A First Order Analysis of Lighting, Shading, and Shadows

Ravi Ramamoorthi, Dhruv Mahajan and Peter Belhumeur

Columbia University

Abstract

The shading in a scene depends on a combination of many factors—how the lighting varies spatially across a surface, how it varies along different directions, the geometric curvature and reflectance properties of objects, and the locations of soft shadows. In this paper, we conduct a complete first order or gradient analysis of lighting, shading and shadows, showing how each factor separately contributes to scene appearance, and when it is important. Gradients are well suited for analyzing the intricate combination of appearance effects, since each gradient term corresponds directly to variation in a specific factor. First, we show how the spatial *and* directional gradients of the light field change, as light interacts with curved objects. This extends the recent frequency analysis of Durand et al. to gradients, and has many advantages for operations, like bump-mapping, that are difficult to analyze in the Fourier domain. Second, we consider the individual terms responsible for shading gradients, such as lighting variation, convolution with the surface BRDF, and the object's curvature. This analysis indicates the relative importance of various terms, and shows precisely how they combine in shading. As one practical application, our theoretical framework can be used to adaptively sample images in high-gradient regions for efficient rendering. Third, we understand the effects of soft shadows, computing accurate visibility gradients. We generalize previous work to arbitrary curved occluders, and develop a local framework that is easy to integrate with conventional ray-tracing methods. Our visibility gradients can be directly used in practical gradient interpolation methods for efficient rendering.

1 Introduction

A theoretical analysis of lighting and shading has many applications in forward and inverse rendering. For example, understanding where the image intensity varies rapidly can be used to determine non-uniform image sampling rates for efficient rendering. Understanding how shading changes in penumbra regions can lead to efficient and robust soft shadow computations, as well as advances in inverse lighting-from-shadow algorithms. In this paper, we seek to address these theoretical questions through a first order or gradient analysis of lighting, shading and visibility.

The appearance of a surface, and its gradients, depends on many factors. The shading is affected by lighting—the *spatial* lighting variation over a flat object surface due to close sources, as well as the *angular* variation in lighting at a point from different directions. Shading also depends on geometric effects like the object's curvature, which determines how much the surface normal or orientation changes between neighboring points. The material properties are also important, since shading is effectively a convolution with the object BRDF [Ramamoorthi and Hanrahan 2001]. These factors can combine in complex ways in an image, and each factor may have less or more importance depending on the situation. For example, the *spatial* variation in lighting over a surface can be primarily responsible for the specular reflections from a near source on a glossy flat table. On the other hand, the *angular* variation in lighting is most important for a highly curved bumpy object on the table—the effect of spatial variation here is often small enough, that the lighting can effectively be treated as distant (see Fig. 5).

By definition, the gradient is usually a sum of terms, each corresponding to variation in a specific factor. Hence, a first order analysis is able to isolate the impact of various shading effects. Our computation of gradients also enables new practical rendering algorithms, such as efficient gradient-based image sampling, and fast and accurate gradient-based interpolation of visibility (Fig. 1). Specifically, we make the following contributions:

Analysis of Light Reflection: First, we analyze the basic conceptual steps in the reflection of light from a curved surface (Sec. 4.1). We develop the theory for both *spatial* and *angular* (or directional) gradients of the light field, since many visual effects involve a rich interplay between spatial and angular information.

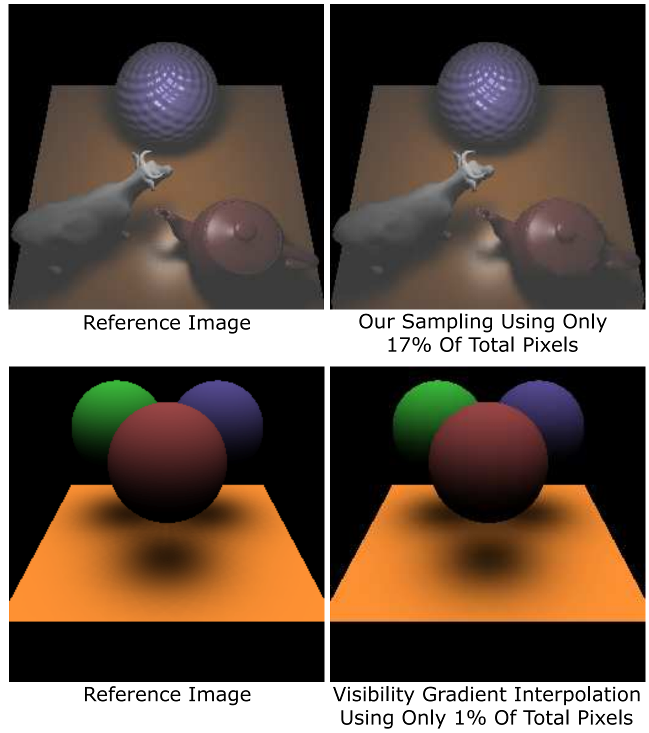


Figure 1: *Our theoretical analysis can be applied to efficient rendering.* **Top:** *Gradient-based image sampling achieves a 6× speedup on a scene with bumpy, diffuse and specular objects, including shadows and near-field lighting.* **Bottom:** *We use visibility gradients for rendering accurate soft shadows from curved objects on the ground plane, evaluating visibility explicitly at only 1% of image pixels. More details are in Figs. 8 and 15.*

Analysis of First Order Terms and Image Sampling: In Sec. 4.3, we determine the gradients for shading on a curved object lit by general spatial and directionally-varying lighting. We combine the basic shading steps in Sec. 4.1, augmenting them with *non-linear* transformations like *bump mapping* (Sec. 4.2). Our final gradient formula can be separated into individual terms that correspond to effects like spatial lighting variation, angular variation, and surface curvature. We analyze the effects of these terms in a variety of situations, understanding which factors are important for the appearance of different scenes (Sec. 4.5). Moreover, we show how to extend the first order analysis to second-order Hessians (Sec. 4.7).

Section 5 (Figs. 1 and 8) applies these ideas to efficient rendering, by adaptively sampling images using a metric based on gradient magnitude. We consider general scenes, with bump maps, glossy reflectance, shadows, and near-field lighting, achieving accurate results using only $10\% - 20\%$ of the effective pixels.

Analysis of Visibility Gradients: We derive new analytic expressions for soft shadow gradients in Secs. 6 and 7. These have usually been neglected in practical gradient techniques [Ward and Heckbert 1992; Annen et al. 2004]. Our analysis is general, and works for arbitrary curved blockers, as well as polygonal objects. Moreover, our formulation is local, based only on analyzing angular discontinuities in visibility at a single spatial location. We demonstrate practical applications to efficient and accurate rendering of soft shadows using gradient-based interpolation (Figs. 1 and 15).

2 Previous Work

This paper builds on a substantial body of previous work on analyzing light transport in a number of different representations.

Operator	Spatial and Angular Domain	Gradient	Fourier	Wavelet
Multiplication	$h(x,\theta) = f(x,\theta)g(x,\theta)$	$\bigtriangledown \mathbf{h} = f\bigtriangledown \mathbf{g} + g\bigtriangledown \mathbf{f}$	$H(\Omega) = F(\Omega)\otimes G(\Omega)$	$H_i = C_{ijk}F_jG_k$
Integration	$h(x) = \int f(x,\theta)\,d\theta$	$h_x = \int f_x(x,\theta)\,d\theta$	$H(\Omega_x) = F(\Omega_x, 0)$	Haar only, $H_i = F_{i0}$
Convolution	$h(x,\theta) = \int f(x,\omega)g(\theta-\omega)\,d\omega$	$\bigtriangledown \mathbf{h} = \bigtriangledown \mathbf{f}\otimes g$	$H(\Omega) = F(\Omega)G(\Omega)$	No simple formula
Linear Transforms	$h(\mathbf{u}) = f(M\mathbf{u})$	$\bigtriangledown \mathbf{h}(\mathbf{u}) = M^T\bigtriangledown \mathbf{f}(M\mathbf{u})$	$H(\Omega) = \frac{1}{\lvert det(M)\rvert}F(M^{-T}\Omega)$	No simple formula
Nonlinear Transforms	$h(\mathbf{u}) = f(T(\mathbf{u}))$	$\bigtriangledown \mathbf{h}(\mathbf{u}) = J^T(\mathbf{u})\bigtriangledown \mathbf{f}(T(\mathbf{u}))$	No simple formula	No simple formula

Figure 2: *The basic mathematical operators of light transport, and the resulting transformations in gradient, Fourier and wavelet representations.*

Frequency Domain Analysis: Frequency domain techniques have been popular for light field analysis, leading to a signal-processing approach. Chai et al. [2000] analyze light field sampling in the Fourier domain. Ramamoorthi and Hanrahan [2001] develop a convolution framework for reflection on curved surfaces using spherical harmonics. Ng [2005] has shown how Fourier analysis can be used to derive a slice theorem for light fields.

Most recently, and closest to our work, Durand et al. [2005] derive a frequency analysis of light transport considering both spatial and angular variation. In Sec. 3.1 (Fig. 2), we directly compare Fourier and gradient analysis in terms of basic mathematical operators (Sec. 8, at the end, has a more detailed discussion of specific steps.) First order analysis has two main benefits for us. The gradient is naturally written as a sum of terms corresponding to specific variations in shading, while keeping other factors fixed. This makes it easier to analyze the importance of various shading effects. Moreover, first order analysis is by definition fully local and can handle general non-linear effects like bump mapping, while Fourier analysis always requires a finite neighborhood and linearization.

Wavelet Analysis: Wavelets have been another popular tool for efficient computations and representation of light transport. Early work in rendering includes wavelet radiosity [Gortler et al. 1993; Gershbein et al. 1994]. More recently, Ng et al. [2004] have analyzed multiplication and triple product integrals using wavelets. However, many of the mathematical operations of light transport currently have no simple analytic interpretation in wavelets (see Sec. 3.1 and Fig. 2). Thus, wavelets seem more useful for efficient practical computation, rather than for deriving theoretical insights.

Differential and Gradient Analysis: Gradient-based methods have been widely used in graphics, starting with the irradiance gradients of Ward and Heckbert [1992]. While we are inspired by these methods, there are some important differences. While Ward and Heckbert [1992] essentially try to find the gradients of the incident light field, we seek to determine how these gradients evolve as light interacts with object surfaces or blockers. Igehy [1999] and Chen and Arvo [2000] find differentials of individual ray paths as certain parameters (like viewpoint or location on the image plane) vary. By contrast, we seek to determine how the gradients over the entire light field transform. Most importantly, this paper is focused more on theoretical analysis, understanding the nature of shading variation by considering the various gradient terms. We are optimistic that our analysis can be used to derive new theoretical bounds and practical algorithms for previous methods.

Visibility Analysis: Shadows are one of the most important visual features and have received significant theoretical attention. Durand et al. [2002] develop a full characterization of visibility events in terms of the visibility complex. Soler and Sillion [1998] and Ramamoorthi et al. [2004] have characterized special cases as convolutions. Arvo [1994] has derived irradiance Jacobians for occluded polyhedral scenes, and applied them to shadow computations based on a global analysis of scene configuration. Holzschuch and Sillion [1998] compute gradients and Hessians of form factors for error analysis. By contrast, our approach is local, using only the visibility information at a single spatial location, and can consider general curved occluders in general complex lighting.

3 Preliminaries

We start by writing the reflection equation at a single point x,

$$B(x,\theta) = \int L(x,\omega)\rho(x,\theta,\omega)V(x,\omega)\cos\omega\,d\omega, \qquad (1)$$

where B is the reflected light field, $L(x,\omega)$ is the incident light field, ρ is the BRDF and V is the visibility. In this paper, light fields such as B or L are expressed in terms of their spatial location x and local angular direction (ω or θ), with respect to the local surface normal.

Our goal is a first order analysis of reflection on a curved surface. We consider *both* spatial and angular gradients, because most physical phenomena involve deep interplay between spatial and angular effects. For example, angular variation in the lighting often leads to spatial variation in the shading on a curved object.

For much of the paper, the derivations are carried out in the 2D plane or flatland for clarity and simplicity. While the 3D extensions (detailed in Secs. 4.4, 7 and Appendix B) are more complicated algebraically, much the same results are obtained. Our analysis is applied practically to efficient rendering of 3D scenes (Sec. 5), and to evaluation of soft shadows from curved blockers in 3D (Sec. 7).

We will be analyzing various parts and generalizations of Equation 1. In this section, we will consider abstractly the result h of the interaction of two functions f and g, which will usually correspond to the lighting and BRDF respectively. From Sec. 4 onwards, we will be more concrete, using notation closer to Equation 1. The partial derivatives will be denoted with subscripts—for example, $f_x(x,\omega) = \partial f(x,\omega)/\partial x$. In Sec. 3.1, we will also compare the first order analysis to Fourier analysis, such as [Durand et al. 2005], presenting a unified framework for both in terms of basic mathematical operations. Sec. 8 at the end of the paper has a more specific discussion and comparison with examples. We denote the Fourier transform of $f(x,\theta)$ as $F(\Omega_x,\Omega_\theta)$, where the subscripts now stand for the spatial (x) or angular (θ) coordinate.

3.1 Mathematical Operations of Light Transport

The interaction of lighting with the reflectance and geometry of objects involves fairly complex effects on the light field, as well as the gradients or Fourier spectra. However, the basic shading steps can all be reduced to five basic mathematical building blocks—multiplication, integration, convolution of functions, and linear and nonlinear transformations on a function's domain. For example, modulation of the shading by a texture map involves multiplication. Adding up the contributions of lighting from every incident direction involves integration. The interaction of lighting and reflectance can usually be written as a convolution with the surface BRDF. We will see that transformations between a global coordinate frame and the local frame of the surface can be written as linear transformations of the spatial and angular coordinates. Complex shading effects like general bump mapping, and visibility computations require nonlinear transformations of the coordinates.

Figure 2 summarizes these mathematical operators for gradient, Fourier and wavelet representations. While many of these formulae are widely available in calculus textbooks, their forms give considerable insight in comparing analysis with different representations.

Multiplication: Canonically, $h(x,\theta) = f(x,\theta)g(x,\theta)$. In the Fourier basis, this is a convolution, $H(\Omega) = F(\Omega)\otimes G(\Omega)$, where the $\otimes$ symbol stands for convolution. For gradients,

$$\bigtriangledown \mathbf{h} = f\bigtriangledown \mathbf{g} + g\bigtriangledown \mathbf{f}. \qquad (2)$$

Integration: Consider $h(x) = \int f(x,\theta)\,d\theta$, where for example f may denote the lighting pre-multiplied by the cosine term (with the result $h(x)$ being the diffuse shading). After a Fourier transform, this corresponds to restricting ourselves to the $\Omega_\theta = 0$ line, i.e. Ω_x axis, so $H(\Omega_x) = F(\Omega_x, 0)$. For first order analysis,

$$h_x = \int f_x(x,\theta)\,d\theta. \qquad (3)$$

Convolution: Canonically, $h(x,\theta) = \int f(x,\omega)g(\theta-\omega)\,d\omega$, where f can be thought of as the incident lighting and g as the homogeneous radially symmetric BRDF. In the Fourier basis, this becomes a multiplication, $H(\Omega) = F(\Omega)G(\Omega)$. For gradient analysis, it is convenient to realize that convolution is a symmetric operation.[1] Thus, derivatives and convolutions commute, so that

$$h = f \otimes g \;\Rightarrow\; \bigtriangledown\mathbf{h} = \bigtriangledown\mathbf{f} \otimes g, \tag{4}$$

where the convolution is only over the angular coordinate.
Linear Transformations: In general, we consider

$$h(\mathbf{u}) = f(M\mathbf{u}), \tag{5}$$

where $\mathbf{u}$ is a $n \times 1$ vector and M is a $n \times n$ matrix. In 2D, the light field has two dimensions, so $n = 2$ and $\mathbf{u} = (x, \theta)^T$. For example, f could be the incident light field in global coordinates, and h could be the lighting in the local coordinate frame of a point, with M being the appropriate transformation of $\mathbf{u} = (x, \theta)^T$.

For Fourier analysis, we can use the general Fourier linear transformation theorem. While the derivation is straightforward, it does not appear to be commonly found in standard texts or well known in the field, so we briefly derive it in Appendix A,

$$H(\Omega) = \frac{1}{\mid det(M) \mid} F(M^{-T}\Omega), \tag{6}$$

where $det(M)$ is the determinant of M.

For gradients, we have a similar linear transformation theorem (also derived in Appendix A). In particular,

$$\bigtriangledown\mathbf{h}(\mathbf{u}) = M^T \bigtriangledown\mathbf{f}(M\mathbf{u}). \tag{7}$$

Nonlinear Transformations: Finally, we come to nonlinear transformations. These are seldom considered in analyses of light transport (and are not treated by Durand et al. [2005] at all), because it is not clear how to handle them with Fourier or wavelet methods.

To apply gradient techniques, we effectively use the chain rule. We assume $h(\mathbf{u}) = f(T(\mathbf{u}))$, where T is a general non-linear and not necessarily invertible transformation. However, T can be locally linearized by computing the Jacobian, to obtain a local linear transformation matrix $J(\mathbf{u})$ (that now depends on $\mathbf{u}$),

$$h(u_i) = f(T_i(\mathbf{u})) \quad J_{ik}(\mathbf{u}) = \frac{\partial T_i}{\partial u_k} \quad \bigtriangledown\mathbf{h}(\mathbf{u}) = J^T(\mathbf{u})\bigtriangledown\mathbf{f}(T(\mathbf{u})). \tag{8}$$

Implications: Besides relating Fourier and gradient techniques, direct application of these formulae simplifies many derivations both in our paper and in previous work. For example, many derivations in [Durand et al. 2005] follow directly from the Fourier linear transformation theorem. The Fourier slice result in [Ng 2005] can be easily derived using a combination of the linear transformation and integration relations. Figure 2 also indicates why certain representations are more commonly used for mathematical analysis. The Fourier basis handles the first four basic operations in a very simple way, making it possible to conduct a full analysis of linear light transport, such as Durand et al. [2005]. Similarly, the simple form of those operations with gradients makes them well suited to the analysis in this paper. Moreover, gradients are often the only available tool when considering nonlinear transformations, for which there is no simple Fourier equivalent. For wavelets, on the other hand, most operations like convolution or linear transforms are very difficult to study analytically (even though there are often efficient computational methods, such as the recent triple product multiplication algorithms [Ng et al. 2004; Clarberg et al. 2005]).

4 Light Reflection from Curved Surfaces

In this section, we first discuss the important conceptual steps for reflection from a homogeneous curved object (with a brief digression to consider spatially-varying materials and analysis in 3D). Then, we analyze non-linear transformations like normal or bump maps, and derive the combined gradient including all effects. Finally, we analyze the effects of individual shading terms, and the sampling of images. In this section, we do not explicitly consider cast shadows, since visibility is analyzed in detail in Secs. 6 and 7.

[1] By symmetry, $h_\theta = f_\theta \otimes g$ in Equation 4 is the same as $h_\theta = f \otimes g_\theta$. This symmetry no longer holds for 3D spherical convolution, where the lighting is a 2D spherical function, while the radially symmetric BRDF is 1D. In that case, we must use $f \otimes g_\theta$ (see Appendix B). However, Equation 4 is still accurate for flatland, and can be used even for 3D sampling.

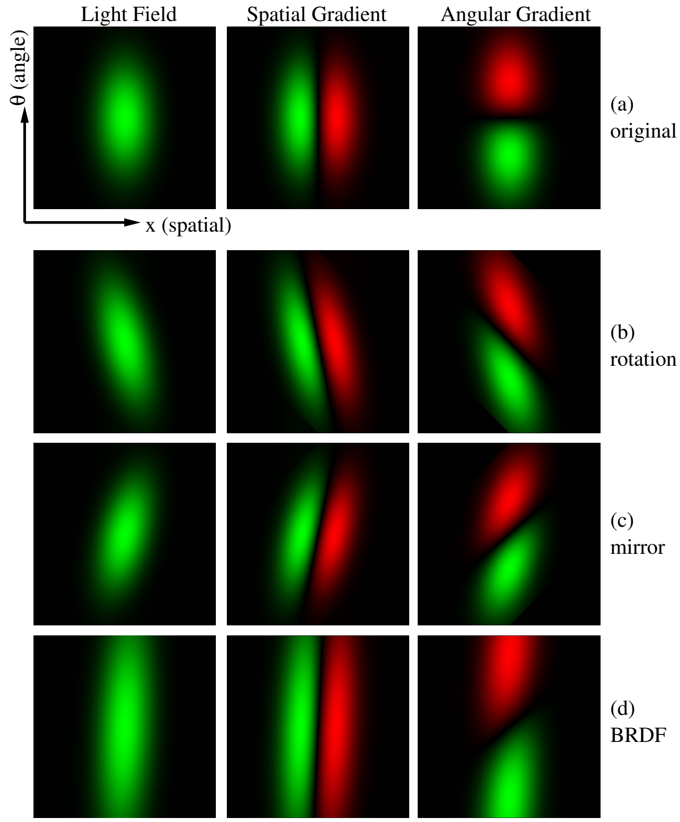


Figure 3: *The light field and its spatial and angular gradients, as a result of the various curved surface shading steps in Sec. 4.1. Green denotes positive values, and red denotes negative values.*

4.1 Basic Shading Steps

To illustrate our ideas, we start with a spatially and directionally-varying light source, showing how the light field and gradients change with various shading steps. As shown in Fig. 3a, the source intensity $L(x, \theta)$ varies as a Gaussian along both spatial (horizontal) and angular (vertical) axes. Besides providing a simple didactic example, one motivation is to consider spatially *and* directionally-varying sources, that have rarely been studied.

We assume the global coordinate frame is aligned so the surface normal at the origin $x = 0$ is pointing straight up (towards $\theta = 0$). The surface is parameterized by the arc-length distance x along it (which is equivalent to the global x coordinate near $x = 0$ and used interchangeably). We linearize the surface about $x = 0$, so that the normal is given by kx, where k is the standard geometric curvature, and we use positive signs for counter-clockwise directions.

Step 1—Per-Point Rotation into Local or Surface Frame: We must perform a rotation at each point to convert global coordinates to local. Let $L(x, \theta)$ be the incident light field in the global frame. As shown in previous work [Ramamoorthi and Hanrahan 2001; Durand et al. 2005], the light field in the local or surface coordinate frame is $L^s(x, \theta) = L(x, \theta + n)$, where n is the surface normal. Noting that $n = kx$, we write $L^s(x, \theta) = L(x, \theta + kx)$. This is a linear transformation of the variables x and θ, that mixes spatial and angular coordinates, shearing the light field along the angular dimension as seen in Fig. 3b. If $\mathbf{u} = (x, \theta)^T$, $L^s(\mathbf{u}) = L(M\mathbf{u})$ with M being

$$\begin{pmatrix} x_{new} \\ \theta_{new} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ k & 1 \end{pmatrix} \begin{pmatrix} x \\ \theta \end{pmatrix} \quad M = \begin{pmatrix} 1 & 0 \\ k & 1 \end{pmatrix} \; M^T = \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix}. \tag{9}$$

Using the linear transformation theorem in Equation 7, and premultiplying by M^T as required,

$$\begin{aligned} L^s(x, \theta) &= L(x, \theta + kx) \\ \bigtriangledown\mathbf{L^s}(x, \theta) &= \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix} \bigtriangledown\mathbf{L}(x, \theta + kx). \end{aligned} \tag{10}$$

This can be written out explicitly as

$$\begin{pmatrix} L^s_x \\ L^s_\theta \end{pmatrix} = \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix} \begin{pmatrix} L_x(x,\theta+kx) \\ L_\theta(x,\theta+kx) \end{pmatrix} = \begin{pmatrix} L_x(x,\theta+kx)+k\cdot L_\theta(x,\theta+kx) \\ L_\theta(x,\theta+kx) \end{pmatrix}, \quad (11)$$

which can be easily verified by differentiating L^s directly, and where we have made the arguments for evaluation explicit. As seen in Fig. 3b, the spatial and angular gradients are sheared in the angular dimension like the light field, because all quantities are evaluated at the sheared coordinates $(x, \theta+kx)$.

From the above equation, the angular gradients L^s_θ have the same form as L_θ. The spatial gradient L^s_x makes explicit that shading variation occurs in two ways—either the incident light field includes spatially varying components L_x, and/or the surface has curvature k (and there is angular lighting variation L_θ). For a distant environment map (so L is independent of x), there is no spatial variation ($L_x = 0$), and L^s_x is only due to curvature. For a flat surface, there is no curvature (and in fact, $L^s = L$ for this step), and spatial gradients only come from the original light field. We can also see how to relate the two components, which have comparable magnitude when $| L_x | \sim | kL_\theta |$. This discussion also immediately shows the benefit of first order analysis, where individual gradient terms correspond directly to different types of shading variation.

Cosine Multiplication: We can now multiply by the cosine term, with the standard multiplication formula for the gradients (Equation 2). Since the cosine effect is relatively subtle and often rolled into Phong-like BRDFs, we will simply incorporate it in the BRDF transport function for the combined analysis in Sec. 4.3.

Step 2–Mirror Reparameterization: For glossy materials, we reparameterize by the mirror direction, setting $L^m(x,\theta) = L^s(x,-\theta)$. The light field and gradients in Fig. 3c are therefore reflected about the $\theta-$axis. The angular gradient is also negated,

$$L^m_\theta(x,\theta) = -L^s_\theta(x,-\theta), \quad (12)$$

or more formally,

$$\begin{aligned} L^m(x,\theta) &= L^s(x,-\theta) \\ \bigtriangledown \mathbf{L^m}(x,\theta) &= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \bigtriangledown \mathbf{L^s}(x,-\theta). \end{aligned} \quad (13)$$

Step 3—BRDF Convolution: Reflection from the surface can be written as a convolution with a radially symmetric BRDF [2] ρ,

$$B^s(x,\theta) = L^m \otimes \rho = \int L^m(x,\omega)\rho(\omega-\theta)\,d\omega. \quad (14)$$

For gradients, we use the gradient convolution rule in Equation 4,

$$\bigtriangledown \mathbf{B^s}(x,\theta) = \bigtriangledown \mathbf{L^m} \otimes \rho = \int \bigtriangledown \mathbf{L^m}(x,\omega)\rho(\omega-\theta)\,d\omega. \quad (15)$$

Since gradients and convolutions commute, we effectively obtain gradients of the convolution by convolving the gradients,

$$\begin{pmatrix} B^s_x \\ B^s_\theta \end{pmatrix} = \begin{pmatrix} L^m_x \otimes \rho \\ L^m_\theta \otimes \rho \end{pmatrix}. \quad (16)$$

Figure 3d shows the results of convolving with a Gaussian for ρ. This is analogous to a Phong or an approximate Torrance-Sparrow BRDF. We would expect the convolution to lead to some blurring along the vertical, or angular direction, and this is in fact the case for *both* the light field, and the spatial and angular gradients.

Step 4—Inverse Per-Point Rotation into Global Frame: So far, we have worked in the local or surface coordinate frame (hence, the superscript s on the reflected light field B^s). If we seek to express the final result in the global frame, we should undo the original per-point rotation, writing, analogous to Equation 10,

$$\begin{aligned} B(x,\theta) &= B^s(x,\theta-kx) \\ \bigtriangledown \mathbf{B}(x,\theta) &= \begin{pmatrix} 1 & -k \\ 0 & 1 \end{pmatrix} \bigtriangledown \mathbf{B^s}(x,\theta-kx). \end{aligned} \quad (17)$$

[2] We use $\rho(\omega-\theta)$ instead of $\rho(\theta-\omega)$ for algebraic simplicity in Sec. 4.3. Since the BRDF is symmetric, this does not matter, and is actually more consistent with our sign conventions.

Spatially Varying Materials: As a brief aside, we consider a generalization of step 3 to spatially-varying materials.[3] In this case,

$$B^s(x,\theta) = \int L^m(x,\omega)\rho(x,\omega-\theta)\,d\omega. \quad (18)$$

Note that the convolution is only over the angular coordinates, while L^m and ρ are multiplied over the spatial coordinates. The gradients are given by

$$\begin{pmatrix} B^s_x \\ B^s_\theta \end{pmatrix} = \begin{pmatrix} L^m_x \otimes \rho + L^m \otimes \rho_x \\ L^m_\theta \otimes \rho \end{pmatrix}. \quad (19)$$

The only additional term is $L^m \otimes \rho_x$ in B^s_x, which corresponds to the spatial gradient or texture in the BRDF.

An interesting special case is texture mapping, where $\rho(x)$ simply multiplies the diffuse shading. In that case, we denote E as the irradiance $\int L^s(x,\omega)\,d\omega$ so that $B^s(x) = E(x)\rho(x)$ and

$$B^s_x = E_x\rho + E\rho_x. \quad (20)$$

For smooth objects, the diffuse shading is low frequency [Ramamoorthi and Hanrahan 2001], so E_x is generally small and $B^s_x \sim E\rho_x$. (A similar result holds even in 3D, with $\bigtriangledown \mathbf{B^s} \sim E \bigtriangledown \rho$. In 3D, the direction of the gradient $\bigtriangledown \mathbf{B^s}$ depends primarily on the direction of the texture gradient $\bigtriangledown \rho$, independent of the lighting or irradiance, while the magnitude is scaled by E. This is one explanation why the direction of the gradient is a good measure for lighting-insensitive recognition in computer vision [Chen et al. 2000].)

Analysis in 3D: In Appendix B, we extend the four basic shading steps to 3D. This requires simple vector calculus and differential geometry. While the algebra is more complex, we obtain very similar results as in the 2D or flatland case. For example, the curvature k simply corresponds to the principal curvatures in 3D. In fact, as we will see in Sec. 4.4, it is possible and simpler to directly use the straightforward 3D analogs of these 2D results for real images.

4.2 Gradients for Normal or Bump Maps

Section 4.1 assumes a local linearization of the surface. We now generalize to arbitrary normal or bump maps, which are nonlinear transformations. In this case, the per-point rotation step involves a general function $n(x)$ for the normal. By differentiating, using the chain rule (or using Equation 8, with the Jacobian of the transform),

$$\begin{aligned} L^s(x,\theta) &= L(x,\theta+n(x)) \\ \bigtriangledown \mathbf{L^s}(x,\theta) &= \begin{pmatrix} 1 & n_x \\ 0 & 1 \end{pmatrix} \bigtriangledown \mathbf{L}(x,\theta+n(x)), \end{aligned} \quad (21)$$

where $n_x = \partial n/\partial x$. Hence, we can define a general per-point curvature, $k(x) = n_x = \partial n/\partial x$, assuming an arc-length parameterization. For normal maps $n(x) = \hat{n}(x) + n_0(x)$, where $\hat{n}$ is the bump map, and $n_0(x)$ is the base normal of the surface. Assuming the bump map has much higher frequencies than the base surface, $k(x) \approx \partial\hat{n}/\partial x$, and depends primarily on the curvature of the bump map. If there is no bump map, $k(x)$ is simply the curvature of the base surface $\partial n_o/\partial x$. The use of the gradient analysis lets us generalize to bump maps very easily, with the general function $k(x) = n_x = \partial n/\partial x$ simply taking the place of k in Equation 10.

4.3 Light Field Gradients

We now combine the four light-surface interaction steps in Sec. 4.1, replacing kx with $n(x)$. From Equations 17 and 14,

$$B(x,\theta) = B^s(x,\theta-n(x)) = \int L^m(x,\omega)\rho(\omega-\theta+n(x))\,d\omega. \quad (22)$$

Upon substituting Equations 13 and 10 for L^m, we obtain $L^m(x,\omega) = L^s(x,-\omega) = L(x,-\omega+n(x))$. Hence,

$$\begin{aligned} B(x,\theta) &= \int L(x,-\omega+n(x))\rho(\omega-\theta+n(x))\,d\omega \\ &= \int L(x,\omega')\rho(2n(x)-\theta-\omega')\,d\omega', \end{aligned} \quad (23)$$

[3] We could also generalize the BRDF model beyond radially symmetric. The gradients would be essentially the same, but with the convolutions replaced by a general integral using the general BRDF $\rho(x,\theta,\omega)$.

	SV $L_x \otimes \rho$	DV $L_\theta \otimes \rho$	CV $2n_x$	CDV $2n_x(L_\theta \otimes \rho)$	SV+CDV B_x
Distant Light	0				CDV
Overcast Sky	0	low		low	low
Near Light	high				SV+CDV
Bump Map			high	high	CDV
Flat Surface			0	0	SV
Flat+Envmap	0		0	0	0
Lambertian		low		low	SV+CDV
Mirror		high			CDV

Figure 4: *Magnitudes of various light field gradient terms, corresponding to a variety of common situations and special cases. Entries not filled in have "normal" values, depending on the specific lighting and BRDF.*

where we set $\omega' = n(x) - \omega$, and we end up with a standard convolution, but evaluated at the "reflected outgoing direction," given by $\theta_r = 2n(x) - \theta$, as one might expect.

Upon making similar substitutions for the gradients (Equations 10, 13, 16 and 17), and combining the linear transforms,

$$\begin{aligned} \triangledown \mathbf{B} &= \begin{pmatrix} 1 & -n_x \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 & n_x \\ 0 & 1 \end{pmatrix} \times \\ & \int \triangledown \mathbf{L}(x, -\omega + n(x)) \rho(\omega - \theta + n(x))\, d\omega \qquad (24) \\ &= \begin{pmatrix} 1 & 2n_x \\ 0 & -1 \end{pmatrix} \int \triangledown \mathbf{L}(x, \omega') \rho(2n(x) - \theta - \omega')\, d\omega'. \end{aligned}$$

Now, we can write down explicitly, using $\theta_r = 2n(x) - \theta$ for the reflected direction, and $\otimes$ for convolution,

$$\boxed{\begin{aligned} B_x(x,\theta) &= \quad (L_x \otimes \rho)(x, \theta_r) + 2 \cdot n_x \cdot \left(L_\theta \otimes \rho\right)(x, \theta_r) \\ B_\theta(x,\theta) &= -\left(L_\theta \otimes \rho\right)(x, \theta_r). \end{aligned}} \qquad (25)$$

This is an overall formula for shading gradients on a curved surface. While the initial derivation in Sec. 4.1 assumed the global coordinate frame was aligned with the surface at $x = 0$, and used a linearization of the surface as a conceptual tool, the final formula is completely local, as expected for gradient analysis. We only need the geometric curvature n_x at a point, and the spatial and angular gradients of the incident light L_x and L_θ, expressed in the local coordinate or tangent frame—where x is a local arc-length parameterization of the surface. We have verified these results for a number of flatland scenes, with analytic examples and numerical evaluation.

For simplicity, we focus on homogeneous objects in this section. However, incorporating spatial BRDF variation is straightforward. First, consider the common case when ρ is a product of the current angular BRDF, and a spatially-varying texture which simply multiplies the final result. We have already studied texture mapping in Equation 20. The spatial gradient B_x involves a modulation of Equation 25 by the texture, and an additional term corresponding to the texture gradient modulated by the image intensity from Equation 23. This latter term can dominate in regions of large texture gradients and corresponds to the observation that high-frequency texture often masks slow shading variations. General spatially-varying BRDFs require a generalization of the BRDF convolution in step 3, as in Equation 19 of Sec. 4.1. The only additional term in Equation 25 is $(L \otimes \rho_x)(x, \theta_r)$, in the spatial gradient B_x.

4.4 Direct Extension to 3D

While our derivations have been in 2D, one can directly use the 3D analogs of these results for many rendering applications and analyses. A formal, accurate 3D derivation of the shading steps is given in Appendix B and is seen to have a very similar form.

To directly extend Equation 25 to 3D, we interpret the convolutions $\otimes$ as 3D convolutions of lighting gradients and the BRDF, over the full sphere of incident lighting directions. The 2D curvature n_x is simply replaced by the Gaussian curvature of the surface. For practical computations, the incident light field's spatial and angular gradients (corresponding to L_x and L_θ) can be determined analytically where possible, or numerically otherwise, and usually relate directly to the variation in intensity of the light sources.

Consider the spatial gradient B_x in 2D. In 3D, we will have two such expressions B_x and B_y. For the gradient magnitude visualizations in Sec. 4.5 or the non-uniform image sampling in Sec. 5, we consider the net magnitude $(B_x^2 + B_y^2)^{1/2}$. These magnitudes are independent of which specific (orthogonal) directions are picked for the axes x and y. For the angular gradients, we treat the direction θ as a unit vector, with B_θ corresponding to two gradients along the directions in the tangent plane to θ. Finally, we consider the net magnitude of these angular gradients in Secs. 4.5 and 5.

4.5 Implications: Analysis of Gradient Terms

We now discuss some implications of Equation 25. Figure 4 shows a number of common situations. To aid our discussion, we label the important terms. We refer to $L_x \otimes \rho$ as the *spatial variation (SV)* term in the lighting. Analogously, $L_\theta \otimes \rho$ is the *directional variation (DV)* term—the directional variation in the reflected light field B_θ is essentially the same as the incident DV. We refer to $2n_x$ as the *curvature (CV)* term, and the product $2n_x(L_\theta \otimes \rho)$ as the *curvature directional variation (CDV)* term. Spatial gradients in the reflected light field B_x are a combination of *SV* and *CDV* terms.

We first describe how various factors (lighting, geometry and materials) affect shading gradients. Fig. 4 summarizes our insights. Then, we use a simple 3D scene to illustrate some of these effects.

Lighting: In distant lighting, there is no spatial lighting variation SV ($L_x = 0$), and spatial gradients B_x are due solely to the curvature and angular lighting variation (CDV). If the environment itself varies little (low DV, small $|L_\theta|$), such as an overcast sky, we get soft shading effects with little spatial variation ($|B_x|$ is small). On the other hand, for a near light source, there is significant spatial variation (large L_x), and both SV and CDV must be considered.

Geometry: A bump-mapped surface has high curvature, so the directional term CDV will be large, and the main contributor to B_x. On the other hand, a flat surface has no curvature, so the CDV term vanishes, and only the spatial variation L_x in the lighting can induce shading changes. A particularly interesting special case is a flat surface in a distant environment map. In this case, we get uniform shading across the surface, and indeed $B_x = 0$.

BRDF: Material properties can also affect the results. For a Lambertian object (or the diffuse lobe of a general material), the BRDF ρ is a low-pass filter that causes the directional shading DV to be low-frequency and smooth. Hence, strong spatial gradients in the lighting (the SV term) can often be important to the overall shading. Moreover, we know that sharp edges cannot come from the DV term, and will either be at geometric discontinuities (very high curvature) or because of strong spatial variation in lighting. On the other hand, for a mirror surface, like a chrome-steel sphere often used to estimate the illumination, we will see the full directional variation in the lighting, and DV will be high.

We can also make some quantitative statements. The spatial term SV and directional CDV will be of roughly the same magnitude when $|L_x| \sim 2|n_x||L_\theta|$. This allows a concept like "far" lighting to formalized as $|L_x| \ll 2|n_x||L_\theta|$. In the simple case when the near light source(s) is isotropic and at a distance d, from trigonometry, $L_x \approx L_\theta/d$ so the condition for far lighting becomes $1/d \ll 2|n_x|$, which relates the distance of the lighting to the surface curvature. This criterion depends on the curvature—a light source that is far for a bump-mapped object may not be classified as far for a flat table. One application is efficient rendering approximation, where light sources could be treated as distant for bump-mapped or other high-curvature surfaces, while being modeled exactly in flat regions based on these criteria. There are similar applications for inverse problems and perception—it will be relatively easier to estimate near-field lighting effects from flatter objects than curved surfaces.

We illustrate some of these ideas with a simple didactic 3D scene in Fig. 5 that includes a nearly (but not with zero curvature) flat table on which sit a diffuse, diffuse+glossy, and bumpy sphere. The scene is lit by a moderately close area source. We use the direct 3D analogs of the 2D gradients, as discussed in Sec. 4.4.

The gradient magnitudes are visualized on a log scale in Figs. 5b-5e. The spatial gradient of the (moderately near) lighting (b) can be large, and is primarily responsible for the highlight on the (nearly) flat table. Indeed, CDV is very low on the table, while being highest on the bumpy sphere. CDV is also responsible for effects like the specular highlight on the glossy sphere. Figure 5e plots the ratio

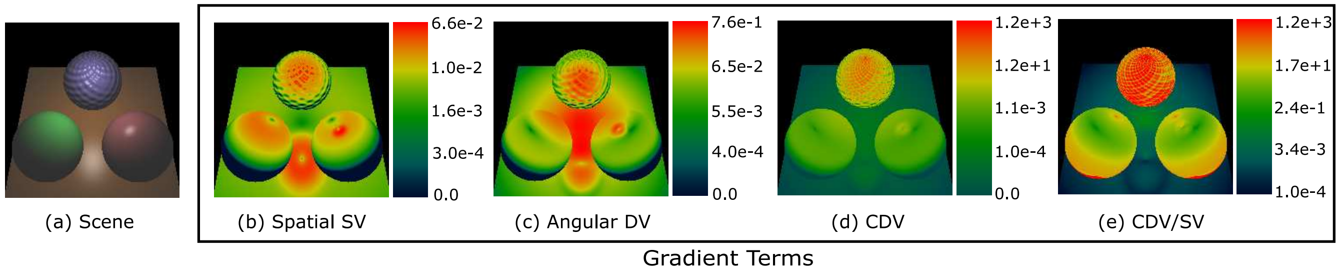


Figure 5: *A scene that shows various shading effects, including a nearly flat table, and diffuse, glossy and bumpy spheres, lit by a moderately close area source. (b), (c) and (d) show various terms of the gradients. (e) shows the ratio of curvature-weighed directional variation (CDV) and spatial variation (SV).*

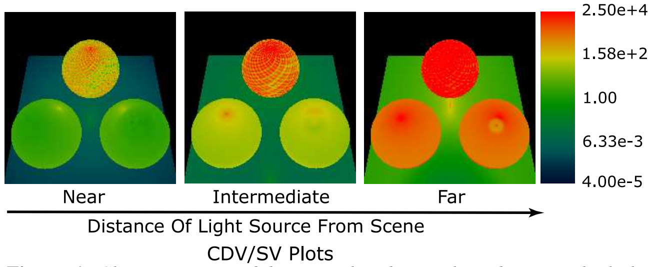


Figure 6: *Change in ratio of directional and spatial gradients as the light moves further away. For near lighting, SV is important, while the relative importance of the angular term (and CDV/SV) increases for far lighting.*

of angular and spatial terms CDV/SV. This ratio is very high for the bump-mapped object, where the angular term dominates, and very low for the table. One insight is that the lighting is effectively distant for the bumpy sphere, but not for the table. In diffuse regions of the Lambertian and diffuse+glossy sphere, there are parts where CDV and SV are comparable.

Finally, Fig. 6 shows the effects of moving the light source further away for the same scene. For near lighting, the spatial gradient SV is quite important. As the lighting becomes more distant, this term becomes less important relative to the directional variation CDV, and CDV/SV increases, as expected.

4.6 Image Gradients

So far, we have found the light field gradients. We now seek the projected image $I(u)$ and image gradients I_u. For this, we assume the perspective projection model, with $u = \gamma x_\perp / z$, where z is the vertical distance to the point, $x_\perp$ is the horizontal distance and γ is the focal length. Using the standard chain rule for gradients,

$$I_u = C^T B \qquad C = \begin{pmatrix} \partial x/\partial u \\ \partial \theta/\partial u \end{pmatrix} \qquad B = \begin{pmatrix} B_x \\ B_\theta \end{pmatrix}. \tag{26}$$

The terms B just correspond to $B_x = \partial B/\partial x$ and $B_\theta = \partial B/\partial \theta$, and are the light field derivatives in Equation 25. The terms C are the camera derivatives. To derive them, we write $u = \gamma x_\perp / z$. The algebra is slightly tricky, but these are standard trigonometric expressions. For brevity, we omit the derivation, stating the result,

$$\frac{\partial x}{\partial u} = \frac{z^2}{\gamma \sqrt{z^2 + x_\perp^2}\,(\mathbf{n_0} \cdot \mathbf{v})} \qquad \frac{\partial \theta}{\partial u} = \frac{1}{\gamma + u^2/\gamma}, \tag{27}$$

where $\mathbf{n_0} \cdot \mathbf{v}$ is the dot product between the viewing ray and the (global, without normal mapping) geometric surface normal.

As an example, consider highlights on a flat surface under distant lighting (but with close viewer). Since there is no curvature or spatial lighting variation, the spatial light field gradient $B_x = 0$. In this case, for $u \ll \gamma$, $\partial\theta/\partial u \approx \gamma^{-1}$, and the image gradient is $I_u \approx \gamma^{-1} B_\theta$, dominated by the *angular* variation in the light field.

At the other extreme, assume the camera is distant (large γ and z). We can neglect $\partial\theta/\partial u$, since θ does not vary much over the image. Moreover, $x_\perp$ is small relative to z, so that we can write $\partial x/\partial u \sim (z/\gamma)(\mathbf{n_0} \cdot \mathbf{v})^{-1}$, and

$$I_u \approx \frac{1}{\gamma} \frac{z}{\mathbf{n_0} \cdot \mathbf{v}} B_x. \tag{28}$$

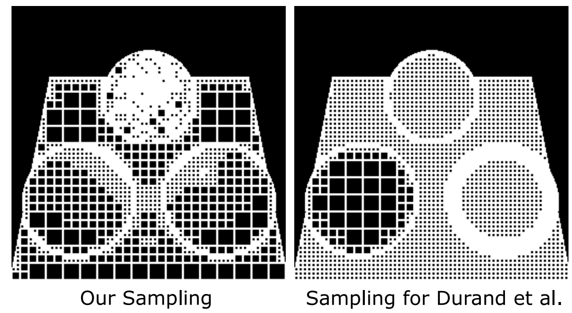


Figure 7: *Comparison of sample locations (using 10% of image pixels) with our gradient algorithm, and the approximation of Durand et al., for the scene in Fig. 5. We correctly place more samples on the bumpy sphere, and in highlights, while Durand et al. focus on the relatively smooth regions near the sphere boundaries, with a coarser sampling for the diffuse object.* Note: To avoid line aliasing, the reader may wish to zoom into the PDF.

This just corresponds to scaling up the spatial light field gradient B_x at grazing angles (when $\mathbf{n_0} \cdot \mathbf{v}$ is small) and for distant objects (large z), when more of the surface projects onto a single pixel.

4.7 Second Order Light Field Analysis

Finally, one benefit of a first order analysis is that it is easy to extend to higher orders. Appendix C differentiates the convolutions in Equation 25 to derive second-order terms or Hessians,

$$\begin{aligned} B_{xx} &= L_{xx} \otimes \rho + 4n_x(L_{x\theta} \otimes \rho) + 4(n_x)^2(L_{\theta\theta} \otimes \rho) + 2n_{xx}(L_\theta \otimes \rho) \\ B_{\theta\theta} &= L_{\theta\theta} \otimes \rho \\ B_{\theta x} = B_{x\theta} &= -L_{x\theta} \otimes \rho - 2n_x(L_{\theta\theta} \otimes \rho). \end{aligned} \tag{29}$$

As expected, these second derivatives involve second derivatives of the incident light field. The angular second derivative $B_{\theta\theta}$ is easy, just corresponding to the second derivative $L_{\theta\theta}$ of the incident light field. Similarly, the mixed partials involve only two terms—the mixed partial $L_{x\theta}$ and the curvature-weighted $L_{\theta\theta}$. This is very similar to the spatial gradient behavior B_x in Equation 25. The spatial second derivative B_{xx} is the most complex, and includes a number of terms, including a curvature derivative n_{xx} in one of them, indicating the intricacies of the reflected light field.

5 Practical Application: Gradient-Based Image Sampling for Efficient Rendering

We now develop a simple prototype application that adaptively samples images. We seek to place more image samples in high-contrast regions with large gradients, to speed up image synthesis. The analysis in the last section immediately confers insight, and one could develop a number of simple heuristics to place samples where SV or CDV terms are expected to be large. One would not even need to formally evaluate the convolutions, and we could simply focus on regions of high spatial or directional change in lighting. We could also use our analysis to sample more finely in high curvature regions, and grazing angles for the camera (low $\mathbf{n_0} \cdot \mathbf{v}$), or to develop a simple metric using a product of these factors. As a proof of concept, in this section, we will show how to use the *full* gradient computation, i.e. I_u from Equation 26, with camera terms in Equation 27, and light field gradients in Equation 25. As in Sec. 4.5, we use the direct 3D analogs of the 2D or flatland gradients.

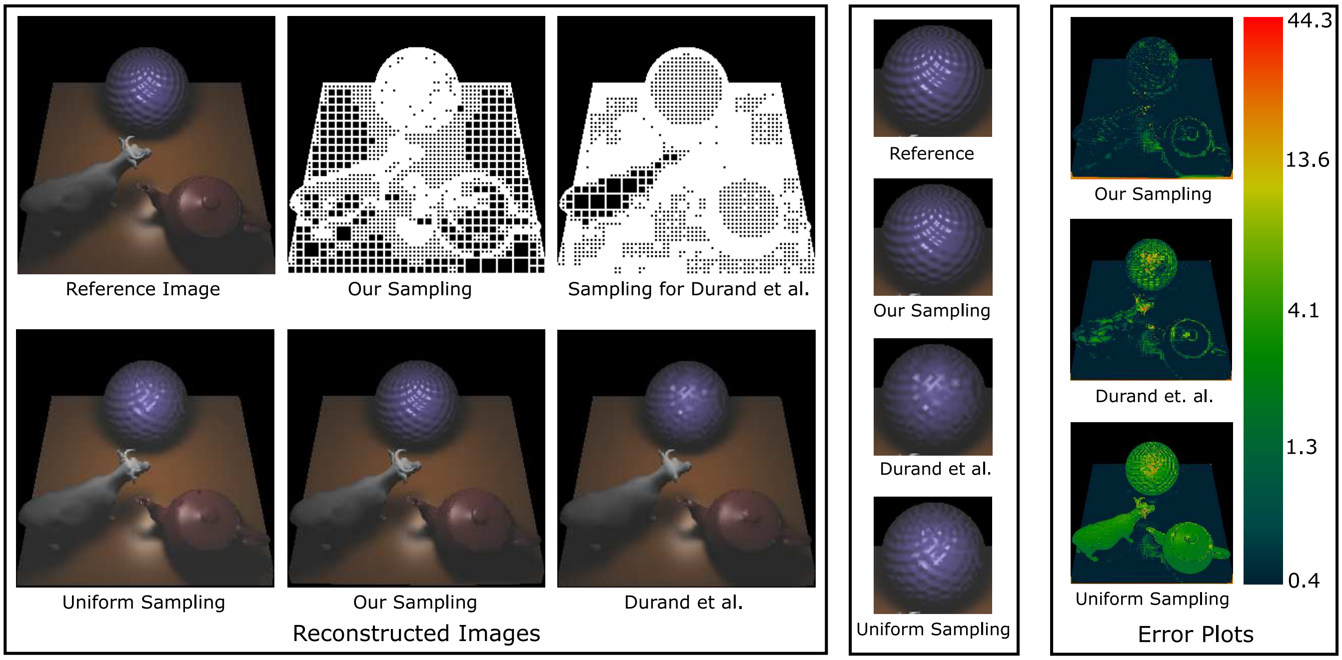


Figure 8: *Comparison of various image sampling strategies for a 3D scene with complex geometry, reflectance and shadows. Our method, based on gradient magnitude, places samples in regions of high-curvature and shading change, closely matching the reference with only* 17% *of the effective pixels. By contrast, both uniform sampling and Durand et al. blur the bumpy sphere, as well as having other regions with large errors.* Note: To avoid on-screen aliasing in the sampling visualizations, the reader may wish to zoom into the figure in the PDF.

For adaptive sampling, we use a method based on quad-trees. We first render a sparse uniform sampling, dividing the image into 8×8 blocks. We treat each sample as the center of a square, and use the gradient magnitude $|I_u|$, multiplied by the area of the square, as an importance metric. We then greedily pick the square of greatest importance, and refine it into four smaller squares, placing four new samples at their centers—this reduces the net importance metric in each sub-square, since the area is reduced to one-fourth. We also always subdivide along object boundaries or silhouettes in image space, to avoid interpolating across different objects. At the end, samples are eventually distributed according to gradient magnitude. Note that when we subdivide to create new samples, those need not lie on the original grid of image pixels, and can be at fractional locations—our final image reconstruction involves a Delaunay triangulation of all the sample locations (including those at each level of subdivision), and interpolates to determine pixel intensities.

This simple adaptive scheme also allows us to directly compare with any other metric. In particular, we consider the frequency-based sampling heuristic of [Durand et al. 2005] (Equation 20 of their paper), which in our notation can be written as

$$I \sim \frac{1}{\gamma} \frac{2kz}{\mathbf{n_0} \cdot \mathbf{v}} \Omega_\rho,$$

where I is the importance given to a pixel, k is the global curvature (without considering nonlinear effects like bump maps), and Ω_ρ is an overall bandlimit for the BRDF (based on [Ramamoorthi and Hanrahan 2001], we use $\Omega_\rho = \sqrt{6s}$ where s is the Phong exponent).

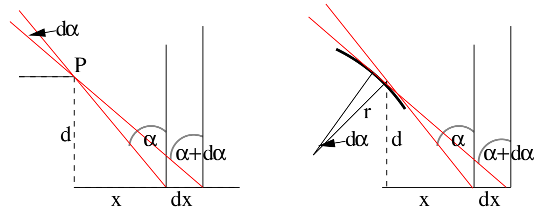
Figure 7 compares the sample distribution for our gradient metric, and that from the above formula for the scene in Fig. 5 (both with 10% of the total image samples). The approach of [Durand et al. 2005] places many samples near object boundaries where $\mathbf{n_0} \cdot \mathbf{v}$ is small. However, these regions are primarily diffuse, with the shading not varying rapidly. Moreover, more weight is given to the glossy spheres than the Lambertian sphere, and no special importance is attached to the bumpy object. By contrast, our approach can explicitly evaluate the gradients. Therefore, it places lots of samples in the bumps, to capture them accurately. Moreover, the glossy sphere has a finer sampling in only the highlight region, but coarser sampling elsewhere. Finally, our approach samples somewhat more densely on the table, between the spheres, where there are interesting near-field lighting effects like highlights.

To extend these results to complex geometric objects, we must address curvature on meshes, complex reflectance and shadows. Gaussian curvatures are assumed given as part of the input, and in practice are precomputed using the TriMesh library [Rusinkiewicz 2004]. For general BRDFs, we simply replace the convolutions with the explicit shading calculation for gradients, as we already must do for the image intensity. Finally, for shadows, we modulate the shading integrals for the gradients by visibility, as for the image itself. Shadow testing is the expensive operation in a ray tracer, especially with soft shadows in complex lighting.[4] Since the same visibility samples are used for image and gradient calculations, our approach introduces minimal overhead.

Figure 8 shows a similar scene, now with complex geometric objects and shadows. We see that with only 17% of the samples, we obtain sharp results comparable to the reference, including on the bumpy sphere. By contrast, an equal number of uniform samples, or those from [Durand et al. 2005] blurs the bumps considerably. Moreover, comparison of sample distributions shows that we place them appropriately, in high-curvature (bumpy) areas and rapid change regions like highlights or shadows. Considering the error plots on the far right, uniform sampling has large errors on all of the objects. [Durand et al. 2005] has large errors on the bumpy sphere, and to some extent in the teapot highlight, as well as on the head of the cow—since less weight is given to diffuse objects.

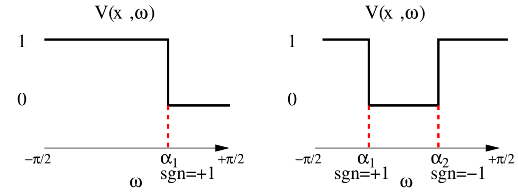
In terms of running time, the reference image took 75.9 minutes. If we perform a gradient calculation at *every pixel*, to check overhead, the additional time is only 1.2 minutes, or less than 2%. The primary overhead actually comes from quad tree construction, and takes about 6 minutes (8%). On this specific scene with 17% of the pixels, since the bumps are easier to ray-trace for shadows (a simple sphere), we actually obtain a slightly super-linear speedup (11.5 minutes for gradient sampling, or a 6.6× improvement), and are also somewhat faster than [Durand et al. 2005], who place fewer samples in the bumps (17.5 minutes). In general, we expect the efficiency improvement to be directly proportional to the number of pixels evaluated for both our method and [Durand et al. 2005].

[4] Coherence-based shadow accelerations like [Agrawala et al. 2000] are not readily applicable, since the lighting cannot be assumed distant or a small area source. Moreover, we also consider complex reflectance effects. Earlier coherence-based techniques like [Guo 1998] are not intended for bumpy surfaces in complex environment illumination with soft shadows.



(a) Poly−Line (point blocker) (b) Curved Surface

Figure 9: We consider the gradient of the blocked region $\alpha_x = d\alpha/dx$. (a) shows the case of a poly-line blocker, where the extremal ray intersects a single point P, and (b) shows a curved blocker.



(a) single discontinuity (b) multiple discontinuities

Figure 10: Representation of the visibility $V(x,\omega)$ at a single point for (a) single and (b) multiple discontinuities. The sign in the visibility equations is positive if going from visible to blocked, and we sum over discontinuities.

6 First Order Analysis of Soft Shadows in 2D

So far, we have not explicitly considered the visibility term $V(x,\omega)$ in Equation 1. In this section, we analyze soft shadow gradients in detail. We start by showing how to incorporate visibility into the shading equations. Then, we compute visibility gradients V_x and V_ω in 2D, followed in Sec. 7 by a 3D analysis.

6.1 Incorporating Visibility

Taking shadows into account, the incident light field L^{shad} at a point is a product of the unshadowed lighting L and binary visibility V,

$$L^{\text{shad}}(x,\theta) = L(x,\theta)V(x,\theta). \tag{30}$$

The gradients are given simply by

$$\bigtriangledown \mathbf{L}^{\text{shad}}(x,\theta) = \bigtriangledown \mathbf{L}(x,\theta)V(x,\theta) + L(x,\theta)\bigtriangledown \mathbf{V}(x,\theta). \tag{31}$$

Our previous formulae are still valid, if we re-interpret L as the shadowed illumination L^{shad}, which is already pre-multiplied by the visibility. Hence, in any gradient formula, we can replace L_x by $L_xV + LV_x$ and L_θ by $L_\theta V + LV_\theta$. The first term (L_xV or $L_\theta V$) simply requires us to modulate the shading integrals or convolutions, such as those in Equation 25, by the visibility V. We have already taken this into account for practical image synthesis and sampling.

The second term (LV_x or LV_θ) requires us to find visibility gradients. Indeed, these gradients are the focus of this section, and have usually been omitted in previous analyses and algorithms. In general, the shading gradients in these cases can be written as

$$\begin{aligned} B(x,\theta) &= \int T(x,\theta,\omega)V(x,\omega)\,d\omega \\ \bigtriangledown \mathbf{B}^V &= \int T(x,\theta,\omega)\bigtriangledown \mathbf{V}(\mathbf{x},\omega)\,d\omega, \end{aligned} \tag{32}$$

where T is a general transport operator encompassing lighting L and BRDF ρ. For simple convolution, $T(x,\theta,\omega) = L(x,\omega)\rho(x,\theta-\omega)$. The superscript in B^V indicates explicitly that we are considering the visibility gradient term (not gradients of the lighting or BRDF). For ease of notation, we will drop this superscript from now on.

6.2 Local Visibility Gradients

The visibility V is a binary function, and the gradients of it are delta functions, being 0 except at discontinuities, where they are infinite. However, the integral above for $\bigtriangledown \mathbf{B}(x,\theta)$ is still finite.[5]

Figure 9 shows the 2D cases we consider, of poly-line and curved blockers. For computing gradients, it suffices to consider the local region around an extremal blocker point P, summing over all such visibility discontinuities. From Fig. 9, $V(x,\omega) = 1$ (visible) when $\omega < \alpha(x)$ and 0 (blocked) when $\omega > \alpha(x)$. Formally,

$$V(x,\omega) = H(\alpha(x)-\omega), \tag{33}$$

where H is the Heaviside step function ($H(u) = 1$ when $u > 0$ and 0 otherwise). If the visibility transition at α is from blocked to visible instead of vice-versa, we will need to change signs, using $1-H$. The derivative of the Heaviside function is the delta function,

$$V_x = \delta(\alpha(x)-\omega)\alpha_x \qquad V_\omega = -\delta(\alpha(x)-\omega). \tag{34}$$

[5]As noted by Arvo [1994] and others, there are degenerate configurations (such as where shadow lines from multiple blockers meet at a point or a visibility ray is tangent to two blockers), when the gradients can actually be infinite. In practice, as seen by our results, these degenerate cases are relatively rare and do not affect the numerics significantly.

Angular Gradients: Plugging into Equation 32, the angular gradient is simple—the delta function evaluates T at $\omega = \alpha(x)$,

$$B_\theta = \sum_j -\text{sgn}_j T(x,\theta,\alpha_j(x)), \tag{35}$$

where we sum over all discontinuities j of the visibility $V(x,\omega)$ for given x, with the appropriate sign. For example, in Fig. 10a, $B_\theta = -T(x,\theta,\alpha_1(x))$. In Fig. 10b, $B_\theta = -T(x,\theta,\alpha_1(x)) + T(x,\theta,\alpha_2(x))$. Note that we only need to observe the visibility discontinuities at a single spatial location x to compute the gradients. If one numerically computes $V(x,\omega)$, such as by ray-tracing, it is easy to determine the discontinuities and apply the equation above.

Spatial Gradients: The spatial gradients are more interesting, since we need to determine $\alpha_x = d\alpha/dx$. This effectively controls how fast the visibility changes as we move along x, and therefore depends on the vertical distance d to the blocker.

First, consider the case where we have a poly-line blocker, and therefore a single extremal point P, as in Fig. 9a. From trigonometry, $\tan\alpha = x/d$, which can be differentiated to give

$$\frac{d\alpha}{dx} = \frac{\cos^2\alpha}{d} = \frac{\cos\alpha}{D}, \tag{36}$$

where d is the vertical distance to the blocker, and D is the total distance, with $D = d/\cos\alpha$. In appendix D, we show that exactly the same result holds even if we consider a curved blocker, as in Fig. 9b. Finally, we can write for the spatial gradient,

$$B_x = \sum_j +\text{sgn}_j T(x,\theta,\alpha_j(x))\frac{\cos\alpha_j}{D_j}, \tag{37}$$

where we sum over all discontinuities, as for the angular gradient.

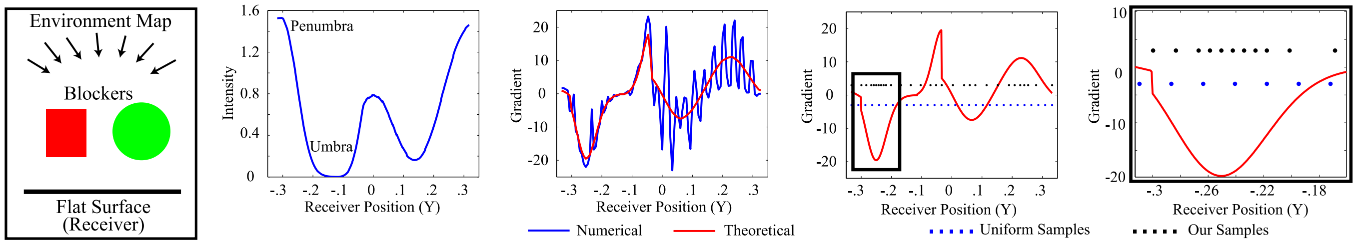
Finally, note that Equations 35 and 37 no longer involve delta functions, and therefore easily allow further differentiation to find second-order Hessians. Moreover, our approach can also apply to other shading situations involving delta-function gradients, such as the sharp edges of area light sources, or mirror reflectance.

Implications and Discussion: These results have several implications. First, the gradient varies inversely with the distance to the blockers, and we must sum over all discontinuities. This explains why the harmonic mean of blocker distances is commonly used as a metric for gradient algorithms and sampling [Ward and Heckbert 1992]. However, we go further in deriving an *exact* formula, which considers general curved or polygonal blockers, and can accurately be used for gradient-based *interpolation*, beyond its use as a metric for sampling. Moreover, we consider the $\cos\alpha$ term in the numerator, with a smaller gradient for blockers at grazing angles.

Also note the rich interplay between spatial and angular effects for visibility, as for the earlier shading formulae. Equation 37 for the *spatial* gradient involves knowledge only of the *angular* discontinuities α_j in visibility at a single spatial location.

6.3 Results in 2D or Flatland

To focus on visibility effects, we consider an environment map (with usual Gaussian variation in lighting) acting on a flat diffuse surface, shadowed by a box (rectangle) and circle. The cosine term can be folded into the lighting, and T corresponds almost directly to the illumination. Figure 11b shows that the shading on the receiver has complex umbra and penumbra regions. As seen in (c), the numerical intensity gradient is very noisy. Because visibility is a binary function, the intensity (and local visibility extremum $\alpha(x)$)



(a) Scene Description (b) Intensity On Receiver (c) Numerical vs Theoretical Gradient (d) Our sampling Scheme vs Uniform Sampling

Figure 11: *Using our formula for visibility gradients and sampling. In (a) we show a schematic of our scene. (b) shows the shading on the receiver, that includes complex umbra and penumbra regions. (c) compares gradients obtained by numerical differentiation, with those from our theoretical formula, indicating the noise in numerical differentiation. (d) (including closeup on right), compares our adaptive sampling based on gradients to uniform sampling.*

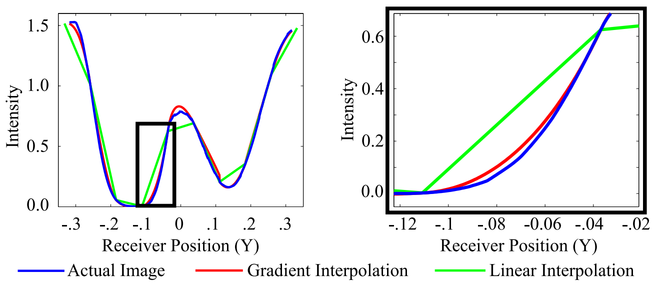


Figure 12: *Comparison of gradient interpolation versus standard linear interpolation for shadows (with uniform sampling). As seen in the closeup on the right, using gradients gives significantly greater accuracy.*

usually changes in a stair-step pattern, depending on specific lighting and image resolutions. While this mild aliasing is not usually a problem for image synthesis, it introduces serious problems for numerical differentiation. By contrast, with our formal treatment of the visibility gradient as a delta function, we calculate a smooth result that precisely matches the analytic value. Figure 11d explores applications to sampling, using an adaptive sampler based on our accurate gradients, which places more samples in high-gradient shadow regions. We compare in this case to uniform sampling since Fourier bandwidth bounds will also give only an (uniform) overall sampling rate, depending on the harmonic distance to blockers.

Our calculations will perhaps be most useful in terms of accurate visibility gradients for gradient-based interpolation in algorithms like [Ward and Heckbert 1992], which currently usually ignore visibility gradients. Figure 12 compares gradient interpolation to standard linear interpolation (for uniform sampling)—it is clear gradient interpolation gives much higher accuracy.

7 First Order Analysis of Soft Shadows in 3D

Unlike for curved surface reflection in Sec. 4, it is not possible to use the direct 3D analog of the 2D results (as in Sec. 4.4). However, we can extend our visibility analysis to 3D, using much the same techniques as in 2D. In this section, for simplicity we assume a flat receiver. We start by considering spatial visibility gradients. Then, we compute gradients for the total visible area of the sphere of directions, followed by gradients in complex lighting. To our knowledge, this is the first derivation of accurate visibility gradients for general curved occluders, and we apply our results to gradient interpolation methods for efficient rendering (Fig. 15).

7.1 Spatial Visibility Gradients

In 3D, the visibility is

$$V(x,y,\theta,\phi) = H(\alpha(x,y,\phi)-\theta), \tag{38}$$

where θ and ϕ are a standard spherical parameterization, in terms of elevation and azimuthal angle, and (x,y) are spatial coordinates. We consider only V_x here, since the derivation for V_y is similar,

$$V_x(x,y,\theta,\phi) = \delta(\alpha(x,y,\phi)-\theta)\alpha_x(x,y,\phi). \tag{39}$$

As in 2D, the difficult part is determining α_x. It is convenient to define new axes u and v, where u is aligned along ϕ, i.e., is given by $(\cos\phi, \sin\phi)$, and v is aligned at 90° given by $(-\sin\phi, \cos\phi)$. In this case, applying the chain rule,

$$\alpha_x(x,y,\phi) = \alpha_x(u,v) = \alpha_u(u,v)u_x + \alpha_v(u,v)v_x, \tag{40}$$

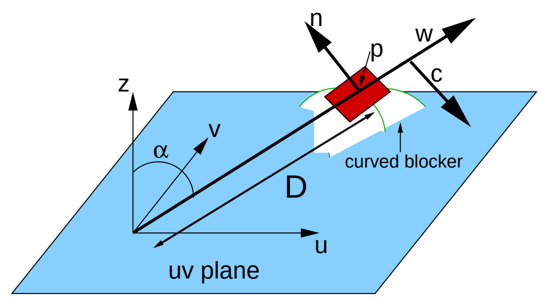


Figure 13: *Local geometry for calculation of α_u and α_v in 3D visibility.*

which can be simplified to

$$\alpha_x(x,y,\phi) = \alpha_u(u,v)\cos\phi - \alpha_v(u,v)\sin\phi. \tag{41}$$

We now derive α_u and α_v for a general curved surface. Figure 13 shows the local geometry in the u-v plane. (Appendix D derives the analogous result for a polygonal object or mesh, which we have numerically verified extensively using general cuboids or boxes.)

The basic idea is to think about the differential change in the point of intersection $\mathbf{p}$, as one moves the spatial location slightly in the u-v plane. We define $\mathbf{w}$ as the direction of the tangent ray to the surface, given in the (u,v,z) frame by $(\sin\alpha, 0, \cos\alpha)$. $\mathbf{c}$ is the transverse direction given in the (u,v,z) frame as $(\cos\alpha, 0, -\sin\alpha)$. $\mathbf{w}$ and $\mathbf{c}$ are orthogonal, forming a coordinate frame for the u-z plane.

If the angle α changes a small amount, the point of intersection moves an amount $\mathbf{dp} = (D\,d\alpha)\,\mathbf{c}$ along $\mathbf{c}$, from basic trigonometry. Similarly, if the distance to the blocker changes a small amount, the point of intersection moves along the tangent ray, $\mathbf{dp} = (dD)\mathbf{w}$. Finally, if the starting spatial location shifts in the u-v plane by an amount dr in the direction $\mathbf{m}$ (where for us $\mathbf{m}$ is either the $u-$ or $v-$axis), the point of intersection will move $\mathbf{dp} = dr\,\mathbf{m}$,

$$\mathbf{dp} = (D\,d\alpha)\,\mathbf{c} + (dD)\mathbf{w} + (dr)\,\mathbf{m}. \tag{42}$$

The new point of intersection still lies on the tangent plane to the surface at $\mathbf{p}$ (shown in red in Fig. 13) to first order. Therefore, $\mathbf{dp}\cdot\mathbf{n} = 0$, where $\mathbf{n}$ is the surface normal at $\mathbf{p}$,

$$\mathbf{dp}\cdot\mathbf{n} = 0 = (D\,d\alpha)\,\mathbf{c}\cdot\mathbf{n} + (dD)\mathbf{w}\cdot\mathbf{n} + (dr)\,\mathbf{m}\cdot\mathbf{n}. \tag{43}$$

Now, the condition of tangency requires that $\mathbf{w}\cdot\mathbf{n} = 0$, so that

$$D\,d\alpha(\mathbf{c}\cdot\mathbf{n}) + dr(\mathbf{m}\cdot\mathbf{n}) = 0 \;\Rightarrow\; \frac{d\alpha}{dr} = -\frac{1}{D}\frac{\mathbf{m}\cdot\mathbf{n}}{\mathbf{c}\cdot\mathbf{n}}. \tag{44}$$

Finding α_u: Let $dr = du$ and $\mathbf{m} = \mathbf{u}$ be a unit vector along u. It is possible to express $\mathbf{u} = \sin\alpha\mathbf{w} + \cos\alpha\mathbf{c}$. Noting that $\mathbf{w}\cdot\mathbf{n} = 0$,

$$\alpha_u = -\frac{1}{D}\frac{\sin\alpha\mathbf{w}\cdot\mathbf{n} + \cos\alpha\,\mathbf{c}\cdot\mathbf{n}}{\mathbf{c}\cdot\mathbf{n}} = -\frac{\cos\alpha}{D}, \tag{45}$$

as expected, since α_u is essentially the flatland 2D case.
Finding α_v: Now, consider α_v with $\mathbf{m} = \mathbf{v}$. Directly from Equation 44,

$$\alpha_v = -\frac{1}{D}\frac{\mathbf{v}\cdot\mathbf{n}}{\mathbf{c}\cdot\mathbf{n}}. \tag{46}$$

Finally, note that as in the 2D case, the spatial gradients depend inversely on the distance to the blocker D. The gradients also depend on the angle α, as well as the angle between the blocker's surface normal $\mathbf{n}$, and the vectors $\mathbf{c}$ and $\mathbf{v}$.

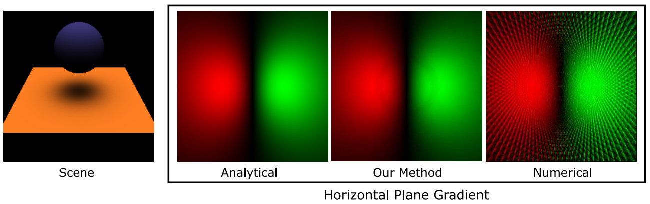


Figure 14: *Accuracy of net visibility gradients. From left to right, the simple scene, the independent analytic gradient for B_x (for a head-on view of the ground plane), the accurate results computed by our method, and the very noisy results obtained by numerical differentiation.*

7.2 Gradients of Net Visibility

Now, we consider gradients of the net visibility, i.e., what fraction of the hemisphere of directions is blocked. This is immediately useful for ambient occlusion [Christensen 2002], and provides a useful background for the complex environments considered later,

$$B(x,y) = \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi/2} V(x,y,\theta,\phi)\ \sin\theta\, d\theta d\phi. \tag{47}$$

To compute gradients B_x, we use the visibility gradients V_x from Equation 39. The delta function causes the θ integral to be evaluated at $\alpha(x,y,\phi)$. Summing over multiple discontinuities as usual,

$$B_x(x,y) = \int_{\phi=0}^{2\pi} \sum_j \left(\operatorname{sgn}_j \frac{\partial \alpha_j}{\partial x}(x,y,\phi) \sin\alpha_j(x,y,\phi) \right) d\phi, \tag{48}$$

where we use the long form $\partial\alpha_j/\partial x$ instead of α_x, to avoid confusion with the subscript j for the j^{th} discontinuity. For simplicity, we assume x and y are coordinates on the receiver. If we seek gradients in the image, we will need to consider camera terms as we did in 2D, and we have implemented these where appropriate.

Efficiency and Accuracy: Our gradient computation is very efficient and accurate.[6] The gradients depend only on discontinuities, not the full visibility function. In particular, Equation 48 is only a 1D integral, as opposed to image evaluation in Equation 47, which requires 2D integration. Figure 14 shows a plane with a single sphere blocker. For a single sphere, it is easy to derive an independent analytic expression for Equation 47, to test accuracy. Indeed, with 400 samples in ϕ, the gradients computed by our method in Fig. 14 are nearly exact for 24 bit RGB images. On the other hand, with even 900 samples, image evaluation shows noticeable variance or bias. While these errors are usually tolerable for image synthesis, they become pronounced when computing gradients by numerical differentiation—which is also compared to in Fig. 14.

7.3 Complex Lighting

We now consider environment map illumination on a flat Lambertian surface. In this case, the cosine term can be folded into the lighting, and the gradients are only due to soft shadows,

$$\begin{aligned} B(x,y) &= \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi/2} L(\theta,\phi) V(x,y,\theta,\phi)\ \sin\theta\, d\theta d\phi \qquad (49) \\ B_x(x,y) &= \int_{\phi=0}^{2\pi} \sum_j \left(\operatorname{sgn}_j L(\alpha_j(x,y,\phi),\phi) \frac{\partial\alpha_j}{\partial x}(x,y,\phi) \sin\alpha_j(x,y,\phi) \right) d\phi. \end{aligned}$$

These visibility gradients can also trivially be extended to spatially-varying lighting, $L(x,y,\theta,\phi)$—however, the intensity gradients B_x would need an additional term corresponding to variation in the lighting itself. Similarly, complex reflectance can be baked into the lighting term for a flat surface with a distant viewer. For a close viewer, the visibility part of the gradient simply replaces L with a general transport function, and there will be an additional term for gradients of the BRDF themselves.

Results: Figure 15 applies these results to image synthesis with complex penumbra regions of a diffuse plane in environment map lighting (with the usual Gaussian lighting variation). The plane shading is computed accurately (with a high resolution 100×100 environment sampling) at a coarse grid of only $1/10$ the resolution of the image in both directions (*or 1% of the total pixels*), and interpolated using gradients. These results are compared to simple linear interpolation, and are clearly seen to be much more accurate. Thus, we can use our gradient formulae to very efficiently render complex shadows with standard gradient interpolation methods.

8 Discussion and Comparison

In this section, we briefly discuss some of our theoretical results, comparing to Fourier analysis [Durand et al. 2005], and previous analyses of visibility [Arvo 1994; Ramamoorthi et al. 2004].

Basic Shading Steps: Section 4.1 conducts a first order analysis of the basic steps for light reflection from a curved surface, and is similar to the frequency analysis of light transport in [Durand et al. 2005]. Most analogies and differences follow directly from the forms of the mathematical operations in Sec. 3.1. For instance, consider the per-point rotation in step 1 with $L^s(x,\theta) = L(x,\theta+kx)$, and gradient formula in Equation 10. The effects on the Fourier spectrum are found from the linear transformation theorem in Equation 6. To apply that, we need M^{-T} and its determinant. M^{-T} is simply M^T in Equation 9, with k replaced by $-k$, and $det(M) = 1$,

$$L^s(\Omega) = L(\Omega') \quad \begin{pmatrix} \Omega'_x \\ \Omega'_\theta \end{pmatrix} = \begin{pmatrix} 1 & -k \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \Omega_x \\ \Omega_\theta \end{pmatrix} = \begin{pmatrix} \Omega_x - k\Omega_\theta \\ \Omega_\theta \end{pmatrix}. \tag{50}$$

Apart from a sign change (we use a different sign convention for k), this is the formula derived from first principles by Durand et al. [2005]. The important differences between Fourier and gradient results are direct consequences of the linear transformation theorems in Equations 7 (gradient) and 6 (Fourier). The function in the gradient case is evaluated at $L(M\mathbf{u})$—in this case $(x, \theta+kx)$. Hence, the gradients will be transformed in the *same* way as the light field—in this case, sheared along the angular dimension. By contrast, the function in the Fourier case is evaluated at $L(M^{-T}\Omega)$—in this case $(\Omega_x - k\Omega_\theta, \Omega_\theta)$. Therefore, the 2D Fourier spectrum is often sheared in the *opposite* way as the light field (and gradients)—in this case, along the spatial dimension.

Also note that [Durand et al. 2005] include additional shading steps at the beginning and end, for reparameterization into the local tangent frame, in their equivalent of section 4.1. Because gradient analysis is fully local, as discussed after Equation 25, our final results do not need these additional reparameterization steps.

Advantages of First Order Analysis: For the theory in this paper, first order analysis easily separates the different factors responsible for shading variation. Moreover, we can accurately consider non-linear transformations like bump maps, aspects of the camera transformations, and visibility. The derivations and insights in Secs. 4.2–4.7 would not be easy to obtain from Fourier analysis.

A concrete example is the relative performance of gradient-based sampling versus a frequency-based metric, in Sec. 5. In fact, one can consider the metric in [Durand et al. 2005] a special case of Equation 28, with the assumption of a distant camera with distant lighting and no bump mapping. In that case, $B_x = 2k(L_\theta \otimes \rho)$, where we substitute the global curvature k for n_x,

$$I_u \approx \frac{1}{\gamma} \frac{2kz}{\mathbf{n_0} \cdot \mathbf{v}} (L_\theta \otimes \rho)(x, \theta_r). \tag{51}$$

If we further approximated the convolution with a constant bandlimit ($(L_\theta \otimes \rho) \sim \Omega_\rho$) derived from the BRDF, we would recover the image sampling criterion in Equation 20 of [Durand et al. 2005].

We emphasize that the gradient approach allows spatially-varying lighting, bump mapping and close cameras. Moreover, we can explicitly evaluate the convolutions like $L_\theta \otimes \rho$, while Fourier methods do not easily lend themselves to practical computation.

[6] The only tricky numerical issue occurs if $\partial\alpha_j/\partial x$ is very large. The denominator $\mathbf{c}\cdot\mathbf{n}$ in Equation 46 can be small when the u-z plane is (nearly) tangent to the surface (so both $\mathbf{c}\cdot\mathbf{n}$ and $\mathbf{w}\cdot\mathbf{n}$ are zero). It can be shown that this is a weak singularity, going as $(\phi-\phi_0)^{-1/2}$, where ϕ_0 is where the u-z plane is tangent to the surface. Therefore, the integral is still well behaved, and we evaluate it efficiently by an adaptive sampling of ϕ. The other potential problem is minor errors when the number of discontinuities α_j changes abruptly, such as from 1 to 2. This is responsible for the slight errors near the projection of the sphere on the ground plane in Fig. 14.

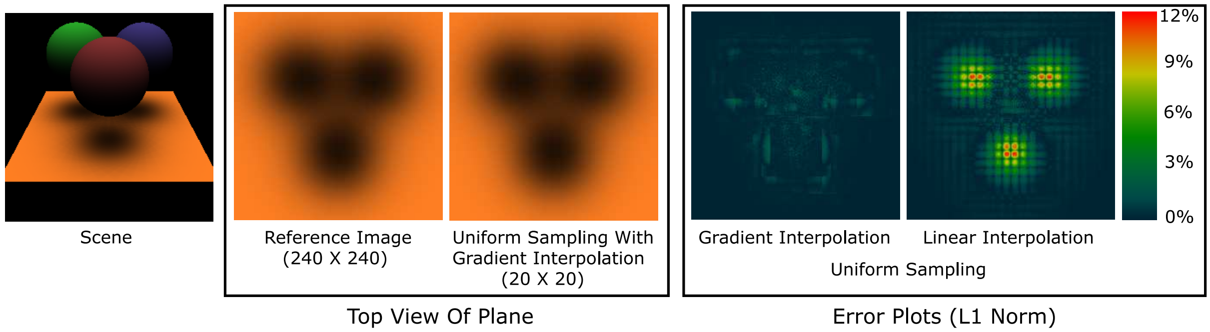


***Figure 15:** Comparison of gradient interpolation with linear interpolation. The leftmost image shows the scene. The other images interpolate shading directly on the ground plane (shown in a head on view), using a uniform 20×20 sampling (*less than 1% of total pixels*). Gradient interpolation with our accurate gradients is very high quality, and substantially lowers the error from simple linear interpolation.*

Visibility: Since visibility is non-linear, frequency analysis can give some insights, but not the precise formulae derived here. Our analysis of 2D visibility is perhaps closest to the shadow convolution relation in [Ramamoorthi et al. 2004]. However, we make an exact calculation of local shadowing and gradients, and generalize to curved occluders. The 3D analysis has some similarities to the irradiance Jacobian of [Arvo 1994] for polygonal scenes, but we also consider general curved occluding surfaces. Moreover, our approach is different, in that all computations are local, depending only on visibility discontinuities at a single point, and being easy and efficient to integrate into a ray-tracing framework. Note that we consider local gradients, and our method is orthogonal to identifying global visibility events [Durand et al. 2002]—a future direction is to combine visibility gradients with global analysis.

9 Conclusions and Future Work

We present a complete first order theory of lighting, shading and shadows. First, we develop a full gradient analysis of the basic shading steps, showing the interplay between spatial and angular effects. Second, we analyze the gradients for general scenes with bump maps, and spatially and directionally-varying lighting. Gradient analysis allows us to separate the effects of the individual terms, and determine under what conditions each factor—lighting variation, surface curvature, and object reflectance—is important. We show how to practically use gradients to adaptively sample images for efficient rendering. Third, we develop novel results for visibility gradients, that generalize much previous work on analysis of soft shadows, and enable gradient-based visibility interpolation.

We see this paper as an important theoretical step in the analysis of light transport. In the future, we are hopeful that methods from here can be used to derive rigorous theoretical bounds and new practical algorithms for many widely used gradient interpolation methods like irradiance gradients, spherical harmonic gradients, ray differentials and path perturbation. We also expect to see the sampling rate analysis applied to other problems involving nonlinear steps, such as the flow of light fields in the full volume or 5D space, as is necessary for applications like shadow fields for precomputed radiance transfer [Zhou et al. 2005]. Finally, gradient-based methods are also likely to be important in inverse problems where we have only local information, such as nearby views.

References

AGRAWALA, M., RAMAMOORTHI, R., HEIRICH, A., AND MOLL, L. 2000. Efficient image-based methods for rendering soft shadows. In *SIGGRAPH 00*, 375–384.

ANNEN, T., KAUTZ, J., DURAND, F., AND SEIDEL, H. 2004. Spherical Harmonic Gradients for Mid-Range Illumination. In *EuroGraphics Symposium on Rendering*, 331–336.

ARVO, J. 1994. The Irradiance Jacobian for Partially Occluded Polyhedral Sources. In *SIGGRAPH 94*, 343–350.

CHAI, J., TONG, X., SHAN, S., AND SHUM, H. 2000. Plenoptic Sampling. In *SIGGRAPH 2000*, 307–318.

CHEN, M., AND ARVO, J. 2000. Theory and application of specular path perturbation. *ACM Transactions on Graphics, Vol. 19*, No. 4, 246–278.

CHEN, H., BELHUMEUR, P., AND JACOBS, D. 2000. In search of illumination invariants. In *CVPR 00*, 254–261.

CHRISTENSEN, P. 2002. Note # 35: Ambient Occlusion, Image-Based Illumination, and Global Illumination. In *Photorealistic RenderMan Application Notes*.

CLARBERG, P., JAROSZ, W., MOLLER, T., AND JENSEN, H. 2005. Wavelet Importance Sampling: Efficiently Evaluating Products of Complex Functions. *ACM Transactions on Graphics (SIGGRAPH 2005), Vol. 24*, No. 3, 1166–1175.

DURAND, F., DRETTAKIS, G., AND PUECH, C. 2002. The 3D Visibility Complex. *ACM Transactions on Graphics, Vol. 21*, No. 2, 176–206.

DURAND, F., HOLZSCHUCH, N., SOLER, C., CHAN, E., AND SILLION, F. 2005. A Frequency Analysis of Light Transport. *ACM Transactions on Graphics (SIGGRAPH 2005), Vol. 25*, No. 3, 1115–1126.

GERSHBEIN, R., SCHRODER, P., AND HANRAHAN, P. 1994. Textures and radiosity: Controlling emission and reflection with texture maps. In *SIGGRAPH 94*, 51–58.

GORTLER, S., SCHRODER, P., COHEN, M., AND HANRAHAN, P. 1993. Wavelet Radiosity. In *SIGGRAPH 93*, 221–230.

GUO, B. 1998. Progressive Radiance Evaluation Using Directional Coherence Maps. In *SIGGRAPH 98*, 255–266.

HOLZSCHUCH, N., AND SILLION, F. 1998. An exhaustive error-bounding algorithm for hierarchical radiosity. *Computer Graphics Forum, Vol. 17*, No. 4, 197–218.

IGEHY, H. 1999. Tracing Ray Differentials. In *SIGGRAPH 99*, 179–186.

NG, R., RAMAMOORTHI, R., AND HANRAHAN, P. 2004. Triple Product Wavelet Integrals for All-Frequency Relighting. *ACM Transactions on Graphics (SIGGRAPH 2004), Vol. 23*, No. 3, 475–485.

NG, R. 2005. Fourier Slice Photography. *ACM Transactions on Graphics (SIGGRAPH 2005), Vol. 25*, No. 3, 735–744.

RAMAMOORTHI, R., AND HANRAHAN, P. 2001. A Signal-Processing Framework for Inverse Rendering. In *SIGGRAPH 01*, 117–128.

RAMAMOORTHI, R., KOUDELKA, M., AND BELHUMEUR, P. 2004. A Fourier Theory for Cast Shadows. In *European Conference on Computer Vision*, I–146–I–162.

RUSINKIEWICZ, S. 2004. Estimating Curvatures and Their Derivatives on Triangle Meshes. In *Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*.

SOLER, C., AND SILLION, F. 1998. Fast Calculation of Soft Shadow Textures Using Convolution. In *SIGGRAPH 98*, 321–332.

WARD, G., AND HECKBERT, P. 1992. Irradiance Gradients. In *Eurographics Rendering Workshop 92*, 85–98.

ZHOU, K., HU, Y., LIN, S., GUO, B., AND SHUM, H. 2005. Precomputed shadow fields for dynamic scenes. *ACM Transactions on Graphics (SIGGRAPH 2005), Vol. 25*, No. 3.

Appendix A: Linear Transformation Theorems

In this appendix, we briefly derive the linear transformation theorem for Fourier transforms, as well as gradient-based methods. For Fourier, note that (with $I = \sqrt{-1}$),

$$F(\Omega) = \int f(\mathbf{u}) \exp\left[-2\pi I \Omega^T \mathbf{u}\right] d\mathbf{u}, \tag{52}$$

where Ω and $\mathbf{u}$ are both vectors. If we now set $\mathbf{u} = M\mathbf{v}$, then

$$\begin{aligned} F(\Omega) &= |\,det(M)\,| \int f(M\mathbf{v}) \exp\left[-2\pi I \Omega^T M\mathbf{v}\right] d\mathbf{v} \\ &= |\,det(M)\,| \int h(\mathbf{v}) \exp\left[-2\pi I (M^T\Omega)^T \mathbf{v}\right] d\mathbf{v}, \\ F(\Omega) &= |\,det(M)\,|\, H(M^T\Omega) \;\Rightarrow\; H(\Omega) = \frac{1}{|\,det(M)\,|} F(M^{-T}\Omega). \end{aligned} \quad (53)$$

Now, if the transform includes translations, and so we have $\mathbf{u} = M\mathbf{v} + \mathbf{c}$, then

$$F(\Omega) = |\,det(M)\,| \exp[-2\pi I \Omega^T \mathbf{c}] \int h(\mathbf{v}) \exp\left[-2\pi I (M^T\Omega)^T \mathbf{v}\right] d\mathbf{v} \quad (54)$$

$$= |\,det(M)\,| \exp[-2\pi I \Omega^T \mathbf{c}] H(M^T\Omega) \;\Rightarrow\; H(\Omega) = \frac{\exp[2\pi I \Omega^T M^{-1}\mathbf{c}]}{|\,det(M)\,|} F(M^{-T}\Omega).$$

For gradients, we can write

$$h(u_i) = f(v_i = \sum_j M_{ij} u_j) \;\Rightarrow\; \frac{\partial h}{\partial u_i} = \sum_k \frac{\partial f}{\partial v_k}\frac{\partial v_k}{\partial u_i} = \sum_k \frac{\partial f}{\partial v_k} M_{ki}, \quad (55)$$

which can be rearranged to Equation 7.

Appendix B: Light Reflection from 3D Surfaces

We briefly extend the light-curved surface interaction steps in Sec. 4.1 to 3D. We define local tangent frames, and gradients with respect to motion in a particular direction. At $\mathbf{x}$, the tangents are $\mathbf{t}$ and $\mathbf{b}$, which along with the normal $\mathbf{n}$, form a coordinate frame. Similarly, ω can be treated as a vector, with tangents $\mathbf{u}$ and $\mathbf{v}$ for L^s. The angular tangent directions of the global L will be $\mathbf{l} = R\mathbf{u}$ and $\mathbf{m} = R\mathbf{v}$, where R is the appropriate rotation (or reflection) matrix. Algebraic simplicity in the formulae requires precisely choosing these tangent vectors, which we will describe shortly. For now, note that a gradient such as L^s_t is defined as $L^s_t = (\partial/\partial\alpha) L^s(\mathbf{x} + \alpha\mathbf{t}, \omega)$.

Rotation (step 1): We first discuss rotations, writing $L^s(\mathbf{x}, \omega) = L(\mathbf{x}, R[\mathbf{n}(\mathbf{x})]\omega)$, where R is a 3×3 rotation matrix in 3D. The angular gradients now become

$$L^s_u = \frac{\partial}{\partial\alpha} L(\mathbf{x}, R(\omega + \alpha\mathbf{u})) \quad L^s_v = \frac{\partial}{\partial\alpha} L(\mathbf{x}, R(\omega + \alpha\mathbf{v})), \quad (56)$$

where R can be treated as a constant matrix, since the spatial location (and hence the normal) is not changing. Since we know that $\mathbf{l} = R\mathbf{u}$ and $\mathbf{m} = R\mathbf{v}$, this simply becomes

$$L^s_u = \frac{\partial}{\partial\alpha} L(\mathbf{x}, R\omega + \alpha\mathbf{l}) = L_l(\mathbf{x}, R\omega) \qquad L^s_v = \frac{\partial}{\partial\alpha} L(\mathbf{x}, R\omega + \alpha\mathbf{m}) = L_m(\mathbf{x}, R\omega), \quad (57)$$

so that the angular gradients behave much like in the 2D case, without transformation.

For the spatial gradients, we write

$$L^s_t = \frac{\partial}{\partial\alpha} L(\mathbf{x} + \alpha\mathbf{t}, R[\mathbf{n}(\mathbf{x} + \alpha\mathbf{t})]\omega) \quad L^s_b = \frac{\partial}{\partial\alpha} L(\mathbf{x} + \alpha\mathbf{b}, R[\mathbf{n}(\mathbf{x} + \alpha\mathbf{b})]\omega), \quad (58)$$

where we must now also account for the change in the rotation matrix. We will only consider L^s_t, with the other term being similar. It can be expanded as

$$L^s_t = L_t(\mathbf{x}, R\omega) + \frac{\partial}{\partial\alpha} L(\mathbf{x}, R[\mathbf{n}(\mathbf{x} + \alpha\mathbf{t})]\omega), \quad (59)$$

where the first term is simply the spatial gradient of the original light field, as in 2D. The second term, that corresponds to curvature and directional variation in 2D, is more interesting to extend to 3D.

We now choose $\mathbf{t}$ and $\mathbf{b}$ to be the maximum and minimum curvature directions, or those directions that diagonalize the shape operator from differential geometry. Using a general set of directions is also possible, but makes the algebra messier. We also define κ as the leading eigenvalue of the shape operator, or principal curvature.

The rotation matrix is simply a projection into the coordinate frame at $\mathbf{n}$. Its rows are therefore simply $\mathbf{t}$, $\mathbf{b}$ and $\mathbf{n}$. The change of the rotation operator can be shown to be

$$R[\mathbf{n}(\mathbf{x} + \alpha\mathbf{t})] \approx \begin{pmatrix} \mathbf{t} + \alpha\kappa\mathbf{n} \\ \mathbf{b} \\ \mathbf{n} - \alpha\kappa\mathbf{t} \end{pmatrix} = R[\mathbf{n}(\mathbf{x})] + \alpha\kappa P \qquad P = \begin{pmatrix} \mathbf{n} \\ 0 \\ -\mathbf{t} \end{pmatrix}. \quad (60)$$

Therefore,

$$L(\mathbf{x}, R[\mathbf{n}(\mathbf{x} + \alpha\mathbf{t})]\omega) = L(\mathbf{x}, R\omega + \alpha\kappa P\omega). \quad (61)$$

It can be easily verified by directly taking dot-products that $R\omega$ and $P\omega$ are orthogonal. Therefore, $P\omega$ lies in the tangent space to $R\omega$, and we can define $\mathbf{l} = P\omega / \parallel P\omega \parallel$, with a corresponding definition for $\mathbf{m}$ using the P derived from $\mathbf{b}$ instead of $\mathbf{t}$. By extension, we can define $\mathbf{u} = R^{-1}\mathbf{l}$ and $\mathbf{v} = R^{-1}\mathbf{m}$. Finally, let $\mu = \parallel P\omega \parallel$.

With these judicious choices for directions and tangent frames, the algebra becomes simple without loss of generality, and

$$\frac{\partial}{\partial\alpha} L(\mathbf{x}, R[\mathbf{n}(\mathbf{x} + \alpha\mathbf{t})]\omega) = \mu\kappa L_l(\mathbf{x}, R\omega). \quad (62)$$

The 3D case now becomes very close to 2D or flatland, with

$$L^s_t = L_t + \mu_1\kappa_1 L_l \quad L^s_b = L_b + \mu_2\kappa_2 L_m, \quad (63)$$

where we have explicitly written the two principal curvatures as κ_1 and κ_2. The final inverse rotation in step 4 can be treated in the same way.

Reflection (step 2): If R is a (now fixed, independent of $\mathbf{x}$) reflection, it simply negates the tangents $\mathbf{u}$ and $\mathbf{v}$, so those gradients (and evaluation location) should be negated, as in 2D, with $L^m_u = -L^s_u$ and $L^m_v = -L^s_v$.

Convolution (step 3): Unlike in the 2D flatland case, we must differentiate the BRDF kernel, rather than the more elegant approach of considering gradients of the lighting. Briefly, the 3D convolution equation for a 1D radially symmetric BRDF is

$$B^s(\mathbf{x}, \theta) = \int_\Omega L^m(\mathbf{x}, \omega)\rho(\theta\cdot\omega)\, d\omega \quad = \quad L^m \otimes \rho. \quad (64)$$

The spatial gradients can proceed as in 2D, with for example

$$B^s_t(\mathbf{x}, \theta) = \int_\Omega L^m_t(\mathbf{x}, \omega)\rho(\theta\cdot\omega)\, d\omega \quad = \quad L^m_t \otimes \rho. \quad (65)$$

For the angular gradients, as before we have to define tangent frames, with

$$B^s_u(\mathbf{x}, \theta) = \frac{\partial}{\partial\alpha} B^s(\mathbf{x}, \theta + \alpha\mathbf{u}) = \frac{\partial}{\partial\alpha} \int_\Omega L^m(\mathbf{x}, \omega)\rho(\theta\cdot\omega + \alpha\mathbf{u}\cdot\omega)\, d\omega, \quad (66)$$

which can easily be simplified, using the derivative ρ' of the 1D BRDF ρ to

$$B^s_u(\mathbf{x}, \theta) = \int_\Omega L^m(\mathbf{x}, \omega)\rho'(\theta\cdot\omega)(\mathbf{u}\cdot\omega)\, d\omega. \quad (67)$$

Appendix C: Second Order Light Field Analysis

We differentiate Equation 25 directly. One tricky issue is that $\theta_r = 2n(x) - \theta$ depends on both x and θ. Therefore,

$$h(x, \theta) = f(x, \theta_r) \;\Rightarrow\; h_x(x, \theta) = f_x(x, \theta_r) + 2\cdot n_x \cdot f_\theta(x, \theta_r) \qquad h_\theta(x, \theta) = -f_\theta(x, \theta_r). \quad (68)$$

Using this relation, we can differentiate the convolutions,

$$\begin{aligned} \left[(L_x \otimes \rho)(x, \theta_r)\right]_x &= (L_{xx} \otimes \rho)(x, \theta_r) + 2n_x(L_{x\theta} \otimes \rho)(x, \theta_r) \\ \left[(L_x \otimes \rho)(x, \theta_r)\right]_\theta &= -(L_{x\theta} \otimes \rho)(x, \theta_r) \\ \left[(L_\theta \otimes \rho)(x, \theta_r)\right]_x &= (L_{x\theta} \otimes \rho)(x, \theta_r) + 2n_x(L_{\theta\theta} \otimes \rho)(x, \theta_r) \\ \left[(L_\theta \otimes \rho)(x, \theta_r)\right]_\theta &= -(L_{\theta\theta} \otimes \rho)(x, \theta_r) \end{aligned} \quad (69)$$

Now, it is easy to derive the result in Equation 29 by (somewhat laborious) differentiation, where we have omitted the parameters (x, θ_r) of evaluation for brevity.

Appendix D: Visibility Analysis

We first elaborate on the curved surface flatland case in Fig. 9b. Assume the blocker object has an instantaneous radius of curvature r. In this case, if we move a distance dx on the surface, the point of tangency will also move. In general, if the original tangent point was $(0, d)$, the new point will be $(-r\sin\alpha d\alpha, d + r\cos\alpha d\alpha)$. This is derived from trigonometry, noting that the length of the arc is $r\,d\alpha$. Now, the coordinates of the new surface point are $(x + dx, 0)$, and we also know that $x = d\tan\alpha$. Hence,

$$\tan(\alpha + d\alpha) = \frac{(x + dx) - (-r\sin\alpha d\alpha)}{d + r\cos\alpha} = \frac{d\tan\alpha + dx + r\sin\alpha d\alpha}{d + r\cos\alpha d\alpha}. \quad (70)$$

We can simplify, multiplying by the denominator, and keeping only first order terms,

$$d\tan\alpha + d\sec^2\alpha d\alpha + r\sin\alpha d\alpha = d\tan\alpha + dx + r\sin\alpha d\alpha, \quad (71)$$

which, upon subtracting $d\tan\alpha + r\sin\alpha d\alpha$ from both sides, can be simplified to exactly the same form as Equation 36. This result is also confirmed by the 3D derivation for α_u (which is essentially the flatland 2D case) in Equation 45.

We now consider Sec. 7.1, extending the curved surface derivation to polygonal blockers. For a polygonal blocker, the occlusion is defined by an extremal line with unit vector $\mathbf{k}$ (the 3D generalization of the extremal point in 2D). Our expression also works for mesh approximations of curved surfaces, where $\mathbf{k}$ can be considered a tangent in the local frame ($\mathbf{k}$, $\mathbf{n}$ and $\mathbf{w}$ form an orthonormal coordinate frame).

For α_u, we are essentially considering the 2D case with a point blocker. Therefore $\mathbf{dp} = \mathbf{0}$ in Equation 42, and by equating components along $\mathbf{c}$ (since $\mathbf{w}$ is orthogonal to $\mathbf{c}$ and $\mathbf{u} = \cos\alpha\mathbf{c} + \sin\alpha\mathbf{w}$), we obtain $\alpha_u = -\cos\alpha/D$ as expected.

For α_v, $\mathbf{dp}$ must lie along $\mathbf{k}$. In this case, $\mathbf{c}$, $\mathbf{w}$ and $\mathbf{v}$ form a coordinate frame. Hence, we can consider the coordinates of $\mathbf{dp}$ in this frame, requiring the condition that $(\mathbf{dp}\cdot\mathbf{c})/(\mathbf{dp}\cdot\mathbf{v}) = \mathbf{k}\cdot\mathbf{c}/\mathbf{k}\cdot\mathbf{v}$. From Equation 42, since $\mathbf{c}$, $\mathbf{w}$ and $\mathbf{v}$ form a coordinate frame, $\mathbf{dp}\cdot\mathbf{c} = D d\alpha$ and $\mathbf{dp}\cdot\mathbf{v} = dv$. Therefore,

$$\frac{D\,d\alpha}{dv} = \frac{\mathbf{k}\cdot\mathbf{c}}{\mathbf{k}\cdot\mathbf{v}} \;\Rightarrow\; \alpha_v = \frac{1}{D}\frac{\mathbf{k}\cdot\mathbf{c}}{\mathbf{k}\cdot\mathbf{v}}. \quad (72)$$

Interestingly, this result reduces to Equation 46, if $\mathbf{k}$ is orthogonal to both $\mathbf{n}$ and $\mathbf{w}$, such as for a polygonal approximation to a curved object. In that case, $\mathbf{k}\cdot\mathbf{c} = -\mathbf{n}\cdot\mathbf{v}$ and $\mathbf{k}\cdot\mathbf{v} = \mathbf{n}\cdot\mathbf{c}$. This indicates that we obtain a consistent result, whether we use an actual curved object, or a polygonal approximation to it.