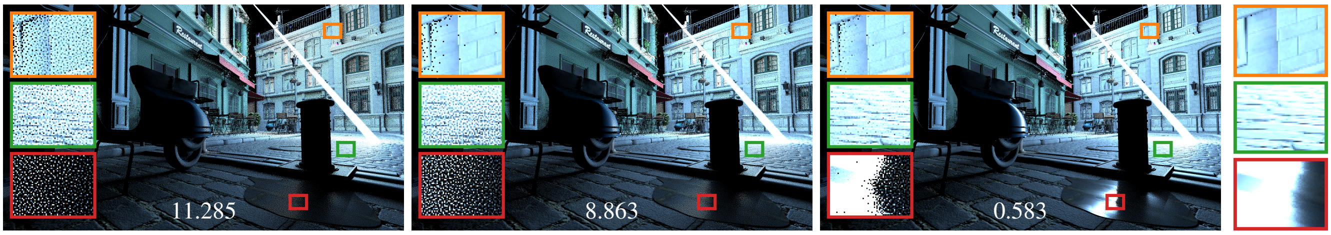
High-Performance Graphics 2021
N. Binder and T. Ritschel
(Guest Editors)

Volume 40 (2021), Number 8

BRDF Importance Sampling for Linear Lights

Christoph Peters[1]

[1]Karlsruhe Institute of Technology



Clipped solid angle sampling, 1.1 ms | Our projected solid angle sampling, 1.1 ms | Our projected solid angle and LTC sampling with MIS, 1.5 ms | Reference

Figure 1: *The bistro exterior (2.9 million triangles), lit by a long linear light source. We compute shading using Monte Carlo integration with ray traced shadows. Taking one sample per pixel proportional to solid angle yields moderate noise throughout the scene. Our projected solid angle sampling achieves clean diffuse shading outside penumbrae but specular highlights remain noisy. If we take a second sample proportional to a linearly transformed cosine [HDHN16] and combine both techniques using clamped optimal MIS [Pet21], noise outside penumbrae becomes weak everywhere. Timings are full frame times at* 1920×1080 *on an NVIDIA RTX 2080 Ti, numbers are RMSEs.*

Abstract

We introduce an efficient method to sample linear lights, i.e. infinitesimally thin cylinders, proportional to projected solid angle. Our method uses inverse function sampling with a specialized iterative procedure that converges to high accuracy in only two iterations. It also allows us to sample proportional to a linearly transformed cosine. By combining both sampling techniques through suitable multiple importance sampling heuristics and by using good stratification, we achieve unbiased diffuse and specular real-time shading with low variance outside penumbrae at two samples per pixel. Additionally, we provide a fast method for solid angle sampling.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture

1. Introduction

Through new graphics hardware, real-time ray tracing has become practical. Nonetheless, real-time renderers have to limit their ray budget to a few rays per pixel. With generic path tracers, the variance at such low sample counts is too high. GPU-friendly importance sampling techniques for specific light transport phenomena are in high demand. We provide such a method for direct lighting with linear lights, which are an excellent idealization of fluorescent tubes.

According to the reflection equation, the reflected radiance in direction $\omega_o \in \Omega$ is

$$L_o(\omega_o) = \int_\Omega L_i(\omega_i) V(\omega_i) f(\omega_i, \omega_o) \langle n, \omega_i \rangle \, \mathrm{d}\omega_i$$

where $\Omega \subset \mathbb{R}^3$ is the hemisphere around the surface normal $n \in \mathbb{R}^3$, f is the bidirectional reflectance distribution function (BRDF), L_i gives incoming radiance due to the light source, $V(\omega_i) \in \{0,1\}$ is light visibility and $\langle n, \omega_i \rangle$ denotes a dot product for the cosine term. A Monte Carlo estimator takes random samples ω_i from Ω proportional to a known density $p(\omega_i)$ and estimates the integral as

$$\frac{L_i(\omega_i) V(\omega_i) f(\omega_i, \omega_o) \langle n, \omega_i \rangle}{p(\omega_i)}.$$

To cancel most of the variance, $p(\omega_i)$ should be nearly proportional to the integrand.

For direct lighting with Lambertian emitters, $L_i(\omega_i)$ is constant within the solid angle of the light source and zero elsewhere. Then techniques that sample this solid angle uniformly

[Wan92, Arv95, UnFK13, Gam16, GUnK*17] give a Monte Carlo estimate that is proportional to $V(\omega_i)f(\omega_i,\omega_o)\langle n,\omega_i\rangle$. No rays are wasted on directions that miss the light but the BRDF and cosine-term may introduce strong variance. Alternatively, we can sample the whole hemisphere proportional to BRDF times cosine [Hd14]. Then the Monte Carlo estimate is nearly proportional to $L_i(\omega_i)V(\omega_i)$. If the light source is small, most samples miss it. In fact, our linear lights are never hit.

An ideal sampling technique would distribute samples proportional to BRDF times cosine but only within the solid angle of the light source. With such a technique, the only remaining source of noise is the visibility term $V(\omega_i)$. Sampling proportional to the visibility term requires global scene knowledge but neglecting it only introduces noise in penumbrae.

To provide such a technique for diffuse BRDFs, we sample linear lights proportional to the cosine term $\langle n,\omega_i\rangle$. In other words, we sample their projected solid angle uniformly. Our method accomplishes this goal by inverse function sampling with a highly optimized numerical inversion (Sec. 3). Our error analysis shows that it always converges to high accuracy in only two iterations. The implementation is thoroughly optimized and some of these ideas also apply to solid angle sampling of linear lights.

Once we can sample linear lights proportional to a cosine, we can also sample them proportional to a linearly transformed cosine (LTC) [HDHN16]. That gives rise to a suitable sampling strategy for specular BRDFs (Sec. 4). We combine this strategy with projected solid angle sampling through clamped optimal MIS [Pet21].

Our sampling techniques enable unbiased shading for linear lights, with ray traced shadows and minimal noise outside of penumbrae (Fig. 1). Blue noise dithering and uniform jittered sampling [RAMN12] are highly effective for the one-dimensional linear lights (Sec. 5.3). Hence, even the shadows have good quality at two samples per pixel (one specular, one diffuse). Two prior works offer similar functionality but they are either prone to branch divergence [LADL18] or slower by a constant factor [Pet21] (Sec. 5.4).

The full source code of our renderer is available.

2. Related Work

There is extensive prior work on linear lights, partly because fluorescent tubes are widely used and partly because integration in one dimension is easier than in two. Early work on diffuse shading [NON85] solves special cases in closed form and relies on quadrature for the general case. Approximate [PA91] and exact [BP93] closed-form solutions for Phong shading followed. Picott [Pic92] presents a closed form for diffuse shading using a slightly different formulation, where the linear light is not an infinitesimally thin cylinder but a sequence of point lights. For specular shading, he proposes a most representative point approach, an approximation that persists in real-time rendering until today [Dro14, dCI17].

These older works advocate variants of shadow volumes [NON85, PA91, BP93, Pic92] whereas our method offers sampling for Monte Carlo integration. Ramamoorthi et. al. [RAMN12] use ray tracing and study the impact of different strategies for stratification on shadows (see Sec. 5.3). As a form of importance sampling, Gamito [Gam16] samples the solid angle of cylinders and disks of finite radius uniformly. The method samples a bounding rectangle using an exact method [UnFK13] and rejects samples outside of the relevant solid angle at the caps. Our method does not reject any samples because it takes the cylinder radius to zero in the limit.

Solid angle sampling is available for all common types of area lights. For spheres [Wan92] and triangles [Arv95, Pet21], there are closed-form solutions. For ellipses and ellipsoids [Hei17] a method involving Newton-Raphson iterations and look-up tables exists [GUnK*17]. Still more effective importance sampling for diffuse shading takes samples proportional to the cosine term $\langle n,\omega_i\rangle$ (like our technique). For spheres that can be done through iterative root finding [UnG18] or in closed form [PD19]. For polygons, there are methods based on recursive subdivision [Un00], Newton's method [Arv01] or special iterative algorithms [Pet21]. All of these could be combined with rejection sampling [Gam16] to sample cylinders of finite size in proportion to projected solid angle.

However, we strive for a faster, more specialized solution. Projected solid angle sampling and LTC importance sampling of linear lights are also useful for differentiable rendering because moving edges make strong contributions to derivatives [LADL18]. The implementation used there employs costly Newton bisection for inverse function sampling. Our method uses a more specialized procedure that always converges in two iterations and is more thoroughly optimized.

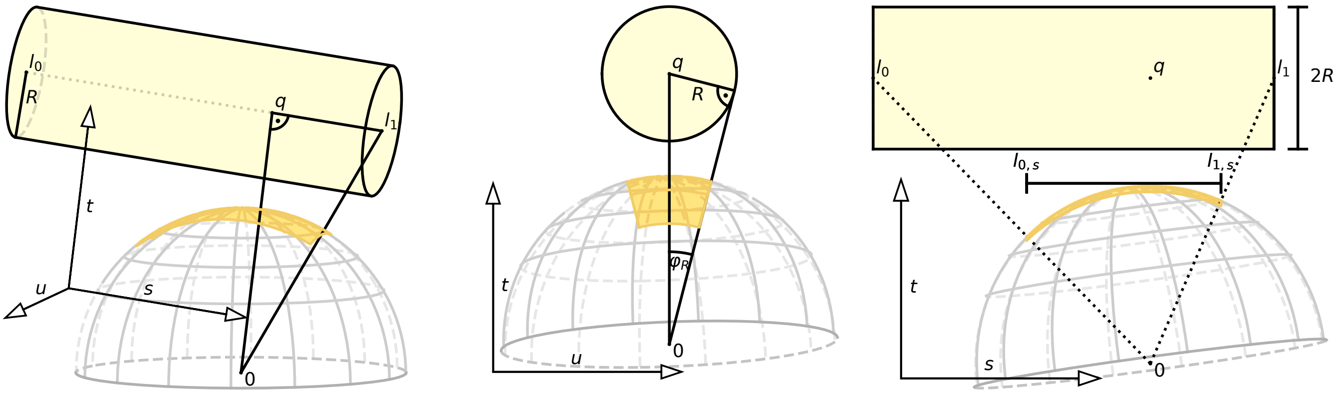
Recently, sampling problems have been studied more fundamentally. The triangle cut parametrization warps samples of a suitable approximate density into samples of another density [Hei20]. In one dimension, it still consumes two random numbers such that stratification is lost. Hart et al. [HPM*20] approximate a target density in primary sample space with linear or quadratic polynomials and sample proportional to those.

LTCs [HDHN16] provide good approximations of many specular BRDFs. With this approximation, computation of unshadowed specular shading reduces to computation of the projected solid angle of the transformed light source. We use them for sampling. They also work for linear lights [HH17a] and disk lights [HH17b]. Dupuy et al. [DHB17] present a similar method for specular importance sampling of spherical lights. It is efficient but struggles with anisotropic highlight shapes.

Moureau et al. [MPC19] describe a GPU-friendly hierarchical data structure to select important lights among thousands of dynamic lights. Most renderers combine strategies for sampling of light sources with methods to sample in proportion to the BRDF [Hd14] through multiple importance sampling (MIS) [VG95]. However, this approach is ineffective for linear lights since all samples miss the light.

3. Sampling Linear Lights for Diffuse Shading

Our method samples the solid angle of linear lights exactly proportional to the cosine term $\langle n,\omega_i\rangle$. Then the Monte Carlo estimator for Lambertian emitters is proportional to $V(\omega_i)f(\omega_i,\omega_o)$. For



(a) *The cylinder in relation to a shading point.* **(b)** *View of the ut-plane.* **(c)** *View of the st-plane.*

Figure 2: *A cylinder and its solid angle. Looking straight at a cap of the cylinder, we see that the solid angle has an opening angle φ_R. A side view reveals that its extent can be described by the projection of its end points onto the sphere in the limit case $R \to 0$.*

diffuse BRDFs that gives low variance and we address specular BRDFs in Sec. 4. In the following, we derive formulas for the solid angle (Sec. 3.1) and projected solid angle of linear lights (Sec. 3.2). To perform inverse function sampling, we introduce an iterative inversion method (Sec. 3.3) with low error (Sec. 3.4). We optimize these methods thoroughly and also apply them for solid angle sampling (Sec. 3.5).

3.1. The Solid Angle of a Line

As starting point, we need a proper definition of a linear light and we need to compute its solid angle. These considerations are not novel [NON85, BP93] but serve to fix the notation and to establish building blocks for an efficient algorithm. We follow the notion (unlike Picott [Pic92]) that a linear light is an infinitesimally thin cylinder. This model is a good fit for fluorescent tubes. Lambertian emitters are the easiest to implement but arbitrary emission profiles are applicable, possibly at the cost of increased variance [Pet21]. We could treat linear lights as limit case of rectangular lights [Pet21] but the self-contained approach presented here leads to an efficient formulation more easily.

Let $l_0, l_1 \in \mathbb{R}^3$ with $l_0 \neq l_1$ be the end points of the cylinder, i.e. the center points of its caps (Figure 2a). We work in a coordinate frame where the shading point is the origin. Let $R > 0$ be the cylinder radius, which we take to zero in the limit. The normalized line direction is given by

$$s := \frac{l_1 - l_0}{\|l_1 - l_0\|} \in \mathbb{S}^2,$$

where $\mathbb{S}^2 \subset \mathbb{R}^3$ denotes the unit sphere. We are interested in the solid angle of this cylinder. To this end, we consider the point q closest to the origin on the infinite central axis of the cylinder. We are also interested in the direction t towards this point:

$$q := l_0 - \langle l_0, s\rangle\, s \in \mathbb{R}^3, \qquad t := \frac{q}{\|q\|} \in \mathbb{S}^2.$$

Together, the directions s, t and $u := s \times t$ form an orthonormal coordinate frame. Viewing the geometric configuration in this frame simplifies the derivation of the solid angle. In the ut-plane, the cylinder appears as circle of radius R (Figure 2b). Its opening angle with respect to the shading point is

$$\varphi_R := \arcsin \frac{R}{\|q\|}.$$

In the st-plane, the cylinder appears as axis-aligned rectangle of height $2R$ (Figure 2c). Since we eventually take R to zero, there is no need to bother with the exact shape of the caps. We capture the extent by storing s-coordinates for the normalized end points of the line:

$$l_{0,s} := \frac{\langle l_0, s\rangle}{\|l_0\|} \in \mathbb{R}, \qquad l_{1,s} := \frac{\langle l_1, s\rangle}{\|l_1\|} \in \mathbb{R}.$$

Now we are prepared to compute the solid angle of the cylinder (with incorrect caps). We write it as integral over the hemisphere in cylindrical coordinates (see [PJH16] chapter 13.6.1):

$$\Omega_R := \int_{l_{0,s}}^{l_{1,s}} \int_{-\varphi_R}^{\varphi_R} 1 \,\mathrm{d}\varphi\,\mathrm{d}\omega_s = 2\varphi_R(l_{1,s} - l_{0,s}). \tag{1}$$

The angle φ is an azimuthal coordinate around the central axis of the cylinder and $\omega_s \in [-1, 1]$ is the s-coordinate of a unit-direction vector. For $R = 0$, this solid angle is zero because $\varphi_R = 0$. Thus, we divide out $R > 0$ before we take the limit:

$$\frac{\mathrm{d}\Omega_R}{\mathrm{d}R} := \lim_{R\to 0} \frac{\Omega_R}{R} = 2\frac{\mathrm{d}\varphi_R}{\mathrm{d}R}(l_{1,s} - l_{0,s}), \qquad \frac{\mathrm{d}\varphi_R}{\mathrm{d}R} := \frac{1}{\|q\|}.$$

According to L'Hôpital's rule, the same result is attained by taking the derivative at $R = 0$. Hence, the choice of notation. We discuss the construction of Monte Carlo estimates in this setting below.

This derivation directly implies a strategy to sample linear lights proportional to solid angle, which is a limit case of prior work [Gam16]. Since the integrand in Equation (1) is constant, we simply sample ω_s uniformly in $[l_{0,s}, l_{1,s}]$, set $\omega_t := \sqrt{1 - \omega_s^2}$ and return $\omega_s s + \omega_t t$ as sampled direction. Sec. 3.5 describes the implementation in more detail.

The limit of the sampled density times radius is the reciprocal of the solid angle per radius $\frac{\mathrm{d}\Omega_R}{\mathrm{d}R}$. We specify the brightness of the linear light through the limit of radiance times radius. Then the integrand of the rendering equation and the term for the density both contain the infinitesimal radius as factor. It cancels out and we are left with a finite radiance. With this convention, the brightness is a physically meaningful quantity with unit $\frac{\mathrm{W}}{\mathrm{sr}\cdot\mathrm{m}}$ (or the photometric counterpart nit $\cdot$ m).

3.2. Sampling the Projected Solid Angle of a Line

To obtain the projected solid angle from the above formulation of the solid angle, we introduce a cosine term for the surface normal $n \in \mathbb{S}^2$. The normal has local coordinates $n_s := \langle n, s\rangle$, $n_t := \langle n, t\rangle$. Recall that the local coordinate $\omega_s \in [l_{0,s}, l_{1,s}]$ corresponds to the normalized direction $\omega_s s + \sqrt{1-\omega_s^2}t$. Thus, the projected solid angle per radius for $R \to 0$ is

$$\begin{aligned}\frac{\mathrm{d}\Omega_r^\perp}{\mathrm{d}R} &:= 2\frac{\mathrm{d}\varphi_R}{\mathrm{d}R}\int_{l_{0,s}}^{l_{1,s}} \left\langle n, \omega_s s + \sqrt{1-\omega_s^2}t \right\rangle \mathrm{d}\omega_s \\ &= 2\frac{\mathrm{d}\varphi_R}{\mathrm{d}R}\int_{l_{0,s}}^{l_{1,s}} n_s\omega_s + n_t\sqrt{1-\omega_s^2}\,\mathrm{d}\omega_s \\ &= \frac{\mathrm{d}\varphi_R}{\mathrm{d}R}(F_{n_s,n_t}(l_{1,s}) - F_{n_s,n_t}(l_{0,s})), \end{aligned} \tag{2}$$

where we use the indefinite integral

$$F_{n_s,n_t}(\omega_s) := n_s\omega_s^2 + n_t\left(\sqrt{1-\omega_s^2}\,\omega_s + \arcsin\omega_s\right). \tag{3}$$

The dot product in this integral must not be negative. Therefore, we clip the line connecting l_0 to l_1 against the tangent plane of the surface.

We intend to use inverse function sampling to sample this projected solid angle uniformly. Our sampling procedure consumes a single uniform random number ξ in $[0,1)$. The sample coordinate ω_s has to be chosen so that the value of the distribution function matches the random number, i.e.

$$\frac{\mathrm{d}\varphi_R}{\mathrm{d}R}(F_{n_s,n_t}(\omega_s) - F_{n_s,n_t}(l_{0,s})) = \xi\frac{\mathrm{d}\Omega_r^\perp}{\mathrm{d}R}. \tag{4}$$

Assuming that we can evaluate the inverse distribution function F_{n_s,n_t}^{-1}, the solution is

$$\omega_s = F_{n_s,n_t}^{-1}\left(\xi\frac{\mathrm{d}R}{\mathrm{d}\varphi_R}\frac{\mathrm{d}\Omega_r^\perp}{\mathrm{d}R} + F_{n_s,n_t}(l_{0,s})\right).$$

The whole sampling procedure, including optimizations described in Section 3.5, is summarized in Algorithm 1.

3.3. Inversion of the Distribution Function

Inversion of F_{n_s,n_t} is challenging. A closed-form solution appears to be impossible. Since scaling of (n_s, n_t) only scales the integral F_{n_s,n_t}, we are dealing with a one-dimensional family of functions to invert. Fig. 3 shows examples. It is possible to use a two-dimensional lookup table but that requires a high resolution at boundaries and memory access at random locations thrashes caches. Instead, we take inspiration from recent work [Pet21] and design a specialized iterative procedure with rapid convergence.

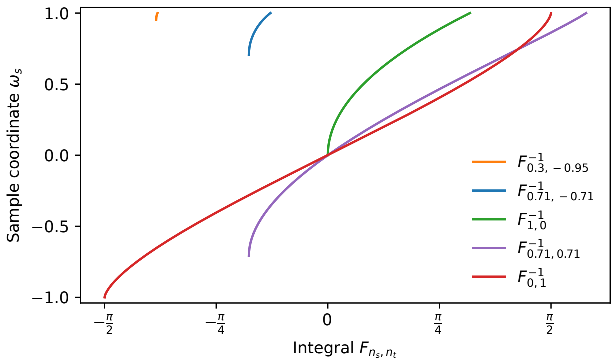


Figure 3: *Plots of the inverse distribution function F_{n_s,n_t}^{-1} for five choices of n_s, n_t. At the left domain boundary, the derivative always approaches infinity. For n_t near -1 (orange, blue), the domain shrinks and the function becomes steep. $F_{1,0}^{-1}$ (green) is simply a square root.*

For our iterative procedure, we make the substitution $\alpha := \arcsin\omega_s$ and consider

$$G_{n_s,n_t}(\alpha) := F_{n_s,n_t}(\sin\alpha) = n_s\sin^2\alpha + n_t(\cos\alpha\sin\alpha + \alpha).$$

Evaluation of this function does not involve costly inverse trigonometric functions and it is more well-behaved. In fact, quadratic Taylor expansions give good local fits of $G_{n_s,n_t}(\alpha)$. Our iterative method exploits that. In each step, it constructs a quadratic Taylor polynomial around the current estimate of α and solves Equation (4) with this approximation. The equation turns into a quadratic. Among the two solutions, we pick the one that is closer to the current estimate. If there are no roots, we take the extremum of the quadratic instead to safeguard against rare numerical issues.

This approach can be thought of as quadratic generalization of Newton's method. It is known as Halley's irrational formula or Laguerre's method [ST95]. Laguerre's method is popular for polynomial root finding but uncommon as general root finding method [PTVF07]. Sec. 3.4 demonstrates that it works well here.

For the initialization, we use solid angle sampling. As derived in Sec. 3.1, that means that we simply set ω_s to

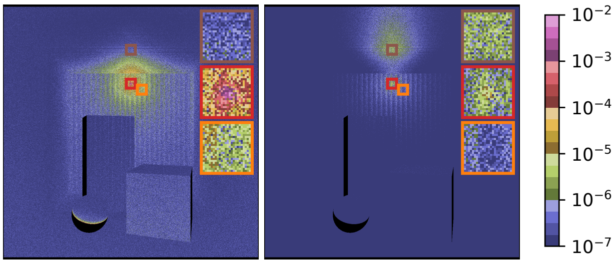
$$l_{0,s} + \xi(l_{1,s} - l_{0,s}).$$

Thus, this initialization is extremely efficient. We proceed to show that it is also accurate enough. Algorithm 2 summarizes our inversion procedure.

3.4. Error Analysis

To avoid branch divergence on GPUs, we want to use a small, fixed iteration count. However, we also want our renderer to be unbiased. Thus, we have to be sure that our method converges to sufficient accuracy in all cases. Since the beginning and the end of the line influence the initialization, the set of all test cases is four-dimensional.

Once again, we take inspiration from recent work [Pet21]. We run the Nelder-Mead optimizer [NM65] in 80-bit float arithmetic to find a line that maximizes the error of our iterative procedure. Since



(a) *Backward error* **(b)** *Backward error times brightness*

Figure 4: *Errors for diffuse samples in a variant of Fig. 6. The linear light is cut in half such that it does not get close to the back wall.*

Nelder-Mead is just a local optimizer, we try 28 billion random initializations. Sometimes the iteration is unstable near the endpoints of the linear light. However, the initialization is nearly perfect there. Thus, we skip the iteration if $\xi < 10^{-5}$ or $\xi > 1 - 10^{-5}$.

As error metric, we use a backward error, namely the perturbation in the random number ξ that suffices to explain the error in the output. This way, we find that the worst possible error after two iterations is $1.58 \cdot 10^{-5}$. On this basis, we consider our method unbiased.

In practice, rounding errors in single-precision arithmetic are far more influential than these theoretical errors (Fig. 4a). The main problem is a cancellation in Equation (2). The indefinite integral F_{n_s,n_t} corresponds to the projected solid angle of a line starting at the closest point q. When the actual projected solid angle is much smaller, we lose precision. However, that only happens in dark regions along the infinite extension of the line. If we take that into account, errors are always low (Fig. 4b). Therefore, we choose not to invest computational resources to avoid this cancellation.

3.5. Optimizations

Conceptually, it is useful to work with the coordinate frame s, t but in practice we skip computation of $t \in \mathbb{S}^2$. We only need it for two purposes: To compute n_t and to construct directions $\omega_s s + \omega_t t$. In both cases, we exploit

$$t = \frac{1}{\|q\|}(l_0 - \langle l_0, s\rangle s).$$

Then

$$\omega_s s + \omega_t t = \left(\omega_s - \frac{\omega_t}{\|q\|}\langle l_0, s\rangle\right)s + \frac{\omega_t}{\|q\|} l_0,$$
$$n_t = \frac{1}{\|q\|}(\langle n, l_0\rangle - n_s \langle l_0, s\rangle).$$

For efficiency reasons, lines are represented by their beginning $l_0 \in \mathbb{R}^3$, their direction $s \in \mathbb{S}^2$ and their length $L := \|l_1 - l_0\|$. We precompute these attributes per linear light. Since $l_1 = l_0 + Ls$, we have $\langle l_1, s\rangle = \langle l_0, s\rangle + L$.

Algorithms 1 and 2 implement our method with these optimizations. Our supplementary code additionally eliminates common subexpressions like $\langle l_0, s\rangle$, $\langle l_0, l_0\rangle$ and $F_{n_s,n_t}(l_{0,s})$. We always clamp coordinates ω_s to $[-1, 1]$ to avoid invalid results. Blinn's quadratic solver [Bli06] makes Algorithm 2 more stable. Besides, it is useful to split Algorithm 1 into a part that executes once per line and another part that runs once per sample.

Minor changes turn Algorithm 1 into a heavily optimized implementation of solid angle sampling. We simply omit all lines that deal with n and always take the else-branch for computation of ω_t.

Algorithm 1 sample_line_projected_solid_angle

Input: Line begin $l_0 \in \mathbb{R}^3$, line direction $s \in \mathbb{S}^2$, line length $L > 0$, shading normal $n \in \mathbb{S}^2$, uniform random number $\xi \in [0, 1)$.

Output: Sampled direction ω_i, density in solid angle measure times radius $p(\omega_i)\mathrm{d}R$.

Clip the line against the plane through the origin with normal n

If the clipped line is empty: Return no sample.

$\|q\|^2 := \langle l_0, l_0\rangle - \langle s, l_0\rangle^2$

$\frac{1}{\|q\|} := \frac{1}{\sqrt{\|q\|^2}}$

$l_{0,s} := \frac{\langle l_0, s\rangle}{\sqrt{\langle l_0, l_0\rangle}}$

$l_{1,s} := \frac{\langle l_0, s\rangle + L}{\sqrt{(\langle l_0, s\rangle + L)^2 + \|q\|^2}}$

$n_s := \langle n, s\rangle$

$n_t := \frac{1}{\|q\|}(\langle n, l_0\rangle - n_s \langle l_0, s\rangle)$

$E := F_{n_s,n_t}(l_{1,s}) - F_{n_s,n_t}(l_{0,s})$ // see Equation (3)

$\frac{\mathrm{d}\Omega_r^\perp}{\mathrm{d}R} := \frac{1}{\|q\|}E$

$\omega_s := l_{0,s} + \xi(l_{1,s} - l_{0,s})$

If $\xi \geq 10^{-5}$ and $\xi \leq 1 - 10^{-5}$:

  $\omega_s, \omega_t :=$ invert_line_sampling_cdf $\left(n_s, n_t, \omega_s, \xi E + F_{n_s,n_t}(l_{0,s})\right)$

else: $\omega_t := \sqrt{1 - \omega_s^2}$ // solid angle sampling

$\omega_i := \left(\omega_s - \frac{1}{\|q\|}\omega_t \langle l_0, s\rangle\right)s + \frac{1}{\|q\|}\omega_t l_0$

Return ω_i, $(\omega_s n_s + \omega_t n_t)\left(\frac{\mathrm{d}\Omega_r^\perp}{\mathrm{d}R}\right)^{-1}$

4. Sampling Linear Lights for Specular Shading

By itself, our projected solid angle sampling gives low variance for diffuse BRDFs but not for specular BRDFs, especially at low roughness (Fig. 1). The same is true for sampling of polygonal lights and we overcome this limitation in the same way [Pet21]. This section briefly describes the necessary steps. For a detailed discussion, we refer to prior work [Pet21].

Algorithm 2 invert_line_sampling_cdf
Input: $n_s, n_t \in \mathbb{R}$, an initialization $\omega_s \in [-1,1]$, $F \in \mathbb{R}$
Output: $\omega_s = F^{-1}_{n_s,n_t}(F) \in [-1,1]$, $\omega_t = \sqrt{1-\omega_s^2}$

$\alpha := \arcsin \omega_s$

$\omega_t := \sqrt{1-\omega_s^2}$

Repeat twice:

$G_\delta := (n_s\omega_s + n_t\omega_t)\omega_s + n_t\alpha - F \qquad // = G_{n_s,n_t}(\alpha) - F$

$G' := 2(n_s\omega_s + n_t\omega_t)\omega_t$

$G'' := 2(n_s\omega_t - n_t\omega_s)\omega_t - 2(n_s\omega_s + n_t\omega_t)\omega_s$

Solve $\frac{G''}{2}\beta^2 + G'\beta + G_\delta = 0$

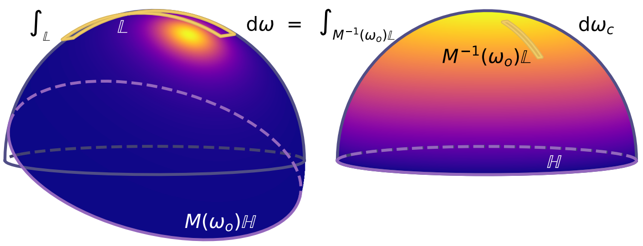
Let β be the root of smaller magnitude (extremum if none exists)

$\alpha := \min\left(\max\left(\alpha+\beta, -\frac{\pi}{2}\right), \frac{\pi}{2}\right)$

$\omega_s := \sin\alpha$

$\omega_t := \cos\alpha$

Return ω_s, ω_t



(a) *LTC integral in world space* **(b)** *Clamped cosine integral*

Figure 5: *Integration over an LTC is equivalent to integrating a clamped cosine over the solid angle of the transformed light source. Note that the horizon $\mathbb{H}$ changes through the transformation. The LTC is zero in parts of the upper hemisphere and non-zero in parts of the lower hemisphere.*

4.1. Sampling Linearly Transformed Cosines

LTCs [HDHN16] are probability density functions of the form

$$p_M(\omega) := \frac{1}{\pi}\max\left(0, \left\langle \frac{M^{-1}\omega}{\|M^{-1}\omega\|}, (0,0,1)^\mathsf{T} \right\rangle\right) \frac{|M^{-1}|}{\|M^{-1}\omega\|^3},$$

where $M \in \mathbb{R}^{3\times 3}$ is chosen so that p_M fits a specular BRDF times cosine for a particular outgoing light direction. By construction, integrating an LTC over a solid angle is the same as integrating a cosine distribution over the linearly transformed solid angle (Fig. 5).

For sampling, we transform the clipped end points of the linear light to cosine space through

$$l_{0,c} := M^{-1} l_0, \qquad l_{1,c} := M^{-1} l_1.$$

Since this transformation changes the horizon (Fig. 5a), we clip the line between $l_{0,c}$ and $l_{1,c}$ a second time. Then we apply our projected solid angle sampling procedure. Resulting samples $\omega_c \in \mathbb{S}^2$ get transformed back to world space through $\frac{M\omega_c}{\|M\omega_c\|}$.

When using LTCs for linear lights, there is a potential pitfall: Linear transformations change the opening angle of the cylinder. To account for this effect, we use the correction factor [HH17a]

$$\frac{\mathrm{d}R}{\mathrm{d}R_c} := \frac{\|M^\mathsf{T}(s\times l_0)\|}{\|s\times l_0\|}. \tag{5}$$

We multiply it onto densities and divide it out of projected solid angles and LTC shading estimates.

4.2. Combining Diffuse and Specular Samples

The LTC density p_M is zero in parts of the upper hemisphere (Fig. 5). Thus, we have to combine the corresponding samples with samples from projected solid angle sampling to get an unbiased estimate. Standard MIS heuristics introduce considerable variance outside of penumbrae. For example, the balance heuristic effectively samples the sum of both densities but this density differs from the BRDF times cosine. Therefore, we use clamped optimal MIS [Pet21], which is designed specifically for this situation.

To use it, we have to compute estimates of unshadowed diffuse and specular shading c_0, c_1 as in the original work on LTCs [HDHN16, HH17a]. Both of these are computed per color channel using the readily available projected solid angle of the linear light. All entries of c_0 must be non-zero, so we clamp diffuse albedos to a minimum of 0.01. Then clamped optimal MIS weights are [Pet21]

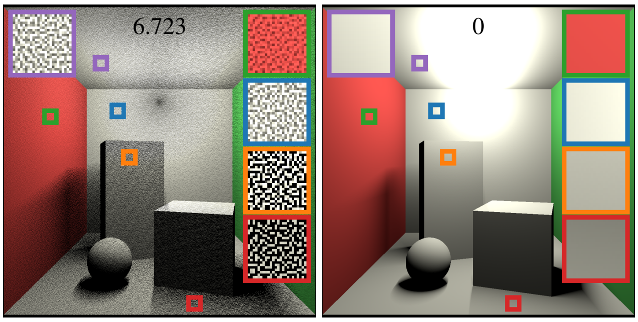
$$w_j(\omega_j) := v\frac{c_j p_j(\omega_j)\mathrm{d}R}{\sum_{k=0}^{1} c_k p_k(\omega_j)\mathrm{d}R} + (1-v)\frac{p_j(\omega_j)\mathrm{d}R}{\sum_{k=0}^{1} p_k(\omega_j)\mathrm{d}R}.$$

where $j \in \{0,1\}$ is the index of the technique that generated the sample $\omega_j \in \mathbb{S}^2$ and $p_k(\omega_j)\mathrm{d}R$ is the density times radius for technique k. The parameter $v \in [0,1]$ blends between the standard balance heuristic and a weighted balance heuristic, which is optimal under idealizing assumptions such as no occlusion [Pet21]. Setting $v = 0.5$ works well in practice.

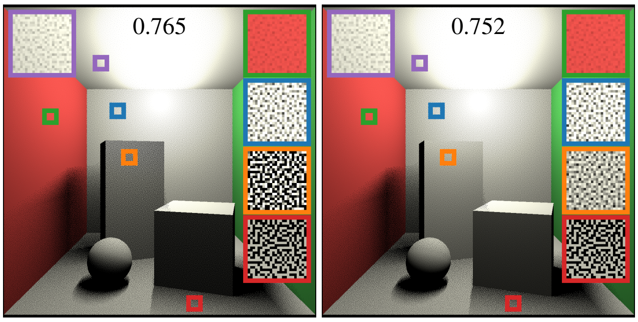
5. Results

In the following, we evaluate the quality of our importance sampling for diffuse (Sec. 5.1) and specular shading (Sec. 5.2) in comparison to prior work. We also make recommendations on stratification (Sec. 5.3) and measure timings (Sec. 5.4).

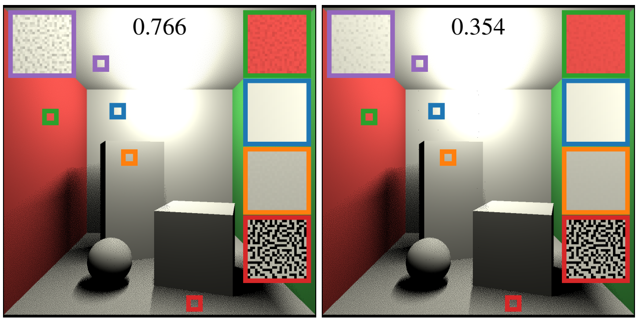
Our Vulkan renderer uses the extension VK_KHR_ray_query to cast shadow rays. It is a deferred renderer with a 32-bit visibility buffer [BH13]. Unless stated otherwise, our experiments use the isotropic Frostbite BRDF [LdR14]. For LTCs, we use a $64\times 64\times 51$ table of transforms M parameterized by roughness, outgoing inclination and Fresnel reflectance at 0°. Support for arbitrary anisotropic BRDFs would require a 5D table, which is hardly viable. We inherit this limitation from LTCs [HDHN16]. Linear lights are displayed with finite extent to convey geometric relations better. Alongside our results, we report root-mean-square errors (RMSEs) computed from HDR frames.



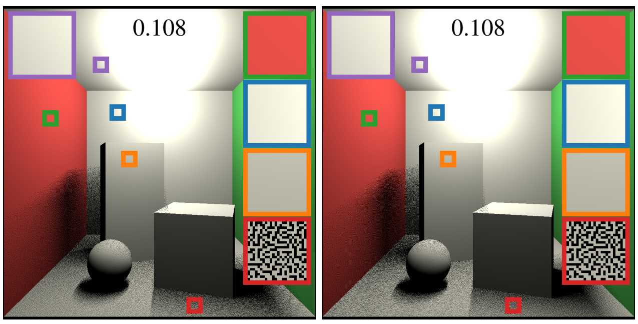
(a) *Area* **(b)** *Reference*



(c) *Solid angle* **(d)** *Clipped solid angle*



(e) *Linear cosine warp [HPM*20]* **(f)** *Quadratic cosine warp [HPM*20]*



(g) *Projected solid angle [LADL18]* **(h)** *Projected solid angle, ours*

Figure 6: *A Lambertian diffuse Cornell box lit by a single linear light just below the ceiling. All techniques use one sample per pixel. Dependent on their geometric relation to the light, different surfaces benefit differently from better sampling. Our projected solid angle sampling has zero variance outside penumbrae.*

5.1. Diffuse Shading

Figure 6 compares different approaches for diffuse shading using a Lambertian diffuse BRDF. Area sampling (Fig. 6a) places samples uniformly along the length of the linear light. The square falloff term and the two cosine terms introduce strong variance, especially on the ceiling and near the light. These regions look darker because sRGB values get clamped at one. Solid angle sampling (Fig. 6c) is far better but the remaining cosine term still causes variance, especially on the white wall, where it ranges down to zero. On the box (orange inset) the linear light is partially below the horizon. Clipping (Fig. 6d) eliminates samples without contribution.

We apply warping of random numbers [HPM*20] on top of clipped solid angle sampling to incorporate the cosine term into the density. With a linear density, this approach is effective on the back wall but barely improves results on the ceiling or the red wall (Fig. 6e). Results deteriorate in the overexposed parts of the ceiling, hence the bad RMSE. A quadratic density helps everywhere, at an increased overhead (Fig. 6f).

As expected, our projected solid angle sampling achieves zero variance outside of penumbrae (Fig. 6h). Noise in penumbrae is not reduced significantly but stratification through blue noise works well (red inset). The method of Li et al. [LADL18] gives identical results at a higher cost.

5.2. Specular Shading

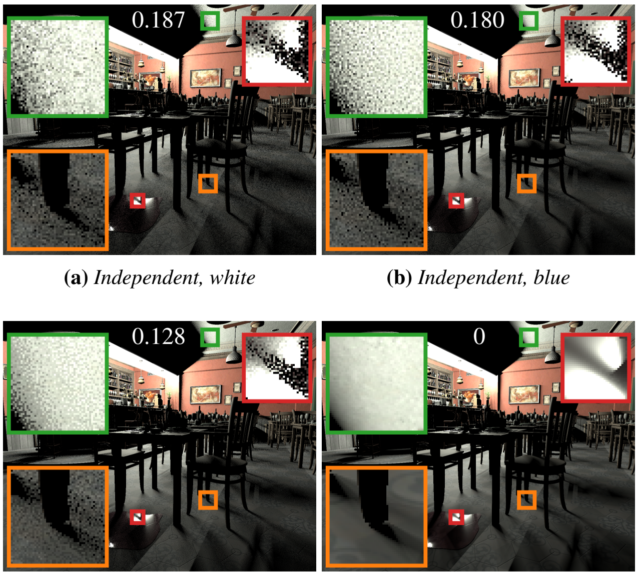
Fig. 1 demonstrates the benefits of our specular importance sampling. The puddle in the foreground (red inset) has low roughness such that solid angle sampling and projected solid angle sampling rarely sample the peak of the specular BRDF. Thus, shading is far from convergence at one sample per pixel. Using an additional specular sample distributed proportional to an LTC through clamped optimal MIS with $v = 0.5$ improves the result drastically. Remaining noise is mostly due to the penumbra of the bollard. Note that the shadow of the bollard looks more like a glossy reflection due to the narrow peak of the BRDF.

5.3. Stratification

We find two established methods to be particularly effective for linear lights. Ramamoorthi et al. [RAMN12] recommend uniform jittered sampling for linear lights. That means that we only consume a single random number ξ on $[0,1)$ per technique per light. If we take $N \in \mathbb{N}$ samples, the random number fed to sampling algorithms for sample $k \in \{0,\ldots,N-1\}$ is $\frac{\xi+k}{N}$. Indeed, this approach gives an appreciable reduction of variance in penumbrae (Fig. 7). Additionally, we use precomputed 64×64 blue noise textures [Uli93]. The blue noise patterns are preserved relatively well in penumbrae.

5.4. Run Time

To measure timings, we use the same setup as previous work [Pet21] such that numbers are comparable. Our test system consists of an NVIDIA Geforce RTX 2080 Ti, an Intel Core i5-9600K and 16 GB RAM. When ray tracing is enabled, our clipped solid angle sampling and our projected solid angle sampling perform identically. The computation is hidden by the latency, even



(a) *Independent, white* **(b)** *Independent, blue*

(c) *Uniform jittered, blue* **(d)** *Reference*

Figure 7: *The bistro interior lit by a long linear light above the counter. We compare white noise against blue noise and independent sampling against uniform jittered sampling. All results use clamped optimal MIS with $v = 0.5$ and six samples per pixel (three diffuse, three specular). Uniform jittered sampling reduces variance, blue noise pushes it into higher frequencies in screen space.*

with the geometrically simple Cornell box (Fig. 6). Adding one sample per pixel at 1920×1080 resolution for the bistro exterior (Fig. 1) takes 0.45 ms.

Therefore, we focus on computational cost and disable ray tracing. We point the camera at a plane and either take 128 samples from 128 different lights or from a single light. Our renderer has an overhead per sample, e.g. to evaluate the BRDF. To measure this overhead separately, we define a cheap baseline sampling technique, namely area sampling without proper density computation.

Table 1 lists the results. Our optimized solid angle sampling is extremely fast, especially regarding the cost per sample. Note however, that the marginal cost of the baseline sampling technique comes on top of that. Even area sampling has a higher cost per sample due to the more complex density. Clipping doubles the cost per light. Warping [HPM*20] benefits from our fast implementation of solid angle sampling. The quadratic variant with closed-form cubic solver performs similarly to our projected solid angle sampling. The linear variant is faster but both of these have inferior quality.

Compared to the method with Newton bisection [LADL18], our projected solid angle sampling costs 2.5 times less per sample and 2 times less per light. However, Newton bisection has a variable iteration count and potentially divergent execution. Therefore, we repeat this experiment on the geometrically more complex scene in Fig. 1 and find that our cost per sample is 4.5 times lower. This gap could grow further in a full path tracer.

Through rejection sampling, rectangle sampling techniques can

Table 1: *Delta timings in milliseconds for rendering a frame at* 1920×1080 *resolution using 128 samples per pixel. The samples are either taken from 128 different linear lights or all from the same light. The baseline timings have been subtracted from each of the timings in the rows above to isolate the cost of sampling itself.*

	128 lights	128 samples
Area	0.47	0.41
Solid angle, ours, Sec. 3.1	0.75	0.17
Clipped solid angle, ours, Sec. 3.1	1.29	0.17
Linear cosine warp [HPM*20]	2.32	0.49
Quadratic cosine warp [HPM*20]	4.78	1.97
Projected solid angle [LADL18]	10.0	5.20
Projected solid angle, ours, Sec. 3	4.40	2.05
+ Baseline	3.25	2.99
Projected solid angle, Fig. 1 [LADL18]	13.6	10.6
Projected solid angle, Fig. 1, ours	4.13	2.34
+ Baseline	3.36	2.55
Rectangle solid angle [UnFK13]	3.65	1.11
Quad projected solid angle [Pet21]	28.9	11.5
+ Baseline	6.46	3.61

sample cylinders [Gam16]. However, our specialized methods for the limit case are considerably faster for solid angle sampling [UnFK13] and projected solid angle sampling [Pet21], respectively.

6. Conclusions

Our work takes importance sampling of linear lights to its natural conclusion. Except for visibility, all terms of the reflection equation are accounted for and the method is stable and inexpensive. It is a valuable addition to any path tracer and also offers efficiency improvements for differentiable rendering [LADL18]. Methodically, our work reinforces the value of tailor-made iterative algorithms for sampling problems in computer graphics. Since we guarantee accurate results with two inexpensive iterations, there is no practical reason to prefer closed-form solutions.

Now that fast projected solid angle sampling and LTC importance sampling are available for polygonal [Pet21] and linear lights, the most important remaining light type are ellipsoids. Spheres have been addressed [PD19] but that is not sufficient for LTC importance sampling. We hope that similar iterative methods will be applicable. Our optimizations may also apply to rectangle solid angle sampling [UnFK13]. Besides, our work further motivates the generalization of LTCs to arbitrary anisotropic BRDFs.

Acknowledgments

We thank Alisa Jung, Johannes Schudeiske, Tobias Zirr, Carsten Dachsbacher and our reviewers for their constructive input. Figs. 1 and 7 use ORCA models courtesy of Amazon Lumberyard. Our renderer ships with additional scenes. The attic is created by Shuprobho Das, Fran Calvente, Asterlil, YopLand, M. Ziemys, Eva11dragon, Rob Tuytel and me. The arcade is made by Nicholas Hull for NVIDIA's Falcor. The living room is made by Jay Artist.

References

[Arv95] ARVO J.: Stratified sampling of spherical triangles. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques* (1995), SIGGRAPH '95, ACM, pp. 437–438. doi:10.1145/218380.218500. 2

[Arv01] ARVO J.: Stratified sampling of 2-manifolds. In *State of the Art in Monte Carlo Ray Tracing for Realistic Image Synthesis (SIGGRAPH 2001 Course Notes)* (Aug. 2001), ACM. 2

[BH13] BURNS C. A., HUNT W. A.: The visibility buffer: A cache-friendly approach to deferred shading. *Journal of Computer Graphics Techniques (JCGT) 2*, 2 (Aug. 2013), 55–69. URL: http://jcgt.org/published/0002/02/04/. 6

[Bli06] BLINN J. F.: How to solve a quadratic equation. part 2. *IEEE Computer Graphics and Applications 26*, 2 (2006), 82–87. doi:10.1109/MCG.2006.35. 5

[BP93] BAO H., PENG Q.: Shading models for linear and area light sources. *Computers & Graphics 17*, 2 (1993), 137–145. doi:10.1016/0097-8493(93)90097-S. 2, 3

[dCI17] DE CARPENTIER G., ISHIYAMA K.: Advances in real-time rendering, part II: Decima engine: Advances in lighting and AA. In *ACM SIGGRAPH 2017 Courses* (2017). doi:10.1145/3084873.3096477. 2

[DHB17] DUPUY J., HEITZ E., BELCOUR L.: A spherical cap preserving parameterization for spherical distributions. *ACM Trans. Graph. (proc. SIGGRAPH) 36*, 4 (July 2017). doi:10.1145/3072959.3073694. 2

[Dro14] DROBOT M.: Physically based area lights. In *GPU Pro 5: Advanced Rendering Techniques* (May 2014), A K Peters/CRC Press, pp. 67–100. 2

[Gam16] GAMITO M. N.: Solid angle sampling of disk and cylinder lights. *Computer Graphics Forum (proc. EGSR) 35*, 4 (June 2016). doi:10.1111/cgf.12946. 2, 3, 8

[GUnK*17] GUILLÉN I., UREÑA C., KING A., FAJARDO M., GEORGIEV I., LÓPEZ-MORENO J., JARABO A.: Area-preserving parameterizations for spherical ellipses. *Computer Graphics Forum (proc. EGSR) 36*, 4 (June 2017). doi:10.1111/cgf.13234. 2

[Hd14] HEITZ E., D'EON E.: Importance sampling microfacet-based BSDFs using the distribution of visible normals. *Computer Graphics Forum (proc. EGSR) 33*, 4 (2014), 103–112. doi:10.1111/cgf.12417. 2

[HDHN16] HEITZ E., DUPUY J., HILL S., NEUBELT D.: Real-time polygonal-light shading with linearly transformed cosines. *ACM Trans. Graph. (proc. SIGGRAPH) 35*, 4 (July 2016). doi:10.1145/2897824.2925895. 1, 2, 6

[Hei17] HEITZ E.: Analytical calculation of the solid angle subtended by an arbitrarily positioned ellipsoid to a point source. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 852* (2017), 10–14. doi:https://doi.org/10.1016/j.nima.2017.02.004. 2

[Hei20] HEITZ E.: Can't invert the CDF? The triangle-cut parameterization of the region under the curve. *Computer Graphics Forum 39*, 4 (2020). doi:10.1111/cgf.14058. 2

[HH17a] HEITZ E., HILL S.: Linear-light shading with linearly transformed cosines. In *GPU Zen: Advanced Rendering Techniques* (2017), Black Cat Publishing Inc., pp. 137–162. URL: https://hal.archives-ouvertes.fr/hal-02155101. 2, 6

[HH17b] HEITZ E., HILL S.: Physically based shading in theory and practice: Real-time line and disk light shading. In *ACM SIGGRAPH 2017 Courses* (2017). doi:10.1145/3084873.3084893. 2

[HPM*20] HART D., PHARR M., MÜLLER T., LOPES W., MCGUIRE M., SHIRLEY P.: Practical product sampling by fitting and composing warps. *Computer Graphics Forum 39*, 4 (2020). doi:10.1111/cgf.14060. 2, 7, 8

[LADL18] LI T.-M., AITTALA M., DURAND F., LEHTINEN J.: Differentiable Monte Carlo ray tracing through edge sampling. *ACM Trans. Graph. (proc. SIGGRAPH Asia) 37*, 6 (Dec. 2018). doi:10.1145/3272127.3275109. 2, 7, 8

[LdR14] LAGARDE S., DE ROUSIERS C.: Physically based shading in theory and practice: Moving Frostbite to PBR. In *ACM SIGGRAPH 2014 Courses* (2014). URL: https://seblagarde.files.wordpress.com/2015/07/course_notes_moving_frostbite_to_pbr_v32.pdf, doi:10.1145/2614028.2615431. 6

[MPC19] MOREAU P., PHARR M., CLARBERG P.: Dynamic many-light sampling for real-time ray tracing. In *High-Performance Graphics - Short Papers* (2019), Steinberger M., Foley T., (Eds.), The Eurographics Association. doi:10.2312/hpg.20191191. 2

[NM65] NELDER J. A., MEAD R.: A simplex method for function minimization. *The Computer Journal 7*, 4 (01 1965), 308–313. doi:10.1093/comjnl/7.4.308. 4

[NON85] NISHITA T., OKAMURA I., NAKAMAE E.: Shading models for point and linear sources. *ACM Trans. Graph. 4*, 2 (Apr. 1985), 124–146. doi:10.1145/282918.282938. 2, 3

[PA91] POULIN P., AMANATIDES J.: Shading and shadowing with linear light sources. *Computers & Graphics 15*, 2 (1991), 259–265. doi:10.1016/0097-8493(91)90079-W. 2

[PD19] PETERS C., DACHSBACHER C.: Sampling projected spherical caps in real time. *Proc. ACM Comput. Graph. Interact. Tech. 2*, 1 (June 2019). doi:10.1145/3320282. 2, 8

[Pet21] PETERS C.: BRDF importance sampling for polygonal lights. *ACM Trans. Graph. (proc. SIGGRAPH) 40*, 4 (July 2021). doi:10.1145/3450626.3459672. 1, 2, 3, 4, 5, 6, 7, 8

[Pic92] PICOTT K. P.: Extensions of the linear and area lighting models. *IEEE Computer Graphics and Applications 12*, 2 (1992), 31–38. doi:10.1109/38.124286. 2, 3

[PJH16] PHARR M., JAKOB W., HUMPHREYS G.: *Physically Based Rendering, 3rd Edition*. Morgan Kaufmann, Nov. 2016. URL: http://www.pbr-book.org. 3

[PTVF07] PRESS W. H., TEUKOLSKY S. A., VETTERLING W. T., FLANNERY B. P.: *Numerical recipes: The art of scientific computing (third edition)*. Cambridge University Press, 2007. 4

[RAMN12] RAMAMOORTHI R., ANDERSON J., MEYER M., NOWROUZEZAHRAI D.: A theory of Monte Carlo visibility sampling. *ACM Trans. Graph. 31*, 5 (Sept. 2012). doi:10.1145/2231816.2231819. 2, 7

[ST95] SCAVO T. R., THOO J. B.: On the geometry of Halley's method. *The American Mathematical Monthly 102*, 5 (1995), 417–426. doi:10.2307/2975033. 4

[Uli93] ULICHNEY R. A.: Void-and-cluster method for dither array generation. *Proc. SPIE 1913* (1993), 1–12. doi:10.1117/12.152707. 7

[Un00] UREÑA C.: Computation of irradiance from triangles by adaptive sampling. *Computer Graphics Forum*, 2 (2000). doi:10.1111/1467-8659.00452. 2

[UnFK13] UREÑA C., FAJARDO M., KING A.: An area-preserving parametrization for spherical rectangles. *Computer Graphics Forum (proc. EGSR) 36*, 4 (June 2013). doi:10.1111/cgf.12151. 2, 8

[UnG18] UREÑA C., GEORGIEV I.: Stratified sampling of projected spherical caps. *Computer Graphics Forum (proc. EGSR) 37*, 4 (2018). doi:10.1111/cgf.13471. 2

[VG95] VEACH E., GUIBAS L. J.: Optimally combining sampling techniques for Monte Carlo rendering. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques* (1995), SIGGRAPH '95, ACM. doi:10.1145/218380.218498. 2

[Wan92] WANG C.: Physically correct direct lighting for distribution ray tracing. In *Graphics Gems III* (1992), Academic Press Professional, pp. 307–313. 2