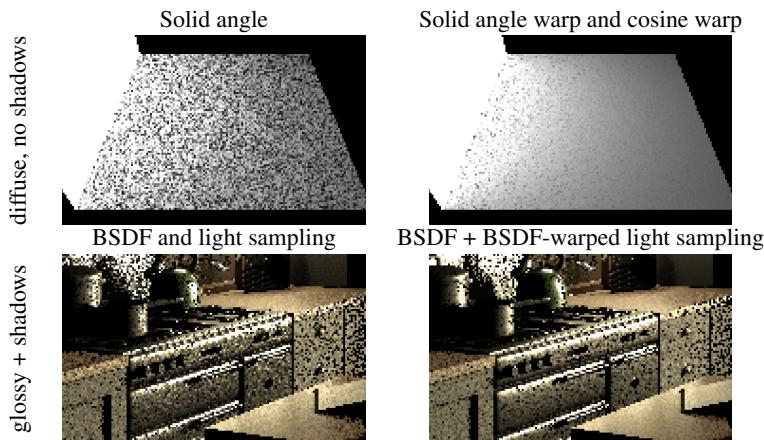


# Practical Product Sampling by Fitting and Composing Warps

D. Hart, M. Pharr, T. Müller, W. Lopes, M. McGuire and P. Shirley  
NVIDIA



A single polygon with a diffuse BRDF and no shadows, 1 sample per pixel. Left: uniform solid angle sampling [UJK13]. Right: our approach, which composes the uniform solid angle warp with a second warp that approximates projected solid angle sampling.

A complex scene with shadows and a variety of BSDFs, 4 samples per pixel. Left: MIS with BSDF and uniform solid angle sampling of the lights. Right: MIS where the light is sampled by composing a second warp that approximates the BSDF and cosine. MRSE (error) is reduced by over 60% with a run-time cost of about 10%.

**Figure 1:** Two examples of fitting and composing warps to improve sampling efficiency.

## Abstract

We introduce a Monte Carlo importance sampling method for integrands composed of products and show its application to rendering where direct sampling of the product is often difficult. Our method is based on warp functions that operate on the primary samples in  $[0, 1]^n$ , where each warp approximates sampling a single factor of the product distribution. Our key insight is that individual factors are often well-behaved and inexpensive to fit and sample in primary sample space, which leads to a practical, efficient sampling algorithm. Our sampling approach is unbiased, easy to implement, and compatible with multiple importance sampling. We show the results of applying our warps to projected solid angle sampling of spherical triangles, to sampling bilinear patch light sources, and to sampling glossy BSDFs and area light sources, with efficiency improvements of over  $1.6\times$  on real-world scenes.

## 1. Introduction

Many rendering problems are formulated as integration problems that solve the Rendering Equation using Monte Carlo integration [PJH16]. A core Monte Carlo variance reduction technique is *importance sampling*, which is based on sampling random points from some desired probability density function (PDF). This sampling process is generally defined in terms of mapping  $n$ -dimensional uniform points  $\mathbf{u} \in [0, 1]^n$  (also known as the *primary sample space* (PSS)) to an  $n$ -dimensional manifold  $\mathcal{D}$  (e.g., the surface of a shape or the unit hemisphere) using a mapping  $[0, 1]^n \rightarrow \mathcal{D}$ .

Given a PDF  $p(\mathbf{x})$ , such a mapping can be found through the

inversion method, where  $p$  is written as the product of 1D PDFs

$$p(\mathbf{x}) = p(x_1)p(x_2|x_1)\cdots p(x_n|x_1, \dots, x_{n-1}), \quad (1)$$

where the CDFs  $P(x_i|x_1, \dots, x_{i-1})$  of each 1-dimensional PDF are derived and subsequently inverted. This approach allows the use of stratified and low-discrepancy PSS points, which generally reduces error. However, for many quantities of interest in rendering it is not possible to compute the closed form of every one of the PDFs, the CDFs, and inverse CDFs.

The collection of inverse CDFs is also known as a *warp*, and in statistics is called a *quantile* function. “Warp” describes the process of displacing uniform random samples non-uniformly to achieve a desired probability distribution; a warp distributes points according to its associated probability density.

It is common practice to choose sampling methods for the individual factors that are analytically invertible, even though they do not match the full distribution of interest, in order to take advantage of the inversion method. For example, Arvo derived a warp that uniformly sampled spherical triangles [Arv95], but no analytic technique for uniformly sampling cosine-weighted spherical triangles is known. The cosine term may vary significantly over the surface of a triangle; in turn, that variation increases variance in Monte Carlo estimates of integrals that include its factor, since it is not accounted for in the importance sampling distribution.

Given a base warp (e.g. one that samples points on a light source), we show that additional warps can be applied to PSS before the base warp in order to better approximate a desired distribution (e.g. cosine-weighted solid angle sampling). Because the composed warp is used for importance sampling, it maintains an unbiased Monte Carlo estimator. The only potential disadvantage of our approach is a loss of efficiency if the choice of fit yields a composed warp that poorly approximates the desired product distribution. However, these composed warps can be used in conjunction with multiple importance sampling [VG95] as an additional sampling technique, thus limiting error if the warp is inaccurate.

We have investigated several PSS warps that can be efficiently fit and evaluated. We find that in practice even simple warps fit with a simple fitting algorithm are effective at reducing variance. The technique is easily added into existing rendering systems. We demonstrate efficiency improvements in complex scenes of over  $2.5 \times$  for diffuse materials and over  $1.6 \times$  for materials with glossy BSDFs.

## 2. Related Work

Importance sampling has been extensively investigated for rendering; here we focus on work directly related to ours. See Pharr et al. [PJH16] for a comprehensive overview.

Algorithms based on the inversion method have been among the most effective importance sampling techniques in practice because they are simple, local and analytic, and they can be used with stratified and low-discrepancy sample points, which further reduce error. Representative examples include Shirley et al.’s algorithms for sampling light sources [SWZ96], Arvo’s spherical triangle sampling [Arv95], and Heitz and d’Eon’s microfacet sampling [Hd14]. Our method builds directly on inversion-method sampling techniques: we augment such techniques by taking into account missing factors via a pre-warp of PSS.

It is often difficult to derive closed-form sampling algorithms, and the variance of the Monte Carlo estimator is high if the sampling distribution is a poor match for the integrand. Multiple sampling distributions can be combined using multiple importance sampling (MIS) [VG95] to bound the variance, but error can still be inferior to that achieved by accurate product sampling. Even though our approach aims at product sampling, it can be combined with defensive sampling strategies using MIS to prevent approximation errors from increasing variance by a large amount.

A variety of approaches have been developed for sampling directly from products of factors, though with limitations such as high memory cost [CAM08], high computational

cost [MMR<sup>+</sup>19], the requirement that BRDFs be precomputed and tabularized [RCL<sup>+</sup>08], or being applicable only to products of certain types of functions [HEV<sup>+</sup>16, HZE<sup>+</sup>19]. Algorithms for environment map product sampling also exist but need to draw multiple light samples per shading point in order to be effective [CJAMJ05, CEL18]. In contrast, our approach requires no additional storage and introduces a small computational overhead.

Resampled importance sampling (RIS) approximates a product sampling distribution [TCE05], though with high variance if the initial sampled factor is a poor match to the product. RIS is complementary to our approach; in future work it would be interesting to investigate the performance of applying RIS to samples generated with our method.

A number of rendering algorithms work from the perspective of primary sample space. Kelemen et al.’s Metropolis light transport algorithm [KSKAC02] recasts Veach and Guibas’s algorithm [VG97] to operate in PSS, which leads to a much simpler implementation. Their insight was that because the mapping from the PSS to points along a light carrying path is mostly continuous, small perturbations in PSS generally correspond to small permutations to a path, and thus Metropolis sampling in PSS is effective. Our approach leverages this property of PSS as well.

PSS has more recently been used as a domain for learning important light-carrying paths during rendering. Guo et al. [GBBE18] developed a path guiding algorithm that learns a high-dimensional warp within PSS using a kd-tree. More closely related to our approach is the work of Müller et al. [MMR<sup>+</sup>19] and Zheng and Zwicker [ZZ19], who apply a composition of warps to primary samples, each of which is parameterized by a neural network and fit using gradient descent. In contrast, we use much simpler warping functions and fit them uniquely at each point being shaded, rather than globally using prior paths. Our method is thus significantly less costly, albeit with larger approximation error.

Our composition of warps is closely related to “normalizing flows” [TV10, TT13], which represent PDFs resulting from a composition of warps that are applied to a base PDF. In fact, our approach is a special case of a normalizing flow: one with uniform base PDF that approximates the product distribution of the rendering integral. However, unlike many techniques relying on normalizing flows [DKB14, RM15, MMR<sup>+</sup>19, ZZ19, MRNK20], we do *not* use neural networks to parameterize our warps and we use closed-form fitting routines as opposed to gradient-based update rules.

Arvo [Arv01] presented an approach for deriving importance sampling algorithms based on starting with a non-uniform mapping to the domain being sampled and then deriving the sample warp that gives a uniform sample distribution when composed with the original mapping. One interpretation of our approach is that we approximately match these warps when closed-form solutions are not available.

Subr and Arvo [SA07] and Portsmouth [Por17] described algorithms for linearly interpolating sampling densities over shapes. Our work generalizes theirs; by operating instead in PSS, we are able to use a wider variety of sampling warps and are able to apply a series of warps in succession to account for multiple factors.

**Table 1:** Notation

Symbol	Definition
$\mathbf{u}$	Uniform $n$ -dimensional PSS point in $[0, 1]^n$ .
$\mathcal{D}$	General $n$ -dimensional domain.
$\mathbf{x}$	Point in $\mathcal{D}$ .
$p(\mathbf{x})$	Probability density function (PDF).
$P(\mathbf{x})$	Probability measure. In 1D, the cumulative distribution function (CDF).
$w(\mathbf{x})$	Warp: a bijection from $\mathbb{R}^n$ to itself used to warp samples $\mathbf{x}$ .
$J_w$	Reciprocal of the absolute determinant of the Jacobian matrix, $\left  \det \left( \frac{\partial w_i(\mathbf{x})}{\partial \mathbf{x}^T} \right) \right ^{-1}$ , of a warp $w$ .

Further, we demonstrate the value of our approach for a range of applications.

Dupuy et al. [DHB17] introduced an approach for computing distributions over spherical caps that can be easily sampled from and integrated. Among other applications, they showed the applicability to BSDF product sampling with spherical light sources. Because our approach operates in PSS, it allows product sampling with any type of area light source that has a parameterization. While we also use an approximation to the analytically unrealizable ideal warp, we do so within a framework of composable warps that can be used with a wide variety of integrands.

To our knowledge, the first application of warping PSS to reduce variance in Monte Carlo integration was described by Booth in the field of neutron transport [Boo86]. He describes an online adaptation of random numbers during a simulation to increase the probability of generating samples in regions of sample space where the integrand’s value is relatively large. Our method uses a more lightweight analytic approach to attack the same goal.

### 3. Warping Primary Sample Space

To derive our sampling and warp-fitting algorithms, we will first formalize the effect of warps on probability distributions and then show characteristics of theoretically ideal warps for importance sampling a given Monte Carlo integrand. Then, we will present our practical approximation of such an ideal warp. See Table 1 for a summary of notation.

#### 3.1. Background

We are interested in the effect of a warp  $w$ —which we define as a continuous, bijective mapping—on the probability density of  $n$ -dimensional points  $\mathbf{x} \in \mathbb{R}^n$  that are distributed according to  $p(\mathbf{x})$ . More precisely, we are interested in expressing the probability density  $p_w(\mathbf{x}')$ , where  $\mathbf{x}' = w(\mathbf{x})$ . Multivariable calculus allows us to express how this PDF transforms under warps via the change-of-variables formula.

$$p_w(\mathbf{x}') = p(\mathbf{x}) \left| \det \left( \frac{\partial w(\mathbf{x})}{\partial \mathbf{x}^T} \right) \right|^{-1} = p(\mathbf{x}) J_w(\mathbf{x}), \quad (2)$$

where  $J_w(\mathbf{x}) := \left| \det \left( \frac{\partial w(\mathbf{x})}{\partial \mathbf{x}^T} \right) \right|^{-1}$  is the *reciprocal* of the *absolute determinant* of the Jacobian matrix of  $w$  at  $\mathbf{x}$ .

Note that in the more general case where the warp maps to/from a manifold embedded in a space with different dimensionality, the square root of the determinant of the Gram matrix would be required, but this is not the case for the warps we consider.

To make intuitive sense of the change-of-variable formula, note that  $\left| \det \left( \frac{\partial w(\mathbf{x})}{\partial \mathbf{x}^T} \right) \right|$  describes the *multiplicative change in  $n$ -dimensional hypervolume* caused by  $w$  in an infinitesimal neighbourhood around  $\mathbf{x}$ . Since  $p(\mathbf{x})$  is a *density*, with units of inverse hypervolume, it scales by the reciprocal of the change in hypervolume: namely by  $J_w(\mathbf{x})$ .

**Relation to the inversion method.**  $J_w(\mathbf{x})$  is a generalization of the reciprocal of the 1-dimensional derivative  $(dw/dx)^{-1}$ , which can be equivalently thought of as the 1-dimensional derivative of the *inverse warp*  $dw^{-1}/dx'$ . The relationship between the derivative of the inverse warp and the sample PDF is widely known in computer graphics as the “inversion method.” The change-of-variable formula is therefore a generalization of the inversion method to higher dimensions. What this means is that whenever we have a 1-dimensional CDF $^{-1}$ -PDF pair that acts on uniform samples, there is a 1:1 correspondence between the CDF $^{-1}$  and  $w$  as well as between the PDF and  $J_w(\mathbf{x})$ ; in other words,  $J_w(\mathbf{x})$  can be computed by evaluating the PDF of  $\mathbf{x}'$ .

#### 3.1.1. Composite Warps and their PDF

Next, we consider the effect of composing multiple warps together. We define a composite warp as  $w = w_m \circ \dots \circ w_1$ . By the chain rule, the Jacobian matrix of the composition of  $m$  warps is the product of the Jacobian matrices of the individual warps. Applying this rule to Equation 2 leads to

$$p_w(\mathbf{x}^m) = p(\mathbf{x}) \left| \det \left( \prod_{i=1}^m \frac{\partial w_i(\mathbf{x}^i)}{\partial \mathbf{x}^{iT}} \right) \right|^{-1}, \quad (3)$$

where  $\mathbf{x}^i$  are the intermediate values of  $\mathbf{x}$  between warps:

$$\begin{aligned} \mathbf{x}^1 &= \mathbf{x} \\ \mathbf{x}^i &= (w_{i-1} \circ \dots \circ w_1)(\mathbf{x}) \text{ for } i \geq 2. \end{aligned} \quad (4)$$

Since the determinant of the product of matrices is equal to the product of their determinants, we can express Equation 3 in terms of the  $J_{w_i}$  of the individual warps  $w_i$ :

$$\begin{aligned} p(\mathbf{x}) \left| \det \left( \prod_{i=1}^m \frac{\partial w_i(\mathbf{x}^i)}{\partial \mathbf{x}^{iT}} \right) \right|^{-1} &= p(\mathbf{x}) \prod_{i=1}^m \left| \det \left( \frac{\partial w_i(\mathbf{x}^i)}{\partial \mathbf{x}^{iT}} \right) \right|^{-1} \\ &= p(\mathbf{x}) \prod_{i=1}^m J_{w_i}(\mathbf{x}^i). \end{aligned} \quad (5)$$

We use uniform samples in PSS, i.e.  $p(\mathbf{x}) = 1$ , and therefore the absolute value of the Jacobian determinant is itself the PDF of the warped samples:

$$p_w(\mathbf{x}^m) = \prod_{i=1}^m J_{w_i}(\mathbf{x}^i) = J_w(\mathbf{x}). \quad (6)$$

We are interested in *augmenting* existing sampling techniques that are based on the inversion method and so we assume an existing sampling technique (e.g. BSDF sampling or uniform emitter sampling)  $\mathbf{x}' = P_s^{-1}(\mathbf{x})$  with associated PDF  $p_s(\mathbf{x})$ . We therefore set the *last* warp of our composite warp to that technique, i.e.  $w_m = P_s^{-1}$  and  $J_{w_m} = p_s$ ; all earlier warps then are PSS warps that operate on  $[0, 1]^n$ .

Algorithm 1 generates warped samples and computes their PDF using Equation 6. Note that the final PDF can easily be computed during sampling using the successive warped values of  $\mathbf{x}^i$  and the product of  $J_{w_i}$  values. To compute the PDF for an arbitrary value  $\mathbf{x} \in \mathcal{D}$  (for example, for multiple importance sampling), the inverse warps  $w_i^{-1}$  can be applied to  $\mathbf{x}$  in reverse order, with their Jacobian determinant values multiplied together along the way; see Algorithm 2.

Note that in Algorithm 2, it is necessary to be able to invert  $w_m$ , the base sampling technique we started with. Thus, techniques such as rejection sampling that cannot be inverted are unusable with our method. For sampling techniques based on the inversion method, the inverse of  $w_m$  is just the collection of CDFs in each dimensions, which are derived along with the sampling technique. The inverses of uniform area sampling algorithms of shapes are straightforward, but the inverses of uniform spherical sampling algorithms [Arv95, UFK13] have not been previously published, to our knowledge. We include implementations in the supplementary material.

### 3.2. Optimal Warps

The Monte Carlo estimator of the integral of a function  $f$  using warped samples  $f(\mathbf{x}^m)/p_w(\mathbf{x}^m)$  achieves our goal of minimizing variance when the PDF of the warped samples is proportional to  $f$ :

$$p_w(\mathbf{x}^m) = \prod_{i=1}^m J_{w_i}(\mathbf{x}^i) \propto f(\mathbf{x}^m). \quad (7)$$

---

**Algorithm 1:** General sample warping algorithm. Given  $m$  warps  $w_i$ , the warps are successively applied to a uniform PSS sample  $\mathbf{u}$ . The final sample value  $\mathbf{x} \in \mathcal{D}$  and its PDF are returned.

---

```

pdf ← 1
x ← u
for i ← 1 to m do
|   x ← w_i(x)
|   pdf ← pdf × J_wi(x)
end
return x, pdf

```

---

Consider the case of having one or more existing warps (e.g.,  $w_m = P_s^{-1}$  and possibly one or more PSS warps) where we'd like to compose an additional warp  $w_i$  (to aim for being proportional to  $f$ ). Simple algebra on Equation 7 yields the optimality condition on  $w_i$ :

$$J_{w_i}(\mathbf{x}^i) \propto \frac{f(\mathbf{x}^m)}{\prod_{j \neq i} J_{w_j}(\mathbf{x}^j)}. \quad (8)$$

---

**Algorithm 2:** Algorithm to compute the PDF of a given value  $\mathbf{x} \in \mathcal{D}$ . Note that the order in which the PSS warps are applied is reversed compared to their use for generating samples.

---

```

pdf ← 1
for i ← m to 1 do
|   pdf ← pdf × J_wi(x)
|   x ← w_i^{-1}(x)
end
return pdf

```

---

Thus, an additional  $i$ -th warp—which can be inserted into the chain of warps at *any* position  $i$ —can, theoretically, perfectly correct any remaining discrepancy w.r.t.  $f$  by having the right inverse Jacobian determinant.

In practice, perfectly correcting all remaining discrepancy is usually impossible, but one can take smaller steps. For example, if pre-existing warps cover *some* factors of  $f$ , such as the BSDF in the rendering equation, then additional warps can be introduced for the remaining factors—e.g. the foreshortening term.

### 3.3. Fitting Warping Functions

We focus on fitting customized warping functions from scratch at each point being shaded: doing so eliminates the need to maintain additional data structures and avoids the complexity and computational expense of on-line learning during rendering. In turn, we must use warping functions that can be efficiently fit and evaluated. Because time spent on fitting and evaluating warps could instead be used to take more samples without warping, warping must be more efficient overall to be worthwhile.

It is often useful to fit an approximation to the right-hand side of Equation 8, for example neglecting expensive factors (such as visibility) or using approximations (for example, to the BSDF.) Furthermore, if we know that a factor  $g$  of  $f$  isn't accounted for by any of the existing warps, we can directly fit  $J_{w_i}(\mathbf{x}^i) \propto g(\mathbf{x}^m)$ .

Although it may seem wise to incorporate all of the factors in the right-hand side of Equation 8 when fitting a warp, we have found the best results by limiting each warp to have responsibility for just part of the overall function  $f$ . We believe that because the warping functions we use have limited expressive power, their fits may be made worse if they also try to include the residual errors from other warps.

To make fitting as simple as possible, we directly approximate the optimal *inverse Jacobian determinant* (the right-hand side of Equation 8) up to a constant factor rather than attempting to derive a good warp in a single step. This idea is similar to that employed by Müller et al.'s piecewise-polynomial warps [MMR<sup>+</sup>19]. We use simple parametric functions that admit closed-form integrals, which allows us to normalize them and obtain their inverse Jacobian determinant, as well as to apply the inversion method to find a corresponding warp; these functions are summarized in Table 2. The simplest, a bilinear function, is defined by  $2 \times 2$  values at the corners of the parametric domain. The biquadratic Bézier is

**Table 2:** 2D functions  $a(\mathbf{x})$  that we used to fit PSS product factors.

	$a(\mathbf{x})$	Parameters
Bilinear	$g(\mathbf{x}_1, g(\mathbf{x}_2, v_{0,0}, v_{1,0}),$ $g(\mathbf{x}_2, v_{0,1}, v_{1,1})),$ with $g(x, c_0, c_1) = (1-x)c_0 + xc_1.$	$v_{i,j}$
Biquadratic Bézier	$b(\mathbf{x}_1, b(\mathbf{x}_2, v_{0,0}, v_{1,0}, v_{2,0}),$ $b(\mathbf{x}_2, v_{0,1}, v_{1,1}, v_{2,1}),$ $b(\mathbf{x}_2, v_{0,2}, v_{1,2}, v_{2,2})),$ with $b(x, c_0, c_1, c_2) =$ $(1-x)^2 c_0 + 2(1-x)xc_1 + x^2 c_2.$	$v_{i,j}$

more expressive with  $3 \times 3$  control points, but requires solving two cubic equations to apply the warp.

Given  $\hat{f}$  that represents either the right hand side of Equation 8, or some approximation of it, we fit the approximation functions  $a(\mathbf{x})$  via point-wise evaluation of  $\hat{f}$  at the corresponding PSS points for their parameters. If we are fitting a warp  $w_i$ , then for each warp parameter  $v$ , we take the corresponding PSS point  $\mathbf{u}$ , compute the corresponding  $\mathbf{x}^m \in \mathcal{D}$  by applying the subsequent warps, and evaluate  $\hat{f}$ :

$$v_{i,j} \leftarrow \hat{f}(\mathbf{x}^m) = \hat{f}((w_m \circ \dots \circ w_{i+1})(\mathbf{u})). \quad (9)$$

As an example, consider using a bilinear function in PSS to approximate for the cosine term for direct lighting from a triangular light source, where Arvo’s uniform triangle solid angle sampling algorithm is being used:  $w_1$  is the warp to fit, and  $w_2$  applies Arvo’s approach. The parameters  $v_{i,j}$  lie at the corners of PSS— $(0,0)$ ,  $(0,1)$ ,  $(1,0)$ , and  $(1,1)$ —and  $w_2$  maps those four PSS values to the three vertices of the triangle, with two PSS corners mapped to one triangle vertex. In turn, the values of the cosine factor at the corresponding triangle vertices give the weights  $v_{i,j}$  for the bilinear approximation function.

More generally, if multiple PSS warps are used, PSS points at which we evaluate  $\hat{f}$  must be warped by all subsequent, already specified warps to find  $\mathbf{x}^m$ , as shown in Equation 9.

Each of our approximation functions can be normalized over  $[0, 1]^2$  to obtain its corresponding inverse Jacobian determinant  $J_w$ . Corresponding warps and inverse warps can then be found directly. In 1D, given  $J_w$ , the warp can be found using the inversion method: integrate  $J_w$  and then invert the result. For a multi-dimensional warp, an infinite number of warps can have  $J_w$ . In that case, we only need a single valid warp, which can be derived by marginalizing over all of the dimensions except one and applying the inversion method to find a 1D warp in that dimension. After applying this newfound warp along the chosen dimension, the warp over the remaining dimensions can be found recursively by the same approach, ignoring all previously warped dimensions, until all dimensions have been warped. Warps, inverse warps, and inverse Jacobian determinants  $J_w$  for our parametric functions are summarized in Table 3.

## 4. Results

We have applied PSS warps to three direct lighting problems: cosine-weighted solid angle sampling of light sources, uniform area sampling of bilinear patches, and sampling the product of BSDFs and light sources. Our experiments were performed with a modified version of the *pbtrt* renderer [PJH16], which is included in the supplemental material. All images were rendered with one sample per pixel using multiple importance sampling with one light sample and one BSDF sample. In the following, we used Celarek et al.’s approach to compute robust estimates of MSE and its per-pixel variance [CJWL19]. We use *Monte Carlo efficiency*, one over the product of MSE and running time, to compare techniques.

### 4.1. Cosine-Weighted Solid Angle Sampling

When estimating direct illumination from emissive geometry with Monte Carlo integration, it is more efficient to uniformly sample the solid angle subtended by the emitter than to uniformly sample its surface area [Arv95, UFK13, Gam16, TWCC06]. Doing so eliminates variation in the integrand due to the inverse squared distance factor and the cosine between the light’s normal vector and the outgoing light direction, which in turn reduces variance. Better is to also incorporate the cosine of the incident direction at the shading point into the sampling distribution. Although there has been progress in doing so for spherical lights [UnG18, PD19], it has remained challenging for other shapes, requiring numerical inversion of CDFs [Arv01] or adaptive refinement algorithms that are not based on a warping and thus cannot make use of stratified or low-discrepancy sample points to reduce variance [Un00].

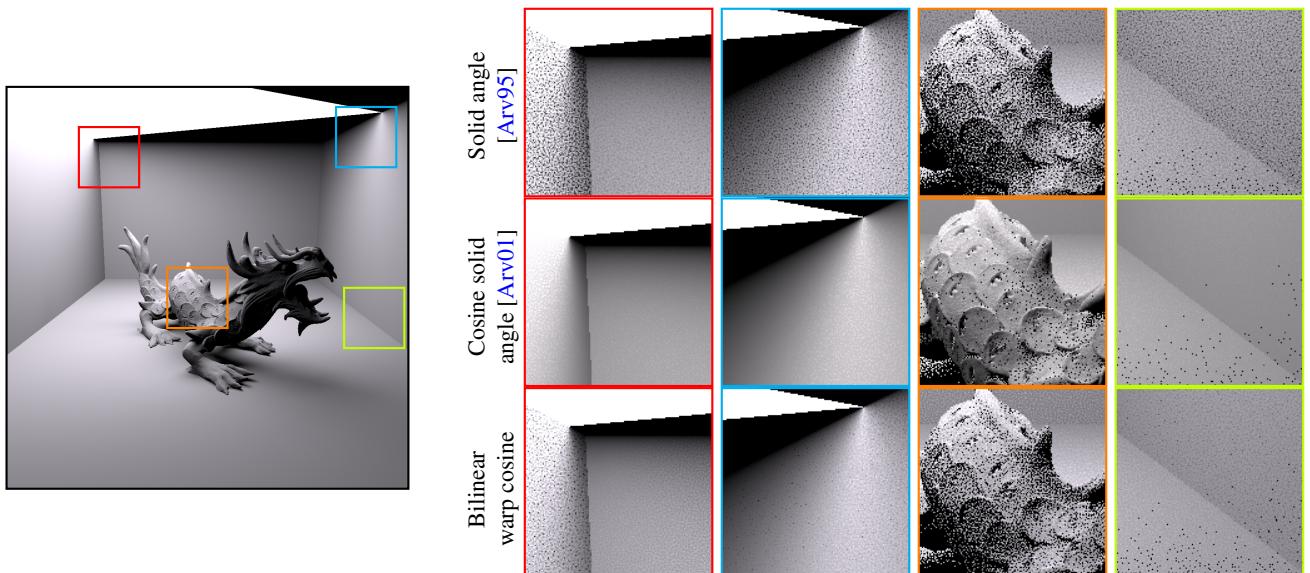
We have applied PSS warps to incorporate this additional cosine factor, transforming it into PSS and fitting a warp to approximate the cosine, as described in Section 3.3. Figure 2 shows a scene where PSS warps were used to account for the cosine factor of a triangular light source sampled using Arvo’s algorithm [Arv95]. Using a bilinear warp for the incident cosine factor reduces MSE by a factor of  $1.75 \times$  compared to uniform solid angle sampling, and a biquadratic warp gives a  $1.91 \times$  MSE improvement. The respective increases in run time are 1% and 6%.

We implemented Arvo’s method for cosine weighted solid angle sampling [Arv01], which requires numerical inversion of the CDF; we used six iterations of bisection, which we found to be faster and more stable than both Newton-Raphson and Arvo’s cubic approximation. This method is optimal for a diffuse BRDF if visibility is not considered in the sample distribution. For the scene in Figure 2, Arvo’s method gives  $3.8 \times$  lower error than uniform solid angle sampling with a relative run time cost of  $1.24 \times$ . Error is  $1.5 \times$  less than with our bilinear warp, at a cost of an increase of  $1.21 \times$  in rendering time; efficiency is thus  $1.28 \times$  that of our linear warp approximation. Our implementation of Arvo’s algorithm requires the evaluation of over 100 transcendental functions for each sample generated and uses  $17 \times$  more time than our bilinear warp.

We have also measured the effectiveness of approximate cosine-weighted sampling with more complex scenes and saw error reductions of  $1.25 \times$  to  $1.48 \times$ . Those results are presented in Section 4.3 along with results for approximate BSDF product warp sampling.

**Table 3:** The associated warps  $w(\mathbf{x})$ , inverse warps  $w^{-1}(\mathbf{x})$ , and inverse Jacobian determinants  $J_w(\mathbf{x})$  for the fitting functions  $a(\mathbf{x})$  in Table 2. (Note: the auxiliary functions  $g$  and  $b$  used below are defined in Table 2.)

Warp	$w(\mathbf{x})$	$w^{-1}(\mathbf{x})$	$J_w(\mathbf{x})$
Bilinear	$\mathbf{x}'_2 = h(\mathbf{x}_2, v_{0,0} + v_{1,0}, v_{0,1} + v_{1,1})$ $\mathbf{x}'_1 = h(\mathbf{u}_1, g(\mathbf{x}'_2, v_{0,0}, v_{0,1}), g(\mathbf{x}'_2, v_{1,0}, v_{1,1}))$ with $h(x, a, b) = \frac{a - \sqrt{g(x, a^2, b^2)}}{a - b}$ .	$\mathbf{x}'_0 = l(\mathbf{x}_0, g(\mathbf{x}_1, v_{0,0}, v_{0,1}), g(\mathbf{x}_1, v_{1,0}, v_{1,1}))$ $\mathbf{x}'_1 = l(\mathbf{x}_1, v_{0,0} + v_{1,0}, v_{0,1} + v_{1,1})$ with $l(x, a, b) = \frac{x(a(2-x)+bx)}{a+b}$	$\frac{4a(\mathbf{x})}{\sum_{i,j} v_{i,j}}$
Biquadratic Bézier	Solve $B(\mathbf{x}'_2, \sum_i v_{i,0}, \sum_i v_{i,1}, \sum_i v_{i,2}) - \mathbf{x}_2 = 0$ for $\mathbf{x}'_2$ , where $B(x, a, b, c) = \frac{(a-2b+c)x^3 + (3b-3a)x^2 + 3ax}{a+b+c}$ Solve $B(\mathbf{x}'_1, c_0, c_1, c_2) - \mathbf{x}_1 = 0$ for $\mathbf{x}'_1$ , where $c_i = b(\mathbf{x}'_2, v_{i,0}, v_{i,1}, v_{i,2})$ .	$\mathbf{x}'_0 = B(\mathbf{x}_0, b(\mathbf{x}_1, v_{0,0}, v_{1,0}, v_{2,0}), b(\mathbf{x}_1, v_{0,1}, v_{1,1}, v_{2,1}), b(\mathbf{x}_1, v_{0,2}, v_{1,2}, v_{2,2}))$ $\mathbf{x}'_1 = B(\mathbf{x}_1, c_0, c_1, c_2)$ where $c_i = \frac{1}{3} \sum_i v_{i,j}$	$\frac{9a(\mathbf{x})}{\sum_{i,j} v_{i,j}}$



**Figure 2:** Close-ups of triangular light test scene, rendered with one sample per pixel and multiple importance sampling using one light sample and one BSDF sample.

#### 4.2. Sampling Bilinear Patches

With the recent development of an efficient ray—bilinear patch intersection algorithm [Res19], bilinear patches are an increasingly appealing primitive for ray tracing. We have applied PSS warps both to uniform area sampling of bilinear patches as well as to approximate cosine-weighted sampling of them.

A bilinear patch can be defined in parametric form over  $[0, 1]^2$  given four points  $\mathbf{p}_{0,0}$ ,  $\mathbf{p}_{1,0}$ ,  $\mathbf{p}_{0,1}$ , and  $\mathbf{p}_{1,1}$  as

$$f(u, v) = (1-u)(1-v)\mathbf{p}_{0,0} + u(1-v)\mathbf{p}_{1,0} + (1-u)v\mathbf{p}_{0,1} + uv\mathbf{p}_{1,1}.$$

In order to generate uniform samples with respect to area on such a patch, points should be sampled with probability proportional to

$$\left\| \frac{\partial f(u, v)}{\partial u} \times \frac{\partial f(u, v)}{\partial v} \right\|, \quad (10)$$

where  $\times$  denotes the cross product and  $\|\cdot\|$  is the length of the

vector  $n$ . While the partial derivatives of a bilinear patch are easily computed, the integral of Equation 10 over the patch cannot be solved in closed form, which precludes deriving a closed-form equi-area sampling method.

The best current area sampling approach is to sample uniformly in  $(u, v)$ . The PDF of a sample is then

$$p(u, v) = 1 / \left\| \frac{\partial f(u, v)}{\partial u} \times \frac{\partial f(u, v)}{\partial v} \right\|. \quad (11)$$

For patches where two vertices are relatively close together, uniform parametric sampling places too many samples near those vertices. In turn, those samples’ PDF values are much larger than samples near the other two vertices, leading to higher variance.

Applying Equation 8, we can see that the ideal PSS warp would cancel out Equation 11’s factor. The first two images in Figure 3 show a test scene rendered with uniform parametric sampling and approximate equi-area sampling with a bilinear warp and numeric



**Figure 3:** Scene with an emissive bilinear patch, rendered with (left) uniform parametric sampling, (middle) approximate uniform area sampling using a bilinear warp, and (right) approximate cosine-weighted solid angle sampling using a composition of three PSS warps.

**Table 4:** Comparison of PSS warp-based sampling techniques to uniform parametric sampling for the scene in Figure 3. All images were rendered using MIS.

Sampling method	MSE factor	Time	Efficiency factor
Uniform parametric	1	1	1
Bilinear approx. area	0.373	1.01	2.66
Biquadratic approx. area	0.370	1.02	2.64
Three-warp approx. cosine	0.151	1.23	5.38

results are shown in Table 4. MSE is reduced by  $2.70\times$  and, thanks to the negligible computational overhead, efficiency is improved by  $2.66\times$ . We saw only marginal benefit from a biquadratic warp over a bilinear warp.

Arvo and Novins showed that for a planar quadrilateral, a bilinear warping gives uniformly-distributed points [AN07]. Our bilinear warp also gives exact equi-areal sampling in that case, while still working well for the more general case of bilinear patches.

Although approximate equi-area sampling is much better than uniform parametric sampling, it is further desirable to approximate cosine-weighted solid angle sampling when computing direct illumination. To do so, four factors must be accounted for: the non-uniformity of the patch parameterization, the inverse squared distance factor, and the cosine factors at the light source and receiving point.

We found that trying to encode the product of all of these into a single warp was ineffective, even with a biquadratic warp. Doing so gave worse results than just using approximate equi-area sampling. In practice these terms are mostly uncorrelated, so it is not surprising that a single simple warp does not have the capacity to encode their variation.

Therefore, we considered multi-stage PSS warping approaches. We found that the inverse squared distance and light source's cosine term are generally closely correlated so encoded them in a single

warp. The other two factors were warped independently in the following sequence:

- A bilinear warp to approximate equi-area sampling.
- A biquadratic warp that encodes the product of inverse squared distance and cosine at the light
- A biquadratic warp that accounts for the cosine at the illuminated point.

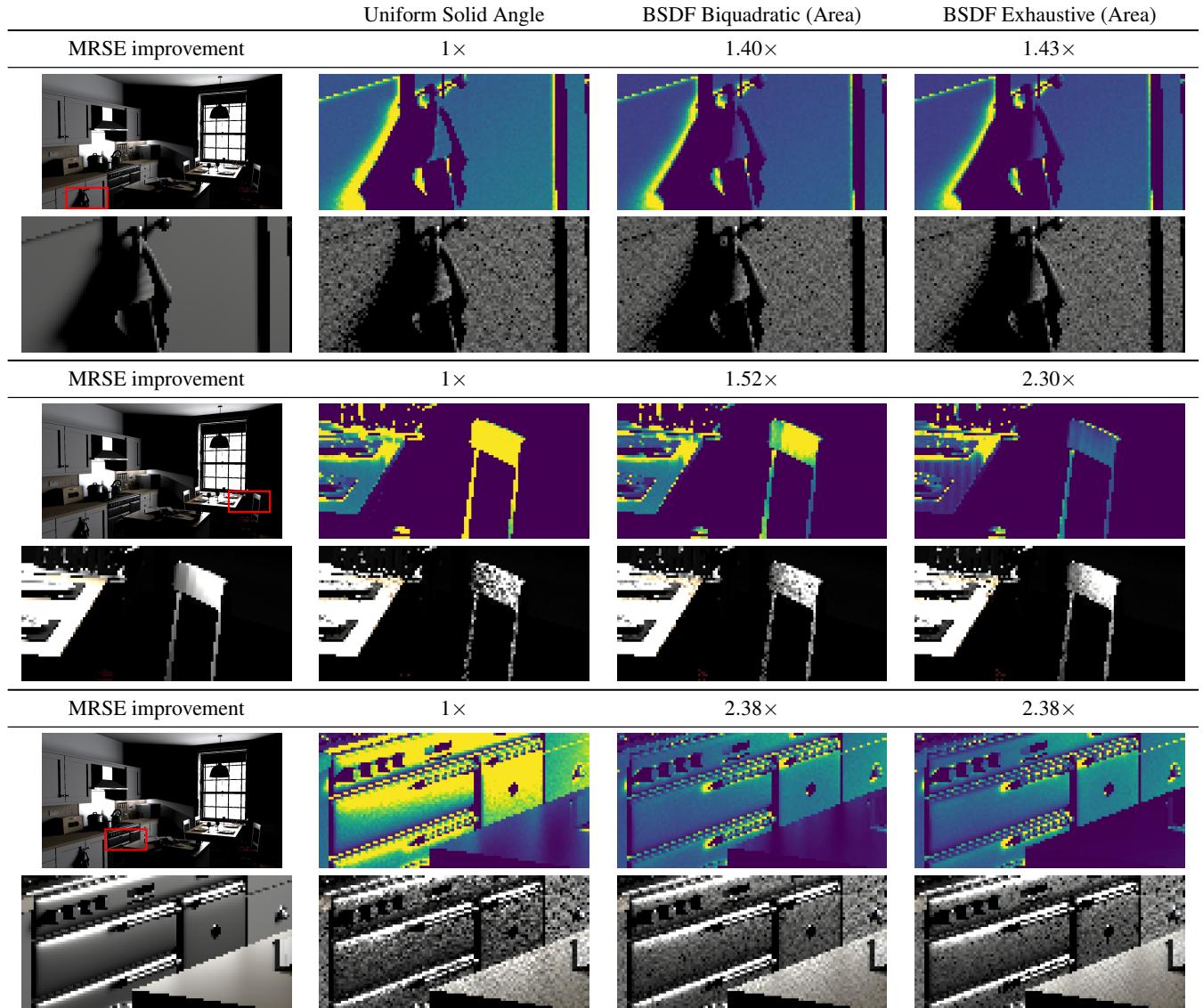
That approach incurred a 23% performance cost, with error almost as good as using four biquadratic warps.

The rightmost image in Figure 3 shows a rendering of the scene using this approach, and Table 4 compares MSE improvement, runtime, and Monte Carlo efficiency. Accounting for the cosine and inverse squared distance terms gives a substantial additional improvement in regions where those factors are important. Even though using one bilinear and two biquadratic warps increases running time by 23%, the  $6.62\times$  reduction in MSE it gives makes it the most efficient approach for this scene.

### 4.3. BSDF Product Warp Sampling

We have also applied PSS warps to BSDF-weighted light source sampling, fitting  $J_w$  to approximate the BSDF and the cosine term. We investigated both bilinear and biquadratic Jacobian fitting functions and measured results using both uniform area sampling and uniform solid angle sampling for the warp from PSS to the light source. Because these scenes include highly specular surfaces with bright highlights, we use mean relative squared error (MRSE) for all results in this section so that relatively small errors in bright pixels do not dominate the results.

We used three scenes with quadrilateral light sources to evaluate this approach for more complex scenes. Table 5 shows numeric results and Figure 4 shows cropped regions of images of the kitchen scene. The supplemental material includes raw MRSE images and similar visualizations for all three scenes as well as images rendered at low sampling rates. With these scenes, our approach gives as much as a  $1.67\times$  improvement in efficiency. The biquadratic



**Figure 4:** The effectiveness of various sampling techniques in different parts of the kitchen scene, at 16 samples per pixel. All results use MIS with a second sample taken based on the BSDF. In the left column is a fully resolved reference image. Middle, uniform solid angle light sampling, and BSDF-weighted light sampling using a biquadratic warp. Right, an exhaustive light sampling algorithm that computes a custom sampling distribution at each pixel, to estimate the best possible improvement of any sampling algorithm. MRSE improvement is measured relative to uniform solid angle sampling.

warp was the most effective at reducing MRSE, but the bilinear warp did surprisingly well and had the best overall efficiency, given that it only added 3–6% to the overall runtime.

The benefits of approximate BSDF product sampling vary across the images; specular highlights due to the specular peak are handled well with regular BSDF sampling, and diffuse regions are handled well by approximate cosine-weighted solid angle sampling alone. However, approximate BSDF product sampling significantly reduces error in regions like the TV screen in the living room scene

and by the metal backsplash behind the stove in the kitchen scene, where it improves by  $2.85 \times$ .

Most of the remaining error in these scenes is in pixels where the light source is partially occluded. It is also evident that using uniform area sampling for the final warp works better in some areas than uniform solid angle sampling, and vice versa. For example, both the kitchen and living room scene have higher error on the ceiling near the window when uniform solid angle sampling is used; the kitchen scene has lower error above the microwave with uniform solid angle sampling. While solid angle sampling works

**Table 5:** Improvement in MRSE and efficiency with cosine warp sampling and BSDF product warp sampling with both a uniform solid angle warp and a uniform area warp for the final warp, compared to uniform solid angle sampling for several complex scenes. Multiple importance sampling was used for all cases.

Scene	Sampling Method	Solid Angle Warp			Area Warp		
		MRSE	Time	Efficiency	MRSE	Time	Efficiency
	Cosine Bilinear Warp	0.780	1.02	1.26	—	—	—
	BSDF Product Bilinear Warp	0.796	1.04	1.21	0.600	1.00	1.67
	BSDF Product Biquadratic Warp	0.804	1.11	1.12	0.597	1.07	1.57
	BSDF Product Reference	0.760	—	—	0.565	—	—
	Cosine Bilinear Warp	0.797	1.03	1.22	—	—	—
	BSDF Product Bilinear Warp	0.676	1.06	1.40	0.598	1.03	1.62
	BSDF Product Biquadratic Warp	0.704	1.09	1.30	0.591	1.06	1.60
	BSDF Product Reference	0.628	—	—	0.600	—	—
	Cosine Bilinear Warp	0.838	1.01	1.18	—	—	—
	BSDF Product Bilinear Warp	0.816	1.03	1.19	0.770	1.01	1.29
	BSDF Product Biquadratic Warp	0.838	1.10	1.08	0.763	1.06	1.24
	BSDF Product Reference	0.802	—	—	0.732	—	—

well for unoccluded lights, if the seemingly important part of the light is actually occluded, more samples will have no contribution and error will increase.

To measure how close our sampling method is to an exact product sampling scheme, we compared to reference BSDF product sampling where, at each point being shaded, we evaluated the BSDF and cosine term at  $32 \times 32$  points on the light source and used those values to build a tabularized distribution to use for sampling points on the light. As expected, when we use the reference BSDF product method, error is nearly zero if the light source is unoccluded, but error remains where the light is partially visible.

For these scenes, our approximate bilinear warp provides much of the benefit of exact product sampling; for some scenes, our approach has slightly lower error. The per-pixel MSE variance images help explain this surprising result: in regions where the light is partially occluded, exact BSDF product sampling algorithm concentrates samples in a part of the light that is not visible; note, for example, the right side of the cabinets in the kitchen scene.

#### 4.4. Discussion

The warps we have considered in this work are efficient but are less expressive than more complex alternatives could be. Richer warping functions could be more effective, especially if combined with reusing warp fits at nearby points in the scene. For example, the warps in “normalizing flows” [TV10, TT13] gain their richness by being parameterized using neural networks [DKB14, RM15] and by being optimized from re-used samples [ZZ19, MMR<sup>+</sup>19, MRNK20]. In future work, one may attempt to import the gradient-based sample re-use and/or alternative warping functions into our framework.

Our PSS warping approach also implicitly assumes that the final

warp to  $\mathcal{D}$  is not discontinuous. If this is not the case, PSS warping is likely to be ineffective. More generally, because our fitting algorithm is based on point-wise evaluation at a small number of points in PSS, it can make no guarantees about the overall quality of the resulting fit. For example, consider sampling a textured light source where the final warp samples a point on the light according to its emitted radiance and where emission is concentrated in a small region of the light. Variance will increase if a PSS warp tries to account for the incident cosine; PSS points near the corners may be warped far across the surface of the light source, potentially causing them to have a much higher cosine factor than expected by the PSS warp’s  $J_{w_i}$  function.

## 5. Conclusion

We have presented a framework for approaching importance sampling by reasoning in terms of simple component warps in the PSS. We have shown the value of this approach for a variety of problems in Monte Carlo ray tracing, with efficiency gains of over  $1.6 \times$  for realistic scenes. The approach has minimal computational overhead and is easily integrated into standard Monte Carlo ray tracers; it is probably not a coincidence that modularizing the algebra of importance sampling to “black box” simple transforms in the PSS leads to natural modularization in the software realization.

Future work includes finding simpler component warps to add to the toolbox of our warps, e.g., an oriented “bump” that still allows stratification, more automated ways of assembling warps in series to approach particular integration problems, and applying the techniques we have presented to other Monte Carlo problems.

## Acknowledgments

Eugene d’Eon, Petrik Clarberg, and Ingo Wald provided numerous helpful suggestions, and Aaron Lefohn, David Luebke, Steven Parker, and NVIDIA Research supported this work. Jay-Artist modeled the kitchen and living room were and Mareck modeled the living room. Benedikt Bitterli converted the model to *pbrt*’s file format.

## References

- [AN07] James Arvo and Kevin Novins. Stratified sampling of convex quadrilaterals. *JGT*, 12(2):1–12, 2007.
- [Arv95] J. Arvo. Stratified sampling of spherical triangles. In *SIGGRAPH*, 1995.
- [Arv01] James Arvo. Stratified sampling of 2-manifolds. SIGGRAPH 2001 Course Notes: State of the Art in Monte Carlo Ray Tracing for Realistic Image Synthesis, 2001.
- [Boo86] Thomas E. Booth. A Monte Carlo learning/biasing experiment with intelligent random numbers. *Nuclear Science and Engineering*, 92(3):465–481, 1986.
- [CAM08] Petrik Clarberg and Tomas Akenine-Möller. Practical product importance sampling for direct illumination. *CGF*, 27(2):681–690, 2008.
- [CEL18] Alejandro Conty Estevez and Pascal Lecocq. Fast product importance sampling of environment maps. In *SIGGRAPH 2018*, SIGGRAPH 2018, 2018.
- [CJAM05] Petrik Clarberg, Wojciech Jarosz, Tomas Akenine-Möller, and Henrik Wann Jensen. Wavelet importance sampling: Efficiently evaluating products of complex functions. In *SIGGRAPH*, pages 1166–1175, New York, NY, USA, 2005.
- [CJWL19] A. Celarek, W. Jakob, M. Wimmer, and J. Lehtinen. Quantifying the error of light transport algorithms. *CGF*, 38(4):111–121, 2019.
- [DHB17] Jonathan Dupuy, Eric Heitz, and Laurent Belcour. A spherical cap preserving parameterization for spherical distributions. *ACM TOG*, 36(4):139:1–139:12, July 2017.
- [DKB14] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv:1410.8516*, October 2014.
- [Gam16] Manuel N. Gamito. Solid angle sampling of disk and cylinder lights. *CGF*, 35(4):25–36, 2016.
- [GBBE18] Jerry Jinfeng Guo, Pablo Bauszat, Jacco Bikker, and Elmar Eisemann. Primary sample space path guiding. In *EGSR*, pages 73–82, 2018.
- [Hd14] E. Heitz and E. d’Eon. Importance sampling microfacet-based bsdfs using the distribution of visible normals. In *EGSR*, pages 103–112, 2014.
- [HEV<sup>+</sup>16] Sebastian Herholz, Oskar Elek, Jiří Vorba, Hendrik Lensch, and Jaroslav Křivánek. Product importance sampling for light transport path guiding. *CGF*, 35(4):67–77, 2016.
- [HZE<sup>+</sup>19] Sebastian Herholz, Yangyang Zhao, Oskar Elek, Derek Nowrouzezahrai, Hendrik Lensch, and Jaroslav Křivánek. Volume path guiding based on zero-variance random walk theory. *ACM TOG*, 38(3):25:1–25:19, 2019.
- [KSKAC02] Csaba Kelemen, László Szirmay-Kalos, György Antal, and Ferenc Csonka. A simple and robust mutation strategy for the metropolis light transport algorithm. *CGF*, 21(3):531–540, 2002.
- [MMR<sup>+</sup>19] Thomas Müller, Brian McWilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. Neural importance sampling. *ACM TOG*, 38(5):145:1–145:19, October 2019.
- [MRNK20] Thomas Müller, Fabrice Rousselle, Jan Novák, and Alexander Keller. Neural control variates. *arXiv:2006.01524*, June 2020.
- [PD19] Christoph Peters and Carsten Dachsbacher. Sampling projected spherical caps in real time. *Proc. ACM Comput. Graph. Interact. Tech.*, 2(1):1:1–1:16, June 2019.
- [PJH16] M. Pharr, W. Jakob, and G. Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann, 2016.
- [Por17] Jamie Portsmouth. Efficient barycentric point sampling on meshes, 2017.
- [RCL<sup>+</sup>08] Fabrice Rousselle, Petrik Clarberg, Luc Leblanc, Victor Ostromoukhov, and Pierre Poulin. Efficient product sampling using hierarchical thresholding. *Vis. Comput.*, 24(7):465–474, July 2008.
- [Res19] A. Reshetov. Cool patches: A geometric approach to ray/bilinear patch intersections. In E. Haines and T. Akenine-Möller, editors, *Ray Tracing Gems*. Apress, Berkeley, CA, 2019.
- [RM15] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538, 2015.
- [SA07] Kartic Subr and James Arvo. Steerable importance sampling. In *Symp.on Inter. Ray Tracing*, pages 133–140, 2007.
- [SWZ96] Peter Shirley, Changyaw Wang, and Kurt Zimmerman. Monte carlo techniques for direct lighting calculations. *ACM TOG*, 15(1):1–36, 1996.
- [TCE05] Justin F. Talbot, David Cline, and Parris Egbert. Importance resampling for global illumination. In *EGSR*, pages 139–146, 2005.
- [TT13] Esteban Tabak and Cristina V. Turner. A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics*, 66(2):145–164, 2013.
- [TV10] Esteban Tabak and Eric Vanden Eijnden. Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, 8(1):217–233, 2010.
- [TWCC06] Yuan-Yu Tsai, Chung-Ming Wang, Chung-Hsien Chang, and Yu-Ming Cheng. Tunable bounding volumes for Monte Carlo applications. In *International Conference on Computational Science and Its Applications*. Springer, 2006.
- [UFK13] C. Ureña, M. Fajardo, and A. King. An area-preserving parametrization for spherical rectangles. In *EUROGRAPHICS*, 2013.
- [Un00] C. Ureña. Computation of irradiance from triangles by adaptive sampling. *CGF*, 19(2):165–171, 2000.
- [UnG18] Carlos Ureña and Iliyan Georgiev. Stratified sampling of projected spherical caps. *CGF*, 37(4):13–20, 2018.
- [VG95] Eric Veach and Leonidas J. Guibas. Optimally combining sampling techniques for monte carlo rendering. In *SIGGRAPH*, pages 419–428, New York, NY, USA, 1995.
- [VG97] Eric Veach and Leonidas J. Guibas. Metropolis light transport. In *SIGGRAPH*, pages 65–76, New York, NY, USA, 1997.
- [ZZ19] Quan Zheng and Matthias Zwicker. Learning to importance sample in primary sample space. *CGF*, 38(2):169–179, 2019.