

PBRT: Monte Carlo Integration

Thursday, January 14, 2021 5:59 PM

13 Monte Carlo integration

Integrators compute radiance along ray paths between lights and the camera.

The integral equations that describe light scattering, like $L_o(p, \omega_o)$, generally don't have analytic solutions (because the incident radiance function $L_i(p, \omega_i)$ can't be computed in closed form because the set Ω of directions usually isn't the full hemisphere or sphere, but one that is "patchy" due to occlusion of subsets of directions).

Standard numerical integration techniques that are very effective at solving low-dimensional smooth integrals are ineffective for rendering because their convergence rate is poor for the higher-dimensional (dimension here refers to the number of variables of integrands) and discontinuous integrals that are common here. (Standard numerical integration techniques: trapezoidal integration, Gaussian quadrature.)

Monte Carlo numerical methods use random sampling to evaluate (estimate) integrals with a convergence rate that is independent of the dimensionality of the integrand.

- What does it mean for an integral to converge?

Judicious use of randomness in algorithm design.

Las Vegas vs Monte Carlos algorithms.

Given the same input, repeated runs of a Monte Carlo algorithm give a result that is statistically very close to the exact value.

Monte Carlo integration can estimate $\int f(x) dx$ by evaluating $f(x)$ at arbitrary points in the domain. $f(x)$ can therefore be discontinuous.

13.1 Background and probability review

Probability of sampling illumination from each light in the scene based on the power Φ_i from each source relative to the total power from all sources:

$$p_i = \frac{\Phi_i}{\sum_j \Phi_j}$$

Note that light sources with higher power have a higher probability of being sampled.

13.1.1 Continuous random variables

In rendering, most random variables taken on values over ranges of continuous domains, e.g. directions on the unit sphere or the surfaces of shapes in the scene.

The canonical uniform random variable ξ takes on all values in $[0,1]$ with equal probability. This

random variable is important because we can generate samples **from arbitrary distributions** by transforming/mapping a sample from ξ .

The **probability density function PDF** of a continuous random variable describes the relative probability of a random variable taking on a particular value. It is the derivative of the *CDF* $P(x)$:

$$p(x) = \frac{dP(x)}{dx}$$

Probability that a random variable lies inside $[a, b]$:

$$P(x \in [a, b]) = \int_a^b p(x) dx$$

13.1.2 Expected values and variance

Monte Carlo integration computes the **expected values** of arbitrary integrals.

Variance is used to quantify the error in a value estimated by a Monte Carlo algorithm.

Given random variables that are **independent**, variance has the property that the sum of the variances is equal to the variance of the their sum.

- TODO: Sum of random variables.

13.2 The Monte Carlo estimator

(See [notes](#). Also [Simulation, Ross](#).)

This estimator approximates the value of an arbitrary integral and is the foundation of PBRT's light transport algorithms.

Given N uniform $[a, b]$ random variables U_i :

$$F_N = \frac{b-a}{N} \sum_{i=1}^N f(U_i)$$

is a **Monte Carlo estimator** and its expected value is the integral of f :

$$E[F_N] = \int_a^b f(x) dx$$

by the strong law of large numbers. This is Monte Carlo integration, the approximation of the value of a definite integral in $[a, b]$ using sums of uniform $[a, b]$ random variables.

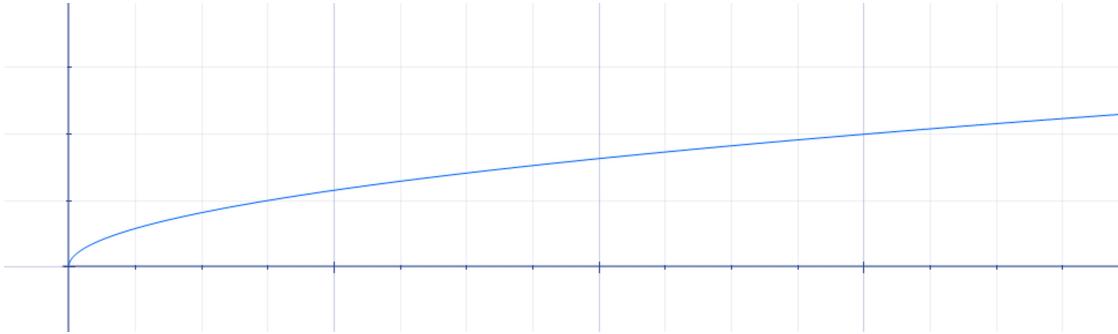
U_i can be computed with *Lerp(rng.UniformFloat(), a, b)*.

We want $Var(F_N)$ to be **small** and using estimators **based on other distributions** more appropriate for the case than the uniform distribution can reduce variance. The Monte Carlo estimator with an arbitrary *PDF* is:

$$F_N = \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{p(X_i)}$$

The number N of samples (that is, of random variables) is independent of the dimensionality of f in Monte Carlo integration. That's not the case with standard numerical quadrature techniques, where N is exponential in the dimension.

The error of F_N decreases with N at a rate of $O(\sqrt{N})$, which means that after some number N_k of samples, the improvement is less and less noticeable.



Note that this rate is independent of the dimensionality of f . This is crucial because the light transport equation used by the path tracing algorithm is an infinite-dimensional integral.

- TODO: what's an estimator?
- TODO: characterization of uniform random variable.
- TODO: simulation uses: the concept of joint density, the strong law of large numbers.

13.3 Sampling random variables

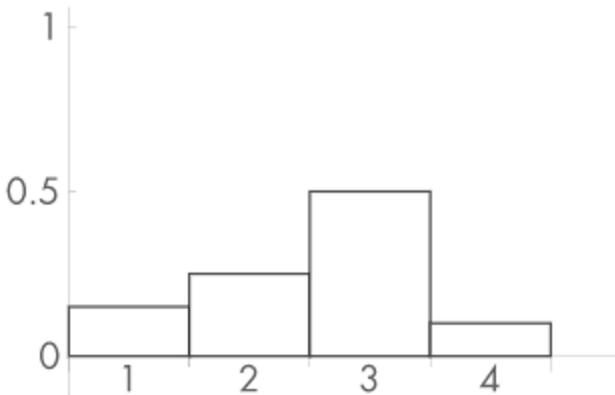
To evaluate the Monte Carlo estimator $F_N = \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{p(X_i)}$ with arbitrary PDF , first we need to know how to generate samples X_i with that PDF .

Ultimately, we want to generate samples from the distributions defined by *BDSFs*, light sources, cameras, and scattering media.

13.3.1 The inversion method

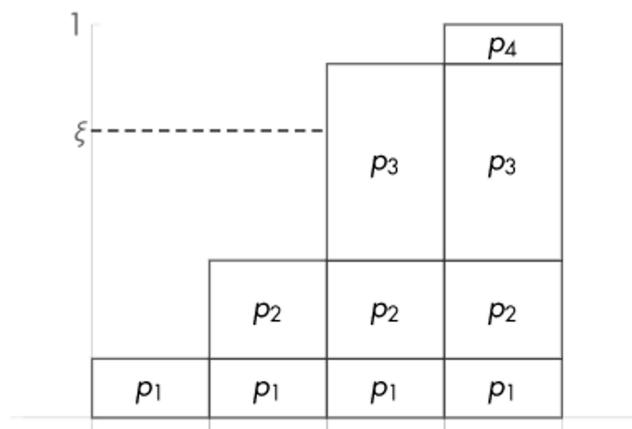
The **inversion method** maps uniform random variables to random variables of the desired distribution.

Take this PMF , which, by the strong law of large numbers, assigns probabilities p_i to 4 different outcomes of an experiment:



Call this random variable X .

The CDF of this PMF_X is shown next. The inversion method samples the range of CDF_X using a uniform $[0,1]$ random variable ξ :



And then maps that sample to the corresponding value x_i of X (out of $\{1,2,3,4\}$, the values it can assume) and its probability p_i (this map is the inverse function of CDF_X). As you can see, outcome x_i is chosen with probability equal to x_i 's own probability as assigned by PMF_X .

Now, let $x_i \rightarrow \infty$, so that X becomes a continuous random variable with PDF_X . CDF_X is then found by integration and the mapping between U and X is the inverse of CDF_X .

1. Compute the CDF :

$$F(x) = \int_0^x f(t) dt$$

2. Compute the inverse $F^{-1}(x)$.
3. Obtain a sample ξ of U .
4. Evaluate the inverse at ξ to obtain X_i :

$$X_i = F^{-1}(\xi)$$

which has the desired $PDF f$.

Here's the inversion method for a **piecewise-constant** 1D function with domain [0,1]:

The function assigns a constant value to all x in the i th subinterval of size $\Delta x = \frac{1}{N}$, where N is the number of subintervals that the domain [0,1] is divided into. These constant values c_i of the **range** of f are **presumably unbounded**.

$$f(x) = \begin{cases} c_0, & x_0 \leq x < x_1 \\ c_1, & x_1 \leq x < x_2 \\ \dots \end{cases}$$

Its definite integral (area under the curve) is:

$$F = \int_0^1 f(x) dx = \sum_{i=0}^{N-1} c_i \Delta x = \sum_{i=0}^{N-1} \frac{c_i}{N}$$

Note that the right-hand side is the exact value of the integral. It is not a Riemann sum. It is expressed as a sum because the definite integral of $f(x)$ lends itself to be expressed that way.

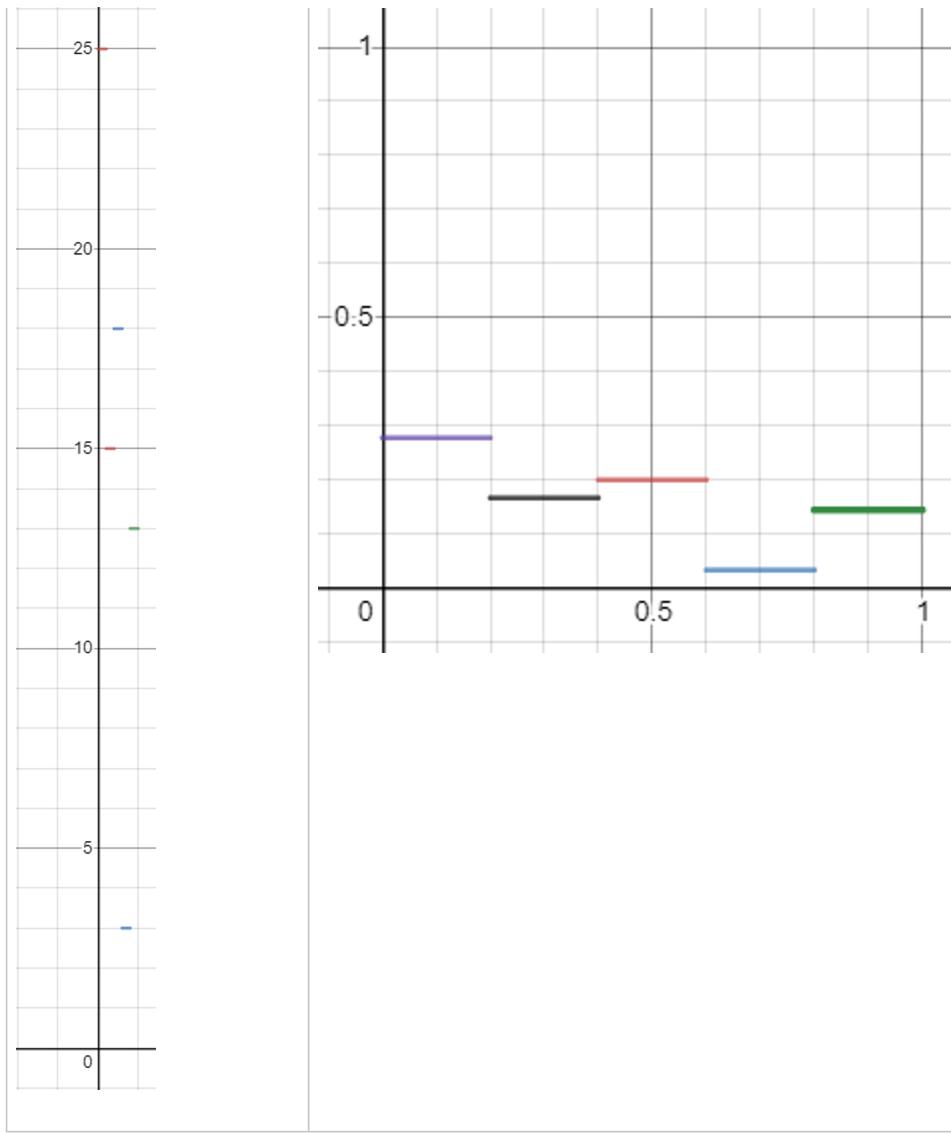
Now, the **key to understand** this is that we want to turn $f(x)$ into a *PDF* $p(x)$. Just as it is, the range of f is not [0,1], like the range of a *PDF* should be; it is, instead, **presumably unbounded**.

To turn $f(x)$ into a *PDF* $p(x)$, we scale it by the definite integral F to map its unbounded range to the interval [0,1]:

$$p(x) = \frac{f(x)}{F}$$

Think of the probability of x_i^* as the fraction of the total area that the i th rectangle $f(x_i^*) \Delta x$ represents as $\Delta x \rightarrow 0$ (as in a Riemann sum).

$f(x)$, unbounded range	$p(x)$, range [0,1], sum to 1
--------------------------	--------------------------------



Now, the inversion method:

Step 1: compute the CDF:

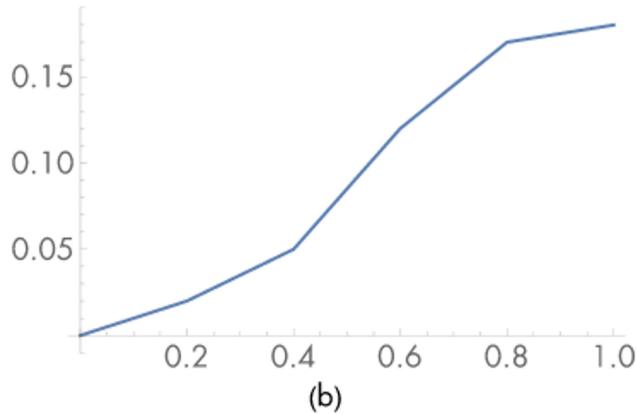
Note that it is defined recursively and that it is piecewise-linear (the i th linear subinterval is $c_i x$):

$$P(x_0) = 0$$

$$P(x_1) = \int_{x_0}^{x_1} p(x) dx = \frac{v_0}{cN} = P(x_0) + \frac{v_0}{cN}$$

$$P(x_2) = \int_{x_0}^{x_2} p(x) dx = \int_{x_0}^{x_1} p(x) dx + \int_{x_1}^{x_2} p(x) dx = P(x_1) + \frac{v_1}{cN}$$

$$P(x_i) = P(x_{i-1}) + \frac{v_{i-1}}{cN}.$$



Step 2: compute the inverse of the *CDF*:

We want to compute the value x such that:

$$\xi = \int_0^x p(t) dt = P(x)$$

That is, we want the outcome x of the sample space $[0,1]$ that has the cumulative probability $P(x) = \xi$, where ξ is the uniform random number we are using to sample the *PDF*.

The interval $[x_i, x_{i+1}]$ of f where this x lies is the one where $P(x_i) < \xi < P(x_{i+1})$, which can be found using a binary search.

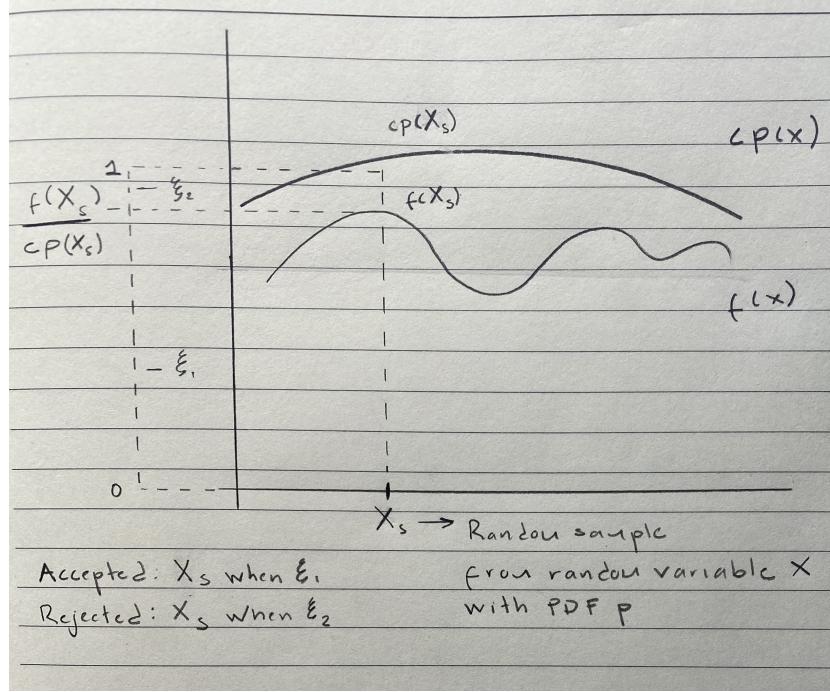
(My code comments finish these thoughts and might even correct them.)

In summary, the inversion method transforms a canonical uniform random variables into a random variable of another distribution.

13.3.2 The rejection method

The rejection method can be used to generate samples according to a function's distribution when that function is not integrable (which prevents us from finding the corresponding *PDF* and *CDF*) or when the inverse of the associated *CDF* can't be found analytically.

Here, X and its *PDF* $p(x)$ are not the ones determined by the function f that we are interested in sampling. Recall that we are doing the rejection method because we can't obtain its *PDF*. So this *PDF* $p(x)$ is some other one that we do know and that has the property that some multiple of it bounds f .



Algorithm:

```

loop forever:
    sample X from p's distribution
    if  $\xi < \frac{f(X)}{cp(X)}$ , then:
        return X
    
```

Rejection sampling isn't actually used in any of the Monte Carlo algorithms currently implemented in PBRT. We will normally prefer to find distributions that are similar to $f(x)$ that can be sampled directly, so that well-distributed points on $[0,1]^n$ can be mapped to sample points that are in turn well-distributed.

Example: rejection sampling a unit circle

Skipped because rejection sampling isn't used in PBRT.

13.4 Metropolis sampling

TODO: advanced content.

13.5 Transforming between distributions

(Understanding this is critical for drawing samples from multidimensional distribution functions.)

The inversion method not only transforms canonical uniform random variables into random variables of another distribution: a random variable X of some other distribution can also be transformed into a random variable Y of another distribution, **provided that they have the same CDF**, which is the case when **Y is a function of X** , and provided that **Y is one-to-one**.

Let's see:

The function y of a random variable X is also a random variable, $Y = y(X)$.

The *CDF* F of Y is equal to the *CDF* of X (much like their [expected values](#)):

$$P\{Y \leq y\} = P\{X \leq x\}$$

$$F_Y(y) = F_Y(y(x)) = F_X(x)$$

Their *PDFs* p_X and p_Y are their derivatives (which must be equal, because the *CDFs* are equal):

$$f_Y(y) \frac{dy}{dx} = f_X(x)$$

by the chain rule.

We are interested in $f_Y(y)$, because we are assuming that we know $f_X(x)$. Solving for f_Y :

$$f_Y(y) = \left(\frac{dy}{dx} \right)^{-1} f_X(x)$$

And we also know $\frac{dy}{dx}$, because we know $y(x)$.

We also know that $y'(x)$ is either strictly positive or strictly negative (because $y(x)$ is required to be one-to-one, it is either a strictly increasing or strictly decreasing function). So the final relationship between the *PDFs* $f_Y(y)$ and $f_X(x)$ is:

$$f_Y(y) = \left| \frac{dy}{dx} \right|^{-1} f_X(x)$$

This a generalization of the [inversion method](#): you can sample Y by sampling X .

13.5.1 Transformation in multiple dimensions

The [general inversion method](#) applies to n-dimensional random variables, and the relationship between the *PDFs* is the same:

$$f_Y(y) = f_Y(T(x)) = \frac{f_X(x)}{|J_{T(x)}|}$$

where X is n-dimensional, $Y = T(X)$ is a bijection (one-to-one and onto function, an [invertible function](#)), and $|J_{T(x)}|$ is the [absolute value of the determinant](#) of the [Jacobian matrix](#) of T .

The [Jacobian matrix](#) of an n-dimensional function is the [matrix](#) of all its [first-order partial derivatives](#).

13.5.2 Polar coordinates

TODO

13.5.3 Spherical coordinates

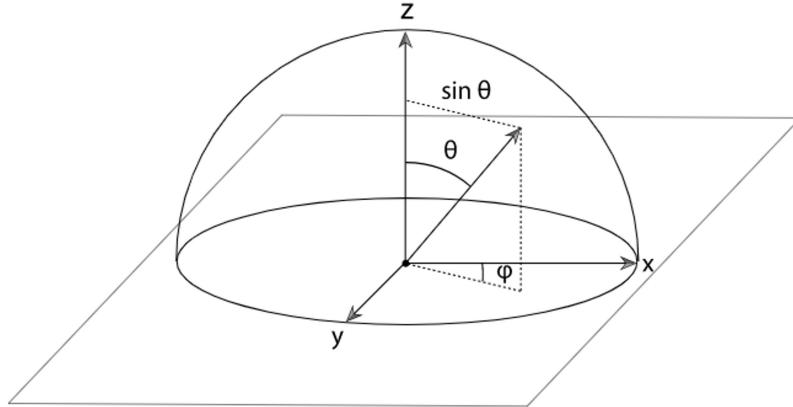
A direction in spherical coordinates is transformed to spherical coordinates as follows:

$$x = r \sin \theta \cos \phi$$

$$y = r \sin \theta \sin \phi$$

$$z = r \cos \theta$$

which is a vector transformation T (not linear, though).



(See 13.5 Transforming between distributions. There, we sample the distribution of X in order to sample the distribution of $Y = T(X)$, a transformation of X ; and the relationship between the *PDFs* is $f_Y(T(x)) = \frac{f_X(x)}{|J_{T(x)}|}$. In what follows, we solve for $f_X(x)$: $f_X(x) = |J_{T(x)}| f_Y(T(x))$, where $f_Y(T(x))$ is the *PDF* of the distribution of directions in cartesian coordinates, T transforms a spherical coordinate into a cartesian coordinate, and $f_X(x)$ is the *PDF* of the distribution of directions in spherical coordinates. In short, we are transforming a *PDF* from cartesian to spherical.)

The relationship between the *PDF* of the spherical coordinate transformation and the *PDF* of the directions in cartesian coordinates is thus:

$$p(r, \theta, \phi) = |J_T| p(x, y, z)$$

$$p(r, \theta, \phi) = r^2 \sin \theta \ p(x, y, z)$$

where $|J_T| = r^2 \sin \theta$ is the determinant of the Jacobian matrix of T , the transformation that maps a spherical coordinate to a cartesian coordinate:

$$x = r \sin \theta \cos \phi$$

$$y = r \sin \theta \sin \phi$$

$$z = r \cos \theta$$

13.6 2D sampling with multidimensional transformations

When a 2D joint density function $p(x, y)$ is not separable (that is, $p(x, y) \neq p_x(x)p_y(y)$), you can't just sample X and Y independently using their own 1D density function p_x and p_y .

To sample X (or Y) from a non-separable *PDF*, you have the **marginal density functions**:

$$p(x) = \int p(x, y) dy$$

which is the *PDF* of X alone defined as the **average density** for a particular value x over all possible y values.

Once you have $p(x)$, you can sample it in 1 dimension to obtain a sample X .

Then, you obtain the **conditional density function** $p(y|x)$ using that sample X :

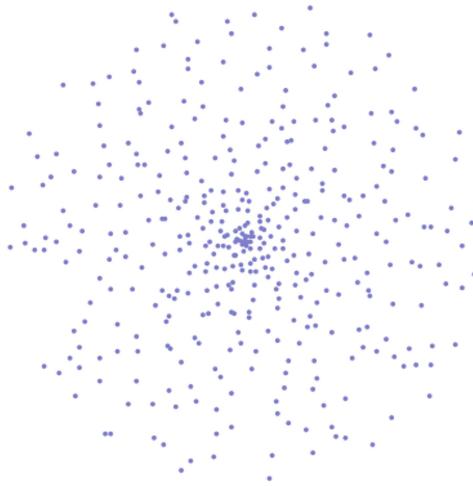
$$p(y|x) = \frac{p(x,y)}{p(x)}$$

and sample it in 1 dimension to obtain Y .

13.6.2 Sampling a unit disk

You can sample the unit disk by generating a random polar coordinate: $r = \xi_1$ and $\theta = 2\pi\xi_2$.

But the samples won't be uniformly distributed, they'll be concentrated near the center:



Recall that a uniform distribution has a **constant PDF**. So the *PDF* of a distribution of points over the area $\pi r^2 = \pi$ of a unit disk is $p(x,y) = \frac{1}{\pi}$, which is constant. Transforming it to polar coordinates, $p(r,\theta) = \frac{r}{\theta}$. Like every *PDF*, it must integrate to 1:

$$\int_0^{2\pi} \int_0^1 \frac{r}{\pi} dr d\theta$$

$$= \int_0^{2\pi} \left. \frac{r^2}{2\pi} \right|_0^1 d\theta$$

$$= \int_0^{2\pi} \frac{1}{2\pi} d\theta$$

$$= \left. \frac{\theta}{2\pi} \right|_0^{2\pi}$$

$$= 1$$

From section 13.6, recall that to sample a 2D joint density function like $p(r,\theta)$, you first obtain the marginal density function of one of the variables:

$$p(r) = \int_0^{2\pi} p(r, \theta) d\theta = 2r \quad (\text{why } 2r? \text{ when } r=1, p(1)=2, \text{ which can't be ...})$$

and then the conditional density for the other:

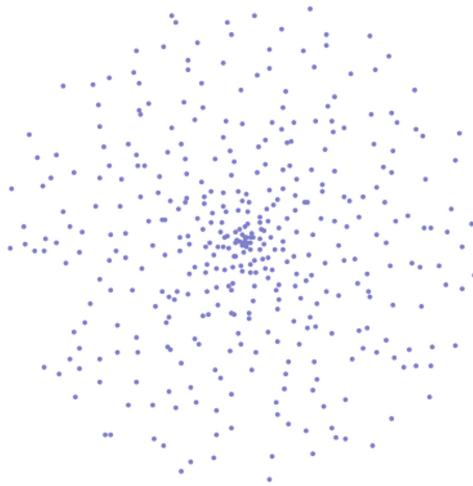
$$p(\theta|r) = \frac{p(r, \theta)}{p(r)} = \frac{1}{2\pi}$$

which is constant because every point at radius r has equal probability of being sampled.

We integrate $p(r)$ to obtain $P(r)$, the *CDF* of r , which with the [inversion method](#) we can invert and use $P^{-1}(r)$ to sample $p(r)$ using uniform random numbers:

$$r = \sqrt{\xi_1}$$

The square root pushes the sample toward the edge of the disk, counteracting the concentration/clumping near the center:



Likewise,

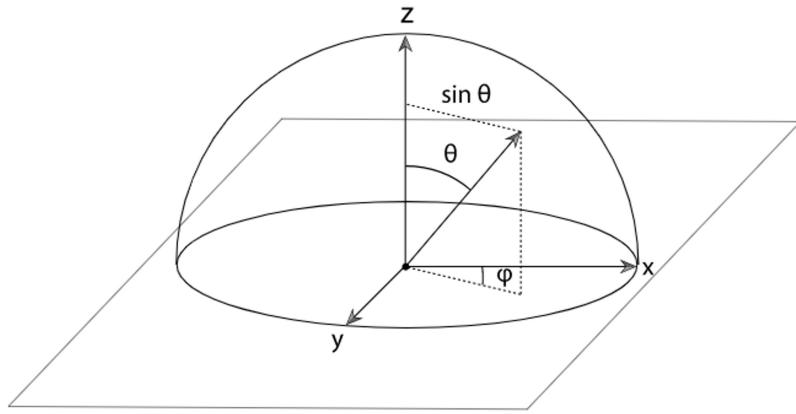
$$\theta = 2\pi\xi_2$$

13.6.3 Cosine-weighted hemisphere sampling

Recall the outgoing radiance equation:

$$L_o(p, \omega_o) = \int_{\Omega=H^2(\mathbf{n})} f_r(p, \omega_o, \omega_i) L_i(p, \omega_i) |\cos \theta_i| d\omega_i$$

and note the cosine factor (see [here](#) for reason). This cosine factor has a larger value the closer the incident direction is to the top of the hemisphere. We can say that L_o is **cosine-weighted** and that such incident directions make larger contributions to L_o .



For some reason I don't understand, we want to be able to sample incident directions ω_i with probability proportional to $\cos \theta_i$:

$$p(\omega) \propto \cos \theta$$

so that the top of the hemisphere has a higher probability of being sampled.

Integrate the hemisphere of cosine-weighted probabilities:

$$\int_{H^2} p(\omega) d\omega = 1$$

$$\int_0^{2\pi} \int_0^{\frac{\pi}{2}} c \cos \theta \sin \theta d\theta d\phi = 1$$

$$c 2\pi \int_0^{\pi/2} \cos \theta \sin \theta d\theta = 1$$

$$c = \frac{1}{\pi}$$

(To understand the integral, section 13.5.3 Spherical coordinates explains that $d\omega = \sin \theta \, d\theta \, d\phi$. We also know that $p(\omega) \propto \cos \theta$, as explained before.)

Section 13.5.3 Spherical coordinates explains that a *PDF* over solid angle ω is mapped to one over spherical coordinates as follows:

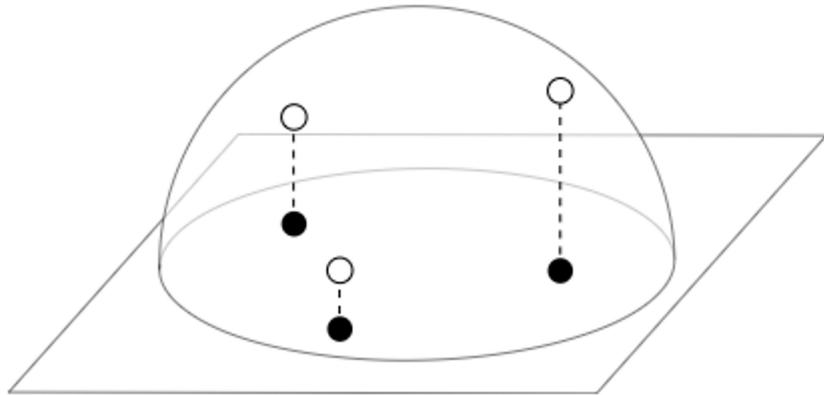
$$p(\theta, \phi) d\theta d\phi = p(\omega) d\omega$$

$$p(\theta, \phi) = \sin \theta p(\omega).$$

Recall that $p(\omega) \propto \cos \theta$. So plugging that into $p(\theta, \phi) = \sin \theta p(\omega)$:

$$p(\theta, \phi) = c \sin \theta \, p(\omega) = \frac{1}{\pi} \sin \theta \cos \theta$$

This requires us to sample solid angle ω , which is not trivial. **Malley's method** shows that we can arrive at the result of the definition of $p(\theta, \phi)$, that is, $\frac{1}{\pi} \sin \theta \cos \theta$ by sampling a unit disk uniformly and projecting the points up to the unit hemisphere above it:



As before, and as in section 13.5, we want to transform between distributions; this time between a uniform distribution of points over the unit disk expressed in polar coordinates (r, ϕ) and a distribution of points over the unit hemisphere. As in section 13.5, the relationship between the 2 PDFs involves the determinant of the Jacobian matrix of a transformation T : in this case T transforms points on the disk in polar coordinates to points on the hemisphere in spherical coordinates.

$$p(\theta, \phi) = |J_T| p(r, \phi) = \cos \theta \frac{r}{\pi} = (\cos \theta \sin \theta)/\pi,$$

where $r = \sin \theta$ is a "vertical projection".

"We have used the transformation method to prove that Malley's method generates directions with a cosine-weighted distribution."

13.7 Russian roulette and splitting

(This is directly used by the [path tracing algorithm](#).)

Russian roulette and **splitting** are 2 methods that improve the **efficiency** of a Monte Carlo estimator. Efficiency is defined formally as:

$$\epsilon[F] = \frac{1}{V[F]T[F]}$$

where $V[F]$ is the variance of the estimator and $T[F]$ is the time it takes to evaluate it. An estimator that takes less time to produce the same variance, or that produces less variance in the same amount of time is relatively more efficient.

What we want is to sample carefully so that the chosen samples are significant:

- Russian roulette helps **avoid** samples that are both **expensive to evaluate** and that make **small contributions**.

Samples are expensive to compute because they involve tracing a shadow ray between a point and a light source to determine visibility. Take the Monte Carlo estimator of the [direct lighting integral equation](#): If there isn't occlusion, but the BSDF returns 0 or a very small value for that direction, the sample's contribution will be 0 or very small too. Or if the direction is far from the surface normal and close to being tangential to the surface, the $|\cos \theta|$ factor will weigh down its contribution. It's better if we can avoid tracing shadow rays for such directions.

The [Monte Carlo estimator](#) is:

$$F_N = \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{p(X_i)}$$

The Russian roulette Monte Carlo estimator is:

$$F' = \begin{cases} \frac{F - qc}{1 - q} & \xi > q \\ c & \text{otherwise} \end{cases}$$

where q is the probability of terminating the sum at the current term (i.e. current sample, from 1 to N) and c is a constant (usually 0) that is returned instead of the value of the integrand $f(x)$. (Is q determined by the current term's sample? So that unimportant samples are indeed skipped?)

The notation of F' seems a little misleading: it doesn't say that the entire estimator for the N samples is not computed; it says that the sum of the N estimates (each corresponding to a sample) may be terminated at some term $k < N$ (or is it for skipping individual samples? If so, does q depend on the sample, so that unimportant samples are indeed skipped?).

Simply terminating the estimator would introduce bias in it. The $\frac{1}{1-q}$ factor by which the value of the integrand is multiplied is supposed to eliminate bias due to the skipped samples/estimates; in fact, the expected value of F' is the same as the expected value of F : $E[F'] = E[F]$, that is, the value of the estimated integral. I don't know what the $-qc$ term in the numerator does.

Russian roulette increases the variance of the estimator, though (see [importance sampling](#) for variance reduction techniques).

- Splitting places more samples in important dimensions of the integral. For example, a pixel (x, y) position and a direction ω are needed to compute outgoing radiance $L(p, \omega) = L(x, y, \omega)$ from the point intersected by a ray passing through pixel position (x, y) and arriving at it from direction ω (as sampled by a shadow ray that goes in the opposite direction). Instead of generating 200 pixel positions and directions, splitting the estimator could allow you to use only 100 pixel position samples (for pixel antialiasing), but the full 200 direction samples for casting shadow rays: 100 pixel positions may be enough for pixel antialiasing, but the full 200 shadow rays may be necessary for sampling, for example, an area light.

The standard Monte Carlo estimator for:

$$\int_A \int_{S^2} L_d(x, y, \omega) d\omega dA$$

(where A is the total pixel surface area and S^2 is the sphere of directions at the first intersection point p) is:

$$\frac{1}{N} \sum_{i=1}^N \frac{L(x_i, y_i, \omega_i)}{p(x_i, y_i, \omega_i)}$$

(I'm not sure about the denominator.) and with splitting it becomes:

$$\frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M \frac{L(x_i, y_i, \omega_{i,j})}{p(x_i, y_i) p(\omega_{i,j})}$$

which allows you to sample each dimension of the function with its own number of samples.

Splitting increases the efficiency of the estimator by allocating samples as needed by each dimension to do a more significant sampling.

13.8 Careful sample placement

TODO

13.9 Bias

TODO

13.10 Importance sampling

Importance sampling is a **variance reduction technique** that exploits the fact that the Monte Carlo estimator converges more quickly if the samples are taken from a distribution $p(x)$ that is similar to the function $f(x)$ of the integrand.

One of the problems that importance sampling avoids is wasting computation. For example, the $\cos \theta$ factor of the **scattering equation** tends to 0 as ω_i turns away from the surface normal, and so does the entire integrand. These directions ω_i contribute very little and evaluating the integrand with them is equally expensive as it is for more significant angle samples. So we want to reduce the likelihood of sampling them and increase the likelihood of sampling directions for which the rest of the factors of the integrand (f , L_i , etc) are relatively large. The latter is achieved by sampling directions with a probability distribution that is **similar in shape** to these factors, that is:

$$p(x) \propto f(x)$$

instead of with the default uniform *PDF*.

Take this example:

$p(x) = cf(x)$ (the probability density is a constant multiple of the function, the integrand) in which case $c = \frac{1}{\int f(x)dx}$ so that the sum of probabilities be 1.

Now look at the Monte Carlo estimator:

$$F_N = \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{p(X_i)}$$

and take one term of the summation, that is, **one estimate**:

$$\frac{f(X_i)}{p(X_i)}$$

With our example density $p(x) = cf(x)$, we get:

$$\frac{f(X_i)}{p(X_i)} = \frac{1}{c} = \int f(x) dx$$

This was for a single sample X_i and the estimate is already the value of the integral. Evaluating the estimator for the N samples, we get:

$$F_N = \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{p(X_i)} = \frac{1}{N} N \int f(x) dx = \int f(x) dx$$

With this *PDF*, the estimator gave us the exact value of the integrand, which is also its expected value:

$$E[F_N] = \int f(x) dx$$

Now recall the definition of variance: a measure of the spread of the values of the random variable or estimator; or a measure of the distance of the estimates (samples) from the expected value. In our case:

$$Var(X) = E[F_N^2] - (E[F_N])^2 = 0$$

variance is 0: no estimate differed from the expected value.

But of course this is unrealistic, because this *PDF* is defined in terms of the value of the integral, a value that we don't know and that we are trying to estimate. But this shows that if the *PDF* $p(x) \propto f(x)$, then variance decreases. It also shows that when variance is low, fewer samples are required to get closer to the expected value of the estimator.

13.10.1 Multiple importance sampling

Note that thus far the estimated integral is of a single-function integrand, but the integrand of the scattering equation and others is a product of functions:

$$\int f(x) g(x) dx$$

To reduce variance, we want:

$$p(x) \propto f(x)g(x)$$

but obtaining such a *PDF* might be difficult.

We can't just use either $p(x) \propto f(x)$ or $p(x) \propto g(x)$: a *PDF* that samples $f(x)$ with low variance might sample $g(x)$ poorly with high variance. Here's a great concrete example:

Consider the direct lighting scattering equation:

"Consider a near-mirror BRDF illuminated by an area light where L_d 's distribution is used to draw samples. Because the BRDF is almost a mirror, the value of the integrand will be close to 0 at

all ω_i directions except those around the perfect specular reflection direction. This means that almost all of the directions sampled by L_d will have 0 contribution, and variance will be quite high. Even worse, as the light source grows large and a larger set of directions is potentially sampled, the value of the PDF decreases, so for the rare directions where the BRDF is non-0 for the sampled direction we will have a large integrand value being divided by a small PDF value. While sampling from the BRDF's distribution would be a much better approach to this particular case, for diffuse or glossy BRDFs and small light sources, sampling from the BRDF's distribution can similarly lead to much higher variance than sampling from the light's distribution."

The **multiple importance sampling MIS** Monte Carlo estimator for $\int f(x)g(x)dx$ is:

$$\frac{1}{n_f} \sum_{i=1}^{n_f} \frac{f(X_i)g(X_i)w_f(X_i)}{p_f(X_i)} + \frac{1}{n_g} \sum_{j=1}^{n_g} \frac{f(Y_j)g(Y_j)}{p_g(Y_j)}$$

Note that one sum uses $p(x) \propto f(x)$ and the other $p(x) \propto g(x)$, each sum uses its own number of samples n_f and n_g , and its own **weighting function** w .

The weighting function eliminates large variance spikes caused by mismatches between $f(x)g(x)$ and $p(x)$. The **balance heuristic** is a good one:

$$w_s(x) = \frac{n_s p_s(x)}{\sum_i n_i p_i(x)}$$

where the denominator is a sum of the PDFs of all the functions in the product integrand, weighted by their respective number of samples.

The **power heuristic** weighting function reduces variance even further:

$$w_s(x) = \frac{n_s p_s(x)^\beta}{\sum_i n_i p_i(x)^\beta}$$