Accurate Analytic Approximations For Real-Time Specular Area Lighting

Pascal Lecocq* Arthur Dufay Gaël Sourimant Jean-Eudes Marvie

Technicolor Research & Innovation

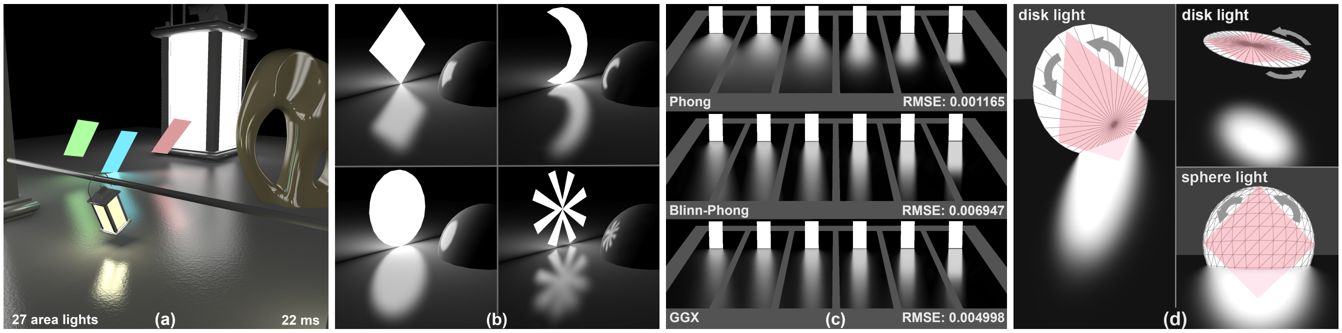

Figure 1: *Our analytic approximations provide real-time performances to specular area lighting at a quality close to the ground truth (a,c). We support polygonal light source shapes of any kind (even animated ones) (b), and surfaces described by Phong or microfacet BRDF models (c). Our method easily extends to spherical and disc area lights with non uniform scaling, using spinning quads (illustrated here in red) (d).*

Abstract

We introduce analytic approximations for accurate real-time rendering of specular surfaces lit by area light sources. Our solution leverages the *Irradiance Tensors* developed by Arvo for the rendering of Phong surfaces lit by a polygonal light source. Using a reformulation of the 1D boundary edge integral, we develop a general framework for approximating and evaluating the integral in constant time using simple peak shape functions. To overcome the Phong restriction, we propose a low cost edge splitting strategy that accounts for the spherical warp introduced by the half vector parametrization. Thanks to this novel extension, we accurately approximate common microfacet BRDFs, providing the first practical method producing specular stretches that closely match ground truth image references in real-time. Finally, using the same approximation framework, we introduce support for spherical and disc area light sources, based on an original polygon spinning method supporting non-uniform scaling operations and horizon clipping. Implemented on a GPU, our method achieves real-time performances without any assumption on area light shape nor surface roughness, with a quality close to the ground truth.

Keywords: area lights, specular, microfacet, real-time, shading

Concepts: •**Computing methodologies** $\rightarrow$ **Computer graphics;**

1 Introduction

Accurate real-time rendering of specular surfaces is a challenging task when considering area light source illumination. The difficulty resides in the evaluation of a two dimensional specular radiance integral for which no practical solution exists, except expensive Monte Carlo based sampling techniques. Most compelling solutions are found using *Most Representative Point (MRP)* approaches, reducing the shading integration problem to a cheap point lighting calculation. However, these methods fail in preserving the specular highlight shape of underlying BRDFs and partial visibility above horizon is complicated to handle.

Finding a solution combining both accuracy and real-time performances is still a challenging problem, with many expectations on high quality demanding applications such as lighting pre-viz tools, game engines or production renderers. However, accurate solutions exist. The *Irradiance Tensors* and its applications developed by Arvo [1995b] provide an exact analytic solution for the shading of glossy surfaces lit by a polygonal light source. But its implementation relies on an expensive contour integration method, and is restricted to Phong surfaces.

In this paper we address these shortcomings by leveraging the *Irradiance Tensors* developed by Arvo with accurate analytic approximations. We further extend the method to overcome the *Phong* restriction enabling support for microfacet BRDF with highlight shape preservation. Finally, we introduce an original polygon spinning method allowing spherical and disc area lighting using the same mathematical framework.

Our contributions are:

- A general framework for approximating and evaluating the edge contour integrals in $O(1)$ time instead of $O(n)$ using simple and integrable peak shape functions.
- A low-cost edge splitting strategy for handling the warp distortion introduced by the half vector parameterization that preserves the highlight shape of microfacet BRDF.
- An original spinning algorithm enabling spherical and disc area lighting that benefits from our approximations and supports non-uniform scale operations.

*{*firstname.lastname*}@technicolor.com

2 Related works

Direct illumination from area light sources has been addressed in various ways in the last decades. We review in this section the related works we think most relevant to our approach, with a focus on the techniques addressing the integration of the specular term with real-time rendering constraints.

Monte Carlo integration. Monte-Carlo integration techniques are a common approach to numerically compute complex integrals based on probabilistic sampling strategies. For direct area light illumination problems, samples are drawn either considering the solid angle sustained by the area shape [Arvo 1995c; Ureña et al. 2013], considering importance sampling of the surface BRDF, or using a combination of both to reduce the variance in presence of specular surfaces. Despite this sampling effort, these methods require a huge amount of samples to converge to a noise free result, hardly compatible with real-time rendering constraints.
Another common approach is to approximate area light sources using a set of *Virtual Point Lights (VPLs)* [Keller 1997], reducing the specular term integration to a many point lights calculation. Clustering methods [Walter et al. 2005] have been proposed to further reduce the algorithm complexity of *VPLs* with successful application in real-time rendering [Nichols and Wyman 2009; Ritschel et al. 2008]. However, these solutions are usually restricted to low frequency illumination problems such as diffuse or weakly glossy surfaces to limit the sampling count and maintain good real-time performances. Rendering high frequency illumination with these methods is still a challenging problem only addressed using huge number of samples or expensive integration techniques far from real-time rendering considerations.

Most Representative Point. *MRP* approaches alleviate the costly sampling techniques by identifying a representative point on the area light that most contributes to the illumination. The method reduces the shading integration problem to a single point lighting calculation providing a practical solution for real-time rendering. Early works on the method can be found in [Picott 1992] for *Phong* area lighting. The *MRP* here is defined as the closest point from the viewing reflection direction. Instead, Drobot [2014] considers a point in the area of intersection between an area light and a cone with aperture parameterized by the surface roughness. Karis [2013] addresses the problem of energy conservation and uses a modification of the specular distribution to better match intensity highlight of specular microfacet models. However, these approaches have several drawbacks. The highlight shape with a *Phong* BRDF is decently approximated, but becomes inaccurate when considering microfacet BRDFs. Horizon handling is yet another issue. The *MRP* approximation works well with simple geometric light emitters but calculation get more complex when the light source is clipped above the horizon plane.

Analytic approaches. Other approaches try to derive an exact analytic solution of the shading integral, or at least a descent approximation. Bao and Peng [1993] approximate the double integral by 2D polynomials using a low degree Taylor series expansion but limited to low exponent *Phong* surfaces. Tanaka and Takahashi [1997] extend the linear area light method of Poulin [1991] and decompose the solid angle into 1D signed integrals along edge great circles. Each 1D integral is then evaluated using a Chebyshev polynomial approximation, restricting the method to low frequency *Phong* surfaces. The *Irradiance Tensors* developed by Arvo [Arvo 1995b] provide an exact analytic solution for the direct illumination of glossy surfaces lit by a polygonal light source. Using tensor theory and Stokes contour integration, the shading integral is decomposed into a sum of signed 1D integrals along the spherical boundary edges of the polygonal light. Each edge integral is then evaluated analytically using a linear time algorithm bound to the *Phong* shininess n. A practical implementation for real-time graphics, including horizon clipping, can be found in [Snyder 1996]. Despite its accuracy, the method only works for *Phong* surfaces and its usage in real-time rendering applications is limited to weakly glossy surfaces due its $O(n)$ time complexity.

Spherical Gaussians (SGs). SGs are spherical functions used in many lighting problems such as *environment lighting* or *global illumination* with subsequent derivation for real-time area light illumination. Wang et al. [2009] approximate a spherical area light using an SG providing a closed-form expression for the integral product with SG approximated BRDFs. To handle microfacet BRDFs, the spherical warp introduced by the half vector transform is approximated using a single isotropic SG. However this method fails to represent the elongated specular stretches at grazing angles. Xu et al. [2013] approximate the spherical warping using *Anisotropic Spherical Gaussians* (ASGs). A practical implementation for spherical light source illumination can be found in [Tokuyoshi 2014]. These methods have two main limitations. First, the spherical warp approximation supposes an isotropic light source emitter. Second, highly glossy surfaces tend to reveal a Gaussian shape due to the area light approximation as an SG. Close to our approach, Xu et al. [2014] propose an analytic solution for integrating an SG over a spherical triangle. The surface integral over the triangle is decomposed as a sum of signed 1D integrals using an original edge parameterization around the SG axis and evaluated using a piece-wise linear approximation. The solution is restricted to isotropic SG only and thus fails to properly render the typical specular stretches of microfacet BRDF models.

3 Our approach

Our method builds upon *Irradiance Tensors* and the contour integration method developed by Arvo [1995b], that we briefly recall in section 4. This approach represents two main challenges.

The first challenge is to get around the $O(n)$ time bottleneck for real-time rendering efficiency. We tackle this problem by rewriting the 1D integrals in a more concise way (section 5) allowing to settle for an accurate $O(1)$ time approximation using a rational peak shape integration framework (section 6). Unlike *Chebyshev* or *Fourier* approximations, our approach is bound to only 1 or 2 rational functions and doesn't suffer from any ringing artifacts.

The second challenge is to overcome the *Phong* BRDF restriction and give support for more plausible BRDFs. The half vector parameterization found in microfacet theory introduces a spherical distortion which can be difficult to predict using non isotropic polygonal light sources. Based on observations from great circle distortions, we can faithfully approximate this spherical warp using a polygonal approach (section 7) in more flexible way than previous methods.

4 Irradiance tensors and the edge integral

The *Irradiance Tensors* developed by Arvo [1995a] provide a useful framework for the analytic integration of polynomials over the sphere S^2. These polynomials correspond to n^{th}order monomial expressions described by an axis-oriented cosine lobe distribution. The integration of this expression over a subset $\Omega_A \subset S^2$ yields to the definition of n^{th}order *axial moment* about an $\mathbf{r}$ axis:

$$M^n(\Omega_A, \mathbf{r}) = \int_{\Omega_A} (\mathbf{u} \cdot \mathbf{r})^n d\mathbf{u} \tag{1}$$

Using tensors product and Stokes theorem, Arvo developed the axial moment into a 1D contour integration over the projected area light boundary Ω_A. Considering a polygonal light source, a closed-form expression for the 1D integrals can be obtained following a

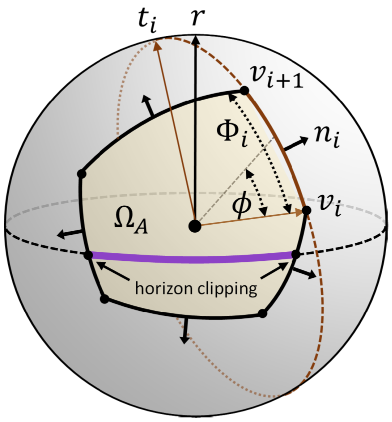

Figure 2: *The integral of a cosine lobe distribution axis, oriented toward* $\mathbf{r}$*, over a spherical region* Ω_A *is reduced to a 1D contour integration over the boundary of* Ω_A*. A closed-form expression is given for polygonal light source using a parameterization of the* i^{th} *spherical edge in the local base* v_i, t_i *(the red dotted line depicts here the great circle supporting the edge* (v_i, v_{i+1})*).*

parameterization of spherical edges along great circles (see figure 2). Let consider a polygon with m boundary edges. Following the notations depicted in figure 2, the closed-form expression is given as follow:

$$(n+1)M^n(\Omega_A, \mathbf{r}) = z\,\Omega_A - \sum_{i=0}^{m} (\mathbf{n_i} \cdot \mathbf{r})\, F(\Phi_i, c_i, \delta_i, n-1) \quad (2)$$

where $c_i = \sqrt{a_i^2 + b_i^2}$, $\delta_i = \tan^{-1}(b_i/a_i)$, $a_i = \mathbf{v_i} \cdot \mathbf{r}$ and $b_i = \mathbf{t_i} \cdot \mathbf{r}$, and where

$$F(\Phi, c, \delta, n) = \sum_{k=0}^{\frac{n-1+z_n}{2}} c^{2k+1-z_n} \int_{-\delta}^{\Phi_i - \delta} (\cos\phi)^{2k+1-z_n}\, d\phi \quad (3)$$

with $z_n = 1 - (n \mod 2)$. For a complete description of *Irradiance Tensors* and how to come to this expression, the reader shall refer to [Arvo 1995a] and [Arvo 1995b]. Note also we use a slightly different notation compared to Arvo to ease the mathematical derivations further developed in the next sections.

The sum of 1D integrals in F is evaluated in closed-form using a recurrence algorithm of complexity $O(n)$ time per edge, n being the *Phong* exponent. Implemented on a GPU, the method works well for weakly glossy surfaces ($n < 40$) but the performance drops as the *Phong* exponent increases, and becomes impractical for highly glossy surface ($n > 1000$). To reduce the evaluation cost for high *Phong* exponents, Arvo suggested early termination of the iteration loop once a desired relative accuracy is reached. But, from our experience, we observe severe performance drop-off, especially at view grazing angles of the surface and caused by a high number of iterations necessary to reach the desired accuracy. The difficulty to predict the performance makes it a nonviable solution for real-time rendering considerations. In a practical GPU implementation, the edge integral F should be ideally evaluated in $O(1)$ whatever the *Phong* exponent.

5 Reformulation of the edge integral

We propose to replace the costly edge integral evaluation of equation 3 by a cheap and accurate analytic approximation that allows constant time evaluation with any *Phong* shininess n. Setting an accurate approximation requires the knowledge or at least the intrinsic

Table 1: *Notations used throughout this paper.*

Symbol	Description
$L(x, \mathbf{v})$	radiance scattered from point x toward direction $\mathbf{v}$
Ω_A	region || solid angle sustained by area light source A
$\mathbf{n}$	surface normal
$\mathbf{v}$	normalized view vector
$\mathbf{r}$	normalized reflected view vector
$\mathbf{h}$	normalized halfway vector given by $(\mathbf{i}+\mathbf{v})/\lvert\mathbf{i}+\mathbf{v}\rvert$
$\mathbf{v_i}$	spherical projection of the i^{th} vertex of the polygonal light
n	cosine lobe exponent
$\mathbf{n_i}$	edge outer normal given by $(\mathbf{v_i} \times \mathbf{v_{i+1}})/\lvert\mathbf{v_i} \times \mathbf{v_{i+1}}\rvert$
$\mathbf{t_i}$	edge tangent vector given by $\mathbf{t_i} = \mathbf{v_i} \times \mathbf{n_i}$
Φ_i	edge arc length
F	spherical edge integral
$\tilde{F}$	approximated spherical edge line integral

characteristics of the integrand function. A common approach is to probe the edge integrand to extract these characteristics. However, in its present form, this requires the evaluation of the sum of the integrand terms. We propose to rewrite the edge integral in a different form in order to get a simpler and more compact expression. To that end, we first introduce a term f and a temporary term q defined as follow:

$$f(\phi, c, n) = \begin{cases} c\ \cos\phi \\ 1 \end{cases} \qquad q = \begin{cases} (n-1)/2 & \text{if } n \text{ is odd} \\ n/2 & \text{if } n \text{ is even} \end{cases}$$

By switching the sum and integral operators and by using the terms introduced above, the edge integral F defined in equation 3 can be re-written as follow:

$$F(\Phi, c, \delta, n) = \int_{-\delta}^{\Phi-\delta} f(\phi, c, n) \sum_{k=0}^{q} (c\ \cos\phi)^{2k}\ d\phi \quad (4)$$

The sum exhibits a geometric series of the form x^{2k} with a generic formula: $\sum_{k=0}^{q} x^{2k} = (x^{2(q+1)} - 1)/(x^2 - 1)$. This allows us to cancel out the sum and get a single function to integrate after substituting the temporary variable q.

$$F(\Phi, c, \delta, n) = \int_{-\delta}^{\Phi-\delta} \frac{(c\cos\phi)^{n+2} - f(\phi, c, n)}{(c\cos\phi)^2 - 1}\, d\phi \quad (5)$$

This reformulation allows the evaluation of the integrand term in constant time. Another advantage is that it enables smooth representation of non n integer values, especially for low n exponents. Though no indefinite integral exists, an accurate analytic approximation can be obtained from our reformulation.

6 Accurate analytic approximations

Let us consider the integrand term from the edge integral in equation 5:

$$I(\Phi, c, n) = \frac{(c\cos\phi)^{n+2} - f(\phi, c, n)}{(c\cos\phi)^2 - 1} \quad (6)$$

According to figure 3, we observe that the shape of I corresponds to symmetric peak shape functions of various *height* and *width* depending on parameters c, n having a minimum reached at $\phi = \pm\pi/2$ and a maximum at $\phi = 0$.

The core idea of our method is to approximate I using peak shape functions described by simple rational expressions with known analytic integration. The approximation relies on a simple fitting procedure that maps a peak shape function to the integrand I characteristics such as the minimum, maximum and width.

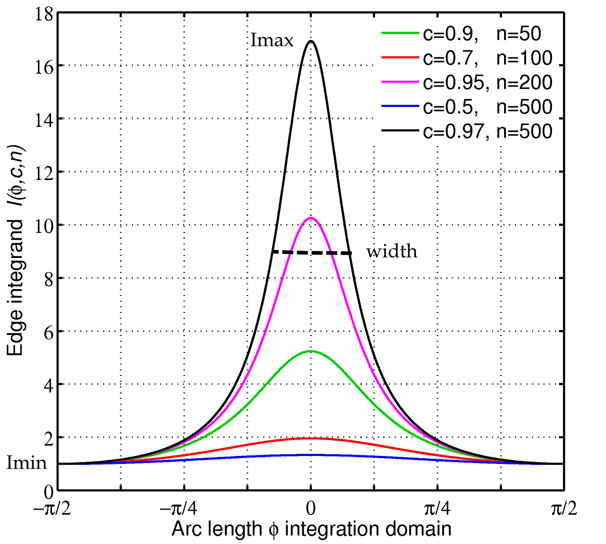

Figure 3: *The edge integrand I reveals peak shape functions of various height and width. The core idea is to approximate I using a simple and integrable peak shape function with same characteristics.*

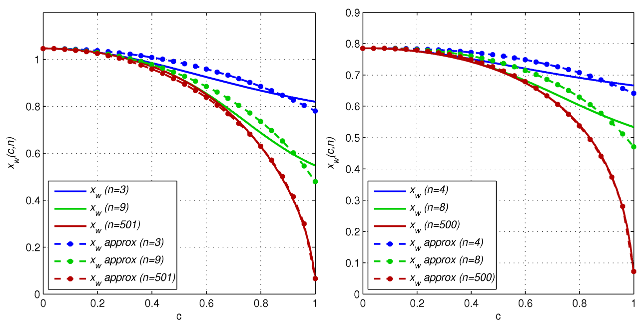

Figure 4: *Approximation of the half width x_w of integrand I for various values of n odd (left) and n even (right).*

Following equation 6, a closed formulation is given for the *minimum* and *maximum* values:

$$I_{\text{min}}(c,n) = \begin{cases} 0 \\ 1 \end{cases} \qquad I_{\text{max}}(c,n) = \begin{cases} \frac{c^{n+2}-1}{c^2-1} & n \text{ odd} \\ \frac{c^{n+2}-c}{c^2-1} & n \text{ even} \end{cases}$$

Half width estimation. The *width* is defined as the *Half Width at Half Maximum (HWHM)* which corresponds to the abscissa x_w such as $I(x_w, c, n) = (I_{\text{max}} - I_{\text{min}})/2$. However, finding a closed-form expression for x_w is somewhat more difficult. Instead, we use an empirical approximation from experimental measurements.

$$x_w(c,n) \approx \begin{cases} \frac{\pi}{3}\sqrt{1-\left(c-\frac{c}{n}\right)^2} & n \text{ odd} \\ \frac{\pi}{4}\left(1-\left(c-\frac{c}{n-1}\right)^{2.5}\right)^{0.45} & n \text{ even} \end{cases} \tag{7}$$

Even if this is a rough estimation (figure 4), the fitting procedure, described in next section, will guarantee that our approximation will pass through the point $(x_w, I(x_w, c, n))$.

6.1 General integration framework

We derive a general framework for approximating and evaluating equation 3 by means of generic peak shape functions. To that end, we first consider a generic peak function P, defined by a minimum P_{min}, a maximum P_{max} and width P_w. An accurate approximation of I can be obtained by adjusting P to the same characteristics of I. The fitting procedure consists in a scaling, offsetting and width adjustment defined as follow:

$$\tilde{I}(\phi, c, n) = \frac{I_{\text{max}} - I_{\text{min}}}{P_{\text{max}} - P_{\text{min}}}(P(\phi, x_w) - P_{\text{min}}) + I_{\text{min}}$$

Note that function parameters have been omitted for brevity. We can further reduce this expression by packing all the constant terms together:

$$\tilde{I}(\phi, c, n) = s\, P(\phi, x_w) + t \tag{8}$$

with $s = (I_{\text{max}} - I_{\text{min}})/(P_{\text{max}} - P_{\text{min}})$ and $t = I_{\text{min}} - s\, P_{\text{min}}$.

A general solution for the evaluation of equation 3 is then given by:

$$\tilde{F}(\Phi, c, \delta, n) = s \int_{-\delta}^{\Phi-\delta} P(\phi, x_w)\, d\phi + t\Phi \tag{9}$$

6.2 Peak-shape functions approximation

We studied several peak shape function families P with indefinite integrals simple enough to avoid time-consuming evaluation and providing an accurate estimate of equation 6. We validated the accuracy of our approximations with ground truth comparison by implementing an energy-conserving single-axis *Phong* model using a single-axial moment evaluation expressed as follow:

$$L(x, \mathbf{v}) = \int_{\Omega_A} f_{\text{Phong}}(\mathbf{i}, \mathbf{v})(\mathbf{n} \cdot \mathbf{i}) d\mathbf{i} = \rho_s \frac{n+1}{2\pi} M^n(\Omega_A, \mathbf{r}) \tag{10}$$

Horizon clipping. Horizon clipping takes into account the energy loss when the area light is partially below the horizon. While the clipping procedure was not explicitly addressed by [Arvo 1995b], a practicable implementation can be found in [Snyder 1996]. We adopt the same procedure in our implementations.

6.2.1 Lorentzian approximation

The simplest approximation can be found by means of a *Lorentzian* peak shape function:

$$P(\phi, c, n) \;=\; \frac{1}{1 + a\phi^2} \;\text{ with }\; \int P = \frac{1}{\sqrt{a}} \tan^{-1}\left(\sqrt{a}\phi\right) \tag{11}$$

We use the equation 7 to compute the fitting point $I(x_w, c, n)$ that roughly corresponds to the half maximum of I. Solving the equation $I(x_w, c, n) = \tilde{I}(x_w, c, n)$ yields to resolution of unknown parameter a.

$$a = \frac{1 - y_w - \frac{4 x_w{}^2}{\pi^2}}{y_w\, x_w{}^2} \quad \text{with} \quad y_w = \frac{I(x_w) - I_{\text{min}}}{I_{\text{max}} - I_{\text{min}}}$$

By replacing the integral term in equation 9 by the one defined in equation 11, we obtain an analytic approximation for F evaluated in constant-time:

$$\tilde{F} = \frac{s}{\sqrt{a}}\left(\tan^{-1}\left(\sqrt{a}(\Phi - \delta)\right) - \tan^{-1}\left(-\delta\sqrt{a}\right)\right) + t\,\Phi$$

Noting that $\tan^{-1} x - \tan^{-1} y = \tan^{-1}\left(\frac{x-y}{1+xy}\right) \bmod \pi$, this expression can be further reduced into a single arctangent evaluation to save GPU instructions.

$$\tilde{F} = \frac{s}{\sqrt{a}} \tan^{-1}\left(\frac{\sqrt{a}\Phi}{1 + a(-\delta)(\Phi - \delta)}\right) \bmod \pi \; + t\,\Phi \tag{12}$$

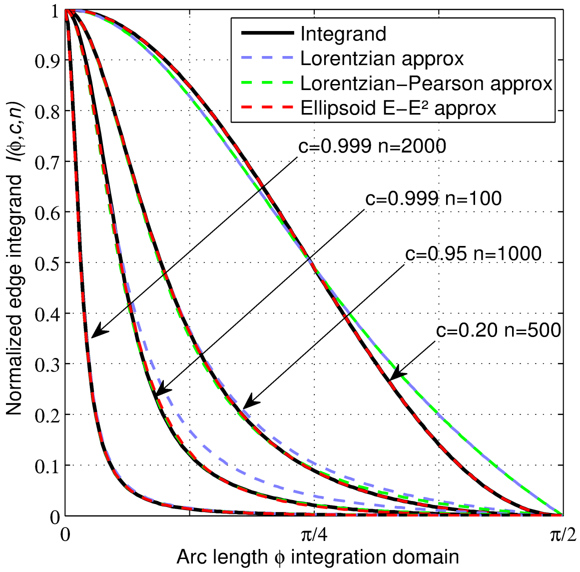

Figure 5: *Approximation of the edge integrand I using peak shape functions for various values of c and n. The Lorentzian function approximates edge integrands fairly well but lacks accuracy in the tail of I. The Lorentzian-Pearson better approximates I, but it remains inaccurate for large width values. The ellipsoid approximation provides the best accuracy whatever the width of the function.*

Error analysis. The figure 6 shows that the Lorentzian approximation is fairly accurate and close to the ground truth whatever the roughness of the surface. However, we can observe slight light leaks around the highlight shape most noticeable when increasing the overall intensity. A careful observation of the Lorentzian approximation plots in figure 5 shows that the error results from an overestimation of the function I around the tail.

6.2.2 Lorentzian - Pearson VII approximation

A better approximation around the tail can be found by combining the *Lorentzian* function with a second peak function with a shorter tail. The idea is to encompass the integrand function in the tail area with these two approximations and find a blending factor from values picked in the tail. The second peak function is defined by a *Pearson VII* function corresponding to a *Lorentzian* function raised to a power m. We chose $m = 2$ which has an indefinite integral simple enough to avoid time-consuming computations:

$$P = \left(\frac{1}{1+b\phi^2}\right)^2 \text{ and } \int P = \frac{\phi}{2(b+\phi^2)} + \frac{\tan^{-1}\left(\frac{\phi}{\sqrt{b}}\right)}{2\sqrt{b}}$$

The *Pearson VII* function behaves exactly like a Lorentzian function on its superior part and has a shorter tail in its bottom part. However, finding the b parameter, such as $\tilde{I}(x_w) = I(x_w)$ requires the resolution a polynomial equation of degree 4 involving complex computations. Fortunately, it turns out that the computation of b can be greatly simplified and save GPU computation time by reusing the a parameter computed for the *Lorentzian* approximation. From our experiments, we found that $b \approx a/2$ always enclosed the target integrand function I.

Linear blending Adjusting I_P to the same width than I gives us another approximation that underestimates I in the tail while preserving the fitting above it. The best approximation then sits between the two functions and can be found using a simple linear blending operation.

$$\tilde{I}_{LP}(\phi) = \alpha\tilde{I}_L(\phi) + (1-\alpha)\tilde{I}_P(\phi) \tag{13}$$

where

$$\alpha = \frac{\tilde{I}_P(x_{\text{tail}}) - I(x_{\text{tail}})}{\tilde{I}_P(x_{\text{tail}}) - \tilde{I}_L(x_{\text{tail}})}$$

The linear blend operation requires the evaluation of the integrand function I at a position x_{tail} located in the tail of the function. However, finding a closed-form expression for x_{tail} represents the same difficulty as for the *half width* estimation. Again, we use instead an empirical approximation:

$$x_{\text{tail}} \approx x_w + 0.3946\, x_w(0)\left(1 - (1 - x_w/x_w(0))^{12}\right) \tag{14}$$

Approximation accuracy. The *Lorentzian-Pearson* approximation greatly improves the overall accuracy of the specular highlight and suppresses most observable artifacts. Although, we still experience subtle light-leaks on areas located outside the specular highlights as shown in figure 6. These leaks are occurring when the peak shape I is very large, *i.e.* when the value c is small. A closer look at the plot in figure 5 shows that the approximation is overestimated at integration domain bounds. Especially, at $\phi = \pm\pi/2$, the first derivative is null while the *Lorentzian-Pearson* approximation is not.

6.2.3 Ellipsoid approximation

A better accuracy, especially at domain bounds, can be obtained using ellipsoid-based peak shape functions. These functions have the interesting property to behave like a Lorentzian but having a null first derivative at $\phi = \pm\pi/2$.

Ellipsoid function:

$$P_E = \frac{a}{1+(a-1)\cos^2(\phi)}\ ;\ \int P_E = \sqrt{a}\tan^{-1}\left(\frac{\tan\phi}{\sqrt{a}}\right) \tag{15}$$

Square Ellipsoid function:

$$P_{E^2} = \left(\frac{b}{1+(b-1)\cos^2(\phi)}\right)^2 ;$$
$$\int P_{E^2} = \frac{\sqrt{b}}{2}(b+1)\tan^{-1}\left(\frac{\tan\phi}{\sqrt{b}}\right) - \frac{b}{2c}(b-1)\sin(2\phi) \tag{16}$$

We follow exactly the same procedure described in sections 6.2.1 and 6.2.2 to fit I_E and I_{E^2} to I and find the best approximation using a linear blending. The parameter a for the first approximation I_E corresponds to:

$$a = \frac{y_w(1-\cos(\phi)^2)}{\cos(\phi)^2(1-y_w)} \tag{17}$$

For I_{E^2}, parameter b roughly follows $b \approx a\left(2.1 + 1.28\frac{x_w}{x_w(0)}\right)$.

Approximation accuracy. The *ellipsoid* approximation provides the best accuracy whatever the *width* of the function with unnoticeable artifacts as illustrated in figure 6.

6.3 Performance vs accuracy analysis

We implemented and tested our approximations on a GPU NVIDIA GTX 580. The table 2 provides the rendering times in milliseconds per edge along with rendering accuracy measurements using a normalized RMSE. Measurements were done considering the processing of all screen pixels, representing the most critical case, at a $720p$ resolution. Note that the timings also includes the double horizon clipping around $\mathbf{n}$ and around $\mathbf{r}$.

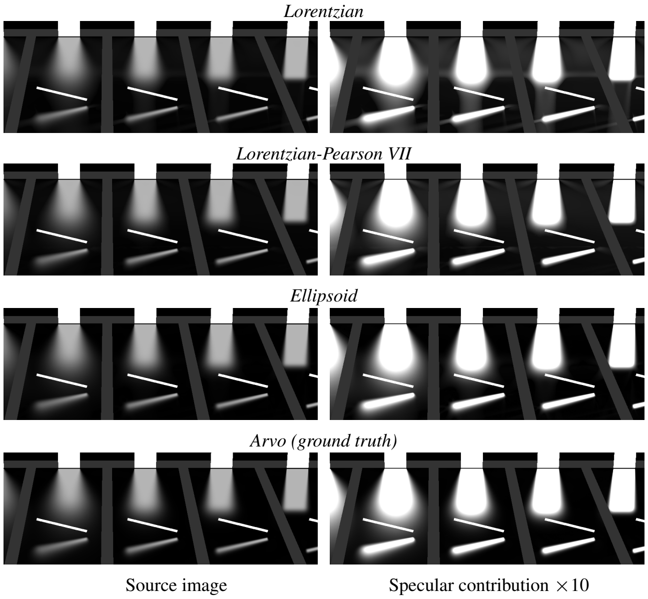

Figure 6: *Comparison of our approximations to the ground truth. The right column depicts a scaled version of the images on left one, so as to make differences visible. While light leaks are clearly visible for the Lorentzian and slightly visible for the Lorentzian-Pearson, they are hardly noticeable on the scaled version for the ellipsoid approximation.*

As expected, the rendering times obtained with Arvo's solution increases with the exponent n, while remaining constant with our approximations. The *Lorentzian* approximation achieves the best performance while the *ellipsoid* is the most accurate with unnoticeable difference with the ground truth with a small computational overhead introduce by a GPU time-consuming tangent evaluation. The *Lorentzian* approximation can be sufficient most of the time for high performance demanding application such as games. For high quality demanding applications such as lighting pre-viz for production rendering, the *Lorentzian-Pearson* or the *ellipsoid* approximation are the best choices.

7 Extension to microfacet BRDFs

The limitation to the *Phong* specular BRDF is a hard constraint for *Irradiance Tensors*. Most production renderers and modern real-time rendering engines makes use of physically based BRDFs built upon the microfacet theory. Rough surfaces rendered with a microfacet BRDF exhibits longer specular stretches, more representative of the real phenomenon. The core of the theory relies on the definition of the half vector $\mathbf{h}$ linking the microgeometry variation with the incoming radiance and the viewing direction. Another key aspect is the definition of the *normal distribution function* $D(\mathbf{h})$, responsible for the shape and the brightness of specular highlights. In this section, we demonstrate that microfacet BRDFs can be well approximated using *Irradiance Tensors* theory. Combined with our approximations, we propose a method that can accurately represent the highlight shape, especially the elongated specular stretches viewed at grazing angle, as predicted by the microfacet theory, and at a quality close to the ground truth.

Let consider the axial moment expressed in the half vector space. Following equation 1, this corresponds to the integration of the well known *Blinn-Phong* distribution D_{Blinn}

$$(n+1)M^n(\Omega'_A, \mathbf{n}) = \int_{\Omega'_A} (\mathbf{h}\cdot\mathbf{n})^n d\mathbf{h} = \int_{\Omega'_A} D_{\text{Blinn}}(\mathbf{h}) d\mathbf{h} \quad (18)$$

Table 2: *Rendering times on a GPU NVIDIA GTX 580 and accuracy measurements for the Phong specular area lighting approximation.*

Method	Exponent	Time/edge (ms)	RMSE
Arvo (exact)	$n = 100$	13.6	n/a
	$n = 500$	49	n/a
	$n = 5000$	476	n/a
Lor approx	$n = 100$	0.25	0.00435485
	$n = 500$	0.25	0.00550697
	$n = 5000$	0.25	0.00412812
Lor-Pear approx	$n = 100$	0.40	0.0036481
	$n = 500$	0.40	0.00309434
	$n = 5000$	0.40	0.00255103
Ellipsoid approx	$n = 100$	0.47	0.00150085
	$n = 500$	0.47	0.00165298
	$n = 5000$	0.47	0.00101419

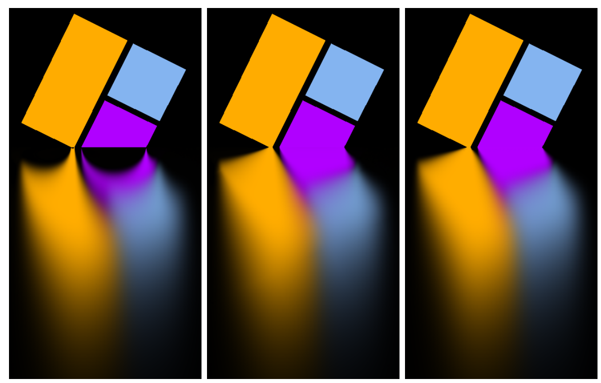

Figure 7: *Left: the area light's vertices projected in half vector space introduce distortions. Middle: our edge splitting strategy overcomes the distortion by best approximating the spherical warp for each shaded pixel using only one split. Right: the reference image.*

Given that $d\mathbf{h} = d\mathbf{i}/(4(\mathbf{h}\cdot\mathbf{v}))$ [Wang et al. 2009], this is equivalent to integrating:

$$\int_{\Omega_A} (n+1)\frac{D_{\text{Blinn}}(\mathbf{h})}{4(\mathbf{h}\cdot\mathbf{v})}\, d\mathbf{i} \quad (19)$$

Integrating the axial moment in half vector space requires the prior knowledge of the transformed spherical region Ω'_A. A naive approach can consist in performing the half vector transform on boundary edge vertices, and evaluate the 1D integral on the newly transformed edges. But as illustrated in figure 7, specular highlights get distorted by the warping distortion introduced by the half vector parameterization. Another possibility is to consider a regular sampling of each edge, but it would require a time-consuming per edge evaluation. Previous methods like [Wang et al. 2009] try to approximate this distortion using anisotropic kernels but it assumes prefect isotropic light emitters only suited for spherical area lights. In our case, the polygonal area lights are not restricted to a specific shape.

7.1 Approximating the half vector warp distortion

Finding a suitable edge parameterization in half vector space, where axial moment computations can apply, is not straightforward. However, a good approximation can be found. Intuitively, we observe that the distortion reaches its maximum at grazing angle, corresponding to situations where the normal $\mathbf{n_i}$ approaches the surface normal axis $\mathbf{n}$.

Algorithm 1: Edge splitting procedure

for *each shading point and each spherical edge* v_i, v_{i+1} **do**
  Orthogonally project $\mathbf{r}$ to the edge plane with normal $\mathbf{n_i}$ at point $\mathbf{p}$
  Normalize $\mathbf{p}$
  Do the half transform of vertices $v_i, v_{i+1} \rightarrow v'_i, v'_{i+1}$
  if $\mathbf{p} \in v_i, v_{i+1}$ **then**
    Split edge at $\mathbf{p}$
    Do the half transform $\mathbf{p} \rightarrow \mathbf{p}'$
    Evaluate edge integral for $v'_i, \mathbf{p}$ and $\mathbf{p}, v'_{i+1}$
  else
    `/* Do not split */`
    Evaluate edge integral v'_i, v'_{i+1}

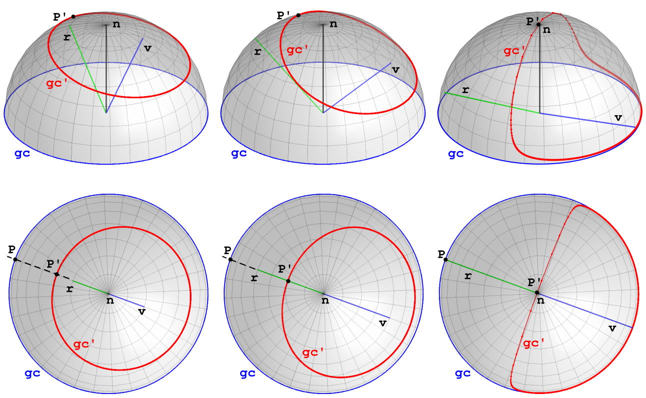

Figure 8: *Illustration of spherical distortion gc' on the great circle gc produced by the half vector transform. The distortion get its maximum for grazing viewing angles at p' which correspond to the transformation of point p, aligned with the viewing reflection* $\mathbf{r}$.

Edge splitting strategy. To give the intuition of our method, let consider the great circle gc sustained by a spherical edge and gc' it's half vector transformation. If we look at the distortion introduced by the half vector transformation in figure 8, we observe that gc' is bent toward the normal axis of gc, with a maximum elevation located at p', and aligned with the viewing vector $\mathbf{v}$. A simple explanation is that the widest angle spawned by gc with the viewing vector $\mathbf{v}$ is found at p. In other words, in the direction of $\mathbf{r}$. This simple observation is the core idea of our edge splitting strategy. Choosing a split position at p will always ensure to get the maximum distortion for an edge in half vector space. The strength of this approach is that a single split is required. Moreover, if the position p is located outside the spherical edge, no split is required and the computational overhead of our solution is greatly reduced. The full edge splitting procedure is described in the algorithm 1.

7.2 Integration of microfacet specular distributions

Other microfacet distribution functions found in literature can be fairly well approximated and integrated by means of axial moment over a spherical region.

Beckmann Approximation. The *Beckmann* distribution is a peak shape that roughly corresponds to a *Blinn-Phong* distribution for roughness values $m < 0.5$. A decent integration approximation, using a singe axial moment, can be obtained by mapping the Beckmann roughness m to the cosine power exponent n. Noting that $n \approx 2/m^2 - 2$, we obtain:

$$\int_{\Omega'_A} D_{\text{Beckmann}}(\mathbf{h})\, dh \approx \left(\frac{2}{m^2} - 1\right) M^n(\Omega'_A, \mathbf{n}) \tag{20}$$

GGX approximation. The *Towbridge-Reitz* distribution (*GGX*) [Walter et al. 2007] corresponds to an ellipsoid peak shape with a wider tail compared to the *Blinn-Phong* and *Beckmann* distributions. Specular highlights are smoother and closely match experimental measurements from real materials. We can reproduce this smooth effect by combining a second axial moment with a wider distribution. Noting that $D_{\text{GGX}}(0) = c^{-2}$ and $D_{\text{GGX}}(1) = c^2$, the integration of the *GGX* distribution term can be approximated as follow:

$$\begin{aligned}\int_{\Omega'_A} D_{\text{GGX}}(\mathbf{h})\, d\mathbf{h} \approx c^2\, \Omega'_A + (\frac{1}{c^2} - c^2)(\alpha M^{n_2}(\Omega'_A, \mathbf{n}) + \\ (1-\alpha) M^{n_1}(\Omega'_A, \mathbf{n}))\end{aligned} \tag{21}$$

From our experiments, we found that with $\alpha = 0.3$, $n_1 = \frac{2}{c^2} - 2$ and $n_2 = n_1/10$ we obtain a fairly good approximation, whatever the eccentricity parameter c.

7.3 Error analysis and performances

We implemented and tested our solution on a GPU NVIDIA GTX 580. Timings and accuracy measurements in table 3 were measured using the *Lorentzian-Pearson* approximation. We also compared our solution with reference images obtained using dense area light sampling. As shown in figure 1-c, the elongated specular stretches predicted by the microfacet theory are faithfully reproduced with an accuracy close to the ground truth. We just note a slight underestimation in terms of brightness in reflection borders for low range n values induced by the one-split approximation. In contrast to *Phong* surfaces, only one horizon clipping is performed around $\mathbf{n}$. Combined with our low-cost edge splitting approach, our solution has a limited computational overhead.

Table 3: *Rendering times per edge and accuracy measurements for microfacet specular distribution using $n = 500$ and the Lorentzian-Pearson approximation*

Spec.D	Time/edge (ms)	RMSE
Phong	0.40	0.00309434
Blinn-Phong	0.51	0.00435485
GGX	0.91	0.0036481

8 Spherical and disc area lights support

We propose a simple extension for specular surfaces lit by a spherical or a disc area light that can be extended also for diffuse surface. Our method is inspired by optical illusions produced by high-speed spinning rotations.

Spinning algorithm. The idea, depicted in figure 9, is to give the illusion of a sphere or a disc by considering the spinning of a k-sided polygon around a central axis. For a sphere, the central axis $\mathbf{n_s}$ corresponds to the direction pointing towards the sphere center (figure 9-left). For a disc, the central axis $\mathbf{n_d}$ corresponds to the centered disc normal (figure 9-right). The orientation of the polygon is computed at each shading point and is aligned with the viewing reflection vector r. This is achieved by setting the position of the first vertex $\mathbf{v_0}$ such as $\mathbf{v_0} = (\mathbf{r_i} - c_A)/|r_i - c_A|$ where r_i is the intersection point between the light plane and the r line.

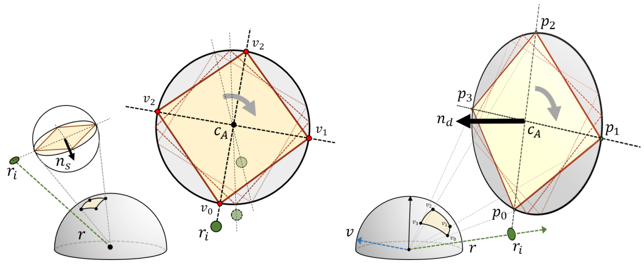

Figure 9: *Description of our spinning approach. We give illusion of a sphere or a disc by considering the spinning of a k-sided polygon around a central axis. The angular spin is driven by the viewing reflection vector r*

Non-uniform scaling operation are simply supported by just considering the inverse light transform operation on the reflection vector $\mathbf{r}$. Note that [Arvo 1995a] gives a closed-form expression for the axial moment over a spherical region but it yields to other mathematical developments. Our solution uses the same polygonal framework taking advantage of the previous approximations with little additional work.

Performance and accuracy. Our spinning approach (see figure 1-d) provides convincing specular highlights for spherical and disc area light sources with $k = 4$. The cost of evaluation is roughly the same as for a 4-sided polygon. However, we observe differences in the highlight intensity at the boundary of the disc due to the area difference between the theoretical disc and the polygonal approximation. Windowed by the specular lobe eccentricity, this difference is even worse as the viewing reflection $\mathbf{r}$ moves toward the border of the disc.

9 Conclusion

We presented efficient and accurate analytic approximations to estimate specular illumination from area light sources for high quality demanding real-time applications. We first demonstrated that the edge integrals of Arvo can be accurately approximated and evaluated in constant-time using a novel integration framework based on rational peak shape functions. We also demonstrated that the Phong restriction can be overcome using an approximation of the half vector warp distortion based on a simple edge splitting strategy. By combining one or several cosine lobe functions, a broad range of microfacet BRDF models can be implemented with very small computational overhead. Finally, using a novel spinning method, we provided support for spherical and disc area light sources at roughly the same cost than 4-sided polygons evaluation.

Some challenges still remain that would be worth exploring in the future. First, soft shadows are ignored with our method. One solution would be to back-project the scene geometry onto the area light and perform a negative contour integration along the geometry silhouette. Textured area lights is also a hard problem for which no satisfying solution exists yet. One possibility with our approach is to modulate the specular term with pre-integrated mip-mapped textures as done in [Drobot 2014]. One other approach would be to look for the varying luminaries derivations introduced by Arvo [1995a] and developed by Chen and Arvo [2000]. Finally, some broader lighting problems such as real-time *environment lighting* or interactive *Global Illumination* would be interesting to address. We believe that our approximation framework can be particularly well adapted to these techniques and may overcome some of the issues encountered with *Spherical Gaussians* or *VPLs* approaches.

References

ARVO, J. 1995. *Analytic methods for simulated light transport*. PhD thesis, Yale University.

ARVO, J. 1995. Applications of irradiance tensors to the simulation of non-lambertian phenomena. In *SIGGRAPH'95*, 335–342.

ARVO, J. 1995. Stratified sampling of spherical triangles. In *SIGGRAPH '95*, ACM, New York, NY, USA, 437–438.

BAO, H., AND PENG, Q. 1993. Shading models for linear and area light sources. *Computers Graphics 17*, 2, 137 – 145.

CHEN, M., AND ARVO, J. 2000. A closed-form solution for the irradiance due to linearly-varying luminaires. In *Eurographics Workshop on Rendering Techniques 2000*, 137–148.

DROBOT, M. 2014. Physically based area lights. In *GPU Pro 5*, 67–100.

KARIS, B. 2013. Real shading in unreal engine 4. In *part of ACM SIGGRAPH 2013 Courses*, SIGGRAPH '13, 22:1–22:8.

KELLER, A. 1997. Instant radiosity. In *SIGGRAPH '97*, 49–56.

NICHOLS, G., AND WYMAN, C. 2009. Direct illumination from dynamic area lights. In *SIGGRAPH '09: Posters*, SIGGRAPH '09, 82:1–82:1.

PICOTT, K. 1992. Extensions of the linear and area lighting models. *IEEE Computer Graphics and Applications 12*, 2, 31–38.

POULIN, P., AND AMANATIDES, J. 1991. Shading and shadowing with linear light sources. *Computers & Graphics 15*, 2.

RITSCHEL, T., GROSCH, T., KAUTZ, J., AND SEIDEL, H.-P. 2008. Interactive Global Illumination based on Coherent Surface Shadow Maps. In *Graphics Interface*, 185–192.

SNYDER, J. M. 1996. Area light sources for real-time graphics. Tech. rep.

TANAKA, T., AND TAKAHASHI, T. 1997. Fast analytic shading and shadowing for area light sources. *Computer Graphics Forum 16*, C231–C240.

TOKUYOSHI, Y. 2014. Virtual spherical gaussian lights for real-time glossy indirect illumination. In *SIGGRAPH Asia 2014 Technical Briefs*, SA '14, 17:1–17:4.

UREÑA, C., FAJARDO, M., AND KING, A. 2013. An area-preserving parametrization for spherical rectangles. *Computer Graphics Forum 32*, 4, 59–66.

WALTER, B., FERNANDEZ, S., ARBREE, A., BALA, K., DONIKIAN, M., AND GREENBERG, D. P. 2005. Lightcuts: A scalable approach to illumination. In *SIGGRAPH 2005*.

WALTER, B., MARSCHNER, S. R., LI, H., AND TORRANCE, K. E. 2007. Microfacet models for refraction through rough surfaces. Eurographics Association, EGSR'07, 195–206.

WANG, J., REN, P., GONG, M., SNYDER, J., AND GUO, B. 2009. All-frequency rendering of dynamic, spatially-varying reflectance. In *SIGGRAPH Asia 2009*, ACM, 133:1–133:10.

XU, K., SUN, W.-L., DONG, Z., ZHAO, D.-Y., WU, R.-D., AND HU, S.-M. 2013. Anisotropic spherical gaussians. *ACM Trans. Graph. 32*, 6 (Nov.), 209:1–209:11.

XU, K., CAO, Y.-P., MA, L.-Q., DONG, Z., WANG, R., AND HU, S.-M. 2014. A practical algorithm for rendering interreflections with all-frequency brdfs. *ACM Trans. Graph. 33*, 1 (Feb.).