A First-Order Analysis of Lighting, Shading, and Shadows

RAVI RAMAMOORTHI, DHRUV MAHAJAN, and PETER BELHUMEUR
Columbia University

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 article, 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 to 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. [2005] 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. Third, we understand the effects of soft shadows, computing accurate visibility gradients, and generalizing previous work to arbitrary curved occluders. As one practical application, our visibility gradients can be directly used with conventional ray-tracing methods in practical gradient interpolation methods for efficient rendering. Moreover, our theoretical framework can be used to adaptively sample images in high-gradient regions for efficient rendering.

Categories and Subject Descriptors: I.3.0 [**Computer Graphics**]: General; [**Image Processing and Computer Vision**]: General

General Terms: Theory

Additional Key Words and Phrases: Gradients, reflectance, shadows, Fourier analysis

ACM Reference Format:

Ramamoorthi, R., Mahajan, D., and Belhumeur, P. 2007. A first-order analysis of lighting, shading, and shadows. *ACM Trans. Graph. 26*, 1, Article 2 (January 2007), 21 pages. DOI = 10.1145/1186644.1186646 http://doi.acm.org/10.1145/1186644.1186646

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 nonuniform 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 article, 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—*spatial* lighting variation over a flat object surface due to close sources, as well as *angular* variation in lighting at a point from different directions. Shading also depends on geometric effects, such as the object's curvature, which determine how much the surface normal or orientation changes between neighboring points. Material properties are also important, since shading is effectively a convolution with the object BRDF [Basri and Jacobs 2001, 2003; Ramamoorthi and Hanrahan 2001, 2004]. 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 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 Figure 5).

For analysis of lighting and shading, 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, as well as fast and accurate gradient-based interpolation of visibility (Figure 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 (Section 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.

—*Analysis of first-order terms:* In Section 4.3, we determine the gradients for shading on a curved object lit by general spatially and directionally-varying lighting. We combine the basic shading steps in Section 4.1, augmenting them with *nonlinear*

This work was supported in part by NSF Grants 0085864, 0325867, 0305322, 0430258, and 0446916, and a Sloan Research fellowship.

Authors' address: R. Ramamoorthi, D. Mahajan, and P. Belhumeur, Computer Science Department, Columbia University, 450 Computer Science Bldg., 500 W 120 St., New York, NY 10027; email: {ravir,dhruv,belhumeur}@cs.columbia.edu.

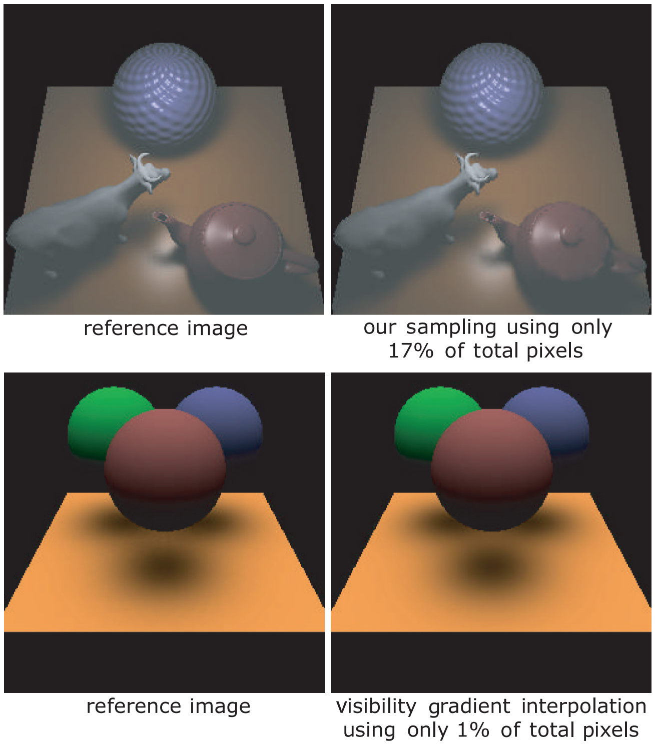

reference image
our sampling using only 17% of total pixels
reference image
visibility gradient interpolation using only 1% of total pixels

Fig. 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 the image pixels. More details are in Figures 13 and 15.

transformations like *bump mapping* (Section 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 (Section 5.2). Moreover, we show how to extend the first-order analysis to second-order Hessians (Section 5.4).

—*Analysis of visibility gradients:* We derive new analytic expressions for soft shadow gradients in Sections 6 and 7, extending the work of Arvo [1994] and Holzschuch and Sillion [1998] to arbitrary curved or polygonal blockers. Moreover, our formulation is local, based only on analyzing angular discontinuities in visibility at a single spatial location.

—*Practical applications:* In Section 8, we use our analytic visibility gradients for efficient and accurate rendering of soft shadows using gradient-based interpolation. As shown in Figures 1 and 13, accurate results can sometimes be obtained by explicitly evaluating only 1% of the pixels. Section 9 (Figures 1 and 15) applies our analysis to efficient rendering by adaptively sampling images using a metric which is 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.

The rest of this article is organized as follows. After a review the of previous work in Section 2, Section 3 reviews the basic preliminaries and mathematical operations of light transport. We then develop our theoretical analysis. First, Section 4 conducts a gradient analysis for light reflection from curved surfaces in flatland. Section 5 discusses extensions to 3D, as well as image gradients and second-order analysis. Thereafter, Section 6 analyzes soft shadow gradients in 2D. Section 7 generalizes these results to 3D, deriving the first general formulae for visibility gradients from curved occluders. Sections 8 and 9 describe practical applications to gradient-based shadow interpolation and gradient-based adaptive image sampling, respectively. Finally, Section 10 compares the theoretical derivations to previous work on Fourier and shadow analysis methods, and Section 11 concludes the article and discusses future work.

2. PREVIOUS WORK

This research builds on a substantial body of previous work on analyzing light transport in a number of different 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, 2004] develop a convolution framework for reflection on curved surfaces with general materials using spherical harmonics. The special case of Lambertian reflection was also analyzed concurrently by [Basri and Jacobs 2001, 2003]. 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. Indeed, our initial analysis of the steps of light reflection from a curved surface (Section 4.1) is inspired by their approach. In Section 3.1 (Figure 2), we directly compare Fourier and gradient analysis in terms of basic mathematical operators (the end of Section 10 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 nonlinear effects like bump mapping, while Fourier analysis always requires a finite neighborhood and linearization.

Wavelet analysis. Wavelets have been another popular tool for efficient computation 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 Section 3.1 and Figure 2). Thus, wavelets seem more useful for efficient practical computation, rather than for deriving theoretical insights.

Differential and gradient analysis of light reflection. 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 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 the differentials of individual ray paths as certain parameters (like viewpoint or location on the image plane) vary. By contrast, we seek to determine how gradients over the entire light field transform. Most importantly, this article is more focused on theoretical analysis, understanding the nature of shading variation by considering various gradient terms. We are

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

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

optimistic that our analysis can be used to derive new theoretical bounds and practical algorithms for previous methods.

Visibility analysis and shadow gradients. Shadows are one of the most important visual features and have received significant theoretical attention. Durand et al. [2002] have developed a full characterization of visibility events in terms of the visibility complex. Soler and Soler and Sillion [1998] and Ramamoorthi et al. [2005] have characterized special cases as convolutions. Stark et al. [1999] derived analytic formulae for the shadowed irradiance in polygonal scenes by explicitly evaluating an integral over the contour of receivers. 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] computed 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,

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

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 article, 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 significant interaction between them. For example, angular variation in the lighting often leads to spatial variation in the shading on a curved object.

For much of the article, the derivations are carried out in the 2D plane, or flatland, for clarity and simplicity. In this domain, the gradient is essentially equivalent to partial derivatives along spatial and angular dimensions, which we give formulae for throughout the work. While the 3D extensions (detailed in Sections 5.1, 7, and Appendix B) are more complicated algebraically, and in some cases involve the definition of appropriate unit vectors for formally relating partial derivatives to gradients, much the same results are obtained. Our analysis is applied practically to evaluation of soft shadows from curved blockers in 3D (Section 8), and to efficient rendering of 3D scenes (Section 9).

We will be analyzing various parts and generalizations of Eq. (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 Section 4 onwards, we will be more concrete, using notation closer to that of Eq. (1). The partial derivatives will be denoted with subscripts—for example, $f_x(x,\omega) = \partial f(x,\omega)/\partial x$. In Section 3.1, we will also obtain insight by comparing the forms of basic mathematical operations for first-order and Fourier analysis—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, written $H(\Omega) = F(\Omega) \otimes G(\Omega)$, where $\otimes$ stands for convolution. For gradients,

$$\bigtriangledown h = f \bigtriangledown g + g \bigtriangledown f. \tag{2}$$

Integration. Consider $h(x) = \int f(x,\theta)\, d\theta$, where, for example, f may denote the lighting premultiplied 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, that is, the Ω_x axis, so $H(\Omega_x) = F(\Omega_x, 0)$. For first-order analysis,

$$h_x = \int f_x(x,\theta)\, d\theta. \tag{3}$$

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

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

where the convolution is only over the angular coordinate.

[1] By symmetry, $h_\theta = f_\theta \otimes g$ in Eq. (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 this case, we must use $f \otimes g_\theta$ (see Appendix B). However, Eq. (4) is still accurate for flatland, and can be used in practice even for 3D sampling.

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,

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

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

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

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

Nonlinear transformations. Finally, we come to nonlinear transformations. These are seldom considered in analyses of light transport 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 that $h(\mathbf{u}) = f(T(\mathbf{u}))$, where T is a general nonlinear 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 $\mathbf{u}$),

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

Implications. Besides relating Fourier and gradient techniques, direct application of these formulae simplifies many derivations, both in our article 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 these operations with gradients makes them well-suited to the analysis in this article. 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 theoretically or analytically. However, recent work shows there are often efficient computational wavelet methods for practical applications, such as triple product algorithms for fast multiplication [Ng et al. 2004; Clarberg et al. 2005], or the recent fast wavelet rotation method [Wang et al. 2006].

4. LIGHT REFLECTION FROM CURVED SURFACES IN 2D

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). Then, we analyze nonlinear transformations like normal or bump maps, and derive the combined gradient, including all effects. In Section 5, we describe a simple direct extension to 3D, and analyze the effects of individual shading terms. In Sections 4 and 5, we do not explicitly consider cast shadows, since visibility is analyzed in detail in Sections 6 and 7.

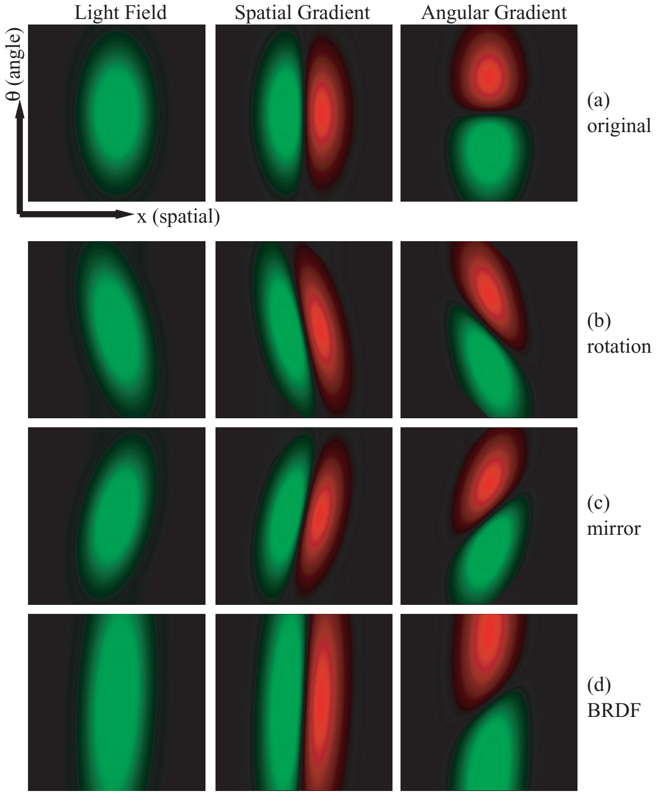


Fig. 3. The light field and its spatial and angular gradients, as a result of the various curved surface shading steps in Section 4.1. Green denotes positive values, and red denotes negative values.

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 Figure 3(a), 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, which have rarely been studied.

We assume the global coordinate frame is aligned such that the surface normal at 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 at $x = 0$, and we use positive signs for counterclockwise directions. Note that this linearization is mainly to simplify the exposition and not actually required, since we only need to consider the local gradients—this will be emphasized in the final results of Sections 4.2 and 4.3.

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. 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, which mixes spatial and angular coordinates, shearing the light field along the angular dimension, as seen in Figure 3(b). If $\mathbf{u} = (x, \theta)^T$, then $L^s(\mathbf{u}) = L(M\mathbf{u})$, with M being

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

Using the linear transformation theorem in Eq. (7), and premultiplying by M^T, as required,

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

This can be written out explicitly as

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

which can be easily verified by differentiating L^s directly, and where we have made the arguments for evaluation explicit. As noted earlier, the subscripts denote partial derivatives with $L^s_x = \partial L^s / \partial x$ and $L^s_\theta = \partial L^s / \partial \theta$. As seen in Figure 3(b), 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 preceding 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 $\mid L_x \mid \sim \mid kL_\theta \mid$. 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 (Eq. (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 Section 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 Figure 3(c) are therefore reflected about the θ-axis. The angular gradient is also negated

$$L^m_\theta(x, \theta) = -L^s_\theta(x, -\theta), \tag{12}$$

or more formally,

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

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

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

Note that we could also generalize the BRDF model beyond radially symmetric. The gradients would be essentially the same, but with the convolution replaced by a general integral using the general BRDF $\rho(\omega, \theta)$.

For determining the gradients of $B^s(x, \theta)$ in Eq. (14), we use the gradient convolution rule in Eq. (4),

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

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

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

Figure 3(d) shows the results of convolving with a Gaussian for ρ. This is analogous to a Phong or 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 Eq. (10),

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

Spatially-Varying materials. As a brief aside, we consider a generalization of Step 3 to spatially-varying materials. In this case,

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

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

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

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 this case, we denote E as the irradiance $\int L^s(x, \omega)\, d\omega$ so that $B^s(x) = E(x)\rho(x)$ and

$$B^s_x = E_x \rho + E \rho_x. \tag{20}$$

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 $\bigtriangledown B^s \sim E \bigtriangledown \rho$. In 3D, the direction of the gradient $\bigtriangledown B^s$ depends primarily on the direction of the texture gradient $\bigtriangledown \rho$, independent

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

of the lighting or irradiance, while the magnitude is scaled by E. This is one explanation for why the direction of the gradient is a good measure for lighting-insensitive recognition in computer vision [Chen et al. 2000].)

4.2 Gradients for Normal or Bump Maps

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 Eq. (8), with the Jacobian of the transform),

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

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 that 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 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 Eq. (10).

4.3 Light Field Gradients

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

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

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

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

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 we might expect.

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

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

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

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

This is an overall formula for shading gradients on a curved surface. While the initial derivation in Section 4.1 assumed that 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, as well as 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 Eq. (20). The spatial gradient B_x involves a modulation of Eq. (25) by the texture, and an additional term corresponding to the texture gradient modulated by the image intensity from Eq. (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 Eq. (19) of Section 4.1. The only additional term in Eq. (25) is $(L \otimes \rho_x)(x,\theta_r)$ in the spatial gradient B_x.

5. EXTENSION TO 3D AND IMPLICATIONS OF GRADIENT ANALYSIS

While our derivations have been in 2D, we can directly use the 3D analogs of these results for many rendering applications and analyses. In this section, we first discuss the direct extension to 3D, and then describe several implications of Eq. (25), as well as extensions to image gradients and second-order analysis.

In Appendix B, we generalize the four basic shading steps of Section 4.1 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, we show in Section 5.1 that it is both possible and simpler to directly use the straightforward 3D analogs of 2D results for real images—the formal 3D derivation in Appendix B is seen to have a very similar form.

5.1 Direct Extension to 3D

To directly extend Eq. (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, respectively) 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 Section 5.2 or the nonuniform image sampling in Section 9, we consider the net magnitude $(B_x^2 + B_y^2)^{1/2}$. These magnitudes are independent of which specific (orthogonal) directions are chosen 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 directions in the tangent plane to θ. Finally, we consider the net magnitude of these angular gradients in Sections 5.2 and 9.

	SV	DV	CV	CDV	SV+CDV
	$L_x \otimes \rho$	$L_\theta \otimes \rho$	$2n_x$	$2n_x(L_\theta \otimes \rho)$	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

Fig. 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.

5.2 Implications: Analysis of Gradient Terms

We now discuss some implications of Eq. (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. Figure 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 due to strong spatial variation in the lighting. On the other hand, for a mirror surface, like the chrome-steel sphere often used to estimate the illumination, we will see 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 be 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 from curved surfaces.

We illustrate some of these ideas with a simple didactic 3D scene in Figure 5 that includes a nearly (but not with zero curvature) flat table on which sit a diffuse, diffuse plus glossy, and bumpy sphere. The scene is lit by a moderately close area source. We use the direct 3D analogs of 2D gradients, as discussed in Section 5.1.

The gradient magnitudes are visualized on a log scale in Figures 5(b)–5(e). The spatial gradient of the (moderately near) lighting (Figure 5(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 5(e) plots the ratio 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 plus glossy sphere, there are parts where CDV and SV are comparable.

Finally, Figure 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 the CDV/SV increases, as expected.

5.3 Image Gradients

So far, we have found the light field gradients. We now seek the projected image $I(u)$ and image gradients I_u. We carry out our derivation only in 2D (with a projected 1D image $I(u)$), since the 3D projection to a 2D image is similar for u and v axes. 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 Eq. (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 a close viewer). Since there is no curvature or spatial lighting variation, the spatial light field gradient $B_x = 0$. In this case, for $u \ll \gamma$, the $\partial \theta / \partial u \approx \gamma^{-1}$, 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 that 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

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

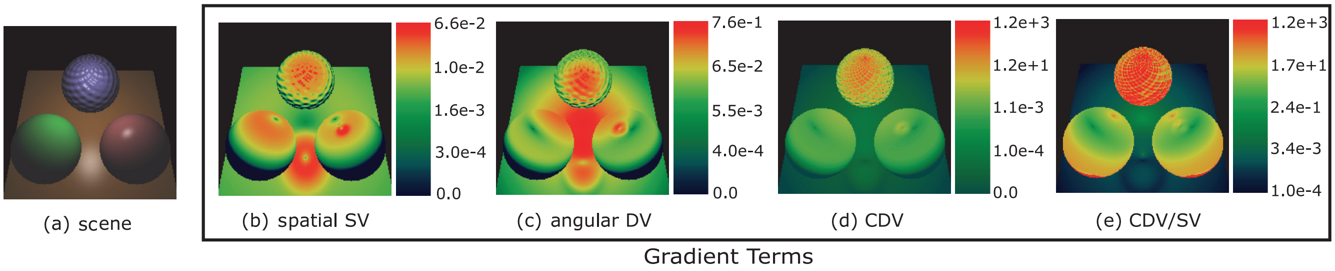


Fig. 5. A scene that shows various shading effects, including a nearly flat table, and diffuse, glossy, and bumpy spheres, lit by a moderately close area source (b), (c), and (d) show various terms of the gradients; (e) shows the ratio of curvature-weighted directional variation (CDV) and spatial variation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

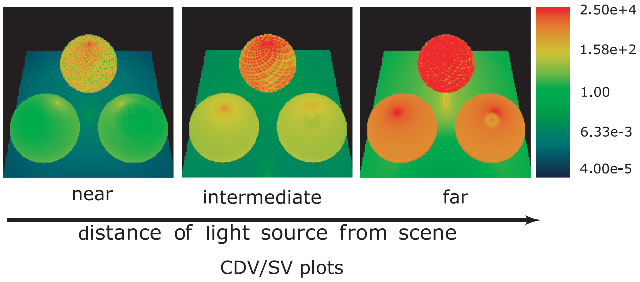


Fig. 6. Change in ratio of directional and spatial gradients as the light moves further away. For near lighting, SV is important, while the relative importance of the angular term (and CDV/SV) increases for far lighting.

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.

5.4 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 Eq. (25) to derive second-order terms, or Hessians,

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

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 Eq. (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.

6. FIRST-ORDER ANALYSIS OF SOFT SHADOWS IN 2D

So far, we have not explicitly considered the visibility term $V(x, \omega)$ in Eq. (1). Indeed, since visibility is a discontinuous function, its "derivatives" can be infinite. However, by taking the shading integral into account, we can derive finite analytic formulae for soft shadow gradients. 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 Section 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

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

Our previous formulae are still valid if we reinterpret L as the shadowed illumination L^{shad}, which is already premultiplied 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 Eq. (25), by the visibility V.

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 often been omitted in previous analyses and algorithms. In general, the shading gradients in these cases can be written as

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

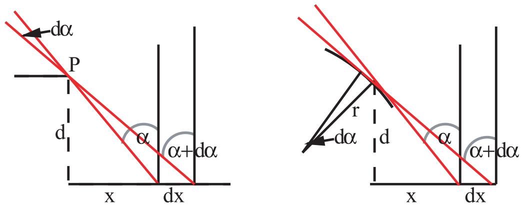
where T is a general transport operator encompassing lighting L and BRDF ρ. For simple convolution, $T(x, \theta, \omega) = L(x, \omega)\rho(x, \theta - \omega)$. The superscript in B^V indicates explicitly that we are considering the visibility gradient term (not gradients of the lighting or BRDF). 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 Dirac delta "functions" or distributions, being zero except at discontinuities, where they are infinite. However, the previous integral for $\nabla B(x, \theta)$ is still finite.[3]

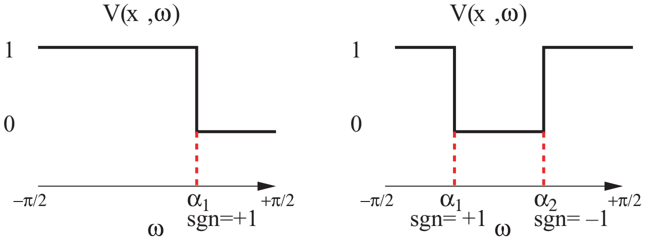
Figure 7 shows the 2D cases of polyline and curved blockers that we consider. For computing gradients, it suffices to consider the local region around an extremal blocker point P, summing over all

[3] 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) where 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.



(a) polyline (point blocker) (b) curved surface

Fig. 7. We consider the gradient of the blocked region $\alpha_x = d\alpha/dx$; (a) a polyline blocker, where the extremal ray intersects a single point P (b) a curved blocker.



(a) single discontinuity (b) multiple discontinuities

Fig. 8. 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.

such visibility discontinuities. From Figure 7, $V(x, \omega) = 1$ (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 $1 - H$.

In practice, a general visibility function at a point can have multiple discontinuities, as shown in Figure 8(b). These general visibility functions can be represented as a sum of the Heaviside functions for each discontinuity $\alpha_j(x)$, and we can consider the net gradient as the sum of the gradients due to each discontinuity. Therefore, in most of the remaining exposition, we analyze a single visibility discontinuity, or blocker, but our final formulae involve sums over all of the visibility discontinuities, and are fully general.

The derivative of the Heaviside function is the Dirac delta

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

Angular gradients. Plugging into Eq. (32), the angular gradient is simple: The delta function evaluates T at $\omega = \alpha(x)$,

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

where we sum over all discontinuities j of the visibility $V(x, \omega)$ for given x, with the appropriate sign sgn_j, as shown in Figure 8. For example, in Figure 8(a), $B_\theta = -T(x, \theta, \alpha_1(x))$. In Figure 8(b), $B_\theta = -T(x, \theta, \alpha_1(x)) + T(x, \theta, \alpha_2(x))$. Note that we only need to observe the visibility discontinuities at a single spatial location x to compute the gradients. If we numerically compute $V(x, \omega)$, such as by ray-tracing, it is easy to determine the discontinuities (simply adjacent angles ω where $V(x, \omega)$ differs) and apply the preceding equation.

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 polyline blocker, and therefore a single extremal point P, as in Figure 7(a). 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 Figure 7(b). Finally, we can write for the spatial gradient

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

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

Finally, note that Eqs. (35) and (37) no longer involve delta functions or infinities, 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.

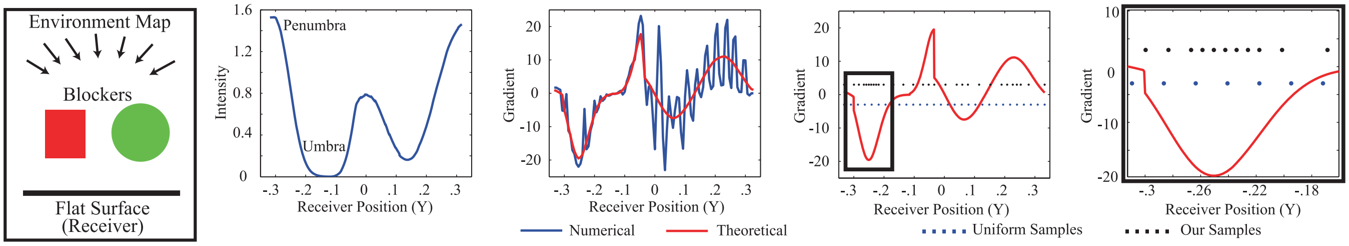
Implications and discussion. These results have several implications. First, the gradient varies inversely with the distance to the blockers. Second, we must sum over all visibility discontinuities at a given spatial location to find the net gradient. This confirms the use of the harmonic mean of blocker distances 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 connection between spatial and angular effects for visibility, as for the earlier shading formulae. Eq. (37) for the spatial gradient at a point involves knowledge only of the angular discontinuities α_j in visibility at this spatial location. Unlike numerical differentiation, we do not need to consider neighboring spatial points, making implementation efficient and robust.

6.3 Evaluation and Verification in 2D or Flatland

To evaluate and verify the accuracy of our formulae, we consider a simple flatland scene of a flat diffuse surface in Figure 9, shadowed by a box (rectangle) and a circle, lit by a distant environment map (with Gaussian variation in lighting, being maximal directly overhead). The cosine term can be folded into the lighting, and T corresponds almost directly to the illumination, allowing us to focus on visibility effects. Figure 9(b) shows that the shading on the receiver has complex umbra and penumbra regions.

To use the preceding analytic formulae, we need to know the visibility discontinuities $\alpha_j(x)$ and other relevant information, such as the distance to the blocker D_j. Note that different discontinuities, j, need not correspond to the same object, and occluding objects can overlap (this is discussed in greater detail in our practical implementation in Section 8.1). The visibility discontinuities can be determined in two ways: For simple objects, the values of $\alpha_j(x)$ can be determined analytically in *object-space*, by considering each object. More generally, we can numerically sample $V(x, \omega)$ at a number of angular locations ω_k, as in standard *image-space* ray-tracing. The discontinuities $\alpha_j(x)$ are then simply those ω_k, where



(a) scene description (b) intensity on receiver (c) numerical vs. theoretical gradient (d) our sampling scheme vs. uniform sampling

Fig. 9. 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.

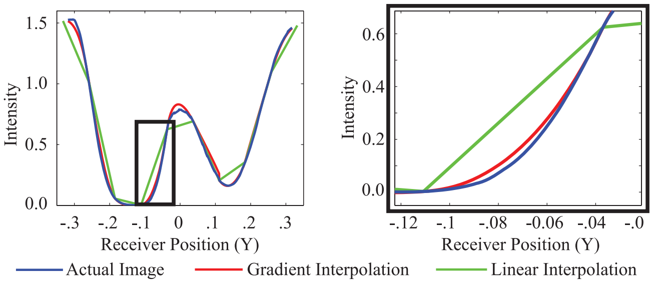


Fig. 10. Comparison of gradient interpolation versus standard linear interpolation for shadows (with uniform sampling). As seen in the closeup on the right, using gradients gives significantly greater accuracy.

adjacent values differ (the visibility boundaries or discontinuities). These numerical α_j are then used in the analytic formulae given by Eqs. (35) and (37). This approach is "semianalytic," since visibility discontinuities α_j (and the related D_j) are determined numerically (but with very high precision, since we typically use a large number of samples, such as 256 or 512, in ω). We also make comparisons with the purely numerical computation of the visibility gradient by standard numerical differentiation of the intensities, without using our analytic formulae at all.

As seen in Figure 9(c), the numerical intensity gradient is very noisy. Because visibility is a binary function, the intensity (and local visibility extrema $\alpha_j(x)$) usually change in a stairstep 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 accurately matches the correct value. Note that at the scale of this figure, fully analytic and "semianalytic" gradients were identical, and are not shown separately.

Figure 9(d) explores applications to adaptive image sampling, using an adaptive sampler based on our accurate gradients which places more samples in high-gradient shadow regions, leading to a much more accurate reconstruction than uniform sampling.

Our calculations will perhaps be most useful in terms of accurate visibility gradients for gradient-based interpolation in algorithms liked those of Ward and Heckbert [1992], which currently often ignore visibility gradients. Figure 10 compares gradient interpolation to standard linear interpolation (for uniform sampling); it is clear that gradient interpolation gives much higher accuracy.

7. FIRST-ORDER ANALYSIS OF SOFT SHADOWS IN 3D

Unlike for curved surface reflection in Section 4, it is not possible to use the direct 3D analog of the 2D results (as in Section 5.1). 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.

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. As in flatland, we are considering a single discontinuity $\alpha(x, y, \phi)$ here for ease of exposition, but we can address general visibility simply by summing the gradients for multiple visibility discontinuities—our final formulae will include this explicit summation.

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, 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,

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

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 11 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 we move 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 $(\sin\alpha, 0, \cos\alpha)$. Here, $\mathbf{c}$ is the transverse direction given in the (u, v, z) frame as $(\cos\alpha, 0, -\sin\alpha)$.

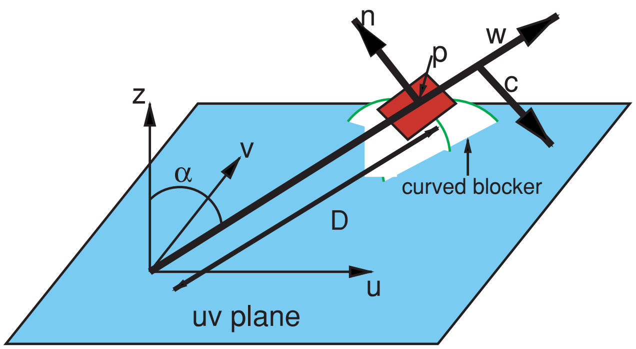


Fig. 11. Local geometry for calculation of α_u and α_v in 3D visibility.

Moreover, **w** and **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 **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 **m** (where for us, **m** is either the u- or v-axis), the point of intersection will move $\mathbf{dp} = dr\,\mathbf{m}$,

$$\mathbf{dp} = (Dd\alpha)\,\mathbf{c} + (dD)\mathbf{w} + (dr)\,\mathbf{m}. \qquad (42)$$

The new point of intersection still lies on the tangent plane to the surface at **p** (shown in red in Figure 11) to first-order. Therefore, $\mathbf{dp}\cdot\mathbf{n} = 0$, where **n** is the surface normal at **p**,

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

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

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

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 $\mathbf{w}\cdot\mathbf{n} = 0$,

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

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 Eq. (44),

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

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 **n** and the vectors **c** and **v**.

7.2 Gradients of Net Visibility

Now, we consider gradients of the net visibility, that is, 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,

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

To compute gradients B_x, we use the visibility gradients V_x from Eq. (39). The delta function causes the θ integral to be evaluated at $\alpha(x, y, \phi)$. Summing over multiple discontinuities as usual (so that we can handle general visibility and multiple discontinuities for each ϕ),

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

where we use the long form $\partial\alpha_j/\partial x$, instead of α_x, to avoid confusion with the subscript j for the jth discontinuity. For simplicity, we assume that 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.

As in flatland, evaluation of the previous equation requires knowing the locations of visibility discontinuities α_j, and related quantities (such as the distance to the blocker at those points). For simple objects and scenes, these can be determined analytically in object-space by considering each object. In general, we can use standard image-space ray-tracing to determine the α_j numerically with high precision, and plug into the preceding formula for a "semianalytic" evaluation, just as in the flatland case.

It is important to note that, unlike in flatland, we must still do a 1D integral over ϕ, which in practice is performed using adaptive numerical integration. Note, however, that the delta functions in visibility have already been resolved, so that this ϕ integral is now well behaved, and efficient to evaluate accurately.[4]

Efficiency and accuracy. Our gradient computation is very efficient and accurate. The gradients depend only on discontinuities, not the full visibility function. In particular, Eq. (48) is only a 1D integral, as opposed to the image evaluation in Eq. (47), which requires 2D integration. Therefore, gradients can be computed very efficiently relative to the actual image. This is true regardless of whether we can analytically determine the discontinuities $\alpha_j(x, y, \phi)$ for each ϕ (as for simple objects in our examples) or need to use a "semianalytic" approach wherein the α_j are found numerically from the full visibility calculation.

Figure 12 shows a plane with a single sphere blocker. For a single sphere, it is easy to derive an independent analytic expression for Eq. (47) to test accuracy. Indeed, with 400 samples in ϕ, the gradients computed by our method in Figure 12 are nearly exact for 24-bit RGB images. On the other hand, even with 900 samples, the 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 in Figure 12.

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,

[4] The only challenging numerical issue occurs if $\partial\alpha_j/\partial x$ is very large. The denominator $\mathbf{c}\cdot\mathbf{n}$ in Eq. (46) can be small when the u-z plane is (nearly) tangent to the surface (so that 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 Figure 12.

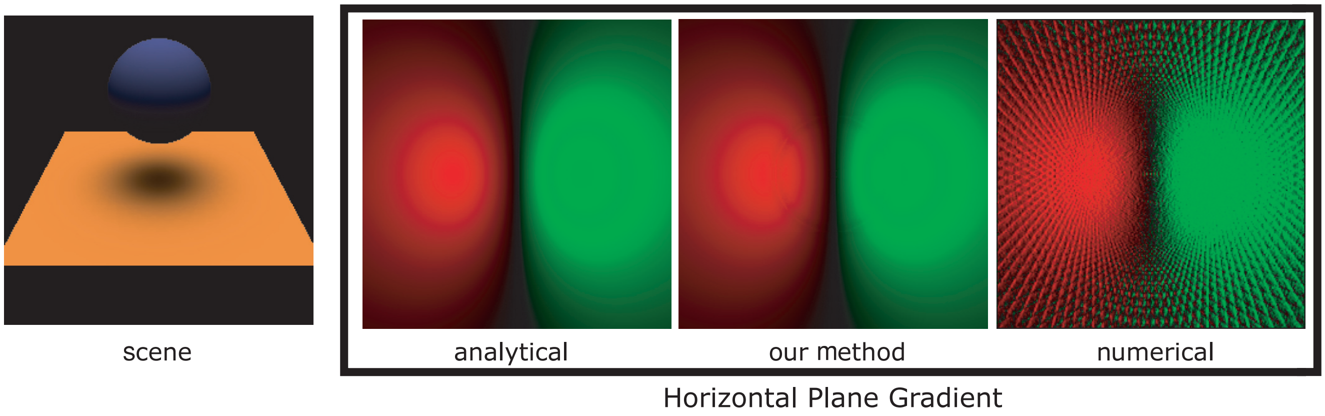


Fig. 12. Accuracy of net visibility gradients. From left-to-right, the simple scene, the independent analytic gradient for B_x (for a head-on view of the ground plane), the accurate results computed by our method, and the very noisy results obtained by numerical differentiation.

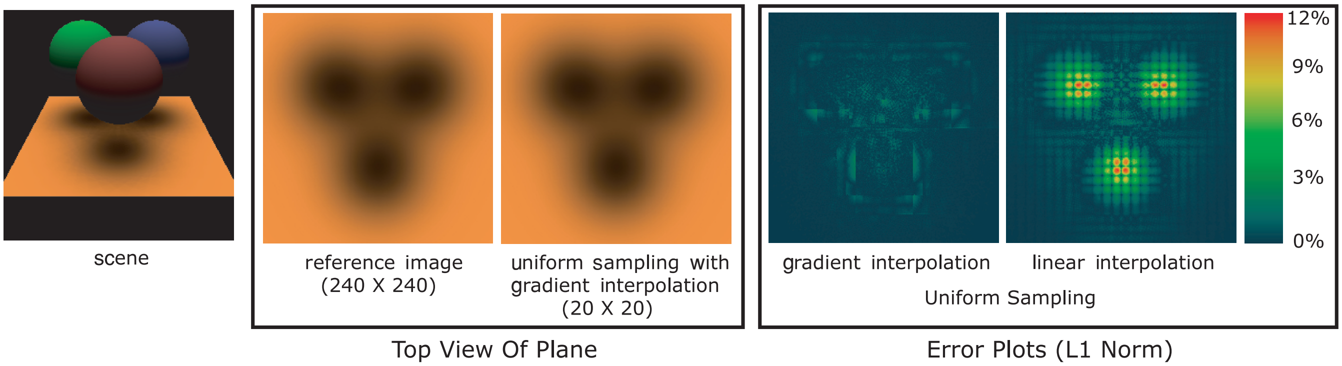


Fig. 13. Comparison of gradient interpolation to linear interpolation. The leftmost image shows the scene. The other images interpolate shading directly on the ground plane (shown in a head-on view), using a uniform 20 × 20 sampling (less than 1% of total pixels). Gradient interpolation with our accurate gradients is very high-quality, and substantially lowers the error from simple linear interpolation.

and the gradients are only due to soft shadows

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

These visibility gradients can also be trivially 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.

8. PRACTICAL APPLICATION: GRADIENT-BASED INTERPOLATION OF SOFT SHADOWS

We now discuss some simple prototype applications, suggesting how gradient analysis may be applied to efficient rendering. In this section, we focus on gradient-based interpolation of soft shadows, while the next section addresses adaptive gradient-based image sampling. We emphasize that these are proof-of-concept results, indicative of the potential of our analysis, with more work required to develop production-quality solutions. Nevertheless, they indicate the potential benefits of a first-order analysis and we expect that many other applications in forward and inverse rendering may benefit from our analysis.

To illustrate our gradient-based interpolation method for efficient rendering of soft shadows, we use the scene in Figure 13, which is lit by an environment map (with the usual Gaussian lighting variation). While the geometry is simple (three spheres on a ground plane), our main focus is on the shadows, which exhibit complex penumbra regions on the ground plane from curved blockers.

The standard approach to rendering computes the image shading and visibility at each pixel (or each location on the receiver plane). Our goal is to instead compute the image and its gradients accurately at a small number of pixels (Figure 13 demonstrates accurate results using only 1% of the pixels). The gradients can then be used to do interpolation very efficiently, instead of explicitly computing the shading at other pixels. This approach is enabled by our accurate analytic formula for soft shadow gradients, and is seen in Figure 13 to be significantly more accurate than simple linear interpolation. Thus, we can use our gradient formulae to very efficiently render complex shadows with standard gradient interpolation methods.

8.1 Implementation

The shading on the receiver plane in Figure 13 is first computed accurately at a coarse spatial grid of only $1/10$ the resolution of the image in both directions (or only 1% of the total pixels). Simultaneously with computing the image, we also compute the gradients at these points, as discussed next. At each pixel on the coarse grid, we use high-resolution $(\theta \times \phi) = 100 \times 100$ environment and visibility sampling.

To compute the spatial gradients for environment map lighting, we can use Eq. (49). Consider a given spatial location (pixel) (x, y) and azimuthal angle ϕ. We will need to determine all discontinuities $\alpha_j(x, y, \phi)$ along θ. Since we sum the gradients due to each discontinuity, we can handle general visibility.

Object-Space analytic method. One approach is to analytically determine the visibility discontinuities $\alpha_j(x, y, \phi)$ for all objects in the scene. We use this object-space approach for our scene since the calculation is simple for a sphere, as well as many other simple parametric objects.[5] We then need to merge the information from all objects in the scene, to determine the net visibility discontinuities observed. For example, let the first sphere be visible between $\alpha_0 = 0$ and $\alpha_1 = \pi/4$, and the second between $\alpha_0 = \pi/6$ and $\alpha_1 = \pi/3$. The receiver point sees only two (and not four) discontinuities in its net visibility at $\alpha_0^* = 0$ (from the first object) and $\alpha_1^* = \pi/3$ (from the second object). Since we are simply dealing with 1D segments (along θ) of visible and blocked regions, this merge is relatively straightforward.

Image-Space semianalytic method. For a scene with several complex objects, it may not be feasible to obtain the α_j analytically, nor practical to do so separately for each object. Fortunately, our ray tracer already has an image-space notion of visibility, computing $V(\theta_k; x, y, \phi)$ for a discrete set of values θ_k for the spatial location (x, y) and azimuthal angle ϕ. We can now easily find the visibility discontinuities α_j (i.e., those θ_k where $V(\theta_k)$ and $V(\theta_{k+1})$ differ). While this calculation is numerical (and hence the overall method is semianalytic), it is fairly precise, since, the visibility is sampled finely. We can then use these α_j values (and auxiliary information like distances to the blockers) in our analytic gradient formulae.

Computation. Once the discontinuities α_j have been determined, we can use the formula for $\partial\alpha_j/\partial x$ from Section 7.1. Determining the lighting term $L(\alpha_j(x, y, \phi), \phi)$ in Eq. (49) is also straightforward. Note that we have thus far considered the calculation for a single value of ϕ. Finally, we need to numerically integrate or add up the contributions due to all ϕ values.

Efficiency and accuracy. The gradient computation at a pixel introduces minimal overhead relative to computing the shading (and shadows) at this pixel. For the object-space analytic technique, finding the visibility discontinuities analytically is very efficient, and does not even require the tracing of rays for visibility. For the image-space method, the same visibility rays as for shading are used to determine discontinuities, so the overhead is minimal—*no new ray tracing is required in either method to compute gradients*. In both cases, the final 1D integral over ϕ is much faster than the 2D lighting integral for the image.

In fact, both object-space analytic and image-space semianalytic methods for computing gradients are much more efficient than straightforward numerical differentation. Standard numerical differentiation would require explicit image computation at neighboring pixels to take finite differences. This computation at nearby pixels (and ray-tracing to compute shadows) is a constant multiplier over computing the shading at a single pixel. On the other hand, analytic or semianalytic gradient evaluation imposes minimal overhead, and, as seen in Figure 12, is significantly more accurate.

Interpolation and results. We now have the image values and gradients computed over a coarse grid. Any of a number of standard gradient-based interpolation schemes [Ward and Heckbert 1992; Annen et al. 2004] can be used to determine the values of intermediate image pixels (we construct a quadratic Bezier curve). As seen in Figure 13, gradient-based interpolation is much more accurate than standard linear interpolation, and one to two orders of magnitude more efficient than explicitly evaluating the shading at each pixel.

8.2 Limitations and Evaluation

Accurate gradient computations in our framework require a fairly dense sampling of the ϕ integrals. In addition, we must have accurate values at the image pixels to enable proper interpolation. This requires a dense sampling of the incident illumination at those pixels that are explicitly evaluated (we use a 100×100 uniform $\theta \times \phi$ sampling, which requires a number of shadow rays to be traced). By contrast, standard image synthesis techniques that compute each pixel independently, without calculating gradients, often do not require very high accuracy (since the final display only has 8 bits). Therefore, they can often use many fewer shadow rays at a pixel [Agarwal et al. 2003]. However, given the dramatically reduced number of pixels our method needs to explicitly evaluate, it is still likely to lead to a speedup.

An issue with all interpolation techniques is that they can be inaccurate when the initial set of samples is too far apart, so that there are important discontinuities that are missed. In these cases, gradient (or linear) interpolation will smooth over the discontinuities. A fully robust system would therefore need to identify significant image-space geometric discontinuities (perhaps using some recent global visibility analysis methods, such as Durand et al. [2002]), and place samples to avoid interpolating across them.

In general, we believe gradient-based soft shadow interpolation provides a powerful tool for fast rendering and analysis of penumbra regions for which general analytic formulae have not previously been available.

9. PRACTICAL APPLICATION: GRADIENT-BASED IMAGE SAMPLING FOR FAST RENDERING

We now develop a simple algorithm that adaptively samples images for efficiency. We seek to place more image samples in high-contrast regions with large gradients. The analysis in Sections 4 and 5 immediately confers insight, and we could develop a number of simple heuristics to place samples where the SV or CDV terms are expected to be large. We would not even need to formally evaluate the convolutions, and 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, that is, I_u from Eq. (26), with camera terms in Eq. (27), and light field gradients

[5] For more complex geometric objects, a computational geometry plane-sweep algorithm may still be possible, since we are essentially intersecting with a plane given by the spatial location (x, y), angle ϕ, and perpendicular direction.

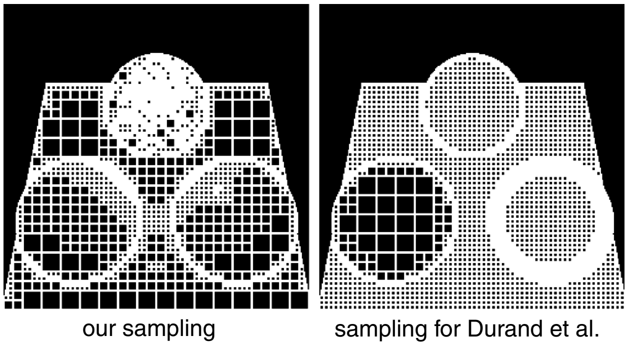


Fig. 14. Comparison of sample locations (using 10% of the image pixels) with our gradient algorithm, and the approximation of Durand et al. [2005], for the scene in Figure 5. We correctly place more samples on the bumpy sphere, and in highlights, while Durand et al. focus on relatively smooth regions 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.*

in Eq. (25). As in Section 5.2, we use direct 3D analogs of 2D or flatland gradients.

9.1 Implementation and Results

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 subsquare, since the area is reduced to 1/4. We also always subdivide along object boundaries or silhouettes in image space so as 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, these 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] (Eq. (20) of their article), 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 band-limit for the BRDF (based on Ramamoorthi and Hanrahan [2001], we use $\Omega_\rho = \sqrt{6s}$, where s is the Phong exponent). Note that for uniformity, we do not compare to the actual implementation in Durand et al. [2005], but only to the aforementioned metric calculated within a single framework. For example, in these comparisons both methods work adaptively, while the actual implementation in Durand et al. [2005] uses a simpler (but potentially more costly) two-pass approach. On the other hand, Durand et al. [2005] actually compute curvature on the basis of finite differences within a normal buffer that, in practice, could take bump maps into account—however, their theoretical development cannot conceptually fully handle bump mapping, and we therefore use the preceding metric.

Figure 14 compares the sample distribution for our gradient metric and that from the previous formula for the scene in Figure 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 to be given as part of the input, and in practice, are precomputed using the TriMesh library [Rusinkiewicz 2004]. For general BRDFs, we simply replace convolutions with the explicit shading calculation for gradients, as we already must do for the image intensity. Finally, for shadows, we modulate shading integrals for the gradients by visibility, as well as for the image itself. Shadow testing is the expensive operation in a ray tracer, especially with soft shadows in complex lighting. Since the same visibility samples are used for image and gradient calculations, our approach introduces minimal overhead.

Figure 15 shows a similar scene, now with 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, blurs the bumps considerably, as does the method of Durand et al. [2005]. 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] have 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 total wall-clock running time, the reference image (at a resolution of 512 × 512), which evaluates each pixel separately, took 75.9 minutes (our ray-tracing software is not optimized). If we simply perform a gradient calculation at *every* pixel so as to check the overhead involved for just the gradient calculation itself (without any adaptive sampling), the additional time is only 1.2 minutes, or less than 2%. The primary overhead in computing the adaptively sampled image actually comes from quad-tree construction, and takes about 6 minutes (8%).

On this specific scene with our adaptive image sampling evaluating 17% of the pixels, the bumps are easier to ray-trace for shadows (a simple sphere) and we actually obtain a slightly superlinear speedup in total wall-clock running time (11.5 minutes for gradient sampling, or a 6.6× improvement over the reference 75.9 minutes), 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].

9.2 Comparison to Alternative Approaches

There are a number of other adaptive image sampling strategies for ray-tracing which we make comparison to briefly here, giving some insights into the strengths and limitations of our approach.

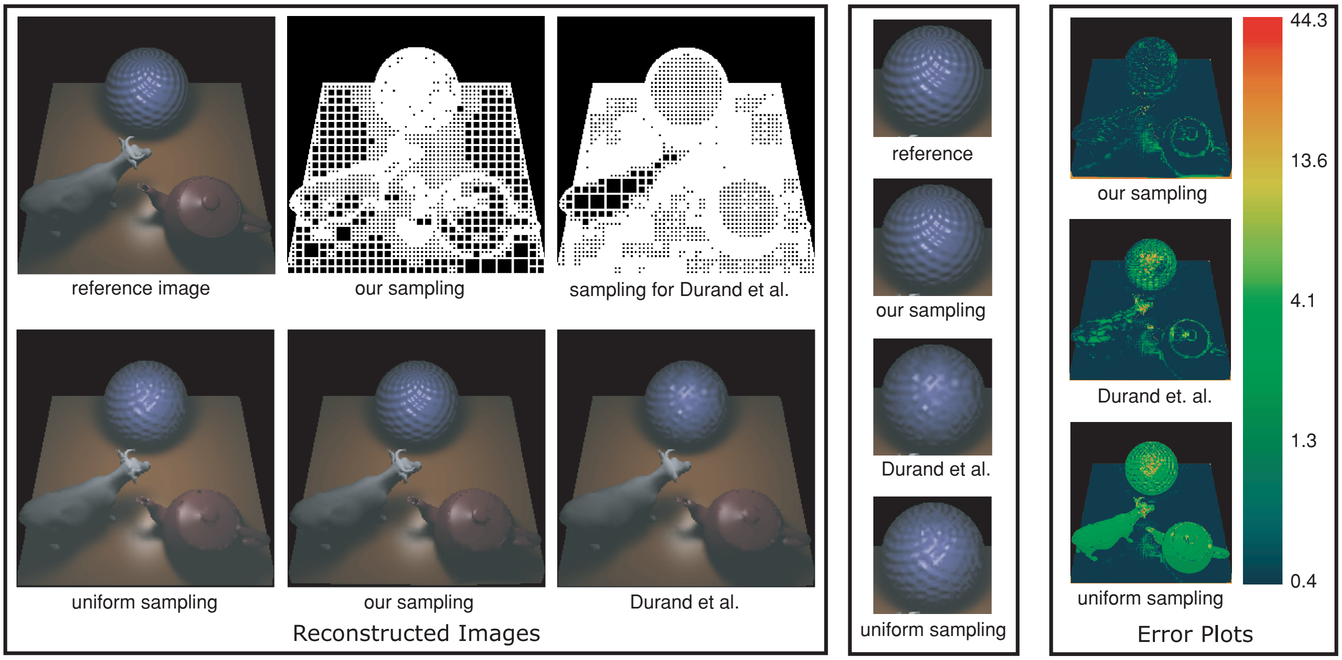


Fig. 15. Comparison of various image sampling strategies for a 3D scene with complex geometry, reflectance, and shadows. Our method, based on gradient magnitude, places samples in regions of high-curvature and shading change, closely matching the reference with only 17% of the effective pixels. By contrast, both uniform sampling and Durand et al. [2005] 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.

As we note that many of these strategies effectively compute gradients numerically in their inner loops, and we believe our method can be integrated into many of these techniques to improve their performance or accuracy.

Adaptive image sampling. Perhaps the most common approach is to consider the variance between nearby samples, adaptively adding more samples in high-variance regions. In effect, this is equivalent to estimating the magnitude of the gradient in a local region numerically. As a simple example, in flatland, the variance or difference in intensity between two neighboring points is just the central difference formula for the derivative at their midpoint. In many respects, this strategy is very similar to our adaptive quad-tree construction for sampling, differing primarily in the numerical (rather than analytically-based) computation of gradients, and confirms the use of a gradient-based sampling metric.

A potential issue with numerical estimation of variance is shown in Figure 16. As an illustrative example, we consider a flatland bumpy surface that is essentially a sine wave. Since the intial samples are placed coarsely, they do not match the frequency of the bumps, and the numerically computed gradient is much lower than the actual (or analytically determined) value. In effect, this is the standard aliasing problem whereby a lower sampling rate effectively sees an incorrect low-frequency or aliased version of the original signal. Note that for adaptive image sampling, we cannot afford to initially sample densely enough to capture all high-frequency details—indeed, doing so would defeat the purpose of adaptive sampling. By contrast, our approach computes gradients analytically and is usually accurate, regardless of the initial grid resolution.

Figure 17 compares our analytic gradient approach to numerically-based adaptive sampling in 2D. We consider bumps whose frequency varies, increasing from left-to-right. This is a very common real-world scenario because of the perspective viewing of surfaces, wherein the frequency content increases for distant regions. Our approach reconstructs a fairly accurate version of the original signal, while numerical adaptive sampling has regions of large error in high-frequency areas. The form of these errors is also interesting, and corresponds to an aliased or low-frequency representation of parts of the image.

Finally, Figure 18 compares images of the bumpy sphere for our method and numerical adaptive sampling with the same number of samples. It can be seen that our gradient-based image sampling technique performs somewhat better in preserving the sharpness, especially in the highlights. Moreover, the sampling patterns show that our approach can focus precisely on the fastest varying regions, even within bumps.

Directional coherence maps. In recent years, several more sophisticated adaptive sampling and image analysis methods have been developed, which we discuss briefly. The most notable work is perhaps that of Guo [1998], which exploits information on edges in the image to construct directional coherence maps for sampling. A number of recent works have shown the importance of identifying edge information [Bala et al. 2003; Sen et al. 2003] in other rendering contexts. In effect, Guo [1998] uses the magnitude of the gradient (estimated numerically from small image blocks) to guide image sampling. However, he goes further in identifying and respecting edge directions. This essentially boils down to computing the gradient *direction*. With a simple extension to our gradient calculation to compute the direction (i.e., B_x and B_y separately), we believe our approach can be integrated into Guo's framework; using an analytically-based gradient could, in many cases, provide more accurate estimates of the behavior of images in local blocks.

Note that for a scene like Figure 15, there are no significant edges except at object boundaries (which we already take into account), so directional coherence maps are unlikely to provide a

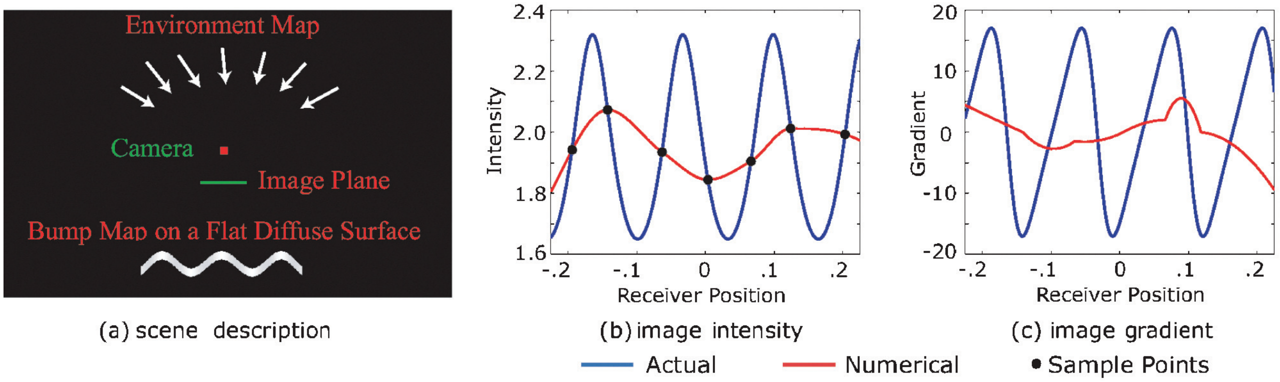


Fig. 16. Potential "aliasing-like" inaccuracies in numerical gradient computation for adaptive image sampling with a coarse initial sample placement. The numerical image gradients in red (effectively used for adaptive image sampling) are significantly lower-frequency than the original signal.

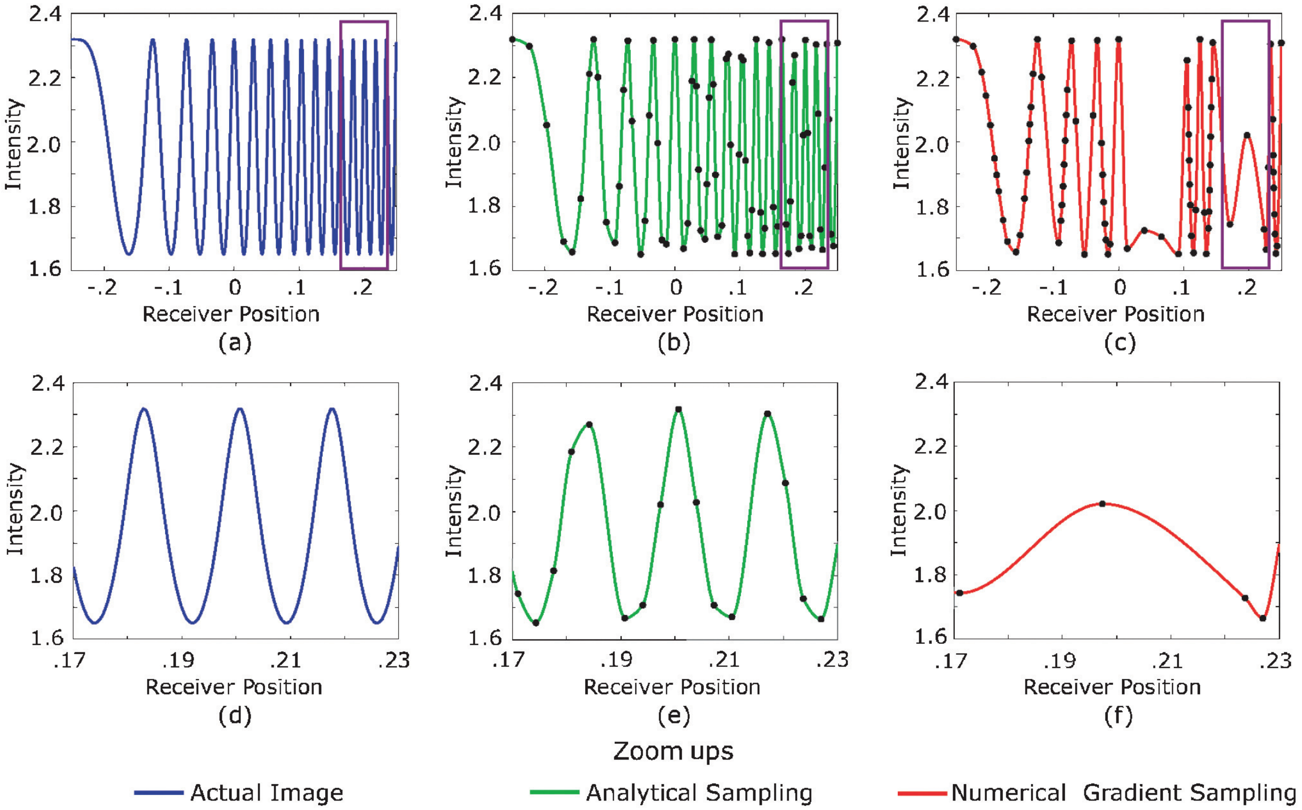


Fig. 17. Common practical situation of spatially-varying frequencies. From left-to-right, the original image or function sampled using our approach with analytic gradients, and using numerical adaptive sampling. The closeups in the bottom row show that our method is quite accurate, while numerical adaptive sampling makes significant errors, and often fails to refine its coarse or "low-frequency" estimate.

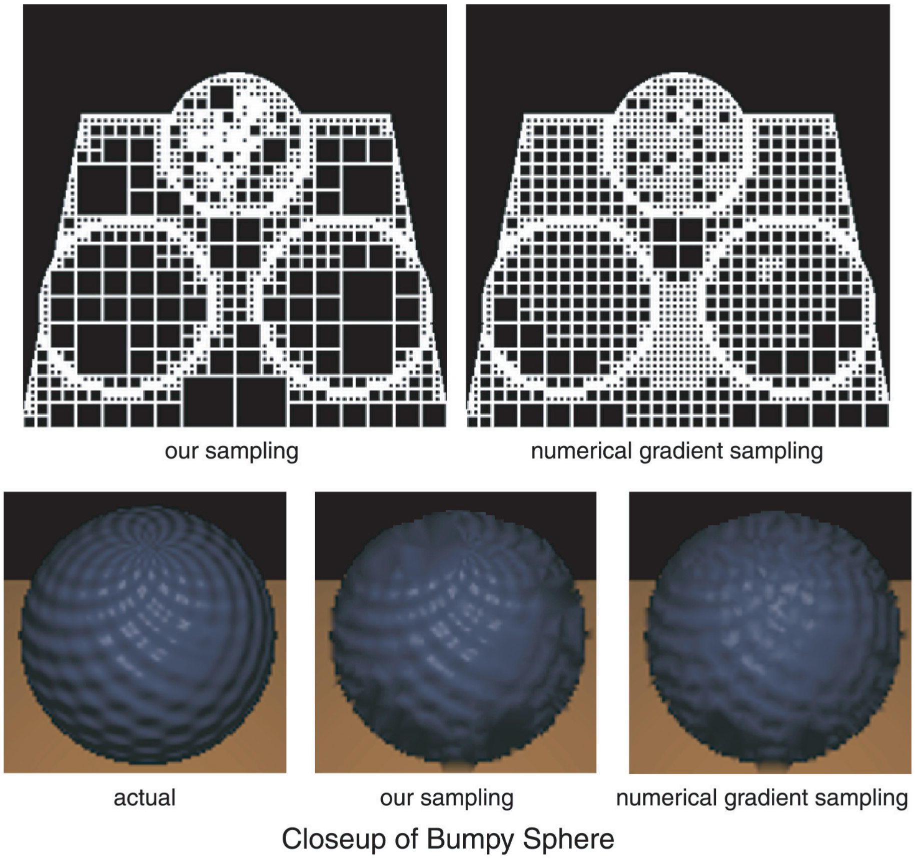


Fig. 18. Comparison of sampling pattern (top) and closeups of bumpy sphere (bottom) for our approach and numerical adaptive image sampling, with the same number of adaptively placed samples. Our method produces a somewhat sharper pattern on the bumps, especially for the specular highlights on the sphere, since adaptive image sampling somewhat underestimates gradients of high-frequency bump map.

significant speedup over standard numerical adaptive sampling. Indeed, coherence-based techniques like Guo [1998] are not intended for bumpy surfaces in complex environment illumination with soft shadows.

Coherence in shadow rays. Much recent work has addressed the exploitation of coherence in shadow ray-tracing [Agrawala et al. 2000; Ben-Artzi et al. 2006], which is usually the dominant cost in rendering with complex lighting. Such coherence-based shadow accelerations are not readily applicable in our case, since the lighting can neither be assumed distant nor a small area source. Moreover, we also consider complex reflectance effects. An interesting future direction is to see if our shadow gradient computations can be used to generalize shadow coherence techniques.

9.3 Limitations and Evaluation

Our current approach has some theoretical and practical limitations. From a theoretical perspective, a gradient or linear interpolation technique for reconstruction can already deal with first-order variation in the image, even if the gradient magnitude is large. Therefore, we should actually be using a second-order analysis to guide image sampling (such as the formulae in Eq. (29). In practice, however, higher-order derivatives are well correlated with the gradient, and gradient-based image sampling is much simpler and more stable than using higher-order schemes. Indeed, as just discussed, most adaptive image sampling methods effectively use some form of gradient sampling. Practically, we need to assess the robustness of our method on more complex scenes. This would be most interesting in the context of a more sophisticated predictor, such as directional coherence maps. Our adaptive sampling metric also does not currently consider shadow gradients; using our visibility gradient formulae may make for more robust sampling in shadowed regions. Nevertheless, we believe our proof-of-concept gradient-based sampler is a viable alternative for adaptive image sampling techniques, and has significant potential for integration into more advanced sampling strategies.

10. DISCUSSION AND COMPARISON

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

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 Section 3.1. For instance, consider the per point rotation in Step 1 with $L^s(x, \theta) = L(x, \theta + kx)$, and the gradient formula in Eq. (10). The effects on the Fourier spectrum are found from the linear transformation theorem in Eq. (6). To apply this, we need M^{-T} and its determinant. Note that M^{-T} is simply M^T in Eq. (9), with k replaced by $-k$, and $det(M) = 1$,

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

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 Eqs. (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 Eq. (25), our final results do not need these additional reparameterization steps.

Advantages of first-order analysis. Frequency analysis has many benefits and insights in a variety of applications. In particular, one obtains a full frequency spectrum, as opposed to the first term of a Taylor expansion. However, for the theory in this article, first-order analysis easily separates the different factors responsible for shading variation. Moreover, we can accurately consider nonlinear transformations like bump maps, aspects of camera transformations, and visibility. The derivations and insights in Sections 4.2–5.4 would not be easy to obtain from Fourier analysis.

A concrete example is the relative performance of gradient-based adaptive image sampling versus a frequency-based metric, in Section 9. In fact, we can consider the sampling metric in Durand et al. [2005] (Eq. (20) of their paper) a special case of Eq. (28), with the assumption of a distant camera with distant lighting and no bump mapping. In this 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 band-limit $((L_\theta \otimes \rho) \sim \Omega_\rho)$ derived from the BRDF, we would recover the image sampling criterion in Eq. (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 convolutions like $L_\theta \otimes \rho$, whereas Fourier methods do not easily lend themselves to practical computation.

Visibility. Since visibility is nonlinear, 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. [2005]. 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.

11. 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 relationship 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. Third, we develop novel results for visibility gradients, which generalize much previous work on analysis of soft shadows. In terms of practical applications, we show how to use gradients to adaptively sample images for efficient rendering, and demonstrate efficient gradient-based visibility interpolation.

We see this article as an important theoretical step in the analysis of light transport. In the future, we are hopeful that methods from this work 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 metrics 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.

More generally, we see this article as an important step towards a comprehensive framework for analyzing light transport using a variety of established mathematical tools. It complements and extends the recent frequency domain and signal processing analyses that use Fourier and spherical harmonic representations. In the future, we expect considerable further effort on a theoretical analysis of lighting and reflection, using other basis representations and operators to give a rich fundamental understanding of the computational nature of light transport.

APPENDIXES

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 $I = \sqrt{-1}$),

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

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

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

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

$$\begin{aligned} F(\mathbf{\Omega}) \;=\; & \mid det(M) \mid \exp[-2\pi I \Omega^T \mathbf{c}] \\ & \int h(\mathbf{v}) \exp\left[-2\pi I (M^T \mathbf{\Omega})^T \mathbf{v}\right] d\mathbf{v} \qquad (54) \\ & = \mid det(M) \mid \exp[-2\pi I \Omega^T \mathbf{c}] H(M^T \mathbf{\Omega}) \\ & \Rightarrow \;\; H(\mathbf{\Omega}) = \frac{\exp[2\pi I \mathbf{\Omega}^T M^{-1} \mathbf{c}]}{\mid det(M) \mid} F(M^{-T} \mathbf{\Omega}). \end{aligned}$$

For gradients, we can write

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

which can be rearranged to Eq. (7).

APPENDIX B: LIGHT REFLECTION FROM 3D SURFACES

We briefly extend the light-curved surface interaction steps in Section 4.1 to 3D. We define local tangent frames and partial derivatives with respect to motion in a particular direction.[6] At $\mathbf{x}$, the tangents are $\mathbf{t}$ and $\mathbf{b}$, which, along with the normal $\mathbf{n}$, form a coordinate frame. Similarly, $\boldsymbol{\omega}$ 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 partial derivative such as L^s_t is defined as $L^s_t = (\partial/\partial\alpha) L^s(\mathbf{x} + \alpha\mathbf{t}, \boldsymbol{\omega})$.

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

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

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

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

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

For the spatial gradients, we write

$$\begin{aligned} L^s_t &= \frac{\partial}{\partial\alpha} L(\mathbf{x} + \alpha\mathbf{t}, R[\mathbf{n}(\mathbf{x} + \alpha\mathbf{t})]\boldsymbol{\omega}) \\ L^s_b &= \frac{\partial}{\partial\alpha} L(\mathbf{x} + \alpha\mathbf{b}, R[\mathbf{n}(\mathbf{x} + \alpha\mathbf{b})]\boldsymbol{\omega}), \qquad (58) \end{aligned}$$

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

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

where the first term is simply the spatial gradient of the original light field, as in 2D. The second term, which 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 (respetively), 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

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

Therefore,

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

It can be easily verified by directly taking dot-products that $R\boldsymbol{\omega}$ and $P\boldsymbol{\omega}$ are orthogonal. Therefore, $P\boldsymbol{\omega}$ lies in the tangent space to $R\boldsymbol{\omega}$, and we can define $\mathbf{l} = P\boldsymbol{\omega} / \parallel P\boldsymbol{\omega} \parallel$, 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 $\mu = \parallel P\boldsymbol{\omega} \parallel$.

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

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

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

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

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 these gradients (and the 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

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

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

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

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

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

[6] Note also that since these directions are unit vectors rather than spherical angular coordinates, the gradient is well-defined based simply on these partial derivatives, without further normalization.

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

$$B_u^s(\mathbf{x}, \boldsymbol{\theta}) = \int_\Omega L^m(\mathbf{x}, \boldsymbol{\omega})\rho'(\boldsymbol{\theta} \cdot \boldsymbol{\omega})(\mathbf{u} \cdot \boldsymbol{\omega})\, d\boldsymbol{\omega}. \qquad (67)$$

APPENDIX C: SECOND-ORDER LIGHT FIELD ANALYSIS

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

$$\begin{aligned} h(x,\theta) = f(x,\theta_r) \;\Rightarrow\; & h_x(x,\theta) = f_x(x,\theta_r) + 2 \cdot n_x \cdot f_\theta(x,\theta_r) \\ h_\theta(x,\theta) = -f_\theta(x,\theta_r). & \end{aligned} \qquad (68)$$

Using this relation, we can differentiate the convolutions,

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

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

APPENDIX D: VISIBILITY ANALYSIS

We first elaborate on the curved surface flatland case in Figure 7b. Assume that 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,

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

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

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

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

We now consider Section 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 Eq. (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 Eq. (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,

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

Interestingly, this result reduces to Eq. (46), if $\mathbf{k}$ is orthogonal to both $\mathbf{n}$ and $\mathbf{w}$, such as for a polygonal approximation to a curved object. In this 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.

ACKNOWLEDGMENTS

This work has a very long history, going back to initial discussions on a differential framework for reflection with Pat Hanrahan more than seven years ago. Sameer Agarwal provided an excellent sounding board for the early ideas in this article, and went well out of his way to read several ill-defined working drafts. We thank Eitan Grinspun for some very helpful early discussions on geometric derivatives for computation of visibility gradients. Todd Zickler and Nolan Goodnight read early drafts of the article, and provided valuable feedback. The derivation of the Fourier linear transformation theorem in Appendix A is largely due to Brian Curless. Finally, we wish to thank the anonymous reviewers for their detailed comments on a very mathematically involved article, and for many helpful suggestions in improving the exposition and analysis.

REFERENCES

AGARWAL, S., RAMAMOORTHI, R., BELONGIE, S., AND JENSEN, H. 2003. Structured importance sampling of environment maps. *ACM Trans. Graph. 22*, 3, 605–612.

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 *Proceedings of the EuroGraphics Symposium on Rendering*. 331–336.

ARVO, J. 1994. The Irradiance Jacobian for partially occluded polyhedral sources. In *SIGGRAPH 94*. 343–350.

BALA, K., WALTER, B., AND GREENBERG, D. 2003. Combining edges and points for interactive high-quality rendering. *ACM Trans. Graph. 22*, 3, 631–640.

BASRI, R. AND JACOBS, D. 2001. Lambertian reflectance and linear subspaces. In *Proceedings of the International Conference on Computer Vision*. 383–390.

BASRI, R. AND JACOBS, D. 2003. Lambertian reflectance and linear subspaces. *IEEE Trans. Pattern Anal. Mach. Intell. 25*, 2, 218–233.

BEN-ARTZI, A., RAMAMOORTHI, R., AND AGRAWALA, M. 2006. Efficient shadows from sampled environment maps. *J. Graph. Tools 11*, 1, 13–36.

CHAI, J., TONG, X., SHAN, S., AND SHUM, H. 2000. Plenoptic sampling. In *Proceedings of the SIGGRAPH Conference*. 307–318.

CHEN, H., BELHUMEUR, P., AND JACOBS, D. 2000. In search of illumination invariants. In *Comput. Vision Pattern Recogn.* 254–261.

CHEN, M. AND ARVO, J. 2000. Theory and application of specular path perturbation. *ACM Trans. Graph. 19*, 4, 246–278.

CHRISTENSEN, P. 2002. Note # 35: Ambient occlusion, image-based illumination, and global illumination. In *Photorealistic RenderMan Appl. Notes*.

CLARBERG, P., JAROSZ, W., MOLLER, T., AND JENSEN, H. 2005. Wavelet importance sampling: Efficiently evaluating products of complex functions. *ACM Trans. Graph. 24*, 3, 1166–1175.

DURAND, F., DRETTAKIS, G., AND PUECH, C. 2002. The 3D visibility complex. *ACM Trans. Graph. 21*, 2, 176–206.

DURAND, F., HOLZSCHUCH, N., SOLER, C., CHAN, E., AND SILLION, F. 2005. A frequency analysis of light transport. *ACM Trans. Graph. 25*, 3, 1115–1126.

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

GORTLER, S., SCHRODER, P., COHEN, M., AND HANRAHAN, P. 1993. Wavelet radiosity. In *Proceedings of the SIGGRAPH Conference*. 221–230.

GUO, B. 1998. Progressive radiance evaluation using directional coherence maps. In *Proceedings of the SIGGRAPH Conference*. 255–266.

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

IGEHY, H. 1999. Tracing ray differentials. In *Proceedings of the SIGGRAPH Conference*. 179–186.

NG, R. 2005. Fourier slice photography. *ACM Trans. Graph. 25*, 3, 735–744.

NG, R., RAMAMOORTHI, R., AND HANRAHAN, P. 2004. Triple product wavelet integrals for all-frequency relighting. *ACM Trans. Graph. 23*, 3, 475–485.

RAMAMOORTHI, R. AND HANRAHAN, P. 2001. A signal-processing framework for inverse rendering. In *Proceedings of the SIGGRAPH Conference*. 117–128.

RAMAMOORTHI, R. AND HANRAHAN, P. 2004. A signal-processing framework for reflection. *ACM Trans. Graph. 23*, 4, 1004–1042.

RAMAMOORTHI, R., KOUDELKA, M., AND BELHUMEUR, P. 2005. A Fourier theory for cast shadows. *IEEE Trans. Pattern Anal. Mach. Intell. 27*, 2, 288–295.

RUSINKIEWICZ, S. 2004. Estimating curvatures and their derivatives on triangle meshes. In *Proceedings of the Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*. 486–493.

SEN, P., CAMMARANO, M., AND HANRAHAN, P. 2003. Shadow silhouette maps. *ACM Trans. Graph. 22*, 3, 521–526.

SOLER, C. AND SILLION, F. 1998. Fast calculation of soft shadow textures using convolution. In *Proceedings of the SIGGRAPH Conference*. 321–332.

STARK, M., COHEN, E., LYCHE, T., AND RIESENFELD, R. 1999. Computing exact shadow irradiance using splines. In *Proceedings of the SIGGRAPH Conference*. 155–164.

WANG, R., NG, R., LUEBKE, D., AND HUMPHREYS, G. 2006. Efficient wavelet rotation for environment map rendering. In *Proceedings of the Eurographics Symposium on Rendering*.

WARD, G. AND HECKBERT, P. 1992. Irradiance gradients. In *Proceedings of the 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 Trans. Graph. 25*, 3, 1196–1201.

Received June 2006; accepted October 2006