The Irradiance Jacobian for Partially Occluded Polyhedral Sources

James Arvo

Program of Computer Graphics*
Cornell University

Abstract

The irradiance at a point on a surface due to a polyhedral source of uniform brightness is given by a well-known analytic formula. In this paper we derive the corresponding analytic expression for the *irradiance Jacobian*, the derivative of the vector representation of irradiance. Although the result is elementary for unoccluded sources, within penumbrae the irradiance Jacobian must incorporate more information about blockers than either the irradiance or vector irradiance. The expression presented here holds for any number of polyhedral blockers and requires only a minor extension of standard polygon clipping to evaluate. To illustrate its use, three related applications are briefly described: direct computation of isolux contours, finding local irradiance extrema, and iso-meshing. Isolux contours are curves of constant irradiance across a surface that can be followed using a predictor-corrector method based on the irradiance Jacobian. Similarly, local extrema can be found using a descent method. Finally, iso-meshing is a new approach to surface mesh generation that incorporates families of isolux contours.

CR Categories and Subject Descriptors: I.3.3 [Computer Graphics]: Picture/Image Generation, I.3.5 [Computational Geometry and Object Modeling]: Geometric Algorithms.

Additional Key Words and Phrases: irradiance gradient, irradiance Jacobian, isolux contours, light field, mesh generation, vector irradiance.

1 Introduction

A perennial problem of computer graphics is the accurate representation of light leaving a surface. In its full generality, the problem entails both local reflection phenomena and the distribution of light reaching the surface. Frequently the problem is simplified by assuming polyhedral environments or Lambertian (diffuse) emitters and reflectors. With these simplifications the remaining challenges are in simulating interreflections and acurately modeling shadows and penumbrae from area light sources.

* 580 Engineering and Theory Center Building, Ithaca, New York 14853, email: arvo@graphics.cornell.edu

SIGGRAPH '94, July 24-29, Orlando, Florida

Many aspects of surface illumination have been studied in order to accurately model features of the reflected light. In previous work, Heckbert [8] and Lischinski et al. [12] identified derivative discontinuities to produce efficient surface meshes. Nishita and Nakamae [14, 15] located penumbrae in polyhedral environments, while Teller [19] computed antipenumbrae for sequences of portals. Ward and Heckbert [22] computed irradiance gradients from random samples of the environment to accurately interpolate irradiance functions. Drettakis and Fiume [5] also estimated gradients from samples and used them to guide subsequent sampling.

The present work introduces a new tool based on the concept of *vector irradiance*, a natural representation of irradiance defined at all points in space. The central contribution of the paper is a closed-form expression for the derivative of the vector irradiance, which we call the *irradiance Jacobian*. The new expression properly accounts for occlusion and subsumes the irradiance gradient as a special case.

In Section 2 we derive the irradiance Jacobian for polygonal sources of uniform brightness starting with an analytic expression for the vector irradiance. The same expression has been used in scalar form by Nishita and Nakamae [14] to accurately simulate polyhedral sources, and by Baum et al. [2] for the computation of form factors. Section 3 introduces a method for characterizing changes in the apparent shape of a source due to differential changes in the receiving point, which is the key to handling occlusions. In Section 4 basic properties of the irradiance Jacobian are discussed, including existence and the connection with gradients.

To illustrate the potential uses of the irradiance Jacobian, Section 5 describes several computations that employ irradiance gradients. We describe a method for direct computation of isolux contours, which are curves of constant irradiance on a surface. Each contour is expressed as the solution of an ordinary differential equation which is solved numerically using a predictor-corrector method. The resulting contours can then be used as the basis of a meshing algorithm. Finding local extrema is a related computation that can be performed using a descent method. Finally, Section 6 describes other potential applications of the irradiance Jacobian.

1.1 Radiometric Preliminaries

All radiometric quantities may be defined in terms of *radiance*: radiant power per unit projected area per unit solid angle [watts/m^{2}sr]. Any collection of light sources in an optical medium uniquely defines a non-negative radiance function over all spatial positions and directions. At the macroscopic level, this function completely specifies the distribution of radiant energy in the medium. We shall denote the radiance at the point r and in the direction ω by $L(r, \omega)$.

Central to the present work is a vector field $\Phi : \mathbb{R}^3 \to \mathbb{R}^3$ known

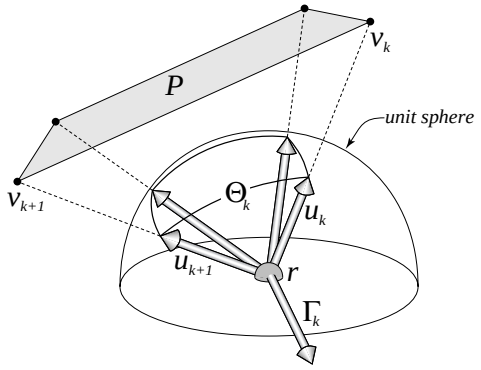


Figure 1: *(a) The vector irradiance at point r due to polygon P can be written in closed-form. (b) The contribution due to edge k is the product of the angle Θ_k and the unit vector Γ_k.*

as the *light field* [7]. In terms of radiance, the light field is given by

$$\Phi(r) \equiv \int_{\mathcal{S}^2} L(r,\omega)\,\omega\,d\sigma(\omega), \tag{1}$$

where $\mathcal{S}^2$ is the unit sphere in $\mathbb{R}^3$, and σ is the canonical measure on the sphere [3]. That is, $\Phi(r)$ is the integral over all unit vectors through r weighted by the radiance in each direction. At each point in space $\Phi(r)$ is a vector quantity known as the *light vector* [7] or the *vector irradiance* [17]; we adopt the latter terminology in this paper.

From vector irradiance we can derive other radiometric quantities, such as the *net flux* [watts/m^2]. At any point r on a hypothetical surface $\mathcal{M}$, the net flux $\phi(r)$ is the net flow of radiant energy across $\mathcal{M}$ per unit area [9]. By definition,

$$\phi(r) \equiv \int_{\mathcal{S}^2} L(r,\omega)\,\cos\theta\,d\sigma(\omega), \tag{2}$$

where θ is the angle of incidence relative to the surface. It follows that ϕ and Φ are related by

$$\phi(r) = -\Phi(r)\cdot\mathbf{n}(r), \tag{3}$$

where $\mathbf{n}(r)$ is the surface normal at the point r. The vector irradiance at r therefore defines a linear mapping that relates surface normals to net flux.

At real surfaces, the net flux accounts for energy arriving from the hemisphere above the surface, in which case $\phi(r)$ is called *irradiance*. Equivalently, the irradiance at a point r on a surface follows from equation (3) if the vector irradiance is computed using sources that are visible and lie above the tangent plane through r.

Although vector irradiance is a very general radiometric quantity defined in any optical medium, it is particularly useful in certain restricted settings. For instance, at diffuse receivers the reflectivity, surface orientation, and vector irradiance completely determine the reflected radiance at every point. Moreover, when the sources are polyhedral, the vector irradiance can be expressed in closed form.

1.2 Polyhedral Sources

For sources of uniform brightness, Φ can be expressed analytically for a number of simple geometries including spheres and infinite strips [7]. Polygonal sources are another important class with known closed-form expressions, and are the focus of this paper. Suppose P is a simple planar polygon in $\mathbb{R}^3$ with vertices $v_1, v_2, \ldots, v_n$. If

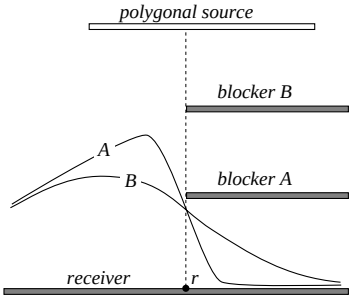


Figure 2: *The irradiance at point r due to source P is the same with either blocker, but the slopes of the irradiance curves are different.*

P is a diffuse source with constant emission M [watts/m^2], then the light field due to P is given by

$$\Phi(r) = \frac{M}{2\pi}\sum_{i=1}^{n}\Theta_i(r)\,\Gamma_i(r), \tag{4}$$

where $\Theta_1, \ldots, \Theta_n$ are the angles subtended by the n edges as seen from the point r, or equivalently, the arclengths of the edges projected onto the unit sphere about r. The vectors $\Gamma_1, \ldots, \Gamma_n$ are unit normals of the polygonal cone with cross section P and apex r. See Figure 1. For any $1 \le k \le n$ the functions Θ_k and Γ_k can be written

$$\Theta_k(r) = \cos^{-1}\left(\frac{v_k - r}{||\,v_k - r\,||}\cdot\frac{v_{k+1} - r}{||\,v_{k+1} - r\,||}\right), \tag{5}$$

and

$$\Gamma_k(r) = \frac{(v_k - r)\times(v_{k+1} - r)}{||\,(v_k - r)\times(v_{k+1} - r)\,||}, \tag{6}$$

where $||\cdot||$ is the Euclidean norm and $v_{n+1} \equiv v_1$. Equation (4) most commonly appears in scalar form [2, 6, 14]. With $M = 1$, the corresponding expression $-\Phi(r)\cdot\mathbf{n}(r)$ is the form factor between a differential patch at r and the polygonal patch P. This scalar expression was first derived by Lambert in the 18$^{\text{th}}$ century [18].

Because the light field is a true vector field, the vector irradiance due to multiple sources may be obtained by summing the contributions from each source individually. Thus, polyhedral sources can be handled by applying equation (4) to each face and summing the resulting vectors. Alternatively, when the faces have equal brightness, equation (4) can be applied to the outer contour of the polyhedron as seen from the point r [14]. Partially occluded sources are handled similarly by summing the contributions of all the visible portions. Determining the visible portions of the sources in polyhedral environments is analogous to clipping polygons for hidden surface removal [23].

The closed-form expression for vector irradiance in equation (4) provides an effective means of computing related expressions, such as derivatives. In the remainder of the paper we derive closed-form expressions for derivatives of the irradiance and vector irradiance due to polyhedral sources in the presence of occluders, and describe several applications.

2 The Irradiance Jacobian

The derivative DF of a differentiable function $F : \mathbb{R}^3 \to \mathbb{R}^3$ is represented by a 3×3 Jacobian matrix. We shall denote the Jacobian

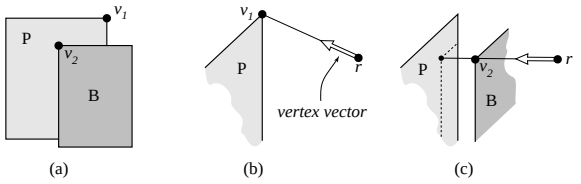


Figure 3: *(a) The view from r of the two types of "intrinsic" vertices. (b) The vertex vector for the unoccluded source vertex v_1. (c) The vertex vector for the blocker vertex v_2 in the interior of source P.*

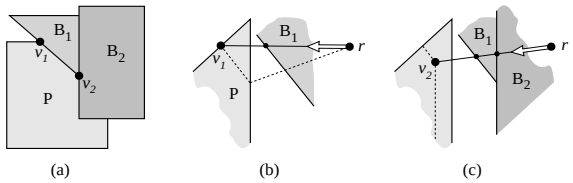


Figure 4: *(a) The view from r of the two types of "apparent" vertices. (b) The vertex vector for v_1 resulting from a blocker edge and a source edge. (c) The vertex vector for v_2 resulting from two blocker edges.*

matrix of F at r by $\mathbf{J}_r(F)$. That is,

$$\mathbf{J}_r(F) \equiv DF(r) = \left[\frac{\partial F_i(r)}{\partial x_j}\right]. \tag{7}$$

In this section we derive the Jacobian matrix of the vector irradiance Φ, or more briefly, the *irradiance Jacobian*. The obvious approach is to differentiate equation (4) with respect to the point r; a straightforward exercise easily performed by a symbolic manipulator. The result holds for unoccluded polygonal sources.

To see why the irradiance Jacobian is more difficult to compute when blockers are present, consider the arrangement in Figure 2. First, observe that the irradiance at the point r can be computed by applying equation (4) to the visible portion of the source, which is the same in the presence of either blocker A or blocker B. Although the resulting expression can be differentiated, this does not result in the irradiance Jacobian. Because the two blockers produce irradiance functions with different slopes at r, the irradiance Jacobians must also differ to account for blocker position.

To derive an expression that applies within penumbrae, we express $\Phi(r)$ in terms of *vertex vectors*, which correspond to vertices of the spherical projection of the polygon, as depicted in Figure 1. Vertex vectors may point toward vertices of two distinct types: *intrinsic* and *apparent*. An intrinsic vertex exists on either the source or the blocker, as shown in Figure 3. An apparent vertex results when the edge of a blocker, as seen from r, crosses the edge of the source or another blocker, as shown in Figure 4. We shall express $\mathbf{J}_r(\Phi)$ in terms of derivatives of the vertex vectors, viewing them as mappings from points in $\mathbb{R}^3$ to unit vectors. The derivative of a vertex vector is a 3×3 matrix, which we call the *vertex Jacobian*. Vertex Jacobians hide the geometric details of each vertex, yielding a relatively simple closed-form expression for $\mathbf{J}_r(\Phi)$.

Let $v_1', v_2', \ldots, v_m'$ be the vertices of P', the source P after clipping away portions that are occluded with respect to the point r. Without loss of generality, we may assume that P' is a single polygon; if it is not, we simply iterate over the pieces. The vertex vectors $u_1(r), u_2(r), \ldots, u_m(r)$ are defined by

$$u_k(r) \equiv \frac{v_k' - r}{||\, v_k' - r \,||}. \tag{8}$$

We also define $w_1(r), \ldots, w_m(r)$ to be the cross products

$$w_k(r) \equiv u_k(r) \times u_{k+1}(r). \tag{9}$$

Henceforth, we assume that u_k and w_k are functions of position and omit the explicit dependence on r. Expressing Θ_k and Γ_k in terms of w_k, we have

$$\Theta_k = \sin^{-1} ||\, w_k \,||, \tag{10}$$

and

$$\Gamma_k = \frac{w_k}{||\, w_k \,||}. \tag{11}$$

The form of equation (10) simplifies the following development, although it is equivalent to equation (5) only for acute angles; that is, only when $u_k \cdot u_{k+1} \geq 0$. This restriction will be removed below.

To compute $\mathbf{J}(\Phi)$ in terms of the vertex Jacobians $\mathbf{J}(u_1), \ldots, \mathbf{J}(u_m)$ we first consider the k^{th} term of the summation in equation (4). Differentiating, we have

$$\mathbf{J}(\Theta_k \Gamma_k) = \Gamma_k \nabla\Theta_k + \Theta_k \mathbf{J}(\Gamma_k), \tag{12}$$

where $\Gamma_k \nabla\Theta_k$ is the outer product of the vector Γ_k and the gradient $\nabla\Theta_k$. We now compute $\nabla\Theta_k$ and $\mathbf{J}(\Gamma_k)$. For brevity, we denote the vertex vectors u_k and u_{k+1} by a and b respectively, and the cross product $a \times b$ by w. Then the gradient of Θ_k with respect to r is

$$\begin{aligned} \nabla\Theta_k &= \nabla \sin^{-1} ||\, w \,|| \\ &= \frac{1}{\sqrt{1 - w^{\mathrm{T}} w}} \left(\frac{w^{\mathrm{T}}}{||\, w \,||}\right) \mathbf{J}(w) \\ &= \left(\frac{w^{\mathrm{T}}}{a^{\mathrm{T}} b}\right) \frac{\mathbf{J}(w)}{||\, w \,||}. \end{aligned} \tag{13}$$

Similarly, differentiating Γ_k with respect to r we have

$$\begin{aligned} \mathbf{J}(\Gamma_k) &= D\left(\frac{w}{||\, w \,||}\right) \\ &= \frac{\mathbf{J}(w)}{||\, w \,||} - \frac{w w^{\mathrm{T}}}{||\, w \,||^3} \mathbf{J}(w) \\ &= \left(\mathbf{I} - \frac{w w^{\mathrm{T}}}{w^{\mathrm{T}} w}\right) \frac{\mathbf{J}(w)}{||\, w \,||}. \end{aligned} \tag{14}$$

From Equations (10)-(14), we obtain an expression for $\mathbf{J}(\Theta_k \Gamma_k)$ in terms of $\mathbf{J}(w)$ and the vertex vectors a and b:

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

If the factor of $\sin^{-1} ||\, w \,||$ is now replaced by the angle between a and b, the expression will hold for all angles, removing the caveat noted earlier. The above expression may be written compactly as

$$\mathbf{J}(\Theta_k \Gamma_k) = \mathbf{E}(a, b)\, \mathbf{J}(a \times b), \tag{15}$$

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

$$\boxed{\mathbf{E}(a, b) \equiv \left(\frac{1}{a^{\mathrm{T}} b}\right) \frac{w w^{\mathrm{T}}}{w^{\mathrm{T}} w} + \frac{\cos^{-1} a^{\mathrm{T}} b}{||\, w \,||} \left(\mathbf{I} - \frac{w w^{\mathrm{T}}}{w^{\mathrm{T}} w}\right).} \tag{16}$$

In equation (16) we have retained w as an abbreviation for $a \times b$. Because the edge matrix contains no derivatives, it can be computed

directly from the vertex vectors a and b. To simplify the Jacobian of $a \times b$, we define another matrix-valued function $\mathbf{Q}$ by

$$\mathbf{Q}(p) \equiv \begin{bmatrix} 0 & p_z & -p_y \\ -p_z & 0 & p_x \\ p_y & -p_x & 0 \end{bmatrix}. \tag{17}$$

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

$$\mathbf{J}(F \times G) = \mathbf{Q}(F)\mathbf{J}(G) - \mathbf{Q}(G)\mathbf{J}(F). \tag{18}$$

Applying the above identity to equation (15), summing over all edges of the clipped source polygon P', and scaling by $M/2\pi$, we arrive at an expression for the irradiance Jacobian due to the visible portion of polygonal source P:

$$\mathbf{J}(\Phi) = \frac{M}{2\pi} \sum_{i=1}^{m} \mathbf{E}(u_i, u_{i+1}) \left[\mathbf{Q}(u_i)\mathbf{J}(u_{i+1}) - \mathbf{Q}(u_{i+1})\mathbf{J}(u_i)\right]. \tag{19}$$

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

$$\boxed{\mathbf{C}(a, b, c) \equiv \mathbf{E}(a, b)\mathbf{Q}(a) - \mathbf{E}(b, c)\mathbf{Q}(c).} \tag{20}$$

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

$$\boxed{\mathbf{J}(\Phi) = \frac{M}{2\pi} \sum_{i=1}^{m} \mathbf{C}(u_{i-1}, u_i, u_{i+1})\ \mathbf{J}(u_i),} \tag{21}$$

where we have made the natural identifications $u_0 \equiv u_m$ and $u_{m+1} \equiv u_1$. Note that each corner matrix $\mathbf{C}$ depends only on the vertex vectors, and not their derivatives; all information concerning apparent motion due to changing the position r is embodied in the vertex Jacobians $\mathbf{J}(u_1), \ldots, \mathbf{J}(u_m)$, which we now examine in detail.

3 Vertex Jacobians

To apply equation (21) we require the vertex Jacobians, which we now construct for both unoccluded and partially occluded polygonal sources. First, observe that each vertex vector $u(r)$ is a smooth function of r almost everywhere; that is, $u(r)$ is differentiable at all $r \in \mathbb{R}^3$ except where two or more edges of distinct polygons appear to coincide, as described in section 4. Differentiability follows from the smoothness of the Euclidean norm and the fact that the apparent point of intersection of two skew lines varies quadratically in r along each of the lines [16]. From this it is evident that the vertex Jacobian exists whenever the real or apparent intersection of two edges exists and is unique.

When the vertex Jacobian exists, it can be constructed by determining its action on each of three linearly independent vectors; that is, by determining the instantaneous change in the vertex vector u as a result of moving r. Differential changes in u are orthogonal to u and collectively define a disk, or in the case of partial occlusion, an ellipse. See Figure 5. The directions that are easiest to analyze are the axes of the ellipse, which are the eigenvectors of the vertex Jacobian. We first treat intrinsic vertices and then generalize to the more difficult case of apparent vertices.

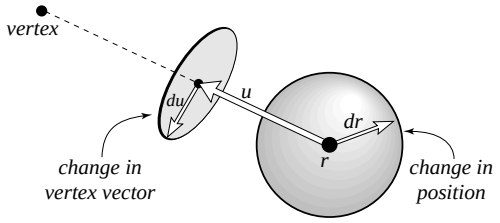


Figure 5: *A differential change in the position r results in a change in the unit vertex vector u. The locus of vectors du forms a disk, or more gnerally, an ellipse in the plane orthogonal to u.*

3.1 Intrinsic Vertices

Suppose that u is the vertex vector associated with an unoccluded source vertex, as shown in Figure 3b. In this case the vertex Jacobian $\mathbf{J}(u)$ is easy to compute since it depends solely on the distance between r and the vertex, which we denote by α. Moving r in the direction of the vertex leaves u unchanged, while motion perpendicular to u causes an opposing change in u. The changes in u are inversely proportional to the distance α. This behavior completely determines the vertex Jacobian. Thus, we have

$$\mathbf{J}(u) = -\frac{1}{\alpha}\left(\mathbf{I} - uu^{\mathrm{T}}\right), \tag{22}$$

where the matrix $\mathbf{I} - uu^{\mathrm{T}}$ is a projection onto the tangent plane of $\mathcal{S}^2$ at the point u, which houses all differential motions of the unit vector u. The same reasoning applies to vertex vectors defined by a blocker vertex, as in Figure 3c. In this case α is the distance along u to the blocker vertex.

3.2 Apparent Vertices

Within penumbrae, apparent vertices may be formed by the apparent crossing of non-coplanar edges. The two distinct cases are depicted in Figure 4. Let u be the vertex vector associated with such a vertex, where the determining edges are segments of skew lines $\mathcal{L}_1$ and $\mathcal{L}_2$. Let s and t be vectors parallel to $\mathcal{L}_1$ and $\mathcal{L}_2$, respectively, as depicted in Figure 6. As in the case of intrinsic vertices, moving r toward the apparent vertex leaves u unchanged, so $\mathbf{J}(u)u = 0$. To account for other motions, we define the vectors $\widehat{s}$ and $\widehat{t}$ by

$$\begin{aligned} \widehat{s} &\equiv (\mathbf{I} - uu^{\mathrm{T}})\, s \\ \widehat{t} &\equiv (\mathbf{I} - uu^{\mathrm{T}})\, t, \end{aligned}$$

which are projections of s and t onto the plane orthogonal to u. Now consider the change in u as r moves parallel to $\widehat{s}$, as shown in Figure 6a. In this case the apparent vertex moves along $\mathcal{L}_1$ while remaining fixed on $\mathcal{L}_2$. Therefore, the change in u is parallel to $\widehat{s}$ but opposite in direction to the change in r. If α_t is the distance to $\mathcal{L}_2$ along u, we have

$$\mathbf{J}(u)\widehat{s} = -\frac{\widehat{s}}{\alpha_t}. \tag{23}$$

Evidently, $\widehat{s}$ is an eigenvector of $\mathbf{J}(u)$ with associated eigenvalue $-1/\alpha_t$. A similar argument holds when r moves along $\widehat{t}$, as shown in Figure 6b. Here the apparent vertex moves along $\mathcal{L}_2$ while remaining fixed at $\mathcal{L}_1$. If α_s is the distance to $\mathcal{L}_1$ along u, we have

$$\mathbf{J}(u)\widehat{t} = -\frac{\widehat{t}}{\alpha_s}, \tag{24}$$

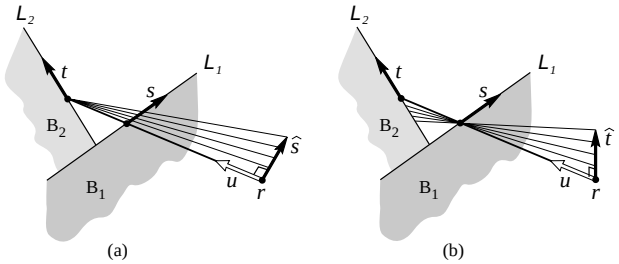


Figure 6: *The vertex Jacobian* $\mathbf{J}(u)$ *with respect to two skew lines* $\mathcal{L}_1$ *and* $\mathcal{L}_2$ *is found by determining how the vertex vector* u *changes as* r *moves parallel to (a) the vector* $\hat{s}$*, and (b) the vector* $\hat{t}$*.*

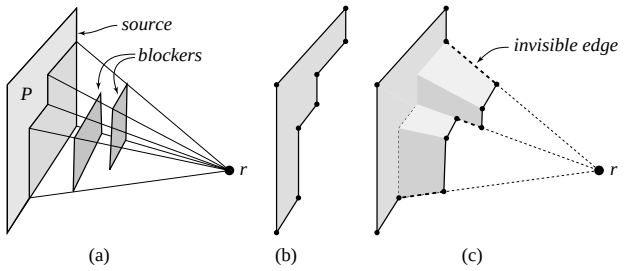


Figure 7: *(a) Source P is partially occluded by two blockers as seen from r. (b) The vector irradiance at r due to P can be computed using the simply-clipped polygon. (c) The irradiance Jacobian at r requires the depth-clipped polygon.*

which provides the third eigenvector and corresponding eigenvalue. Collecting these relationships into a matrix equation, we have

$$\mathbf{J}(u) \begin{bmatrix} \hat{s} & \hat{t} & u \end{bmatrix} = \begin{bmatrix} -\frac{\hat{s}}{\alpha_t} & -\frac{\hat{t}}{\alpha_s} & 0 \end{bmatrix}. \tag{25}$$

It follows immediately that whenever the lines $\mathcal{L}_1$ and $\mathcal{L}_2$ are distinct and non-colinear as viewed from the point r, then

$$\mathbf{J}(u) = \mathbf{A} \begin{bmatrix} -1/\alpha_t & & \\ & -1/\alpha_s & \\ & & 0 \end{bmatrix} \mathbf{A}^{-1} \tag{26}$$

where $\mathbf{A} \equiv \begin{bmatrix} \hat{s} & \hat{t} & u \end{bmatrix}$. Note that equation (26) reduces to equation (22) when $\alpha_s = \alpha_t$. Equation (26) therefore suffices for all vertex vectors, but the special case for intrinsic vertices can be used for efficiency.

3.3 Polygon Depth-Clipping

To compute the irradiance or vector irradiance at a point r, it suffices to clip all sources against all blockers, as seen from r, and apply equation (4) to the resulting vertex lists. This operation is also sufficient to compute the corner matrices and the vertex Jacobians at unoccluded source vertices. However, the vertex Jacobians for the cases illustrated in Figures 3c, 4b, and 4c all require information about the blockers that is missing from traditionally-clipped polygons. Specifically, the distances to blocker edges defining each vertex are needed to form the matrices in Equations (22) and (26).

Thus, additional depth information must be retained along with the clipped polygons for use in computing vertex Jacobians. We propose a simple mechanism, called *depth clipping*, by which the required information appears as additional vertices. The idea is to construct the clipped polygon using segments of source and blocker edges and joining them by segments called *invisible edges*, which cannot be seen from the point r. See Figure 7. The resulting non-planar contour is identical to that of the traditionally-clipped polygon when viewed from r. Each invisible edge produces a vertex Jacobian of the form in equation (26); its end points encode the distances from r while the adjacent edges provide the two directions. Each vertex not adjacent to an invisible edge produces a vertex Jacobian of the form in equation (22).

The depth-clipped polygon and the emission M completely specify the irradiance Jacobian. Most polygon clipping algorithms can be extended to generate this representation using the plane equation of each blocker. The depth-clipped polygon also clearly illustrates the information required for irradiance Jacobians.

4 Properties of the Irradiance Jacobian

In this section we list some of the basic properties of the irradiance Jacobian, beginning with existence. By definition, the Jacobian $\mathbf{J}_r(\Phi)$ exists wherever Φ is differentiable, which requires the existence of each directional derivative at r. Because we consider only area sources, the variation of Φ is *continuous* along any line except when a blocker is in contact with the receiving surface. Instantaneous occlusion causes discontinuous changes in the vector irradiance. In the absence of contact occlusions, the variation of Φ is not only continuous but *differentiable* everywhere except along lines where edges appear to coincide; that is, points at which a source or blocker edge appears to align with another blocker edge [12]. For instance, when both blockers are present simultaneously in Figure 2, the irradiance curve coincides with curve B to the left of r, and with curve A to the right. Therefore, the irradiance at r has a discontinuity in the first derivative. Only contact occlusion and edge-edge alignments cause the Jacobian to be undefined; other types of events cause higher-order discontinuities in the vector irradiance, but are first-order smooth.

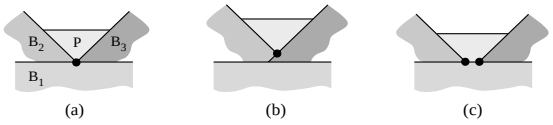


Figure 8: *(a) The vertex Jacobian does not exist at the intersection of three edges. A small change can produce (b) a single apparent vertex, or (c) two apparent vertices.*

From equation (21) it would appear that the irradiance Jacobian does not exist if any one of the vertex Jacobians fails to exist; this is not always so. A vertex Jacobian may be undefined because the vertex lies at the intersection of three edges, as shown in Figure 8a. In cases such as this, a minute change in r can lead to several possible configurations with different vertex Jacobians. See Figures 8b and 8c. However, the unoccluded area of the source still changes smoothly despite such a difficulty at a single vertex. To ensure that equation (21) is valid wherever Φ is differentiable, we simply restrict the edges that are used in computing the vertex Jacobians to those that actually bound the clipped source. Thus, in Figure 8, blocker B_1 is ignored until it makes its presence known by the addition of a new edge, as in Figure 8c.

One of the most useful properties of the Jacobian matrix is its connection with directional derivatives. For any $\xi \in \mathcal{S}^2$, the directional derivative of Φ at r in the direction ξ is

$$D_\xi \Phi(r) = \mathbf{J}_r(\Phi)\, \xi. \tag{27}$$

Although directional derivatives of Φ may be approximated to second order with central differences, using the irradiance Jacobian has several advantages. First, all directional derivatives of $\Phi(r)$ are easily obtained from the irradiance Jacobian at r, which requires a single global clipping operation; that is, sources need only be clipped against blockers once. In contrast, difference approximations require at least two clipping operations per directional derivative. More importantly, directions of maximal change follow immediately from the Jacobian but require multiple finite differences to approximate.

A final property, which we build upon in the next section, is the connection with the rate of change of surface irradiance. Differentiating equation (3) with respect to position, we have

$$\nabla\phi = \Phi^{\mathrm{T}}\,\mathbf{J}(\mathbf{n}) + \mathbf{n}^{\mathrm{T}}\,\mathbf{J}(\Phi), \tag{28}$$

which associates the irradiance gradient with the irradiance Jacobian. Note that $\mathbf{J}(\mathbf{n})$ is related to the curvature of the surface at each point $r \in \mathcal{M}$. For planar surfaces $\mathbf{J}(\mathbf{n}) \equiv 0$, so equation (28) reduces to

$$\nabla\phi = \mathbf{n}^{\mathrm{T}}\,\mathbf{J}(\Phi), \tag{29}$$

which is the form we shall use to compute isolux contours on polygonal receivers. When evaluating equation (29) several optimizations are possible by distributing the vector multiplication across the terms of equation (21), which changes the summation of matrices into a summation of row vectors.

5 Applications of the Irradiance Jacobian

In the first portion of the paper, we have seen how to compute the irradiance Jacobian and the irradiance gradient. The steps can be summarized as follows:

Matrix IrradianceJacobian(**Point** r)

Matrix J $\leftarrow 0$
for each source P with emission M
  begin
  $\widehat{P} \leftarrow P$ depth-clipped against all blockers, as seen from r
  for each i: $\mathbf{J}_i \leftarrow$ vertex Jacobian for the i^{th} vertex of $\widehat{P}$
  for each i: $\mathbf{E}_i \leftarrow$ edge matrix for the i^{th} edge of $\widehat{P}$
  for each i: $\mathbf{C}_i \leftarrow$ corner matrix using $\mathbf{E}_{i-1}$ and $\mathbf{E}_i$
  $\mathbf{J} \leftarrow \mathbf{J} + \frac{M}{2\pi}$ (sum of all $\mathbf{C}_i\,\mathbf{J}_i$)
  end
return J

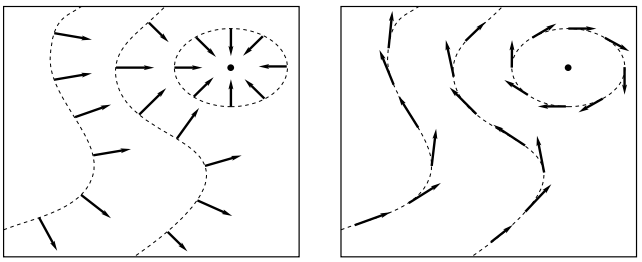
Here the inner loops all refer to the vertices as seen from r; pairs of vertices associated with invisible edges are counted as one. Gradients can then be computed using equation (28) or equation (29). The procedure above is a general-purpose tool with many applications, several of which are described in the remainder of this section.

5.1 Finding Local Extrema

The first application we examine is that of locating irradiance extrema on surfaces, which can be used in computing bounds on the transfer of energy between surfaces [11]. Given the availability of gradients, the most straightforward approach to locating a point of maximal irradiance is with an ascent method of the form

$$r^{i+1} \equiv r^i + \gamma_i \left(\mathbf{I} - \mathbf{n}\mathbf{n}^{\mathrm{T}}\right) \nabla\phi^{\mathrm{T}}(r^i), \tag{30}$$

where r^0 is a given starting point, and the factor γ_i is determined by a line search that insures progress is made toward the extremum. For example, the line search may simply halve γ_i until an increase in irradiance is achieved. The extremum has been found when no further progress can be made. Minima are found similarly. The principle drawbacks of this method are that it finds only local extrema, and convergence can be very slow when the irradiance function is flat. In the absence of a global method for locating all extrema, seed points near each of the relevant extrema must be supplied.



(a) (b)

Figure 9: *(a) Projecting the gradient onto a surface defines a 2D vector field everywhere orthogonal to the level curves. (b) Rotating the projected gradients by $-\pi/2$ creates a vector field whose flow lines are isolux contours. Local maxima are then encircled by clockwise loops.*

5.2 Direct Computation of Isolux Contours

Curves of constant irradiance over surfaces are known as *isolux contours* [20]. Applications of isolux contours in computer graphics include visualizing irradiance distributions [21, 15] and simplifying shading [4] and sampling [5]. In computer vision isolux contours have been used to perform automatic image segmentation [10]. In this section we show how isolux contours can be computed directly.

Every isolux contour on a surface $\mathcal{M}$ can be represented by a function $r : [0, \infty) \rightarrow \mathcal{M}$ that satisfies

$$\phi(r(s)) = \phi(r(0)) \tag{31}$$

for all $s \geq 0$. To compute such a curve we construct a first-order ordinary differential equation (ODE) to which it is a solution, and solve the ODE numerically.

The direction of most rapid increase in $\phi(r)$ at a point $r \in \mathcal{M}$ is given by the gradient $\nabla\phi(r)$, which generally does not lie in the tangent plane of the surface. The projection of the gradient onto the surface is a tangent vector that is orthogonal to the isolux curve passing through its origin. See Figure 9a. If the projected gradient is rotated by 90 degrees, we obtain a direction in which the irradiance remains constant to first order. See Figure 9b. Thus, we define the *isolux differential equation* by

$$\dot{r} = \mathbf{P}(r)\,\nabla\phi^{\mathrm{T}}(r), \tag{32}$$

with the initial condition $r(0) = r_0$, where

$$\mathbf{P}(r) \equiv \mathbf{R}(\mathbf{n}(r))\left[\mathbf{I} - \mathbf{n}(r)\mathbf{n}^{\mathrm{T}}(r)\right], \tag{33}$$

and $\mathbf{R}(z)$ is a rotation by $-\pi/2$ about the vector z. The matrix $\mathbf{P}(r)$ is constant for planar surfaces. The solution of this ODE is an isolux contour with irradiance $c = \phi(r_0)$.

5.2.1 Solving the Isolux Differential Equation

Any technique for solving first-order ordinary differential equations can be applied to solving the isolux differential equation. The overriding consideration in selecting an appropriate method is the number of irradiance values and gradients used in taking a step

along the curve. Obtaining this information involves a global clipping operation, which is generally the most expensive part of the algorithm.

Multistep methods are particularly appropriate for solving the isolux ODE since they make efficient use of the recent history of the curve. For example, Milne's predictor-corrector method is a multistep method that predicts the point $r_{k+1} \equiv r(s_{k+1})$ by extrapolating from the three most recent gradients and function values using a parabola. When the matrix $\mathbf{P}$ is fixed, Milne's predictor is given by

$$r^0_{k+1} \equiv r_{k-3} + \frac{4h}{3}\mathbf{P}\,(2g_{k-2} - g_{k-1} + 2g_k)\,, \tag{34}$$

where g_k denotes the gradient at the point r_k, and h is the step size [1]. Given the predicted value, a corrector is then invoked to find the nearest point on the curve. Because the contour is the zero set of the function $\phi(r) - c$, the correction can be performed very efficiently using Newton's method. Beginning with the predicted point r^0_k, a Newton corrector generates the sequence $r^1_k, r^2_k, \ldots$ by

$$r^{i+1}_k \equiv r^i_k + \left[c - \phi(r^i_k)\right] \frac{\nabla\phi^{\mathrm{T}}(r^i_k)}{||\,\nabla\phi^{\mathrm{T}}(r^i_k)\,||^2}, \tag{35}$$

which converges quadratically to a point on the curve. The iteration is repeated until

$$|c - \phi(r^i_k)| \leq \epsilon, \tag{36}$$

where ϵ is a preset tolerance. With this corrector, accurate polygonal approximations can be generated for arbitrarily long isolux contours. This would not be possible with the traditional Milne corrector, for example, which would eventually drift away from the curve. With a good predictor, very few correction steps are required, which saves costly gradient evaluations.

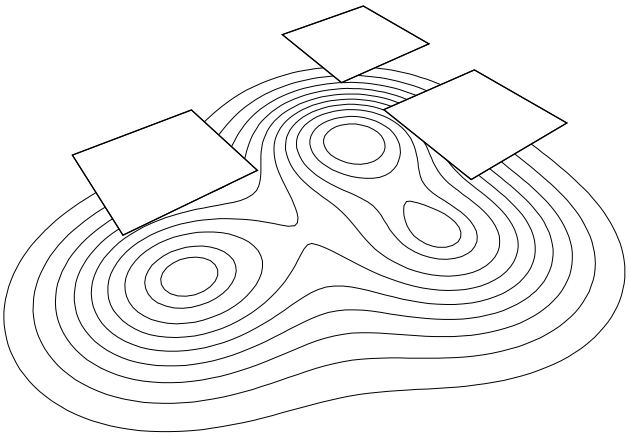


Figure 10: *A family of isolux contours for three unoccluded sources.*

5.2.2 Examples of Isolux Contours

The predictor-corrector method described above was used to compute isolux contours for simple test cases with both unoccluded and partially occluded sources. The step size h and the tolerance for the corrector were user-supplied parameters. Use of the Newton corrector made the curve follower fairly robust; even abrupt turns at or near derivative discontinuities in the irradiance function were automatically compensated for.

To generate a family of curves depicting equal steps in irradiance, similar to a topographic map, we must find starting points for each curve with the desired irradiance values $c_1 > c_2 > \cdots > c_k$. The Newton corrector can be used to find a point on the $(k+1)^{\text{st}}$ curve by finding a root of the equation $\phi(r) - c_{k+1}$ beginning at any point on the k^{th} curve. The curve families in Figures 10 and 11 were automatically generated in this way. Figure 10 shows a family of isolux contours resulting from three unoccluded rectangular sources. Three distinct families were generated, starting at each of the three local maximuma, which were found by the ascent method described in section 5.1. Figure 11 shows a family of isolux contours resulting from a rectangular source and a simple blocker. These contours surround both a peak and a valley.

Because distinct isolux contours cannot cross, any collection of closed contours has an obvious partial ordering defined by containment. To display filled contours, as shown in Figures 12a and 12b, the regions can be painted in back-to-front order after sorting according to the partial order.

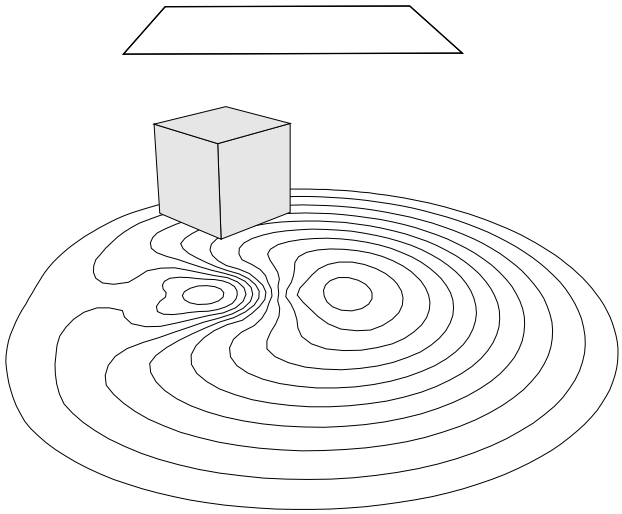


Figure 11: *Isolux contours on a planar receiver due to a rectangular source and simple blocker above the plane of the receiver.*

5.3 Iso-Meshing

Because the isolux contours described in the previous section are generated by direct computation rather than by post-processing an image, they may be used in the image generation process. For example, isolux contours can be used to drive a meshing algorithm for global illumination.

The idea is similar to that of discontinuity meshing [8, 13], which can identify important discontinuities in the radiance function over diffuse surfaces. Isolux contours provide additional information about radiance functions, and can be employed for mesh generation either in a preprocessing step for modeling direct illumination, or as part of a radiosity post-process to create a high-quality mesh for rendering a final image [13].

To best exploit the information in the contours, the mesh elements of an *iso-mesh* should follow the contours. To generate a mesh with this property from isolux contours, we have applied the constrained Delaunay triangulation algorithm used earlier by Lischinski et al. [13] for discontinuity meshing. This approach forces the edges of the mesh elements to align with the isolux contours rather than crossing them. It also creates triangles with good aspect ratios. Figure 13 shows the result of applying this algorithm to the families of isolux contours shown in Figure 12. Meshes of varying coarseness can be generated by selecting subsets of the points along the contours.

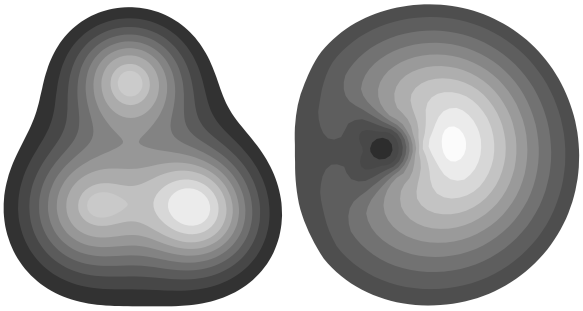


Figure 12: *Filled isolux contours corresponding to the previous figures. Each region is shaded according to the constant irradiance of its contour.*

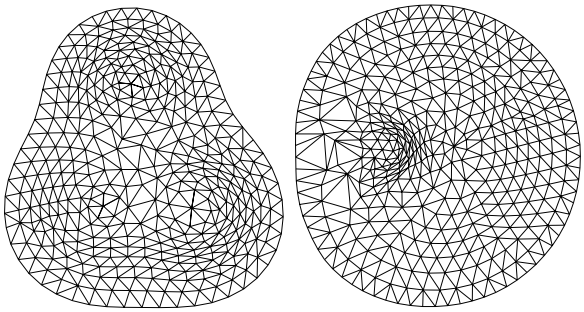


Figure 13: *Iso-meshes generated from families of isolux contours using constrained Delaunay triangulation.*

6 Conclusions and Future Work

We have presented a closed-form expression for the irradiance Jacobian due to polygonal sources of uniform brightness in the presence of arbitrary polygonal blockers. The expression is closely related to the well-known analytic formula for point-to-patch form factors, and is evaluated in much the same way when blockers are present; only a minor extension of standard polygon clipping is required.

Several applications that make use of gradients obtained from the irradiance Jacobian have been demonstrated, including the computation of isolux contours and local irradiance extrema, both in the presence of occluders.

Isolux contours provide a useful characterization of irradiance in regions away from the derivative discontinuities that can be handled with discontinuity meshing. We have demonstrated how a family of isolux contours can form the basis of a mesh generated with constrained Delaunay triangulation. This is one means of interpolating isolux contours, and also demonstrates a new approach to meshing for global illumination.

The irradiance Jacobian may also find other applications in global illumination. In approaches that do not employ an explicit mesh, gradients can be used to guide sampling in a spirit similar to previous approaches [22, 5], but using analytically computed gradients. Alternatively, irradiance gradients can be used to define higher-order interpolants within the elements of an existing mesh.

Acknowledgments

The author owes many thanks to Ken Torrance for his advice and encouragement, and to Steve Westin for his invaluable suggestions. Thanks also to Dani Lischinski for useful discussions and for providing the Delaunay triangulation code, and to Don Greenberg, Brian Smits, Jim Ferwerda, Jed Lengyel, and Erin Shaw for their helpful comments. This work was supported by the NSF grant "Interactive Computer Graphics Input and Display Techniques" (CCR-8617880), and by the NSF/ARPA Science and Technology Center for Computer Graphics and Scientific Visualization (ASC-8920219). Hewlett-Packard Corporation generously provided the workstations used in this research.

REFERENCES

[1] ACTON, F. S. *Numerical Methods that Work*. Harper & Row, New York, 1970.

[2] BAUM, D. R., RUSHMEIER, H. E., AND WINGET, J. M. Improving radiosity solutions through the use of analytically determined form-factors. *Computer Graphics 23*, 3 (July 1989), 325–334.

[3] BERGER, M. *Geometry, Volume II*. Springer-Verlag, New York, 1987. Translated by M. Cole and S. Levy.

[4] CONWAY, D. M., AND COTTINGHAM, M. S. The isoluminance contour model. In *Proceedings of Ausgraph '88* (Melbourne, Australia, September 1988), pp. 43–50.

[5] DRETTAKIS, G., AND FIUME, E. L. Concrete computation of global illumination using structured sampling. In *Proceedings of the Third Eurographics Workshop on Rendering, Bristol, United Kingdom* (1992).

[6] FOK, V. A. The illumination from surfaces of arbitrary shape. *Transactions of the Optical Institute, Leningrad 28* (1924), 1–11. (Russian).

[7] GERSHUN, A. The light field. *Journal of Mathematics and Physics 18*, 2 (May 1939), 51–151. Translated by P. Moon and G. Timoshenko.

[8] HECKBERT, P. S. *Simulating Global Illumination Using Adaptive Meshing*. PhD thesis, University of California, Berkeley, June 1991.

[9] HOPF, E. *Mathematical Problems of Radiative Equilibrium*. Cambridge University Press, New York, 1934.

[10] LIFSHITZ, L. M., AND PIZER, S. M. A multiresolution hierarchical approach to image segmentation based on intensity extrema. *IEEE Transactions on Pattern Analysis and Machine Intelligence 12*, 6 (June 1990), 529–540.

[11] LISCHINSKI, D., SMITS, B., AND GREENBERG, D. P. Bounds and error estimates for radiosity. In *Computer Graphics Proceedings* (1994), Annual Conference Series, ACM SIGGRAPH.

[12] LISCHINSKI, D., TAMPIERI, F., AND GREENBERG, D. P. Discontinuity meshing for accurate radiosity. *IEEE Computer Graphics and Applications 12*, 6 (November 1992), 25–39.

[13] LISCHINSKI, D., TAMPIERI, F., AND GREENBERG, D. P. Combining hierarchical radiosity and discontinuity meshing. In *Computer Graphics Proceedings* (1993), Annual Conference Series, ACM SIGGRAPH, pp. 199–208.

[14] NISHITA, T., AND NAKAMAE, E. Half-tone representation of 3-D objects illuminated by area sources or polyhedron sources. In *Proceedings of the IEEE Computer Software and Applications Conference* (Chicago, November 1983), pp. 237–242.

[15] NISHITA, T., AND NAKAMAE, E. Continuous tone representation of 3-D objects taking account of shadows and interreflection. *Computer Graphics 19*, 3 (July 1985), 23–30.

[16] PLANTINGA, H., AND DYER, C. R. Visibility, occlusion, and the aspect graph. *International Journal of Computer Vision 5*, 2 (1990), 137–160.

[17] PREISENDORFER, R. W. *Hydrologic Optics, Volume II. Foundations*. National Oceanic & Atmospheric Administration, Honolulu, Hawaii, 1976. (Available as NTIS PB-259 794).

[18] SCHRÖDER, P., AND HANRAHAN, P. On the form factor between two polygons. In *Computer Graphics Proceedings* (1993), Annual Conference Series, ACM SIGGRAPH, pp. 163–164.

[19] TELLER, S. Computing the antipenumbra of an area light source. *Computer Graphics 26*, 2 (July 1992), 139–148.

[20] TROTTER, A. P. *Illumination: Its Distribution and Measurement*. The Macmillan Company, London, 1911.

[21] VERBECK, C. P., AND GREENBERG, D. P. A comprehensive light-source description for computer graphics. *IEEE Computer Graphics and Applications 4*, 7 (July 1984), 66–75.

[22] WARD, G. J., AND HECKBERT, P. S. Irradiance gradients. In *Proceedings of the Third Eurographics Workshop on Rendering, Bristol, United Kingdom* (May 1992).

[23] WEILER, K., AND ATHERTON, P. Hidden surface removal using polygon area sorting. *Computer Graphics 11*, 3 (1977), 214–222.