

Stratified Sampling of 2-Manifolds

James Arvo

State of the Art in Monte Carlo Ray Tracing for Realistic Image Synthesis
SIGGRAPH 2001 Course Notes, volume 29
August 2001

Contents

1	Stratified Sampling of 2-Manifolds	5
1.1	Introduction	5
1.2	A Recipe for Sampling Algorithms	8
1.3	Analytic Area-Preserving Parametrizations	11
1.3.1	Sampling Planar Triangles	12
1.3.2	Sampling the Unit Disk	12
1.3.3	Sampling the Unit Hemisphere	13
1.3.4	Sampling a Phong Lobe	14
1.3.5	Sampling Spherical Triangles	15
1.4	Sampling Projected Spherical Polygons	19
1.4.1	The Cumulative Marginal Distribution	21
1.4.2	The Sampling Algorithm	23
2	Combining Sampling Strategies	27
2.1	Introduction	27
2.2	Using Multiple PDFs	28
2.3	Possible Weighting Functions	30
2.4	Obvious is also Nearly Optimal	32
	References	33

Chapter 1

Stratified Sampling of 2-Manifolds

1.1 Introduction

Monte Carlo techniques arise in image synthesis primarily as a means to solve integration problems. Integration over domains of two or higher dimensions is ubiquitous in image synthesis; indirect global illumination is expressed as an integral over all paths of light, which entails numerous direct illumination problems such as the computation of form factors, visibility, reflected radiance, subsurface scattering, and irradiance due to complex or partially occluded luminaires, all of which involve integration.

The Monte Carlo method stems from a very natural and immediate connection between integration and *expectation*. Every integral, in both deterministic and probabilistic contexts, can be viewed as the expected value (mean) of a random variable; by averaging over many samples of the random variable, we may thereby approximate the integral. However, there are infinitely many random variables that can be associated with any given integral; of course, some are better than others.

One attribute that makes some random variables better than others for the purpose of integration is the ease with which they can be sampled. In general, we tend to construct Monte Carlo methods using only those random variables with convenient and efficient sampling procedures. But there is also a competing attribute. One of the maxims of Monte Carlo integration is that the probability density function of the random variable should mimic the integrand as closely as possible. The closer the match, the smaller the variance of the random variable, and the more reliable (and efficient) the estimator. In the limit, when the samples are generated with a density that is exactly proportional to the (positive) integrand, the variance

of the estimator is identically zero [9]. That is, a single sample delivers the exact answer with probability one.

Perhaps the most common form of integral arising in image synthesis is that expressing either irradiance or reflected radiance at a surface. In both cases, we must evaluate (or approximate) an integral over solid angle, which is of the form

$$\int_S f(\vec{\omega})(\vec{\omega} \cdot \vec{n}) d\omega, \quad (1.1)$$

where S is subset of the unit sphere, $\vec{\omega}$ is a unit direction vector, and \vec{n} is the surface normal vector, or over surface area, which is of the form

$$\int_A f(x) \frac{(x \cdot \vec{n})(x \cdot \vec{n}')}{\|x\|^2} dx, \quad (1.2)$$

where A is a surface and x is a point in \mathbb{R}^3 . Technically, these integrals differ only by a change of variable that results from the pullback of surface differentials to solid angle differentials [1]. The function f may represent the radiance or emissive power of a luminaire (in the case of irradiance) or it may include a BRDF (in the case of reflected radiance). In all cases, visibility may be included in the function f , which can make the integrand arbitrarily discontinuous, thereby drastically reducing the effectiveness of standard numerical quadrature methods for computing the integral.

To apply Monte Carlo integration most effectively in image synthesis, we seek sampling algorithms that match the geometries and known light distributions found in a simulated scene to the extent possible. Consequently, a wide assortment of sampling algorithms have been developed for sampling both the surfaces of and the solid angles subtended by various scene geometries [10] so that both forms of the integral above can be accommodated. In this chapter we will see how to construct random variables for specific geometries: that is, random variables whose range coincides with some bounded region of the plane or some bounded surface in \mathbb{R}^3 , and whose probability density function is constant. For instance, we will see how to generate uniformly distributed samples over both planar and spherical triangles, and projected spherical polygons.

All of the sampling algorithms that we construct are based on mappings from the unit square, $[0, 1] \times [0, 1]$, to the regions or surfaces in question that preserve uniform sampling. That is, uniformly distributed samples in the unit square are mapped to uniformly distributed samples in the range. Such mappings also preserve *stratification*, also known as *jitter sampling* [5], which means that uniform partitionings of the unit square map to uniform partitionings of the range. The ability to apply stratified sampling over various domains is a great advantage, as it is

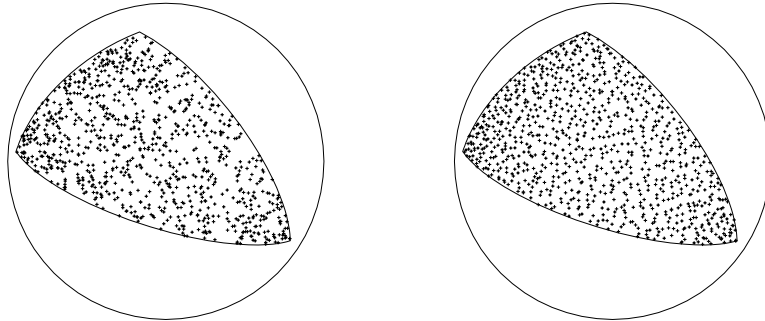


Figure 1.1: A spherical triangle with uniform samples (left) and stratified samples (right). Both sets of samples were generated using an area-preserving parametrization for spherical triangles, which we derive below.

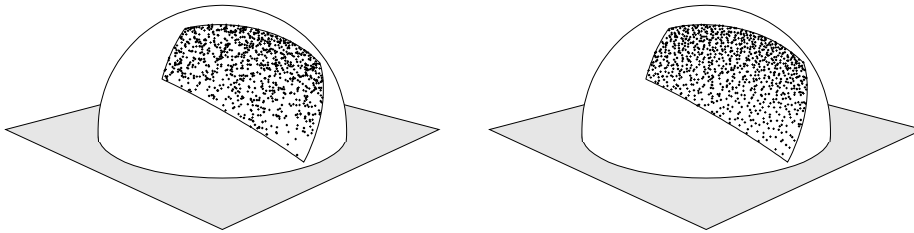


Figure 1.2: A projected spherical polygon with uniform samples (left) and stratified samples (right). Projection onto the plane results in more samples near the north pole of the sphere than near the equator. Both sets of samples were generated using an area-preserving parametrization for spherical polygons, which we derive below.

often a very effective variance reduction technique. Figure 1.1 shows the result of applying such a mapping to a spherical triangle, both with and without stratified (jittered) sampling. Figure 1.2 shows the result of applying such a mapping to a projected spherical polygon, so that the samples on the original spherical polygon are “cosine-distributed” rather than uniformly distributed.

All of the resulting algorithms depend upon a source of uniformly distributed random numbers in the interval $[0, 1]$, which we shall assume is available from some unspecified source: perhaps “drand48,” or some other pseudo-random number generator.

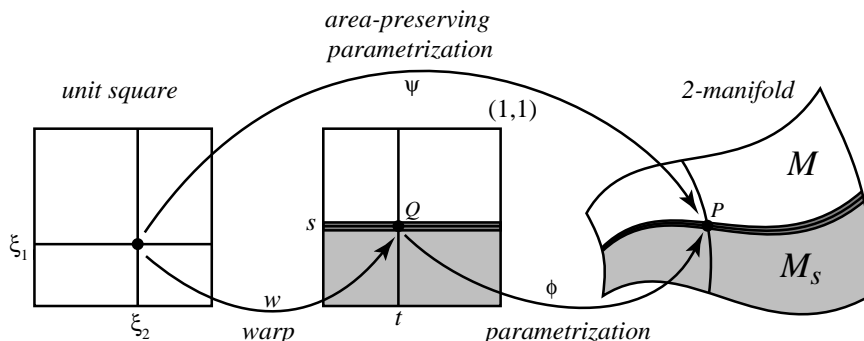


Figure 1.3: An arbitrary parametrization ϕ for a 2-manifold \mathcal{M} can be converted into an area-preserving parametrization, which is useful for uniform and stratified sampling, by composing it with a warping function. The warping function can be derived directly from ϕ by following a precise procedure.

1.2 A Recipe for Sampling Algorithms

Although there is a vast and mature literature on Monte Carlo methods, with many texts describing how to derive sampling algorithms for various geometries and density functions (see, for example, Kalos and Whitlock [6], Spanier and Gelbard [11], or Rubinstein [9]), these treatments do not provide step-by-step instructions for deriving the types of algorithms that we frequently require in computer graphics. In this section we present a detailed “recipe” for how to convert an arbitrary parametrization $\phi : [0, 1]^2 \rightarrow \mathcal{M}$, from the unit square to a 2-manifold, into an *area-preserving* parametrization, $\psi : [0, 1]^2 \rightarrow \mathcal{M}$. That is, a mapping ψ with the property that

$$\mathbf{area}(A) = \mathbf{area}(B) \implies \mathbf{area}(\psi[A]) = \mathbf{area}(\psi[B]), \quad (1.3)$$

for all $A, B \in [0, 1] \times [0, 1]$. Such mappings are used routinely in image synthesis to sample surfaces of luminaires and reflectors. Note that ψ may in fact shrink or magnify areas, but that all areas undergo exactly the same scaling; hence, it is area-preserving in the strictest sense only when $\mathbf{area}(\mathcal{M}) = 1$. A parametrization with this property will allow us to generate uniformly distributed and/or stratified samples over \mathcal{M} by generating samples with the desired properties on the unit square (which is trivial) and then mapping them onto \mathcal{M} . We shall henceforth consider area-preserving parametrizations to be synonymous with *sampling algorithms*.

Let \mathcal{M} represent a shape that we wish to generate uniformly distributed samples on; in particular, \mathcal{M} may be any 2-manifold with boundary in \mathbb{R}^n , where n is typically 2 or 3. The steps for deriving a sampling algorithm for \mathcal{M} are summa-

rized in Figure 1.4. These steps apply for all dimensions $n \geq 2$; that is, \mathcal{M} may be a 2-manifold in any space.

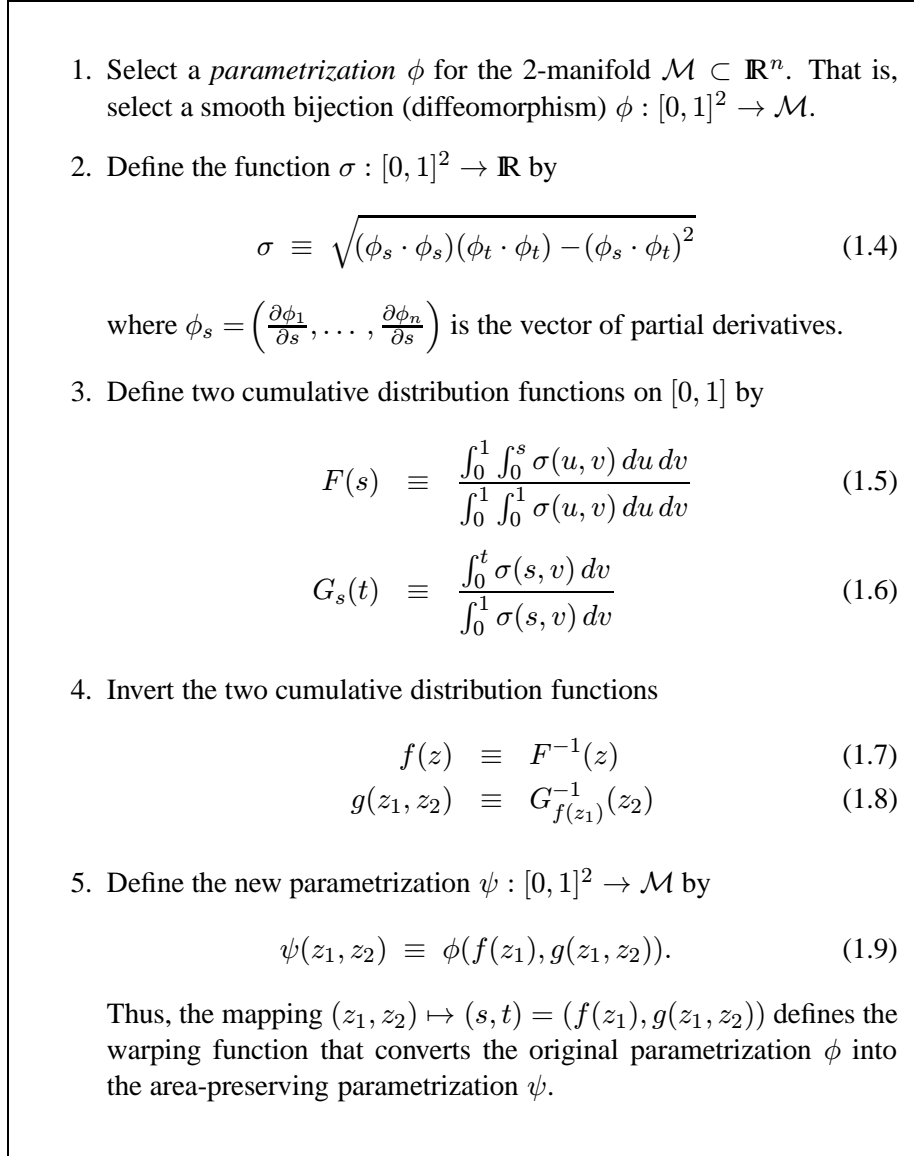


Figure 1.4: Five steps for deriving an area-preserving parametrization ψ from $[0, 1] \times [0, 1]$ to any bounded 2-manifold $\mathcal{M} \subset \mathbb{R}^n$, beginning from an arbitrary parametrization ϕ for \mathcal{M} . The function ψ is suitable for stratified sampling of \mathcal{M} . Step 2 simplifies in the common two- and three- dimensional cases. Step 4 is often the only impediment to finding a closed-form expression for the area-preserving parametrization.

Step 1 requires that we select a smooth bijection ϕ from $[0, 1] \times [0, 1]$ to the 2-manifold \mathcal{M} . Such a function is referred to as a *parametrization* and its inverse is called a *coordinate chart*, as it associates unique 2D coordinates with (almost) all points of \mathcal{M} . In reality, we only require ϕ to be a bijection *almost everywhere*; that is, on all but a set of measure zero, such as the boundaries of \mathcal{M} or $[0, 1] \times [0, 1]$. In theory, any smooth bijection will suffice, although it may be impractical or impossible to perform some of the subsequent steps in Figure 1.4 symbolically (particularly step 4) for all but the simplest functions.

Step 2 defines a function $\sigma : [0, 1]^2 \rightarrow \mathbb{R}$ that links the parametrization ϕ to the notion of surface area on \mathcal{M} . More precisely, for any region $A \subseteq [0, 1]^2$, the function σ satisfies

$$\int_A \sigma = \mathbf{area}(\phi[A]). \quad (1.10)$$

That is, the integral of σ over any region A in the 2D parameter space is the surface area of the corresponding subset of \mathcal{M} under the mapping ϕ . Equation (1.4) holds for all $n \geq 2$. For the typical cases of $n = 2$ and $n = 3$, however, the function σ can be expressed more simply. For example, if \mathcal{M} is a subset of \mathbb{R}^2 then

$$\sigma(s, t) \equiv \det(D_{(s,t)}\phi), \quad (1.11)$$

where $D_{(s,t)}\phi$ is 2×2 Jacobian matrix of ϕ at the point (s, t) . On the other hand, if \mathcal{M} is a subset of \mathbb{R}^3 , then

$$\sigma(s, t) \equiv \|\phi_s(s, t) \times \phi_t(s, t)\|, \quad (1.12)$$

which is a convenient abbreviation for equation (1.4) that holds only when $n = 3$, as the two partial derivatives of ϕ are vectors in \mathbb{R}^3 in this case. Non-uniform sampling can also be accommodated by including a weighting function in the definition of σ in step 2.

Step 3 can often be carried out without the aid of an explicit expression for σ . For example, the cumulative distributions can often be found by reasoning directly about the geometry imposed by the parametrization rather than applying formulas (1.4), (1.5) and (1.6), which can be tedious. Let \mathcal{M}_s denote the family of sub-manifolds of \mathcal{M} defined by the first coordinate of ϕ . That is,

$$\mathcal{M}_s = \phi \left[[0, s] \times [0, 1] \right]. \quad (1.13)$$

See Figure 1.3. It follows from the definition of F and equation (1.10) that

$$F(s) = \frac{\mathbf{area}(\mathcal{M}_s)}{\mathbf{area}(\mathcal{M})}, \quad (1.14)$$

which merely requires that we find an expression for the surface area of \mathcal{M}_s as a function of s . Similarly, by equation (1.7) we have

$$s = f\left(\frac{\text{area}(\mathcal{M}_s)}{\text{area}(\mathcal{M})}\right). \quad (1.15)$$

Thus, f is the map that recovers the parameter s from the fractional area of the sub-manifold \mathcal{M}_s . Equation (1.15) can be more convenient to work with than equation (1.14), as it avoids an explicit function inversion step. While $G_s(\cdot)$ and $g(\cdot, \cdot)$ do not admit equally intuitive interpretations, they can often be determined from the general form of σ , since many of the details vanish due to normalization. A good example of how this can be done is provided by the area-preserving parametrization derived for spherical triangles, which we discuss below.

Step 4 above is the only step that is not purely mechanical, as it involves function inversion. When this step can be carried out symbolically, the end result is a closed-form area-preserving transformation ψ from $[0, 1]^2$ to the manifold \mathcal{M} . Closed-form expressions are usually advantageous, both in terms of simplicity and efficiency. Of the two inversions entailed in step 4, it is typically equation (1.5) that is the more troublesome, and frequently resists symbolic solution. In such a case, it is always possible to perform the inversion numerically using a root-finding method such as Newton’s method; of course, one must always weigh the cost of drawing samples against the benefits conferred by the resulting importance sampling and stratification. When numerical inversion is involved, the area-preserving transformation is less likely to result in a net gain in efficiency.

The steps outlined in Figure 1.4 generalize very naturally to the construction of volume-preserving parametrizations for arbitrary k -manifolds. For any $2 \leq k \leq n$, step 3 entails a sequence of k cumulative distribution functions, each dependent upon all of its predecessors, and step 4 requires the cascaded inversion of all k distributions, in the order of their definition. Step 5 entails a k -way function composition. In the remainder of these notes, we will consider only the case where $k = 2$ and $n \in \{2, 3\}$; that is, we will only consider the problem of generating samples over 2-manifolds (surfaces) in \mathbb{R}^2 or \mathbb{R}^3 .

1.3 Analytic Area-Preserving Parametrizations

In this section we will apply the “recipe” given in Figure 1.4 to derive a number of useful area-preserving parametrizations. Each will be expressed in closed form since the functions F and G_s will be invertible symbolically; however, in the case of spherical triangles it will not be trivial to invert F .

1.3.1 Sampling Planar Triangles

As a first example of applying the steps in Figure 1.4, we shall derive an area-preserving parametrization for an arbitrary triangle ABC in the plane. We begin with an obvious parametrization from $[0, 1] \times [0, 1]$ to a given triangle in terms of barycentric coordinates. That is, let

$$\phi(s, t) = (1 - s)A + s(1 - t)B + stC. \quad (1.16)$$

It is easy to see that ϕ is a smooth mapping that is bijective except when $t = 0$, which is a set of measure zero. Since the codomain of ϕ is \mathbb{R}^2 , σ is simply the Jacobian of ϕ . After a somewhat tedious computation, we obtain

$$\det(D\phi) = 2cs, \quad (1.17)$$

where c is the area of the triangle. From equations (1.5) and (1.6) we obtain

$$F(s) = s^2 \quad \text{and} \quad G_s(t) = t. \quad (1.18)$$

In both cases the constant c disappears due to normalization. These functions are trivial to invert, resulting in

$$f(z) = \sqrt{z} \quad \text{and} \quad g(z_1, z_2) = z_2. \quad (1.19)$$

Finally, after function composition, we have

$$\begin{aligned} \psi(z_1, z_2) &= \phi(f(z_1), g(z_1, z_2)) \\ &= (1 - \sqrt{z_1})A + \sqrt{z_1}(1 - z_2)B + \sqrt{z_1}z_2C. \end{aligned} \quad (1.20)$$

Figure 1.5 shows the final algorithm. If ξ_1 and ξ_2 are independent random variables, uniformly distributed over the interval $[0, 1]$, then the resulting points will be uniformly distributed over the triangle.

1.3.2 Sampling the Unit Disk

Next, we derive an area-preserving parametrization for a unit-radius disk D in the plane, centered at the origin. We start with the parametrization from $[0, 1] \times [0, 1]$ to D given by

$$\phi(s, t) = s [\cos(2\pi t)\mathbf{x} + \sin(2\pi t)\mathbf{y}], \quad (1.21)$$

where \mathbf{x} and \mathbf{y} are the orthogonal unit vectors in the plane. Again, ϕ is a smooth mapping that is bijective except when $s = 0$. Computing the Jacobian of ϕ , we obtain

$$\det(D\phi) = 2\pi s. \quad (1.22)$$

```

SamplePlanarTriangle( real  $\xi_1$ , real  $\xi_2$  )
  Compute the warping function  $(\xi_1, \xi_2) \mapsto (s, t)$ .
   $s \leftarrow \sqrt{\xi_1}$ ;
   $t \leftarrow \xi_2$ ;
  Plug the warped coords into the original parametrization.
   $\mathbf{P} \leftarrow (1 - s)A + s(1 - t)B + stC$ ;
  return  $\mathbf{P}$ ;
end

```

Figure 1.5: Algorithm for computing an area-preserving parametrization of the triangle with vertices A , B , and C . This mapping can be used for uniform or stratified sampling.

The remaining steps proceed precisely as in the case of the planar triangle; in fact, the distributions F and G turn out to be identical. Thus, we obtain

$$\begin{aligned}
 \psi(z_1, z_2) &= \phi(\sqrt{z_1}, z_2) \\
 &= \sqrt{\xi_1} [\cos(2\pi\xi_2)\mathbf{x} + \sin(2\pi\xi_2)\mathbf{y}].
 \end{aligned} \tag{1.23}$$

The resulting algorithm for sampling the unit disk is exactly analogous to the algorithm shown in Figure 1.5 for sampling planar triangles.

1.3.3 Sampling the Unit Hemisphere

As a first example of applying the steps in Figure 1.4 to a surface in \mathbb{R}^3 , we shall derive the well-known area-preserving parametrization for the unit-radius hemisphere centered at the origin. First, we define a parametrization using spherical coordinates:

$$\phi(s, t) = \begin{bmatrix} \sin\left(\frac{\pi s}{2}\right) \cos(2\pi t) \\ \sin\left(\frac{\pi s}{2}\right) \sin(2\pi t) \\ \cos\left(\frac{\pi s}{2}\right) \end{bmatrix}. \tag{1.24}$$

Here the parameter s defines the polar angle and t defines the azimuthal angle. Since the codomain of ϕ is \mathbb{R}^3 , we can apply equation (1.12). Since

$$\phi_s(s, t) \times \phi_t(s, t) = \pi^2 \begin{bmatrix} \cos\left(\frac{\pi s}{2}\right) \cos(2\pi t) \\ \cos\left(\frac{\pi s}{2}\right) \sin(2\pi t) \\ -\sin\left(\frac{\pi s}{2}\right) \end{bmatrix} \times \begin{bmatrix} -\sin\left(\frac{\pi s}{2}\right) \sin(2\pi t) \\ \sin\left(\frac{\pi s}{2}\right) \cos(2\pi t) \\ 0 \end{bmatrix},$$

we obtain

$$\begin{aligned}\sigma(s, t) &= \|\phi_s(s, t) \times \phi_t(s, t)\| \\ &= \pi^2 \sin\left(\frac{\pi s}{2}\right).\end{aligned}\tag{1.25}$$

It then follows easily that

$$F(s) = 1 - \cos\left(\frac{\pi s}{2}\right) \quad \text{and} \quad G_s(t) = t,$$

which are trivial to invert, resulting in

$$f(z) = \frac{2 \cos^{-1}(1 - z)}{\pi} \quad \text{and} \quad g(z_1, z_2) = z_2.$$

Composing f and g with ϕ results in

$$\psi(z_1, z_2) = \begin{bmatrix} \sqrt{z_1(2 - z_1)} \cos(2\pi z_2) \\ \sqrt{z_1(2 - z_1)} \sin(2\pi z_2) \\ 1 - z_1 \end{bmatrix}.\tag{1.26}$$

Here the s coordinate of the parametrization simply selects the z -plane from $z = 1$ and $z = 0$, while the t coordinate parameterizes the resulting circle in the z -plane. The form of ψ can be simplified somewhat by substituting $1 - z_1$ for z_1 , which does not alter the distribution.

1.3.4 Sampling a Phong Lobe

Now suppose that we wish to sample the hemisphere according to a Phong distribution rather than uniformly; that is, with a density proportional to the cosine of the polar angle to a power. To do this we simply include a weighting function in the definition of σ given in equation (1.25). That is, we let

$$\sigma(s, t) = \pi^2 \sin\left(\frac{\pi s}{2}\right) \cos^k\left(\frac{\pi s}{2}\right),\tag{1.27}$$

where k is the Phong exponent. It follows that

$$F(s) = \cos^{k+1}\left(\frac{\pi s}{2}\right) \quad \text{and} \quad G_s(t) = t,$$

which implies that

$$f(z) = \frac{2}{\pi} \cos^{-1} z^{\frac{1}{k+1}} \quad \text{and} \quad g(z_1, z_2) = z_2.$$

It follows that

$$\psi(z_1, z_2) = \begin{bmatrix} \sqrt{1 - z_1^{\frac{2}{k+1}}} \cos(2\pi z_2) \\ \sqrt{1 - z_1^{\frac{2}{k+1}}} \sin(2\pi z_2) \\ z_1^{\frac{1}{k+1}} \end{bmatrix}. \quad (1.28)$$

1.3.5 Sampling Spherical Triangles

We shall now derive an area-preserving parametrization for an arbitrary spherical triangle, which is significantly more challenging than the cases we've considered thus far. Let T denote the spherical triangle with vertices \mathbf{A} , \mathbf{B} , and \mathbf{C} , as shown in Figure 1.6. Such a triangle can be parameterized by using the first coordinate s to select the edge length b_s , which in turn defines sub-triangle $T_s \subset T$, and the second coordinate t to select a point along the edge \mathbf{BC}_s , as shown in Figure 1.6. This parameterization can be expressed as

$$\phi(s, t) = \text{slerp}(\mathbf{B}, \text{slerp}(\mathbf{A}, \mathbf{C}, s), t), \quad (1.29)$$

where $\text{slerp}(\mathbf{A}, \mathbf{C}, s)$ is the *spherical linear interpolation* function that generates points along the great arc connecting \mathbf{A} and \mathbf{C} (according to arc length) as s varies from 0 to 1. The slerp function can be defined as

$$\text{slerp}(\mathbf{x}, \mathbf{y}, s) = \mathbf{x} \cos(\theta s) + [\mathbf{y} | \mathbf{x}] \sin(\theta s), \quad (1.30)$$

where $\theta = \cos^{-1} \mathbf{x} \cdot \mathbf{y}$, and $[\mathbf{y} | \mathbf{x}]$ denotes the normalized component of the vector \mathbf{y} that is orthogonal to the vector \mathbf{x} ; that is

$$[\mathbf{y} | \mathbf{x}] \equiv \frac{(\mathbf{I} - \mathbf{x}\mathbf{x}^T) \mathbf{y}}{\|(\mathbf{I} - \mathbf{x}\mathbf{x}^T) \mathbf{y}\|}, \quad (1.31)$$

where \mathbf{x} is assumed to be a unit vector. From the above definition of ϕ it is now possible to derive the function σ using equation (1.12). We find that σ is of the form

$$\sigma(s, t) = h(s) \sin(a_s t) \quad (1.32)$$

for some function h , where a_s is the length of the moving edge \mathbf{BC}_s as a function of s . The exact nature of h is irrelevant, however, as it will not be needed to compute F , and it is eliminated from G_s by normalization. Thus, we have

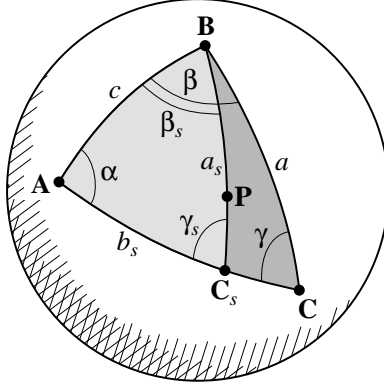


Figure 1.6: Parameter s controls the edge length b_s , which determines the vertex C_s , and consequently sub-triangle T_s . Parameter t then selects a point P along the arc between C_s and B . Not shown is the length of the arc AC , which is b .

$$F(s) = \frac{\text{area}(T_s)}{\text{area}(T)} \quad (1.33)$$

$$G_s(t) = \frac{1 - \cos(a_s t)}{1 - \cos a_s}. \quad (1.34)$$

It follows immediately from inversion of equation (1.34) that

$$g(z_1, z_2) = \frac{1}{a_{z_1}} \cos^{-1} \left[1 - z_2(1 - \cos a_{z_1}) \right]. \quad (1.35)$$

However, solving for f , which is the inverse of F , is not nearly as straightforward. Our approach will be to derive an expression for the function f directly, using equation (1.15), rather than starting from F and inverting it. To do this, we require several elementary identities from spherical trigonometry. Let \mathcal{A} denote the surface area of the spherical triangle T with vertices A , B and C . Let a , b , and c denote the edge lengths of T ,

$$\begin{aligned} a &= \cos^{-1} \mathbf{B} \cdot \mathbf{C}, \\ b &= \cos^{-1} \mathbf{A} \cdot \mathbf{C}, \\ c &= \cos^{-1} \mathbf{A} \cdot \mathbf{B}, \end{aligned}$$

and let α , β , and γ denote the three internal angles, which are the dihedral angles between the planes containing the edges. See Figure 1.6. Listed below are a few

well-known identities for spherical triangles:

$$\mathcal{A} = \alpha + \beta + \gamma - \pi \quad (1.36)$$

$$\frac{\sin \alpha}{\sin a} = \frac{\sin \beta}{\sin b} = \frac{\sin \gamma}{\sin c} \quad (1.37)$$

$$\cos \alpha = -\cos \beta \cos \gamma + \sin \beta \sin \gamma \cos a \quad (1.38)$$

$$\cos \beta = -\cos \gamma \cos \alpha + \sin \gamma \sin \alpha \cos b \quad (1.39)$$

$$\cos \gamma = -\cos \beta \cos \alpha + \sin \beta \sin \alpha \cos c \quad (1.40)$$

Each of these identities will be employed in deriving area-preserving parametrizations, either for spherical triangles or projected spherical polygons, which will be described in the following section. Equation (1.36) is known as Girard's formula, equation (1.37) is the spherical law of sines, and equations (1.38), (1.39), and (1.40) are spherical cosine laws for angles [4].

Our task will be to construct $f : [0, 1] \rightarrow \mathbb{R}$ such that $f(\mathcal{A}_s/\mathcal{A}) = s$, where the parameter $s \in [0, 1]$ selects the sub-triangle T_s and consequently determines the area \mathcal{A}_s . Specifically, the sub-triangle T_s is formed by choosing a new vertex \mathbf{C}_s on the great arc between \mathbf{A} and \mathbf{C} , at an arc length of $b_s = sb$ along the arc from \mathbf{A} , as shown in Figure 1.6. The point \mathbf{P} is finally chosen on the arc between \mathbf{B} and \mathbf{C}_s , according to the parameter t .

To find the parameter s that corresponds to the fractional area $\mathcal{A}_s/\mathcal{A}$, we first solve for $\cos b_s$ in terms of \mathcal{A}_s and various constants associated with the triangle. From equations (1.36) and (1.39) we have

$$\begin{aligned} \cos b_s &= \frac{\cos \gamma_s \cos \alpha + \cos \beta_s}{\sin \gamma_s \sin \alpha} \\ &= \frac{-\cos(\mathcal{A}_s - \alpha - \beta_s) \cos \alpha + \cos \beta_s}{-\sin(\mathcal{A}_s - \alpha - \beta_s) \sin \alpha} \\ &= \frac{\cos(\Delta - \beta_s) \cos \alpha - \cos \beta_s}{\sin(\Delta - \beta_s) \sin \alpha}, \end{aligned} \quad (1.41)$$

where we have introduced $\Delta \equiv \mathcal{A}_s - \alpha$. We now eliminate β_s to obtain a function that depends only on area and the fixed parameters: in particular, we shall construct a function of only Δ , α , and c . We accomplish this by using spherical trigonometry to find expressions for both $\sin \beta_s$ and $\cos \beta_s$. From equation (1.36) and plane trigonometry it follows that

$$\cos \gamma_s = -\cos(\Delta - \beta_s) = \sin \Delta \sin \beta_s - \cos \Delta \cos \beta_s. \quad (1.42)$$

Combining equation (1.42) with equation (1.40) we have

$$(\cos \Delta - \cos \alpha) \cos \beta_s + (\sin \Delta + \sin \alpha \cos c) \sin \beta_s = 0. \quad (1.43)$$

SampleSphericalTriangle(real ξ_1 , real ξ_2)

Use one random variable to select the new area.

$\mathcal{A}_s \leftarrow \xi_1 \mathcal{A}$;

Save the sine and cosine of the angle Δ .

$p \leftarrow \sin(\mathcal{A}_s - \alpha)$;

$q \leftarrow \cos(\mathcal{A}_s - \alpha)$;

Compute the pair (u, v) that determines $\sin \beta_s$ and $\cos \beta_s$.

$u \leftarrow q - \cos \alpha$;

$v \leftarrow p + \sin \alpha \cos c$;

Compute the s coordinate as normalized arc length from \mathbf{A} to \mathbf{C}_s .

$s \leftarrow \frac{1}{b} \cos^{-1} \left[\frac{(v q - u p) \cos \alpha - v}{(v p + u q) \sin \alpha} \right]$;

Compute the third vertex of the sub-triangle.

$\mathbf{C}_s \leftarrow \text{slerp}(\mathbf{A}, \mathbf{C}, s)$;

Compute the t coordinate using \mathbf{C}_s and ξ_2 .

$t \leftarrow \frac{\cos^{-1} \left[1 - \xi_2 (1 - \mathbf{C}_s \cdot \mathbf{B}) \right]}{\cos^{-1} \mathbf{C}_s \cdot \mathbf{B}}$;

Construct the corresponding point on the sphere.

$\mathbf{P} \leftarrow \text{slerp}(\mathbf{B}, \mathbf{C}_s, t)$;

return P;

end

Figure 1.7: An area-preserving parametrization for an arbitrary spherical triangle \mathbf{ABC} . This procedure can be easily optimized to remove the inverse cosines used to compute the warped coordinates s and t , since the **slerp** function uses the cosine of its scalar argument.

Consequently, $\sin \beta_s = -ru$ and $\cos \beta_s = rv$ where

$$u \equiv \cos \Delta - \cos \alpha,$$

$$v \equiv \sin \Delta + \sin \alpha \cos c,$$

and r is a common factor that cancels out in our final expression, so it is irrelevant. Simplifying equation (1.41) using these new expressions for $\sin \beta_s$ and $\cos \beta_s$, we obtain an expression for $\cos b_s$ in terms of Δ , u , v , and α . It then follows that

$$s = \frac{1}{b} \cos^{-1} \left[\frac{(v \cos \Delta - u \sin \Delta) \cos \alpha - v}{(v \sin \Delta + u \cos \Delta) \sin \alpha} \right], \quad (1.44)$$

since $s = b_s/b$. Note that $\cos b_s$ determines b_s , since $0 < b_s < \pi$, and that b_s in turn determines the vertex \mathbf{C}_s . The algorithm shown in Figure 1.7 computes an area-preserving map from the unit square onto the triangle T ; it takes two variables ξ_1 and ξ_2 , each in the unit interval, and returns a point $\mathbf{P} \in T \subset \mathbb{R}^3$. If ξ_1 and ξ_2 are uniformly distributed random variables in $[0, 1]$, the algorithm will produce a random variable \mathbf{P} that is uniformly distributed over the surface of the spherical triangle T .

The procedure in Figure 1.7 explicitly warps the coordinates (ξ_1, ξ_2) into the coordinates (s, t) in such a way that the resulting parametrization is area-preserving. If implemented exactly as shown, the procedure performs a significant amount of unnecessary computation. Most significantly, all of the inverse cosines can be eliminated by substituting the equation (1.30) for the `slerp` function and then simplifying [3]. Also, $\cos \alpha$, $\sin \alpha$, $\cos c$, and $[\mathbf{C} | \mathbf{A}]$, which appears in the expression for `slerp`($\mathbf{A}, \mathbf{C}, s$), need only be computed once per triangle rather than once per sample.

Results of the algorithm are shown in Figure 1.1. On the left, the samples are identically distributed, which produces a pattern equivalent to that obtained by rejection sampling; however, each sample is guaranteed to fall within the triangle. The pattern on the right was generated by partitioning the unit square into a regular grid and choosing one pair (ξ_1, ξ_2) uniformly from each grid cell, which corresponds to stratified sampling. The advantage of stratified sampling is evident in the resulting pattern; the samples are more evenly distributed, which generally reduces the variance of Monte Carlo estimates based on these samples. The sampling algorithm can be applied to spherical polygons by decomposing them into triangles and performing stratified sampling on each component independently, which is analogous to the method for planar polygons described by Turk [12]. This is one means of sampling the solid angle subtended by a polygon. We discuss another approach in the following section.

1.4 Sampling Projected Spherical Polygons

In this section we will see an example in which the inversion of the F function can not be done symbolically; consequently, we will resort to either approximate inversion, or inversion via a root finder.

The dot product $\vec{\omega} \cdot \vec{n}$ appearing in equation (1.1) is the ubiquitous ‘‘cosine’’ factor that appears in nearly every illumination integral. Since it is often infeasible to construct a random variable that mimics the full integrand, we settle for absorbing the cosine term into the sampling distribution; this compromise is a useful special

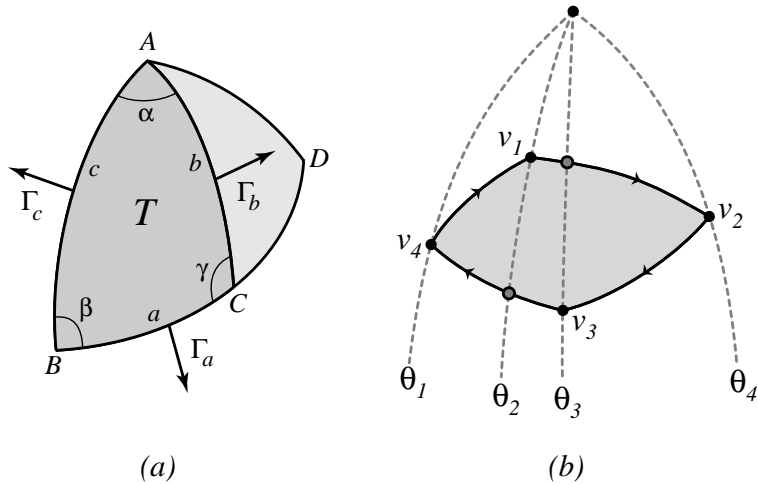


Figure 1.8: (a) Spherical triangle T . We consider the projected area of triangle T as a function of α , keeping vertices A and B , and angle β fixed. (b) Partitioning a spherical polygon by great circles passing through the poles and the vertices.

case of *importance sampling*. In this section we address the problem of generating stratified samples over the solid angle subtended by arbitrary polygons, while taking the cosine weighting into account, as shown in Figure 1.2. The combination of stratification and importance sampling, even in this relatively weak form, can significantly reduce the variance of the associated Monte Carlo estimator [3, 9].

We now describe a new technique for Monte Carlo sampling of spherical polygons with a density proportional to the cosine from a given axis which, by Nusselt's analogy, is equivalent to uniformly sampling the projection of the spherical polygon onto the $z = 0$ plane. The technique handles polygons directly, without first partitioning them into triangles, and is ideally suited for stratified sampling. The Jacobian of the bijection from the unit square to the polygon can be made arbitrarily close to the cosine density, making the statistical bias as close to zero as desired. After preprocessing a polygon with n vertices, which can be done in $O(n^2 \log n)$ time, each sample can be generated in $O(n)$ time.

Let \mathcal{P} denote a spherical polygon. To help in defining the mapping $\psi : [0, 1]^2 \rightarrow \mathcal{P}$, we first derive several basic expressions that pertain to spherical triangles. Let T be a spherical triangle, and consider the family of sub-triangles shown in Figure 1.8a, where the unit vectors A and B and the internal angle β are all fixed, but the internal angle α is allowed to sweep from 0 to the last vertex. Our task in this section is to express a (which is the length of edge BC) and $\cos b$ as functions of

α . From equation (1.38) we have

$$\cos a = \frac{\cos \alpha + \cos \beta \cos \gamma}{\sin \beta \sin \gamma}. \quad (1.45)$$

Let Γ_a , Γ_b , and Γ_c denote the outward unit normals for each edges of the triangle, as shown in Figure 1.8a. Then $\cos \gamma = -\Gamma_a \cdot \Gamma_b$, where Γ_b can be expressed as

$$\Gamma_b = -\Gamma_c \cos \alpha + (\Gamma_c \times A) \sin \alpha. \quad (1.46)$$

Also, from equation (1.37) we have $\sin \gamma = \sin c \sin \alpha / \sin a$. Therefore,

$$a(\alpha) = \tan^{-1} \left(\frac{\sin \alpha}{c_1 \cos \alpha - c_2 \sin \alpha} \right), \quad (1.47)$$

where the constants c_1 and c_2 are given by

$$c_1 = \frac{(\Gamma_a \cdot \Gamma_c) \cos \beta + 1}{\sin \beta \sin c}, \quad c_2 = \frac{(\Gamma_a \cdot \Gamma_c \times A) \cos \beta}{\sin \beta \sin c}. \quad (1.48)$$

These constants depend only on the fixed features of the triangle, as the vectors Γ_a and Γ_c do not depend on α . It is now straightforward to find $\cos b$ as a function of α , which we shall denote by $z(\alpha)$. Specifically,

$$z(\alpha) = (B \cdot \mathbf{N}) \cos a(\alpha) + (D \cdot \mathbf{N}) \sin a(\alpha), \quad (1.49)$$

where D is a point on the sphere that is orthogonal to B , and on the great circle through B and C . That is,

$$D = (\mathbf{I} - BB^T)C. \quad (1.50)$$

We now show how to sample an arbitrary spherical polygon according to a cosine distribution. The function $a(\alpha)$ will be used to invert a cumulative marginal distribution over the polygon, as a great arc sweeps across the polygon, while $z(\alpha)$ will be used to sample one-dimensional vertical slices of the polygon.

1.4.1 The Cumulative Marginal Distribution

We break the problem of computing the bijection $\psi : [0, 1]^2 \rightarrow \mathcal{P}$ into two parts. First, we define a sequence of sub-polygons of \mathcal{P} in much the same way that we parameterized the triangle T above; that is, we define $\mathcal{P}(\theta)$ to be the intersection of \mathcal{P} with a lune¹ whose internal angle is θ , and with one edge passing through an

¹A *lune* is a spherical triangle with exactly two vertices, which are antipodal.

extremal vertex of \mathcal{P} . Next we define a *cumulative marginal distribution* $F(\theta)$ that gives the area of polygon $\mathcal{P}(\theta)$ projected onto the plane orthogonal to \mathbf{N} , which is simply the cosine-weighted area of \mathcal{P} . Then F is a strictly monotonically increasing function of θ . By inverting this function we arrive at the first component of our sampling algorithm. That is, if ξ_1 is a uniformly distributed random variable in $[0, 1]$, and if $\hat{\theta}$ is given by

$$\hat{\theta} = F^{-1}(\rho \xi_1), \quad (1.51)$$

then $\hat{\theta}$ defines the great circle from which to draw a sample.

To find F , we first consider the spherical triangle T and its family of sub-triangles. The projected area of the triangle T , which we denote by ρ , follows immediately from Lambert's formula for computing the irradiance from a polygonal luminaire [2]. That is

$$\rho = -(a\Gamma_a + b\Gamma_b + c\Gamma_c) \cdot \mathbf{N}, \quad (1.52)$$

where Γ_a , Γ_b , and Γ_c are outward normals of the triangle T , as shown in Figure 1.8a. If we now constrain T to be a *polar triangle*, with vertex A at the pole of the hemisphere ($A = \mathbf{N}$), then ρ becomes a very simple function of α . Specifically,

$$\rho(\alpha) = -a(\alpha) (\Gamma_a \cdot \mathbf{N}), \quad (1.53)$$

where $\Gamma_a \cdot \mathbf{N}$ is fixed; this follows from the fact that both Γ_b and Γ_c are orthogonal to \mathbf{N} . Equation (1.53) allows us to easily compute the function $F(\alpha)$ for any collection of spherical polygons whose vertices all lie on the lune with vertices at A and $-A$ as shown in Figure 1.10, where we restrict our attention to the *positive* or upper half of the lune. Thus,

$$F(\theta) = \sum_{i=1}^k \eta_i a_i (\theta - \theta_k), \quad (1.54)$$

for $\theta \in [\theta_k, \theta_{k+1}]$, where the constants $\eta_1, \eta_2, \dots, \eta_k$ account for the slope and orientation of the edges; that is, edges that result in clockwise polar triangles are positive, while those forming counter-clockwise triangles are negative.

We now extend $F(\theta)$ to a general spherical polygon \mathcal{P} by slicing \mathcal{P} into lunes with the above property; that is, we partition \mathcal{P} into smaller polygons by passing a great arc through each vertex, as shown in Figure 1.8b. Then for any spherical polygon, we can evaluate $F(\theta)$ exactly for any value of θ by virtue of equation (1.47). The resulting function F is a piecewise-continuous strictly monotonically increasing function with at most $n - 2$ discontinuities, where n is the number

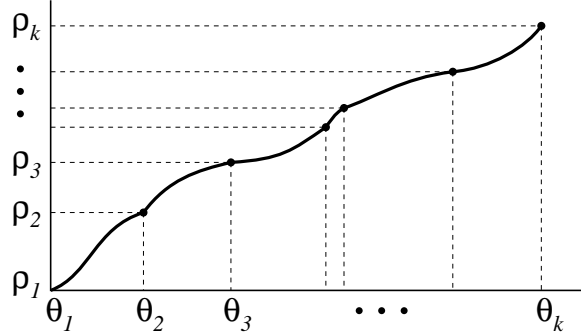


Figure 1.9: The cumulative marginal distribution function F as a function of the angle θ . At each value of θ_i , the abscissa is the form factor from the origin to the polygon that is within the range $[\theta_1, \theta_i]$. This function is strictly monotonically increasing, with at most $n - 2$ derivative discontinuities, where n is the number of vertices in the polygon. The fluctuations in F have been greatly exaggerated for purposes of illustration.

of vertices in the polygon. See Figure 1.9. This function is precisely the *cumulative marginal distribution function* that we must invert to perform the first stage of cosine-weighted sampling. Because it is monotonically increasing, its inverse is well-defined.

1.4.2 The Sampling Algorithm

Given two variables ξ_1 and ξ_2 in the interval $[0, 1]$ we will compute the corresponding point $\mathbf{P} = \psi(\xi_1, \xi_2)$ in the polygon \mathcal{P} . We use ξ_1 to determine an angle $\hat{\theta}$, as described above, and ξ_2 to select the height \hat{z} according to the resulting *conditional density* defined along the intersection of the polygon \mathcal{P} and the great circle at $\hat{\theta}$.

To compute $\hat{\theta}$ using equation (1.51), we proceed in two steps. First, we find the lune from which $\hat{\theta}$ will be drawn. This corresponds to finding the integer k such that

$$\frac{\rho_k}{\rho_{tot}} \leq \xi_1 \leq \frac{\rho_{k+1}}{\rho_{tot}}. \quad (1.55)$$

Next, we must invert F as it is defined on this interval. Given the nature of F , as defined in equations (1.47) and (1.54), it is unlikely that this can be done symbolically in general, so we seek a numerical approximation. This is the *only* step in the algorithm which is not computed exactly; thus, any bias that is introduced in the sampling is a result of this step alone.

Approximate numerical inversion is greatly simplified by the nature of F within each lune. Since F is extremely smooth and strictly monotonic, we can approximate F^{-1} directly to high accuracy with a low-order polynomial. For example, we

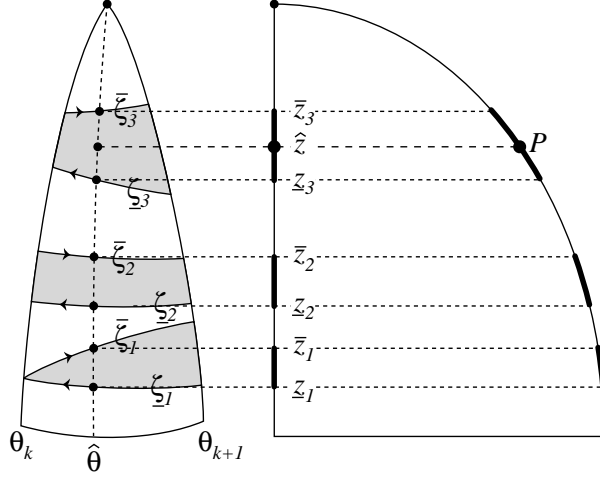


Figure 1.10: On the left is an illustration of a single lune with a collection of arcs passing through it, and the points at which a great circle at $\hat{\theta}$ intersects them. On the right is a cross-section of the circle, showing the heights z_1, \bar{z}_1 , corresponding to these intersection points.

may use

$$F^{-1}(x) \approx a + bx + cx^2 + dx^3, \quad (1.56)$$

where we set

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = V^{-1} \begin{bmatrix} \theta_k \\ \theta_k + \delta_1 \\ \theta_k + \delta_2 \\ \theta_{k+1} \end{bmatrix}. \quad (1.57)$$

Here V is the Vandermonde matrix formed from $F(\theta_k)$, $F(\theta_k + \delta_1)$, $F(\theta_k + \delta_2)$, and $F(\theta_{k+1})$. Coupling this approximation with a Newton iteration can, of course, compute the function inverse to any desired numerical accuracy, making the bias effectively zero. However, such a high degree of accuracy is not warranted for a typical Monte Carlo simulation.

Once the angle $\hat{\theta}$ has been computed using ξ_1 , we then compute \hat{z} using ξ_2 . This corresponds to sampling \hat{z} according to the *conditional density function* corresponding to the choice of $\hat{\theta}$. This conditional density function is defined on the intervals

$$[z_1, \bar{z}_1] \cup [z_2, \bar{z}_2] \cup \cdots \cup [z_n, \bar{z}_n],$$

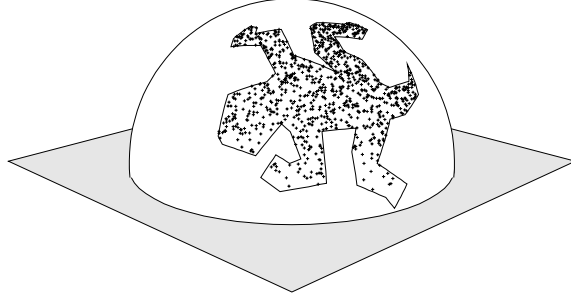


Figure 1.11: A non-convex spherical polygon with cosine-weighted samples generated with the proposed mapping.

which correspond to the intersection points shown in Figure 1.10. These intervals are computed using equation (1.49). The conditional density is proportional to z^2 within these intervals, which distributes the samples vertically according to the cosine of the angle from the pole. The most costly part of sampling according to this density is normalization. We define

$$Z_j \equiv \sum_{i=1}^j (\bar{z}_i^2 - z_i^2). \quad (1.58)$$

Then Z_n is the normalization constant. The random variable ξ_2 then selects the interval by finding $1 \leq \ell \leq n$ such that

$$\frac{Z_{\ell-1}}{Z_n} \leq \xi_2 \leq \frac{Z_\ell}{Z_n} \quad (1.59)$$

where $Z_0 \equiv 0$. Finally, the height of $\mathbf{P} = \psi(\xi_1, \xi_2)$ is

$$\hat{z} = \sqrt{\xi_2 - Z_{\ell-1} + z_\ell}, \quad (1.60)$$

and the point itself is

$$\mathbf{P} = (\omega \cos \hat{\theta}, \omega \sin \hat{\theta}, \hat{z}), \quad (1.61)$$

where $\omega = \sqrt{1 - \hat{z}^2}$.

The algorithm described above also works for spherical polygons \mathcal{P} that surround the pole of the sphere. In this case, each lune has an odd number of segments crossing it, and $\bar{z}_n = 1$ must be added to the list of heights defined by each $\hat{\theta}$ in sampling from the conditional distribution.

The algorithm described above is somewhat more costly than the algorithm for uniform sampling of spherical triangles [3] for two reasons: 1) evaluating piecewise continuous functions requires some searching, and 2) the cumulative marginal

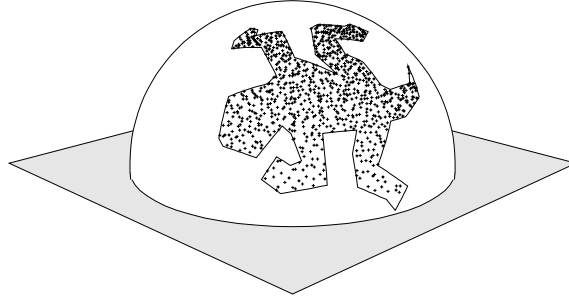


Figure 1.12: A non-convex spherical polygon with cosine-weighted and stratified samples generated with the proposed mapping.

distribution cannot be inverted exactly. Furthermore, the sampling algorithm requires some preprocessing to make both of these operations efficient and accurate.

Pre-processing includes partitioning the polygon into lunes, computing the constants c_1 and c_2 defined in equation (1.48) for each resulting edge, and sorting the line segments within each lune into increasing order. In the worst case, there may be $n - 2$ lunes, with $\Omega(n)$ of them containing $\Omega(n)$ segments. Thus, creating them and sorting them requires $O(n^2 \log n)$ in the worst case. For convex polygons, this drops to $O(n)$, since there can be only two segments per lune.

Once the pre-processing is done, samples can be generated by searching for the appropriate $[\theta_k, \theta_{k+1}]$ interval, which can be done in $O(\log n)$ time, and then sampling according to the conditional distribution, which can be done on $O(n)$ time. The latter cost is the dominant one because all of the intervals must be formed for normalization. Therefore, in the worst case, the cost of drawing a sample is $O(n)$; however, for convex polygons this drops to $O(\log n)$.

Figure 1.2 shows 900 samples in a spherical quadrilateral, distributed according to the cosine distribution. Note that more of the samples are clustered near the pole than the horizon. Stratification was performed by mapping “jittered” points from the unit square onto the quadrilateral. Figures 1.11 and 1.12 show 900 samples distributed according to the cosine density within a highly non-convex spherical polygon. These samples were generated without first partitioning the polygon into triangles. In both of the test cases, the cumulative marginal distribution function F is very nearly piecewise linear, and its inverse can be computed to extremely high accuracy with a piecewise cubic curve.

Chapter 2

Combining Sampling Strategies

2.1 Introduction

In this chapter we explore the idea of constructing effective random variables for Monte Carlo integration by combining two or more simpler random variables. For instance, suppose that we have at our disposal a convenient means of sampling the solid angle subtended by a luminaire, and also a means of sampling a brdf; how are these to be used in concert to estimate the reflected radiance from a surface? While each sampling method can itself serve as the basis of an importance sampling scheme, in isolation neither can reliably predict the shape of the resulting integrand. The problem is that the shape of the brdf may make some directions “important” (i.e. likely to make a large contribution to the integral) while the luminaire, which is potentially orders of magnitude brighter than the indirect illumination, may make other directions “important.” The question that we shall address is how to construct an importance sampling method that accounts for all such “hot spots” by combining available sampling methods, but without introducing statistical bias. The following discussion closely parallels the work of Veach [13], who was the first to systematically explore this idea in the context of global illumination.

To simplify the discussion, let us assume that we are attempting to approximate some quantity \mathcal{I} , which is given by the integral of an unknown and potentially ill-behaved function f over the domain D :

$$\mathcal{I} = \int_D f(x) dx. \quad (2.1)$$

For instance, f may be the product of incident radiance (direct and indirect), a reflectance function, and a visibility factor, and D may be either a collection of surfaces or the hemisphere of incident directions; in cases such as these, \mathcal{I} may

represent reflected radiance. In traditional *importance sampling*, we select a *probability density function* (pdf) p over D and rewrite the integral as

$$\mathcal{I} = \int_D \left[\frac{f(x)}{p(x)} \right] p(x) dx = \left\langle \frac{f(\mathbf{X})}{p(\mathbf{X})} \right\rangle, \quad (2.2)$$

where \mathbf{X} denotes a random variable on the domain D distributed according to the pdf p , and $\langle \cdot \rangle$ denotes the *expected value* of a random variable. The second equality in equation (2.2) is simply the definition of expected value. It follows immediately that the *sample mean* of the new random variable $f(\mathbf{X})/p(\mathbf{X})$ is an estimator for \mathcal{I} ; that is, if

$$\mathcal{E} = \frac{1}{N} \sum_{i=1}^N \frac{f(\mathbf{X}_i)}{p(\mathbf{X}_i)}, \quad (2.3)$$

for $N \geq 1$, where $\mathbf{X}_1, \dots, \mathbf{X}_N$ are iid (independent identically distributed) random variables, each distributed according to the pdf p , then $\langle \mathcal{E} \rangle = \mathcal{I}$. Consequently, $\mathcal{E} \approx \mathcal{I}$, and the quality of the approximation can be improved by increasing N , the number of samples, and/or by increasing the similarity between the original integrand f and the pdf p . Since evaluating $f(\mathbf{X}_i)$ is potentially very costly, we wish to pursue the second option to the extent possible. This is precisely the rationale for importance sampling.

2.2 Using Multiple PDFs

Now let us suppose that we have k distinct pdfs, p_1, p_2, \dots, p_k , that each mimic some potential “hot spot” in the integrand; that is, each concentrates samples in a region of the domain where the integrand may be relatively large. For instance, p_1 may sample according to the brdf, concentrating samples around specular directions of glossy surfaces, while p_2, \dots, p_k sample various luminaires or potential specular reflections. Let us further suppose that for each p_i we draw N_i iid samples, $\mathbf{X}_{i,1}, \mathbf{X}_{i,2}, \dots, \mathbf{X}_{i,N_i}$, distributed according to p_i . Our goal is to combine them into an estimator \mathcal{E} that has several desirable properties. In particular, we wish to ensure that

1. $\langle \mathcal{E} \rangle = \mathcal{I}$,
2. \mathcal{E} is relatively easy to compute,
3. $\text{var}(\mathcal{E})$ is small.

That is, we wish to have the expected value of \mathcal{E} match the actual value of the integral, \mathcal{I} , to pay a low computational price for drawing each sample, and to reduce the variance of \mathcal{E} as much as possible, thereby reducing the number of samples required to attain a reliable approximation. The first requirement ensures that the estimator is *unbiased*. Unbiased estimators have the highly desirable property that they allow us to converge to the exact answer by taking a sufficiently large number of samples. As a general rule, this is the first property that any random variable designed for Monte Carlo integration should possess [7, 8].

As we shall see, there is a large family of functions ϕ that meet property 1, leaving much flexibility in choosing one that meets both properties 2 and 3. We begin by identifying such a class of estimators, then imposing the other constraints. First, consider an estimator of the form

$$\mathcal{E}_\phi \equiv \sum_{i=1}^k \sum_{j=1}^{N_i} \phi_i(\mathbf{X}_{i,j}) f(\mathbf{X}_{i,j}), \quad (2.4)$$

for some suitable choice of the functions ϕ_i . That is, let us allow a different function ϕ_i to be associated with the samples drawn from each pdf p_i , and also allow the weight of each sample to depend on the sample itself. Equation (2.4) is extremely general, and also reasonable, as we can immediately ensure that \mathcal{E}_ϕ is unbiased by constraining the functions ϕ_i to be of the form

$$\phi_i(x) \equiv \frac{w_i(x)}{N_i p_i(x)}, \quad (2.5)$$

where for all $x \in D$ and $1 \leq i \leq k$, the *weighting functions* w_i satisfy

$$w_i(x) \geq 0, \quad (2.6)$$

$$w_1(x) + \cdots + w_k(x) = 1. \quad (2.7)$$

To see that the resulting estimator is unbiased, regardless of the choice of the weighting functions w_i , provided that they satisfy constraints (2.6) and (2.7), let us define \mathcal{E}_w to be the estimator \mathcal{E}_ϕ where the ϕ_i are of the form shown in equa-

tion (2.5), and observe that

$$\begin{aligned}
\langle \mathcal{E}_w \rangle &= \left\langle \sum_{i=1}^k \sum_{j=1}^{N_i} \phi_i(\mathbf{X}_{i,j}) f(\mathbf{X}_{i,j}) \right\rangle \\
&= \sum_{i=1}^k \sum_{j=1}^{N_i} \langle \phi_i(\mathbf{X}_{i,j}) f(\mathbf{X}_{i,j}) \rangle \\
&= \sum_{i=1}^k \sum_{j=1}^{N_i} \int_D \phi_i(x) f(x) p_i(x) dx \\
&= \sum_{i=1}^k \int_D \frac{w_i(x)}{p_i(x)} f(x) p_i(x) dx \\
&= \int_D f(x) \left[\sum_{i=1}^k w_i(x) \right] dx \\
&= \int_D f(x) dx \\
&= \mathcal{I}.
\end{aligned}$$

Thus, by considering only estimators of the form \mathcal{E}_w , we may henceforth ignore property 1 and concentrate strictly on selecting the weighting functions w_i so as to satisfy the other two properties.

2.3 Possible Weighting Functions

In some sense the most obvious weighting functions to employ are given by

$$w_i(x) \equiv \frac{c_i p_i(x)}{q(x)}, \quad (2.8)$$

where

$$q(x) \equiv c_1 p_1(x) + \cdots + c_k p_k(x), \quad (2.9)$$

is a pdf obtained by taking a convex combination of the original pdfs; that is, the constants c_i satisfy $c_i \geq 0$ and $c_1 + \cdots + c_k = 1$. Clearly, these w_i are positive and sum to one at each x ; therefore the resulting estimator is unbiased, as shown above. This particular choice is “obvious” in the sense that it corresponds exactly

to classical importance sampling based on the pdf defined in equation (2.9), when a very natural constraint is imposed on N_1, \dots, N_k . To see this, observe that

$$\begin{aligned}
\mathcal{E}_w &= \sum_{i=1}^k \sum_{j=1}^{N_i} \frac{w_i(\mathbf{X}_{i,j})}{N_i p_i(\mathbf{X}_{i,j})} f(\mathbf{X}_{i,j}) \\
&= \sum_{i=1}^k \left[\frac{1}{N_i} \sum_{j=1}^{N_i} \frac{c_i}{q(\mathbf{X}_{i,j})} f(\mathbf{X}_{i,j}) \right] \\
&= \sum_{i=1}^k \frac{c_i}{N_i} \left[\sum_{j=1}^{N_i} \frac{f(\mathbf{X}_{i,j})}{q(\mathbf{X}_{i,j})} \right]. \tag{2.10}
\end{aligned}$$

Now, let $N = N_1 + \dots + N_k$ be the total number of samples, and let us further assume that the samples have been partitioned among the pdfs p_1, \dots, p_k in proportion to the weights c_1, \dots, c_k , that is, with $N_i = c_i N$. Then the ratio c_i/N_i is constant, and equation (2.10) simplifies to

$$\mathcal{E}_w = \frac{1}{N} \sum_{i=1}^k \sum_{j=1}^{N_i} \frac{f(\mathbf{X}_{i,j})}{q(\mathbf{X}_{i,j})}. \tag{2.11}$$

Note that in equation (2.11) all samples are handled in exactly the same manner; that is, the weighting of the samples does not depend on i , which indicates the pdfs they are distributed according to. This is precisely the formula we would obtain if we began with q as our pdf for importance sampling. Adopting Veach's terminology, we shall refer to this particular choice of weighting functions as the *balance heuristic* [13]. Other possibilities for the weighting functions, which are also based on convex combinations of the original pdfs, include

$$w_i(x) = \begin{cases} 1 & \text{if } c_i p_i(x) = \max_j c_j p_j(x) \\ 0 & \text{otherwise} \end{cases}, \tag{2.12}$$

and

$$w_i(x) = c_i p_i^m(x) \left[\sum_{j=1}^k c_j p_j^m(x) \right]^{-1}, \tag{2.13}$$

for some exponent $m \geq 1$. Again, we need only verify that these weighting functions are non-negative and sum to one for all x to verify that they give rise to unbiased estimators. Note, also, that each of these strategies is extremely simple to compute, thus satisfying property 2 noted earlier.

2.4 Obvious is also Nearly Optimal

Let $\widehat{\mathcal{E}}_w$ denote an estimator that incorporates the balance heuristic, and let \mathcal{E}_w be an estimator with any other valid choice of weighting function. Veach has shown [13] that

$$\text{var}\left(\widehat{\mathcal{E}}_w\right) \leq \text{var}(\mathcal{E}_w) + \mathcal{I}^2 \left[\frac{1}{N_{\min}} - \frac{1}{N} \right], \quad (2.14)$$

where $N_{\min} = \min_i N_i$. Inequality (2.14) indicates that the variance of the estimator $\widehat{\mathcal{E}}_w$ compares favorably with the optimal strategy, which would be infeasible to determine in any case. In fact, as the number of samples of the least-sampled pdf approaches to infinity, the balance heuristic approaches optimality.

Fortunately, the balance heuristic is also extremely easy to apply; it demands very little beyond the standard requirements of importance sampling, which include the ability to generate samples distributed according to each of the original pdfs p_i , and the ability to compute the density of a given point x with respect to each of the original pdfs [8]. This last requirement simply means that for each $x \in D$ and $1 \leq i \leq k$, we must be able to evaluate $p_i(x)$. Thus, $\widehat{\mathcal{E}}_w$ satisfies all three properties noted earlier, and is therefore a reasonable heuristic in itself for combining multiple sampling strategies.

Bibliography

- [1] James Arvo. *Analytic Methods for Simulated Light Transport*. PhD thesis, Yale University, December 1995.
- [2] James Arvo. Applications of irradiance tensors to the simulation of non-Lambertian phenomena. In *Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH*, pages 335–342, August 1995.
- [3] James Arvo. Stratified sampling of spherical triangles. In *Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH*, pages 437–438, August 1995.
- [4] Marcel Berger. *Geometry*, volume II. Springer-Verlag, New York, 1987. Translated by M. Cole and S. Levy.
- [5] Robert L. Cook. Stochastic sampling in computer graphics. *ACM Transactions on Graphics*, 5(1):51–72, 1986.
- [6] M. H. Kalos and Paula A. Whitlock. *Monte Carlo Methods*, volume I, *Basics*. John Wiley & Sons, New York, 1986.
- [7] David Kirk and James Arvo. Unbiased sampling techniques for image synthesis. *Computer Graphics*, 25(4):153–156, July 1991.
- [8] David Kirk and James Arvo. Unbiased variance reduction for global illumination. In *Proceedings of the Second Eurographics Workshop on Rendering*, Barcelona, May 1991.
- [9] R. Y. Rubinstein. *Simulation and the Monte Carlo Method*. John Wiley & Sons, New York, 1981.
- [10] Peter Shirley, Changyaw Wang, and Kurt Zimmerman. Monte Carlo methods for direct lighting calculations. *ACM Transactions on Graphics*, 15(1):1–36, January 1996.

- [11] Jerome Spanier and Ely M. Gelbard. *Monte Carlo Principles and Neutron Transport Problems*. Addison-Wesley, Reading, Massachusetts, 1969.
- [12] Greg Turk. Generating random points in triangles. In Andrew S. Glassner, editor, *Graphics Gems*, pages 24–28. Academic Press, New York, 1990.
- [13] Eric Veach and Leonidas J. Guibas. Optimally combining sampling techniques for Monte Carlo rendering. In *Computer Graphics Proceedings, Annual Conference Series*, ACM SIGGRAPH, pages 419–428, August 1995.