Analytic Approximations for Real-Time Area Light Shading

Pascal Lecocq*, Arthur Dufay, Gaël Sourimant, Jean-Eudes Marvie

Abstract—We introduce analytic approximations for accurate real-time rendering of surfaces lit by non-occluded area light sources. Our solution leverages the Irradiance Tensors developed by Arvo for the shading 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 parameterization. Thanks to this novel extension, we accurately approximate common microfacet BRDFs, providing a practical method producing specular stretches that closely match the ground truth 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.

Index Terms—area light, shading, analytic, microfacet, axial moment, real-time

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 [1] [2], 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. Arvo [3] provides an exact analytic solution for the shading of glossy surfaces lit by a non-occluded polygonal light source. But its implementation relies on an expensive contour integration method, and is restricted to Phong surfaces. A recent and concurrent approach tackles this problem using Linearly Transformed Cosine distributions (LTC) [4]. However, the solution requires per-brdf precomputed tables built upon an expensive minimization technique.

Finding a solution combining accuracy, flexibility and real-time performances is a challenging problem with many expectations on high quality demanding applications such as lighting pre-viz tools, game engines or production renderers.

In this paper we address these shortcomings by leveraging the Irradiance Tensors developed by Arvo with accurate analytic approximations (Figure 1). We further extend the method to handle multiple axis-oriented Cosine lobes and overcome the Phong restriction, enabling support for microfacet BRDFs. Finally, we introduce an original polygon spinning method allowing surfaces shaded by spherical and disc area lights 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.
- An analytical approximation for the multiple product of axis-oriented Cosine lobes that enables the integration of more complex BRDFs over spherical polygons.
- A low-cost edge splitting strategy for handling the warp distortion introduced by the half vector parameterization that enables microfacet BRDF support.
- An original spinning algorithm enabling spherical and disc area lighting and leveraging our approximations, that supports non-uniform scale operations.

2 RELATED WORK

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 [5], [6], 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) [7], reducing
the specular term integration to a many point lights calculation. Clustering methods [8] have been proposed to further reduce the algorithm complexity of VPLs with successful application in real-time rendering [9], [10]. 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 [11] for Phong area lighting. The MRP here is defined as the closest point from the viewing reflection direction. Instead, Drobot [1] considers a point in the area of intersection between an area light and a cone with aperture parameterized by the surface roughness. Karis [2] 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 decent approximation. Bao and Peng [12] approximate the double integral with 2D polynomials using a low degree Taylor series expansion, limiting their method to low exponent Phong surfaces. Tanaka and Takahashi [13] extend the linear area light method of Poulin [14] and decompose the solid angle into 1D signed integrals along edge great circle. 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 [3] 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 [15]. 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 derivations for real-time area light illumination. Wang et al. [16] approximate a spherical area light with an SG providing a closed-form expression for the integral product with an SG-approximated BRDF. 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. [17] approximate the spherical warping using Anisotropic Spherical Gaussians (ASGs). A practical implementation for spherical light source illumination can be found in [18]. 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, Wang et al. [19] use a piece-wise linear approximation of the SG for polygonal visibility evaluation, reducing the 1D edge integrals to analytic expressions. Conversely, Xu et al. [20] use an edge parameterization on the parallel plane to derive 1D edge integral expressions evaluated using a piece-wise linear approximation. Both method require a piece-wise decomposition of the integration domain. Furthermore, they are restricted to isotropic SGs only and fail to represent the anisotropy of microfacet distributions.

**Linear Transformed Cosine.** A recent and concurrent approach proposed by Heitz et al. [4] uses linear transforms of a clamped Cosine distribution (referred hereafter as LTC) to approximate isotropic BRDFs including microfacets. As a
result, by applying the inverse transform on the polygon, the shading operation is reduced to an analytical form factor calculation. LTC is accurate and simpler to evaluate than ours. However, matching a BRDF requires the pre-computation and storage of the transformation matrices for the set of incident directions and roughness values. The need for storage and memory access is not always desirable especially on low-end GPUs. Our solution is fully analytic and does not require any pre-computation step or storage. Furthermore, our method enables integration in the half-vector space, providing higher degree of liberty. In that sense, both methods are complementary.

3 OUR APPROACH

Our method builds upon Irradiance Tensors and the contour integration method developed by Arvo [3], that we briefly recall in Section 4. This approach represents several 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.

Second, the integration of more complex distributions over a spherical polygon combining several axis-oriented Cosine lobes is yet another challenge. By borrowing operators from Spherical Gaussians, we derive simple analytic expressions (section 7) approximating accurately the multiple product of Cosine lobes.

Finally, the last 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 8) in a more flexible way than previous methods.

4 IRRADIANCE TENSORS AND THE EDGE INTEGRAL

The Irradiance Tensors developed by Arvo [21] 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 spherical region \(\Omega_A \subset S^2\) yields to the definition of \(n^{th}\)order axial moment about an \(r\) axis:

\[
M^n(\Omega_A, r) = \int_{\Omega_A} (\mathbf{u} \cdot \mathbf{r})^n d\mathbf{u}
\]  

(1)

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

\[
(n + 1)M^n(\Omega_A, r) = z_n\Omega_A \sum_{i=0}^{m} (\mathbf{n}_i \cdot \mathbf{r}) F(\Phi_i, c_i, \delta_i, n - 1)
\]  

(2)

with

\[
c_i = \sqrt{a_i^2 + b_i^2}; \quad \delta_i = \tan^{-1}(b_i/a_i);
\]

\[
a_i = \mathbf{v}_i \cdot \mathbf{r}; \quad b_i = \mathbf{t}_i \cdot \mathbf{r}
\]

and

\[
F(\Phi, c, \delta, n) = \sum_{k=0}^{\frac{n-1}{2}} c^{2k+1-z_n} \int_{-\delta}^{\Phi_k - \delta} (\cos \phi)^{2k+1-z_n} d\phi
\]  

(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 [21] and [3]. Note also that 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 grazing view angles of the surface because of 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.

### TABLE 1
Notations used throughout this paper.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L(x, v)$</td>
<td>radiance scattered from point $x$ toward direction $v$</td>
</tr>
<tr>
<td>$\Omega_A$</td>
<td>region \parallel solid angle sustained by area light source $A$</td>
</tr>
<tr>
<td>$n$</td>
<td>surface normal</td>
</tr>
<tr>
<td>$v$</td>
<td>normalized view vector</td>
</tr>
<tr>
<td>$r$</td>
<td>normalized reflected view vector</td>
</tr>
<tr>
<td>$h$</td>
<td>normalized halfway vector given by $(i + v) /</td>
</tr>
<tr>
<td>$m$</td>
<td>number of boundary edges on the polygonal light</td>
</tr>
<tr>
<td>$v_i$</td>
<td>spherical projection of the $i^{th}$ vertex of the polygonal light</td>
</tr>
<tr>
<td>$n$</td>
<td>cosine lobe exponent</td>
</tr>
<tr>
<td>$n_i$</td>
<td>edge outer normal given by $(v_1 \times v_{i+1}) /</td>
</tr>
<tr>
<td>$t_i$</td>
<td>edge tangent vector given by $t_i = v_i \times n_i$</td>
</tr>
<tr>
<td>$\Phi_i$</td>
<td>edge arc length</td>
</tr>
<tr>
<td>$F$</td>
<td>spherical edge integral</td>
</tr>
<tr>
<td>$\tilde{F}$</td>
<td>approximated spherical edge line integral</td>
</tr>
</tbody>
</table>

## 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 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 follows:

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

By switching the sum and integral operators and by using the terms introduced above, the edge integral $F$ defined in equation 3 can be rewritten as follows:

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

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 $q$.

\[
  F(\Phi, c, \delta, n) = \int_{-\Phi}^{\Phi} (c \cos \phi)^{n+2} / (c \cos \phi)^2 - 1 \, d\phi 
\]

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:

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

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.

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

\[
  I_{\text{min}}(c, n) = \begin{cases} 
  0 & \text{if } n \text{ is even} \\
  1 & \text{if } n \text{ is odd}
  \end{cases}
\]

**Half width estimation.** The width is defined as the Half Width at Half Maximum (HWHM) which corresponds to the abscissa $x_w$ such as

\[
  I(x_w, c, n) = (I_{\text{max}} - I_{\text{min}}) / 2 - I_{\text{min}}
\]

However, finding a closed-form expression for $x_w$ is somewhat more difficult. One approach could consist in storing pre-computed values for $x_w$ in a 2D table for discrete entries $(c, n)$. Another approach is to settle an analytic approximation. Following experimental measurement studies, we found that $x_w$ can be empirically approximated as follow:

\[
  x_w(c, n) \approx \begin{cases} 
  \frac{n}{4} \sqrt{1 - \left(\frac{c}{n}\right)^2} & \text{if } n \text{ is odd} \\
  \frac{n}{4} \left(1 - \left(\frac{c}{n}\right)^2\right)^{2.5} & \text{if } n \text{ is even}
  \end{cases}
\]

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 $I_{\text{min}}$, a maximum $I_{\text{max}}$ 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:

\[
  \tilde{I}(\phi, c, n) = s \cdot P(\phi, x_w) + t
\]

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

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

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

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

(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:

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

(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 [3], a practicable implementation can be found in [15]. 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:

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

(11)

We use 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 $a$.

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

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:

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

(12)

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:

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

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 $I(x_w) = \tilde{I}(x_w)$ 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.

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

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 $x_{\text{tail}}$ located in the tail of the function. However, finding a closed-form expression for $x_{\text{tail}}$ represents the same difficulty as for the half width estimation. Again, we use instead an empirical approximation:

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

**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: $P_E = a \frac{1}{1 + (a - 1) \cos^2(\phi)}$ \hfill (15)

Indefinite integral: $\int P_E = \sqrt{a} \tan^{-1}\left(\frac{\tan \phi}{\sqrt{a}}\right)$ \hfill (16)

Square Ellipsoid function:

$$P_{E^2} = \left(\frac{b}{1 + (b - 1) \cos^2(\phi)}\right)^2$$

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

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:

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

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 include the double horizon clipping around $n$ and around $r$.

As expected, the rendering time 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 and a small computational overhead introduced 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 previz for production rendering, the Lorentzian-Pearson or the ellipsoid approximation are the best choices.

7 Multiple-axes moments evaluation

The Irradiance tensors allow evaluation of multiple axes using decomposition of tensor product. Arvo [3] proposed a closed form expression for the double-axis moment described by the product of two cosine lobes of order $n$ and $1$. However, the generalization for arbitrary orders combining
An interesting property of SGs is that the product of two SGs is another SG, computed exactly using simple analytic formulas. For a consistent notation with SG, let us introduce the spherical function $C$ to represent a Cosine lobe with a magnitude $\mu$.

$$C(u, r, n, \mu) = \mu(u \cdot r)^n$$  \hspace{1cm} (20)

It turns out that a Cosine lobe can be fairly well approximated by an SG in most situations.

$$C(u, r, n, \mu) \approx G(u, r, \lambda, \mu)$$  \hspace{1cm} (21)

We have also observed that the product of two Cosine lobes closely behaves like the product of two SGs due to the shape similarities. Our idea is to borrow product operators from SGs to derive a single Cosine lobe approximation from the product of two Cosine lobes. Our method consists in mapping an SG on each Cosine lobe and evaluate the parameters of the product in the SG domain. Then, we back-transform the results in the Cosine lobe domain by mapping a Cosine lobe on the resulting SG.

The mapping of a Cosine lobe from/to an SG is achieved by solving the equations such as $C$ and $G$ have the same width at half maximum. These parameters are computed exactly as follows:

$$\lambda = \frac{-\ln 2}{\sqrt{2} - 1} ; \quad n = \frac{-\ln 2}{\ln \frac{\ln 2 + \lambda}{\lambda}}$$  \hspace{1cm} (22)

The product of two Cosine lobes $C_1$ and $C_2$ is then approximated as follows:

$$C_1(r_1, n_1, \mu_1)C_2(r_2, n_2, \mu_2) \approx G_1(r_1, \lambda_1, \mu_1)G_2(r_2, \lambda_2, \mu_2)$$
$$\approx G(r_p, \lambda_p, \mu_p)$$
$$\approx C(r_p, \lambda_p, \mu_p)$$

where

$$r_p = \frac{p_m}{\|p_m\|} ; \quad p_m = \frac{\lambda_1 r_1 + \lambda_2 r_2}{\lambda_1 + \lambda_2}$$
$$\lambda_p = (\lambda_1 + \lambda_2)\|p_m\|$$
$$\mu_p = C_1(r_p, r_1, n_1, \mu_1)C_2(r_p, r_2, n_2, \mu_2)$$

Note that we do not evaluate explicitly the SGs but only borrow their product operators for deriving our approximation. The only difference lies in the computation of magnitude $\mu_p$ computed exactly as the product of the two Cosine lobes at $r_p$. Note that the same reasoning could be employed to approximate the product of anisotropic Cosine lobes by borrowing operators derived in [17]. We left this derivation for future works.

### 7.2 Results and error analysis

We compared our lobe product approximation $C$ against an exact product of $C_1$ and $C_2$ for various exponents and angles in polar coordinates (see Figure 7). In most situations, our single Cosine lobe approximation closely matches the product of two Cosine lobes. However, when the angle between the two lobes is very large, we observe a misalignment between the theoretical product and our approximation. Although, in this situation, the error is balanced by the very low magnitude of the product, close to 0 except when the two lobes have a low exponent.

**Double-axis Phong implementation.** We further validated our approach by implementing the energy-conserving double-axis Phong model. According to the formulas given...
Fig. 7. Polar plots of the product (in black) of two Cosine lobes (in red and blue) against our analytic approximation (in green) following the formulas given in 7.1 for various eccentricity and angle configurations. Our single lobe approximation closely matches the product of two lobes in most situations. The critical scenario arises when the two lobes are far apart. In this case, we observe a misalignment of the two lobes but the error remains unnoticeable due to the low magnitude of the product as illustrated in the rendered pictures.

in Section 7.1, the model can be approximated by a single axial moment as follow:

\[
L(x, v) = \rho_s \frac{n + 2}{2\pi} \int_{\Omega_A} (i, r)^n (n \cdot i) d\Omega
= \rho_s \frac{n + 2}{2\pi} \int_{\Omega_A} C_1(i, r, n, l) C_2(i, n, 1, 1) d\Omega
\approx \rho_s \frac{n + 1}{2\pi} \int_{\Omega_A} C(i, r, n, l, \mu_p, \mu_p) d\Omega
\approx \rho_s \frac{n + 1}{2\pi} M^{np}(\Omega_A, r_p)
\]

This approximation is energy-conserving thanks to the new normalization factor resulting from the Cosine product parameterization and the axial moment normalization.

Images in Figure 8 show a rendering comparison of the double-axis Phong approximation against a ground truth solution. The visual difference is unnoticeable whatever the surface Phong exponent or the angular configuration. As predicted, the error occurring when the angle of the two lobes is large is visually dismissed by the low magnitude of the product.

\[n + 2 \quad \frac{M^n(\Omega_A, n)}{2\pi} = \int_{\Omega_A} \frac{n + 2}{2\pi} (h \cdot n)^n dh = \int_{\Omega_A} D_{\text{Blinn}}(h) dh \]

Given that \(dh = d\Omega/(4(h \cdot v))\), this is equivalent to integrating:

\[\int_{\Omega_A} \frac{D_{\text{Blinn}}(h)}{4(h \cdot v)} d\Omega\]

Integrating the axial moment in the half vector space requires the prior knowledge of the transformed spherical region \(\Omega'_A\). 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 9, specular highlights get distorted by the warping distortion introduced by the half vector parameterization. Another possibility is to sample
Fig. 8. Comparison of a two-axis Phong approximation (top) against a ray-traced ground truth solution (bottom).

Table 2
Rendering times in milliseconds per edge on a GPU NVIDIA GTX 580 and GTX 980Ti with rendering accuracy for the three peak shape approximations.

<table>
<thead>
<tr>
<th>Method</th>
<th>Exponent</th>
<th>Time/edge (ms)</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arvo (exact)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>n = 100</td>
<td>13.6</td>
<td>0.41</td>
<td>n/a</td>
</tr>
<tr>
<td>n = 500</td>
<td>49</td>
<td>1.57</td>
<td>n/a</td>
</tr>
<tr>
<td>n = 5000</td>
<td>476</td>
<td>9.8</td>
<td>n/a</td>
</tr>
<tr>
<td>Lor approx</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>n = 100</td>
<td>0.25</td>
<td>0.12</td>
<td>0.004354</td>
</tr>
<tr>
<td>n = 500</td>
<td>0.25</td>
<td>0.12</td>
<td>0.005906</td>
</tr>
<tr>
<td>n = 5000</td>
<td>0.25</td>
<td>0.12</td>
<td>0.004128</td>
</tr>
<tr>
<td>Lor-Pear approx</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>n = 100</td>
<td>0.40</td>
<td>0.125</td>
<td>0.003641</td>
</tr>
<tr>
<td>n = 500</td>
<td>0.40</td>
<td>0.125</td>
<td>0.003094</td>
</tr>
<tr>
<td>n = 5000</td>
<td>0.40</td>
<td>0.125</td>
<td>0.002551</td>
</tr>
<tr>
<td>Ellipsoid approx</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>n = 100</td>
<td>0.47</td>
<td>0.127</td>
<td>0.001500</td>
</tr>
<tr>
<td>n = 500</td>
<td>0.47</td>
<td>0.127</td>
<td>0.001652</td>
</tr>
<tr>
<td>n = 5000</td>
<td>0.47</td>
<td>0.127</td>
<td>0.001014</td>
</tr>
</tbody>
</table>

each edge, but it would require a time-consuming per edge evaluation. Previous methods like [16] try to approximate this distortion using anisotropic kernels but it supposes perfect isotropic light emitters only suited for spherical area lights. In our case, the polygonal area lights are not restricted to a specific shape.

8.1 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 angles, corresponding to situations where the normal \( n_i \) approaches the surface normal axis \( n \).

Edge splitting strategy. To give the intuition of our method, let consider the great circle \( gc \) sustained by a spherical edge and \( gc' \) its half vector transformation. If we look at the distortion introduced by the half vector transformation in Figure 10, 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 \( v \). A simple explanation is that the widest angle spawned by \( gc \) with the viewing vector \( v \) is found at \( p \). In other words, in the direction of \( r \). 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 Algorithm 1.

8.2 Approximation of microfacet specular distributions
A broad range of microfacet distribution functions found in the literature can be fairly well approximated and integrated.
Algorithm 1: Edge splitting procedure

for each shading point and each spherical edge \(v_i, v_{i+1}\) do

Orthogonally project \(r\) to the edge plane at normal \(n_i\) at point \(p\)
Normalize \(p\)
Do the half transform of vertices \(v_i, v_{i+1} \rightarrow v'_i, v'_{i+1}\)

if \(p \in v_i, v_{i+1}\) then

Split edge at \(p\)
Do the half transform \(p \rightarrow p'\)
Evaluate edge integral for \(v'_i, p\) and \(p, v'_{i+1}\)
else

/* Do not split
Evaluate edge integral \(v'_i, v'_{i+1}\) */

end

The resulting integral is normalized so that \(\int D h \cdot n = 1\)

8.3 Results and improvements

We implemented and tested our solution on a GPU NVIDIA GTX 580 and a GTX 980Ti. The table 3 gives the timing overhead compared to Phong distribution in contrast to Phong, only one horizon clipping is performed around \(n\). As a result, combined with our low-cost edge splitting approach, our solution has a limited computational overhead.

We also compared our solution with reference images obtained with a ray-traced solution. As shown in Figure 11, the elongated specular stretches predicted by the microfacet theory are faithfully reproduced with an accuracy close to the reference in most situation. However, under certain roughness and geometric configurations we can notice a lack of brightness, especially when the viewing reflection is close to an edge border and when the surface is rough (see Figure 13). This phenomenon tends to disappear as the roughness goes to 0 \((n \to \infty)\). These artifacts result from an underestimation of the theoretical solid angle in the half-vector space. As shown in Figure 12, our edge-splitting strategy fails to capture important distortion of the great circle sustained by an edge.

Edge-split balancing strategy. To reduce the visual artifacts, we propose a simple balancing strategy that captures the inflection points on the half-transformed great circle. The idea is to balance the split position between the mid point

Fig. 10. Illustration of the spherical distortion \(g_{c'}\) of the great circle \(g_c\) 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 \(r\).

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 single 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:

\[
\int_{\Omega'_A} D_{\text{Beckmann}}(h) \, dh \approx \frac{1}{\pi m^2} M^n(\Omega'_A, n) \tag{25}
\]

GGX approximation. The GGX/Torrbridge-Reitz distribution [23] corresponds to an ellipsoid peak shape function producing smoother specular highlights that better match experimental measurements from real materials. At the difference to Blinn-Phong, the distribution has a smoother falloff, converging to \(c^2\) at the domain bound when \(h \cdot n = 0\). To mimic this behavior, we split the distribution into one constant term \(c_0 = c^2\) and one lobe term corresponding to the GGX distribution shifted down to 0. The integration of the constant term reverts to the calculation of the solid angle sustained by the area-light in the half vector domain. The integration of the second term is approximated by a weighted sum of two axial moments of order \(n_1\) and \(n_2\) where \(n_2\) has a wider eccentricity compared to \(n_1\) to reproduce the GGX smoothness.

\[
\int_{\Omega'_A} D_{\text{GGX}}(h) \, dh \approx \frac{1}{\pi} \left( c_0 \Omega'_A + c_1 \frac{n_1 + 2}{2} M^{n_1}(\Omega'_A, n) + c_2 \frac{n_2 + 2}{2} M^{n_2}(\Omega'_A, n) \right) \tag{26}
\]

The resulting integral is normalized so that \(\int D h \cdot n = 1\) so the weights \(c_1\) and \(c_2\) are chosen accordingly such as \(c_0 + c_1 + c_2 = 1\). Using a least square fitting method, we found that

\[
n_1 = \frac{2}{c^2} - 2 \quad n_2 = \frac{n_1}{10} \quad \text{with weights} \quad c_1 = 0.7 \quad (1 - c^2) \quad c_2 = 0.3 \quad (1 - c^2)
\]

provide a decent approximation whatever the eccentricity parameter \(c\).
Fig. 11. Surface lit with a Blinn-Phong and GGX microfacet distribution rendered using our analytic approximations compared against a reference ray-traced solution.

Fig. 12. Solid projection of a rectangular area light on the hemisphere (in blue) and its corresponding theoretical projection in the half-vector space (in red). On the left, the edge split approximation (in green) underestimates the theoretical projection when the surface is rough and when \( r \) is at grazing angle against the edge plane. Using a simple balancing heuristic, controlled by \( r \) (middle) and by the roughness (right), we better recover the theoretical solid angle of the half-vector transformed polygon.

Algorithm 2: Edge-split balancing procedure

\[
\begin{align*}
m & = \sqrt{2.0/(n+2)} \\
k & = 2 \cdot |t_i \cdot v| \\
P_m & = (v_i + v_{i+1})/\|v_i + v_{i+1}\| \\
p & = (1 - k m) \ p + k \ m \ P_m \\
p & = p/\|p\|
\end{align*}
\]

As illustrated in Figure 13, the balancing strategy overcomes most of the visual artifacts. The main advantage is that no additional split is required. However, in some extreme cases, the single split approach still shows some differences at extreme grazing angles, or when the light source is very large.

9 **Spherical and disc area lights support**

The analytic approximations we have developed so far are restricted to polygonal light emitters. Light emitters based on analytic shapes such as *Sphere* and *Disc*, represent an interesting class of area-lights that extends the set of luminaries representation. Note that Arvo [21] proposed another closed form expression to evaluate the axial moment over a spherical light. However, the solution is based on another parameterization and recurrence evaluation solved in \( O(n) \) time. Furthermore, the method cannot handle non-uniform scaling operation.

We propose a simple method to shade specular surfaces by spherical and disc area lights. Our method leverages the polygonal approach with our approximations, supporting the partial visibility of the lights and non-uniform scaling operations.

9.1 **Spinning polygon strategy**

Our approach is inspired by optical illusions produced by high-speed spinning rotations. The idea, depicted in Figure 14, is to give the illusion of a sphere or a disc by considering the spinning of a \( k \)-sided polygon around a unit disc normal axis and evaluate the axial moment over the resulting polygon. The orientation \( \theta_m \) of the polygon is computed at each shading point and should theoretically be chosen such as

\[
\theta_m = \arg \max \ M^n(P(\theta), r)
\]

**Disc area light.** Determining the best orientation \( \theta_m \) is a nontrivial maximization problem that depends on several parameters difficult to solve in real-time. However, according to experimental measurements, we noticed that when the roughness goes to 0 \( (n \rightarrow \infty) \), the best orientation is found toward \( r' \), the intersection of the \((p, r)\) line with the disc plane. Conversely, when the roughness goes to 1 \((n \rightarrow 1)\) the best orientation is found towards \( p' \) the perpendicular projection of \( p \) on the disc plane. By setting-up a linear blending between the two positions driven by the roughness \( m \) we can obtain a decent approximation for
the best orientation. Another problem is the area difference between the polygon and the disc resulting in an underestimation of the brightness when the surface is rough. Windowed by the Cosine lobe eccentricity, the difference is reduced when the surface is highly specular. To reduce the difference in all scenarios, we linearly scale the polygon such as area(P) = area(D) as the roughness goes to 0.

**Sphere area light.** For a sphere light, the procedure is roughly the same as for the disc. The main difference lies in the orientation of the disc, facing the shading point \( p \). Also, a scaling factor \( s \) is applied to the unit disc to take into account of the solid angle sustained by the unit sphere (see Figure 15). For both luminaries, non-uniform scaling operations are simply supported by transforming \( p \) and \( r \) into the area light local space prior the orientation estimation. The resulting polygon is then back-transformed into the world space. The full procedure for evaluating the shading from disc and sphere area lights is described in Algorithm 3.

### 9.2 Results and limitations

Our spinning approach (see Figure 1-d) provides convincing specular highlights for spherical and disc area light sources with \( k = 4 \). The partial visibility is properly handled and non-uniform scaling operations allows the representation of ellipses and ellipsoid shaped area lights. In terms of performance, we didn’t notice a significant difference compared to the quad evaluation. However, our approach has some limitations. First, our method to compensate for the area difference is just an approximation. As we do not integrate the circular shape we observe slight brightness differences especially with moderate glossy surfaces. Second, when the predicted position to orient the polygon is too close to

---

**Algorithm 3: Disc and Sphere light shading**

\[
m = \sqrt{2/(n+2)}
\]

\[
s = 1 /* default disc scale */
\]

**for each shading point \( p \) do**

- /* handle non uniform-scaling */
- Transform \( p \) and \( r \) in area light local space
- **if Sphere then**
  - \( \mathbf{n}_D = \text{normalize}(\mathbf{p}) \)
  - \( d = \text{length}(\mathbf{p}) \)
  - \( s = d/\sqrt{d^2 - 1} \)
  - \( k = |\mathbf{n}_D \cdot \mathbf{p}|/d /* prevent early clip */
  - \( s = s (1 - m k + m k \sqrt{\pi/2}) /* +area diff */
  - \( r' = \text{intersection with the disc plane in the } \mathbf{r} \text{ direction} \)
  - \( \mathbf{p}' = \text{orthogonal projection of } \mathbf{p} \text{ to the disc plane} \)
  - \( \mathbf{v}_0 = s \text{ normalize}((m - 1)r' + m \mathbf{p}') \)
  - \( \mathbf{v}_1 = \mathbf{v}_0 \times \mathbf{n}_D \)
  - \( \mathbf{v}_2 = -\mathbf{v}_0 \)
  - \( \mathbf{v}_3 = -\mathbf{v}_1 \)
  - Back-transform \( \mathbf{v}_0 \rightarrow \mathbf{v}_3 \) in world space
  - Compute the axial moment on polygon \( \mathbf{v}_0 \rightarrow \mathbf{v}_3 \)

---

**Fig. 14.** Description of our spinning approach. We give illusion of a disc by considering the spinning of a \( k \)-sided polygon \( P \) around the disc normal \( \mathbf{n}_D \). The orientation of \( P \) (here a quad) in the unit disc is chosen such as \( \arg \max M(P, \mathbf{r}, \mathbf{n}) \). The shading at point \( p \) is then estimated by evaluating the axial moment on the resulting polygon \( P \).

**Fig. 15.** The axial moment over a spherical light is estimated by facing the unit disc toward the shading point \( p \) (left). The disc (and hence the polygon \( P \)) is then scaled by \( s \) to take into account of the solid angle of the sphere (right).
the disc center, the excessive rotations of the polygon on close pixels results in perceptible brightness variations. With a Phong distribution, this variation is mostly perceptible when the disc highlight is viewed at grazing angle. With microfacet distributions, the phenomenon is highly perceptible due to the distortion introduced by the half vector transform. This variation disappears as $n$ goes to infinity.

Fig. 17. Limitations of our polygon spinning strategy. In some particular cases, the excessive rotation of the polygon in a close area introduces undesirable brightness variations. With Phong distribution (left), the phenomenon is perceptible only with discs when viewed from grazing angle. With microfacet distributions (middle + right), the phenomenon is perceptible in many configuration.

10 Conclusion & Future Work

We presented efficient and accurate analytic approximations for the surface shading from polygonal light sources. Our method is flexible and fast enough for high quality demanding real-time applications. In particular, we showed that the edge integrals of Arvo can be accurately approximated and evaluated in constant-time and that the integration of multiple axis-oriented lobes can be easily approximated using derivation from $SG$ operators. We also demonstrated that the Phong restriction can be overcome by approximating the half vector warp distortions using a single edge-split strategy. However, in some extreme configurations, the edge-splitting strategy may exhibits undesirable artifacts. To overcome these defects, we plan to explore alternative parameterization of the great circle in the half vector space. Our goal is to avoid the splitting and better match the distortions introduced by the spherical warp. For spherical and disc area light sources, our spinning quads produce convincing results for Phong distribution at roughly the same cost as the quad. But in some configurations, the excessive rotation of the quad introduces undesirable brightness variations we need to address. We found for instance that finding the optimal orientation can be reduced to a 1D problem by just considering the integration along the diagonal of the quad.

Other 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 mipmapped textures as done in [1] and [4]. One other approach would be to look for the varying luminaries derivations introduced by Arvo [21] and developed by Chen and Arvo [26]. 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**


