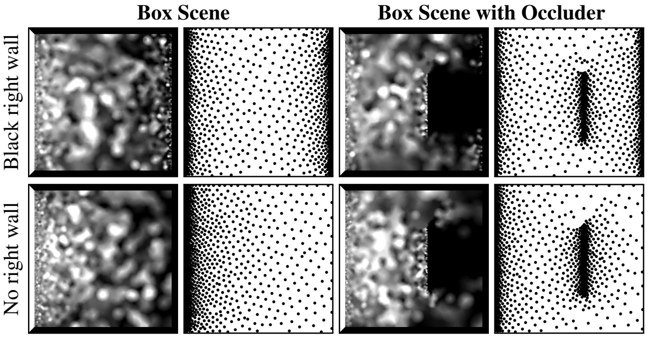


Fig. 21: This example illustrates the robustness of the geometric Hessian-based radius estimation even in the presence of under-sampling and strong noise. We only trace 16 gather rays per cache point and multiply each gather ray's radiance by a uniform random value between 0 and 2. While the irradiance estimate is very noisy, the maximum eigenvalue of the Hessian can already be estimated robustly, leading to a good distribution of cache points.

8.4 Anisotropic Error Metric for Irradiance Caching

The Hessian in Equation (68) in fact describes a quadric surface which gives us a natural way to obtain anisotropic (elliptical) cache points. Working in the coordinate system defined by the principle curvatures (eigenvectors $\mathbf{v}_1, \mathbf{v}_2$ of the Hessian matrix with corresponding eigenvalues/curvatures λ_1 and λ_2), we can express the integrated error in the vicinity of the cache point as:

$$\hat{\varepsilon}^t = \frac{1}{2} \iint_A \left(|\lambda_1| x^2 + |\lambda_2| y^2 \right) \, dy\, dx. \tag{72}$$

The shape that maximizes the surface area of the cache point while minimizing the integrated error is an ellipse with axes $\mathbf{v}_1$ and $\mathbf{v}_2$. Inverting this equation provides us with the radii along these axes:

$$\left(R_i^{\lambda_1}, R_i^{\lambda_2} \right) \approx \sqrt[4]{\frac{4\hat{\varepsilon}^t}{\pi}} \left(\sqrt[4]{\frac{1}{\lambda_1}}, \sqrt[4]{\frac{1}{\lambda_2}} \right). \tag{73}$$

Discussion. In Figure 22 we demonstrate the benefits of anisotropic irradiance caching. We use the geometric Hessian from the previous section and extract elliptical cache points using Equation (73). We visualize the cache points as well as their circular and elliptical regions and the resulting error over the image. Using roughly 256 cache points, anisotropic cache points provides an additional 40% reduction in error.

The idea of anisotropic cache points was first suggested by Herzog et al. [2009] for accelerating gathering from Lightcuts [Walter et al. 2005]. They used the split-sphere heuristic combined with neighbor clamping [Křivánek et al. 2006] but reduced the radius along the gradient direction to produce elliptical cache points. Unfortunately since their anisotropic cache points are obtained from the gradient, the eccentricity of all cache points must be set by the user. Our Hessian-based approach could easily be applied within their context and deduces the eccentricity automatically in a principled fashion by minimizing error.

[2] To retain the physical meaning of the error threshold ε^t we could also estimate the maximum radiance in the scene by e.g. using the maximum of the precomputed photon radiances in a photon map [Christensen 1999].

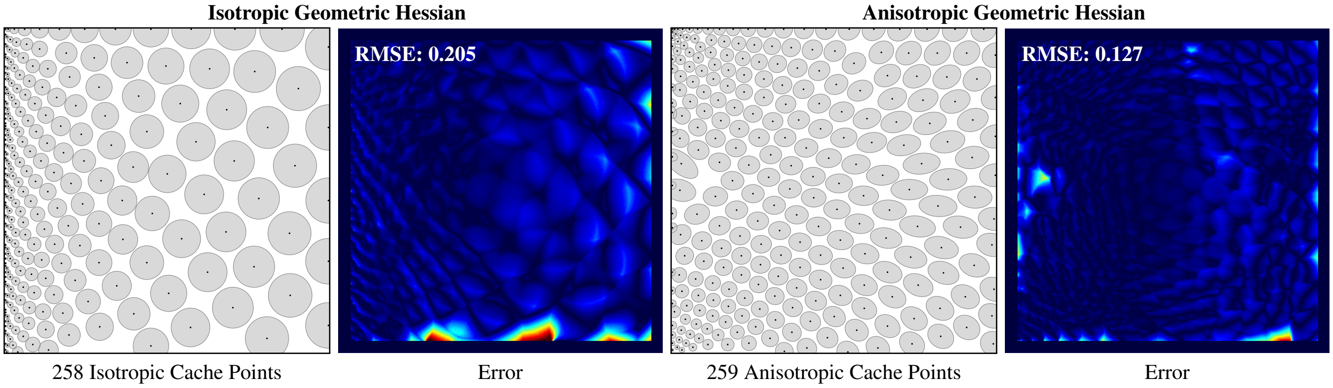


Fig. 22: A comparison of isotropic (left) and anisotropic (right) cache points using our geometric Hessian error metric for the box scene in Figure 19. We visualize the distribution and shape of the cache points and also show the resulting per-pixel error. With the same number of cache points, anisotropic cache points produce lower RMS error over the image. For visualization with minimum overlap we scale the radii of the cache points by half. In reality all areas of the domain are covered by at least one cache point.

9. CONCLUSION AND FUTURE WORK

We presented a comprehensive theory of 2D global illumination which allowed us to implement and easily analyze standard global illumination algorithms in 2D. We used our framework to perform a second-order analysis of global illumination and extended this analysis to obtain practical improvements for irradiance caching in 3D. We showed, for the first time, that a principled error metric derived from a second-order Taylor expansion of the irradiance provides practical benefits over the long-standing split-sphere heuristic. Our analysis suggests future work in a number of directions.

We performed derivations and initial analysis of our Hessian-based error metrics on simple scenes in 2D and 3D. We are confident that our initial exploration will result in a more practical irradiance caching algorithm after more careful analysis on a large collection of complex scenes. We leave this practical investigation as future work to our theoretical analysis.

Our process of extending 2D Hessians to 3D highlighted a previously overlooked issue in the stratified gradient formulation which prevents generalization to Hessians. Though our Hessian-based error metrics compare favorably to the split-sphere approaches, our 2D analysis shows that occlusion-aware Hessians could provide further improvements. We believe the most promising approach for obtaining an occlusion-aware Hessian is to consider the movement of the corners of each stratum instead of the edges. This was done previously for irradiance Jacobians for polygonal scenes [Arvo 1994] and Hessians of radiosity [Holzschuch and Sillion 1995; Holzschuch 1996; Holzschuch and Sillion 1998].

Our error metric computes radii by assuming a first-order Taylor expansion from cache points and using the second-order expansion only to approximate the remaining error. Though we found that second-order Taylor expansion provides some benefits in reconstruction error, care must be taken when computing the maximum radii to avoid over-shooting the ground-truth solution. To use second-order expansion, the principled approach would be to derive a third-order derivative of the irradiance (a rank-2 tensor) to approximate the remaining error. Unfortunately, the resulting complexity would likely outweigh the advantages. Instead, we could naïvely use second-order expansion together with our radius calculation, though overshooting would need to be resolved. Investigating a practical solution for second-order expansion could be a promising avenue for future work.

Another avenue of future work could investigate different definitions of error minimized by the radius computation. We investigated a number of choices for error, including average error, average squared error, relative error, and integrated squared difference, but found that minimizing total error produced results with the lowest RMS error over the image. However, though we minimize total integrated error for each cache point, we reported the RMS error for the image. These two metrics are related but distinct. A exciting possibility for future work is to couple the error metric used to access the quality of the rendered image with the error threshold minimized by the radius computation. This could confirm that the radius calculation in fact minimizes the desired error over the image. It may also be possible to incorporate elements of perceptual error metrics such as visible difference predictors [Daly 1993] and structural similarity [Wang et al. 2004] directly into the radius calculation.

ACKNOWLEDGMENTS

This study was supported in part by the Deutsche Forschungsgemeinschaft (DFG, IRTG 1328). We would like to thank Derek Nowrouzezahrai and Peter-Pike Sloan for proof reading, discussions, and helpful comments.

REFERENCES

Abbott, E. A. 1884. *Flatland: A romance of many dimensions*. Dover Publications.

Annen, T., Kautz, J., Durand, F., and Seidel, H.-P. 2004. Spherical harmonic gradients for mid-range illumination. In *Rendering Techniques 2004 (Proceedings of the Eurographics Symposium on Rendering)*. 331–336.

Arvo, J. 1994. The irradiance jacobian for partially occluded polyhedral sources. In *Proceedings of SIGGRAPH 94*. Computer Graphics Proceedings, Annual Conference Series. 343–350.

Arvo, J. and Kirk, D. B. 1990. Particle transport and image synthesis. In *Computer Graphics (Proceedings of SIGGRAPH 90)*. 63–66.

Arvo, J., Torrance, K., and Smits, B. 1994. A framework for the analysis of error in global illumination algorithms. In *Proceedings of SIGGRAPH 94*. Computer Graphics Proceedings, Annual Conference Series. 75–84.

Christensen, P. H. 1999. Faster photon map global illumination. *Journal of Graphics Tools 4,* 3, 1–10.

DALY, S. 1993. *Digital images and human vision*. MIT Press, Cambridge, MA, USA, Chapter The visible differences predictor: an algorithm for the assessment of image fidelity, 179–206.

DURAND, F., HOLZSCHUCH, N., SOLER, C., CHAN, E., AND SILLION, F. X. 2005. A frequency analysis of light transport. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2005) 24,* 3 (Aug.), 1115–1126.

DURAND, F., ORTI, R., RIVIÈRE, S., AND PUECH, C. 1996. Radiosity in flatland made visibly simple: Using the visibility complex for lighting simulation of dynamic scenes in flatland. In *Symposium on Computational Geometry*. V–11–V–12.

DUTRÉ, P., BALA, K., AND BEKAERT, P. 2006. *Advanced Global Illumination*, 2 ed. A K Peters, Natick, USA.

EDELSBRUNNER, H., OVERMARS, M. H., AND WOOD, D. 1983. Graphics in flatland: A case study. *Advances in Computing Research 1*, 35–59.

GORTLER, S. J., SCHRÖDER, P., COHEN, M. F., AND HANRAHAN, P. 1993. Wavelet radiosity. In *Proceedings of SIGGRAPH 93*. Computer Graphics Proceedings, Annual Conference Series. 221–230.

HECKBERT, P. 1992. Radiosity in flatland. *Computer Graphics Forum (Proceedings of Eurographics 1992) 11,* 3 (Sept.), 181–192.

HERZOG, R., MYSZKOWSKI, K., AND SEIDEL, H.-P. 2009. Anisotropic radiance-cache splatting for efficiently computing high-quality global illumination with lightcuts. *Computer Graphics Forum (Proceedings of Eurographics 2009) 28,* 2 (Apr.), 259–268.

HOLZSCHUCH, N. 1996. Le contrôle de l'erreur dans la méthode de radiosité hiérarchique. Ph.D. thesis, Université Joseph Fourier (Grenoble I).

HOLZSCHUCH, N. AND SILLION, F. 1995. Accurate computation of the radiosity gradient for constant and linear emitters. In *Rendering Techniques 1995 (Proceedings of the Eurographics Workshop on Rendering)*. Springer-Verlag, 186–195.

HOLZSCHUCH, N. AND SILLION, F. X. 1998. An exhaustive error-bounding algorithm for hierarchical radiosity. *Computer Graphics Forum 17,* 4, 197–218.

JAROSZ, W., DONNER, C., ZWICKER, M., AND JENSEN, H. W. 2008. Radiance caching for participating media. *ACM Transactions on Graphics 27,* 1 (Mar.), 7:1–7:11.

JAROSZ, W., ZWICKER, M., AND JENSEN, H. W. 2008. Irradiance gradients in the presence of participating media and occlusions. *Computer Graphics Forum (Proceedings of Eurographics 2008) 27,* 4 (June), 1087–1096.

JENSEN, H. W. 2001. *Realistic image synthesis using photon mapping*. A. K. Peters, Ltd., Natick, MA, USA.

KAJIYA, J. T. 1986. The rendering equation. In *Computer Graphics (Proceedings of SIGGRAPH 86)*. 143–150.

KŘIVÁNEK, J., BOUATOUCH, K., PATTANAIK, S., AND ZÁRA, J. 2006. Making radiance and irradiance caching practical: Adaptive caching and neighbor clamping. In *Rendering Techniques 2006 (Proceedings of the Eurographics Symposium on Rendering)*. 127–138.

KŘIVÁNEK, J., GAUTRON, P., BOUATOUCH, K., AND PATTANAIK, S. 2005. Improved radiance gradient computation. In *Spring Conference on Computer Graphics 2005*. 155–159.

KŘIVÁNEK, J., GAUTRON, P., PATTANAIK, S., AND BOUATOUCH, K. 2005. Radiance caching for efficient global illumination computation. *IEEE Transactions on Visualization and Computer Graphics 11,* 5 (Sept./Oct.), 550–561.

LARSON, G. W. AND SHAKESPEARE, R. A. 1998. *Rendering with radiance: the art and science of lighting visualization*. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.

ORTI, R., RIVIÉRE, S., DURAND, F., AND PUECH, C. 1996. Radiosity for dynamic scenes in flatland with the visibility complex. *Computer Graphics Forum 15,* 3 (Aug.), 237–248.

POCCHIOLA, M. 1990. Graphics in flatland revisited. In *SWAT 90*. Lecture Notes in Computer Science, vol. 447. Springer Berlin / Heidelberg, New York, NY, USA, 85–96.

RAMAMOORTHI, R., MAHAJAN, D., AND BELHUMEUR, P. 2007. A first-order analysis of lighting, shading, and shadows. *ACM Transactions on Graphics 26,* 1 (Jan.), 2:1–2:21.

REICHERT, M. C. 1992. A two-pass radiosity method driven by lights and viewer position. M.S. thesis, Cornell University.

SHIRLEY, P. S. 1991. Time complexity of monte carlo radiosity. In *Eurographics '91*. 459–465.

TABELLION, E. AND LAMORLETTE, A. 2004. An approximate global illumination system for computer generated films. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2004) 23,* 3 (Aug.), 469–476.

VEACH, E. AND GUIBAS, L. J. 1997. Metropolis light transport. In *Proceedings of SIGGRAPH 97*. Computer Graphics Proceedings, Annual Conference Series. 65–76.

WALTER, B., FERNANDEZ, S., ARBREE, A., BALA, K., DONIKIAN, M., AND GREENBERG, D. P. 2005. Lightcuts: a scalable approach to illumination. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2005) 24,* 3 (Aug.), 1098–1107.

WANG, Z., BOVIK, A. C., SHEIKH, H. R., AND SIMONCELLI, E. P. 2004. Image quality assessment: From error visibility to structural similarity. *IEEE transactions on image processing 13,* 4, 600–612.

WARD, G. J. AND HECKBERT, P. 1992. Irradiance gradients. In *Eurographics Workshop on Rendering*. 85–98.

WARD, G. J., RUBINSTEIN, F. M., AND CLEAR, R. D. 1988. A ray tracing solution for diffuse interreflection. In *Computer Graphics (Proceedings of SIGGRAPH 88)*. 85–92.

APPENDIX

A. GRADIENT OF 2D GEOMETRY TERM

Applying the product rule to the 2D geometry term in Equation 26 yields:

$$\nabla_{\mathbf{x}} G_{2D} = \cos\theta_{\mathbf{x}}\cos\theta_{\mathbf{y}}\nabla_{\mathbf{x}}\frac{1}{r} + \frac{\cos\theta_{\mathbf{x}}}{r}\nabla_{\mathbf{x}}\cos\theta_{\mathbf{y}} + \frac{\cos\theta_{\mathbf{y}}}{r}\nabla_{\mathbf{x}}\cos\theta_{\mathbf{x}}, \qquad (74)$$

where $r = \|\mathbf{x}-\mathbf{y}\|$ and we assume without loss of generality that $\mathbf{x}$ is the origin and $\vec{\mathbf{n}}_{\mathbf{x}} = \binom{0}{1}$, as illustrated in Figure 23.

The gradients of the individual terms are:

$$\nabla_{\mathbf{x}}\mathbf{y} = \nabla_{\mathbf{x}}\begin{pmatrix}y_1\\ y_2\end{pmatrix} = \begin{pmatrix}-1\\ -1\end{pmatrix}, \qquad (75)$$

$$\nabla_{\mathbf{x}}\frac{1}{r^n} = -\frac{n}{r^{n+1}}\nabla_{\mathbf{x}} r = \frac{n}{r^{n+2}}\mathbf{y}, \qquad (76)$$

$$\nabla_{\mathbf{x}}\cos\theta_{\mathbf{x}} = \nabla_{\mathbf{x}}\frac{\vec{\mathbf{n}}_{\mathbf{x}}\cdot\mathbf{y}}{r} = \frac{1}{r}\nabla_{\mathbf{x}}(\vec{\mathbf{n}}_{\mathbf{x}}\cdot\mathbf{y}) + (\vec{\mathbf{n}}_{\mathbf{x}}\cdot\mathbf{y})\nabla_{\mathbf{x}}\frac{1}{r} = -\frac{\vec{\mathbf{n}}_{\mathbf{x}}}{r} + \frac{\cos\theta_{\mathbf{x}}}{r^2}\mathbf{y}, \qquad (77)$$

$$\nabla_{\mathbf{x}}\cos\theta_{\mathbf{y}} = -\nabla_{\mathbf{x}}\frac{\vec{\mathbf{n}}_{\mathbf{y}}\cdot\mathbf{y}}{r} = \frac{\vec{\mathbf{n}}_{\mathbf{y}}}{r} + \frac{\cos\theta_{\mathbf{y}}}{r^2}\mathbf{y}. \qquad (78)$$

Inserting Equations 75–78 with $n = 1$ into Equation 74 yields the expression in Equation 38.

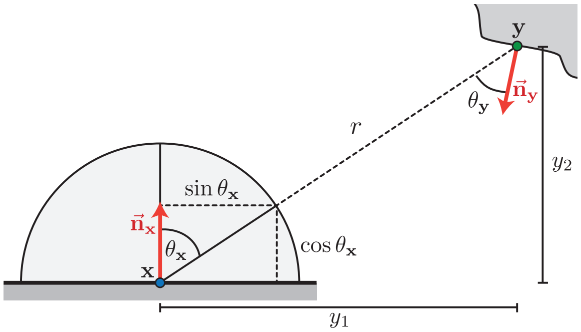

$\mathbf{y}$
$\theta_{\mathbf{y}}$
$\vec{\mathbf{n}}_{\mathbf{y}}$
r
y_2
$\sin\theta_{\mathbf{x}}$
$\vec{\mathbf{n}}_{\mathbf{x}}$
$\theta_{\mathbf{x}}$
$\cos\theta_{\mathbf{x}}$
$\mathbf{x}$
y_1

Fig. 23: The variables and geometric relationship involved in computing the translational derivative of the 2D and 3D geometry terms.

B. HESSIAN OF 2D GEOMETRY TERM

We obtain the Hessian of the geometry term by differentiating the terms in Equation 38:

$$\begin{aligned}
\mathbf{J}_{\mathbf{x}}\left(\frac{\cos\theta_{\mathbf{x}}\cos\theta_{\mathbf{y}}}{r^3}\mathbf{y}\right) &= (\cos\theta_{\mathbf{x}}\cos\theta_{\mathbf{y}}\mathbf{y})\,\nabla_{\mathbf{x}}^{\mathsf{T}}\frac{1}{r^3} + \left(\frac{\cos\theta_{\mathbf{x}}}{r^3}\mathbf{y}\right)\nabla_{\mathbf{x}}^{\mathsf{T}}\cos\theta_{\mathbf{y}} \\
&\quad + \left(\frac{\cos\theta_{\mathbf{y}}}{r^3}\mathbf{y}\right)\nabla_{\mathbf{x}}^{\mathsf{T}}\cos\theta_{\mathbf{x}} + \left(\frac{\cos\theta_{\mathbf{x}}\cos\theta_{\mathbf{y}}}{r^3}\right)\mathbf{J}_{\mathbf{x}}^{\mathsf{T}}(\mathbf{y}) \quad (79) \\
&= 5\frac{\cos\theta_{\mathbf{x}}\cos\theta_{\mathbf{y}}}{r^5}\mathbf{y}\mathbf{y}^{\mathsf{T}} - \frac{\cos\theta_{\mathbf{y}}}{r^4}\mathbf{y}\vec{\mathbf{n}}_{\mathbf{x}}^{\mathsf{T}} \\
&\quad + \frac{\cos\theta_{\mathbf{x}}}{r^4}\mathbf{y}\vec{\mathbf{n}}_{\mathbf{y}}^{\mathsf{T}} - \frac{\cos\theta_{\mathbf{x}}\cos\theta_{\mathbf{y}}}{r^3}\mathbf{I}_2, \quad (80) \\
\mathbf{J}_{\mathbf{x}}\left(\frac{\cos\theta_{\mathbf{y}}}{r^2}\vec{\mathbf{n}}_{\mathbf{x}}\right) &= 3\frac{\cos\theta_{\mathbf{y}}}{r^4}\vec{\mathbf{n}}_{\mathbf{x}}\mathbf{y}^{\mathsf{T}} + \frac{1}{r^3}\vec{\mathbf{n}}_{\mathbf{x}}\vec{\mathbf{n}}_{\mathbf{y}}^{\mathsf{T}}, \quad (81) \\
\mathbf{J}_{\mathbf{x}}\left(\frac{\cos\theta_{\mathbf{y}}}{r^2}\vec{\mathbf{n}}_{\mathbf{y}}\right) &= 3\frac{\cos\theta_{\mathbf{x}}}{r^4}\vec{\mathbf{n}}_{\mathbf{y}}\mathbf{y}^{\mathsf{T}} - \frac{1}{r^3}\vec{\mathbf{n}}_{\mathbf{y}}\vec{\mathbf{n}}_{\mathbf{x}}^{\mathsf{T}}, \quad (82)
\end{aligned}$$

where $\mathbf{J}_{\mathbf{x}}$ denotes the Jacobian operator taken with respect to $\mathbf{x}$. Combining these terms according to Equation 38 results in Equation 49.

C. GRADIENT OF 3D GEOMETRY TERM

Applying the product rule to the standard 3D geometry term yields nearly the same result as in 2D:

$$\nabla_{\mathbf{x}}G_{3D} = \cos\theta_{\mathbf{x}}\cos\theta_{\mathbf{y}}\nabla_{\mathbf{x}}\frac{1}{r^2} + \frac{\cos\theta_{\mathbf{x}}}{r^2}\nabla_{\mathbf{x}}\cos\theta_{\mathbf{y}} + \frac{\cos\theta_{\mathbf{y}}}{r^2}\nabla_{\mathbf{x}}\cos\theta_{\mathbf{x}}. \quad (83)$$

Inserting Equations 75–78 with $n = 2$ into Equation 83 yields the expression in Equation 58.

D. HESSIAN OF 3D GEOMETRY TERM

Differentiating the terms of the geometry gradient in Equation 58 gives:

$$\begin{aligned}
\mathbf{J}_{\mathbf{x}}\left(\frac{\cos\theta_{\mathbf{x}}\cos\theta_{\mathbf{y}}}{r^4}\mathbf{y}\right) &= (\cos\theta_{\mathbf{x}}\cos\theta_{\mathbf{y}}\mathbf{y})\nabla_{\mathbf{x}}^{\mathsf{T}}\frac{1}{r^4} + \left(\frac{\cos\theta_{\mathbf{x}}}{r^4}\mathbf{y}\right)\nabla_{\mathbf{x}}^{\mathsf{T}}\cos\theta_{\mathbf{y}} \\
&\quad + \left(\frac{\cos\theta_{\mathbf{y}}}{r^4}\mathbf{y}\right)\nabla_{\mathbf{x}}^{\mathsf{T}}\cos\theta_{\mathbf{x}} + \left(\frac{\cos\theta_{\mathbf{x}}\cos\theta_{\mathbf{y}}}{r^4}\right)\mathbf{J}_{\mathbf{x}}^{\mathsf{T}}(\mathbf{y}) \quad (84) \\
&= 6\frac{\cos\theta_{\mathbf{x}}\cos\theta_{\mathbf{y}}}{r^6}\mathbf{y}\mathbf{y}^{\mathsf{T}} - \frac{\cos\theta_{\mathbf{y}}}{r^5}\mathbf{y}\vec{\mathbf{n}}_{\mathbf{x}}^{\mathsf{T}} \\
&\quad + \frac{\cos\theta_{\mathbf{x}}}{r^5}\mathbf{y}\vec{\mathbf{n}}_{\mathbf{y}}^{\mathsf{T}} - \frac{\cos\theta_{\mathbf{x}}\cos\theta_{\mathbf{y}}}{r^4}\mathbf{I}_3, \quad (85) \\
\mathbf{J}_{\mathbf{x}}\left(\frac{\cos\theta_{\mathbf{y}}}{r^3}\vec{\mathbf{n}}_{\mathbf{x}}\right) &= 4\frac{\cos\theta_{\mathbf{y}}}{r^5}\vec{\mathbf{n}}_{\mathbf{x}}\mathbf{y}^{\mathsf{T}} + \frac{1}{r^4}\vec{\mathbf{n}}_{\mathbf{x}}\vec{\mathbf{n}}_{\mathbf{y}}^{\mathsf{T}}, \quad (86) \\
\mathbf{J}_{\mathbf{x}}\left(\frac{\cos\theta_{\mathbf{y}}}{r^3}\vec{\mathbf{n}}_{\mathbf{y}}\right) &= 4\frac{\cos\theta_{\mathbf{x}}}{r^5}\vec{\mathbf{n}}_{\mathbf{y}}\mathbf{y}^{\mathsf{T}} - \frac{1}{r^4}\vec{\mathbf{n}}_{\mathbf{y}}\vec{\mathbf{n}}_{\mathbf{x}}^{\mathsf{T}}, \quad (87)
\end{aligned}$$

which results in Equation 66 when combined according to Equation 58.

Received February 2011; accepted October 2011
March 2012