PBRT: Light Transport I: Surface Reflection

Wednesday, March 3, 2021 9:41 AM

14 Light transport I: surface reflection

Ray tracing algorithms + radiometric concepts + Monte Carlo sampling algorithms.

2 integrators that compute scattered radiance from surfaces in the scene.

Integrators evaluate the light transport equation, which is an integral equation that describes the equilibrium distribution of radiance in an environment.

The scattering equation can be estimated with Monte Carlo:

$$L_o(p, \omega_o) = \int_{S^2} f(p, \omega_o, \omega_i)\, L_i(p, \omega_i)\, |\cos\theta_i|\, d\omega_i$$
$$\approx \frac{1}{N} \sum_{j=1}^{N} \frac{f(p, \omega_o, \omega_j)\, L_i(p, \omega_j)\, |\cos\theta_j|}{p(\omega_j)}$$

where the function in the denominator is a PDF and N is the number of samples.

"In practice, we'll want to take some samples from a distribution that approximates the BSDF, some from a distribution that approximates the incident radiance from light sources, and then weight the samples with multiple importance sampling."

So we need methods for sampling from BSDFs and light sources.

Is it correct to say that $DirectLightingIntegrator$ and $PathIntegrator$ obtain the samples that they integrate by following ***light-carrying paths*** starting from the camera and bouncing off of surfaces?

14.1 Sampling reflection functions

BxDF::Sample_f samples f, the $BxDF$, using the inversion method, which simulates the $BxDF$ distribution using a uniform distribution sampled using stratified sampling. BxDF::Sample_f samples f for a given outgoing direction ω_o and a computed/sampled incident direction ω_i. The computed/sampled ω_i is obtained with probability $p(\omega_i)$ according to a PDF (Sample_f returns ω_i along with its probability).

BxDF is a base class and gives a default implementation for Sample_f, which samples the unit hemisphere above the surface with a cosine-weighted distribution (which is great for ***Lambertian*** BRDFs and the Oren-Nayar model). There are at least 2 cases where this implementation is not valid: for $BTDFs$, because transmittance needs to sample the hemisphere *below* the surface, and for surfaces with delta distribution, like perfectly specular ones, which map w_o to a single ω_i (as opposed to any one in the hemisphere sampled at random).

The *PDF of a cosine-weighted distribution* is $p(\omega) = p(\theta, \phi) = \frac{\cos\theta}{\pi}$ when sampling a unit hemisphere ($r = 1$). BxDF::Sample_f returns this probability with which it sampled the returned inciden direction ω_i corresponding to the input ω_o.

14.1.1 Microfacet BxDFs

TODO

14.1.2 FresnelBlend

TODO

14.1.3 Specular reflection and transmission

TODO

14.1.4 Fourier BSDF

TODO

14.1.5 Application: estimating reflectance

TODO

14.1.6 Sampling BSDFs

A BSDF object is a collection of BxDFs of different types (some are BRDFs, some are BTDFs, some are delta/specular, some are diffuse, etc.).

A BSDF is only ever sampled for a given BxDF type at a time. That is, a sample from a BSDF is a sample from the BxDFs that match the requested type.

To sample the incident direction ω_i corresponding to the input ω_o, BSDF::Sample_f samples 1 of the BxDFs, that is, it chooses 1 of them uniformly at random and then samples that BxDF. We don't care about the BxDFs value (the radiometric spectrum it returns), only abou the sample incident direction.

Intuitively, each matching BxDF represents a set of direction from which to sample ω_i. So even though a single BxDF will end up being sampled directly (with BxDF->Sample_f), we actually sampled all the matching BxDFs when we chose that BxDF at random out of all of them. So the PDF of the $BSDF$ is the average of all the $PDFs$ of these matching BxDFs:

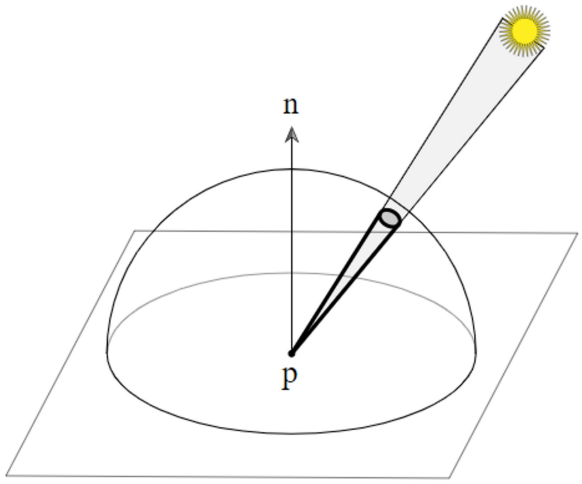
$$p(\omega) = \frac{1}{N}\sum_{i}^{N} p_i(\omega)$$

And the value of the BSDF (a radiometric spectrum) is the sum of the values of all the matching BxDFs for the sampled incident direction ω_i.

14.2 Sampling light sources

It looks like we sample a surface's BSDF at a point primarily in the case of bounced rays. For direct illumination, sampling the BSDF for an incident direction may not yield one that arrives directly from the light source (out of the set of directions of the BSDF, only a subset are directions that arrive from the light source).

Here, for example, the cone of directions that the spherical light subtends is a much better distribution to sample from than the hemisphere.



The Light class's Sample_Li samples this set of directions with respect to a given point of incidence.

14.2.1 Lights with singularities

Light sources described by a delta distribution (similar to perfect specular reflection and transmission), that is, for a given input, they return a single output.

Point lights have a delta distribution because they illuminate a given point only from a single direction (as opposed to area lights, that illuminate it from several directions).

14.2.2 Sampling shapes

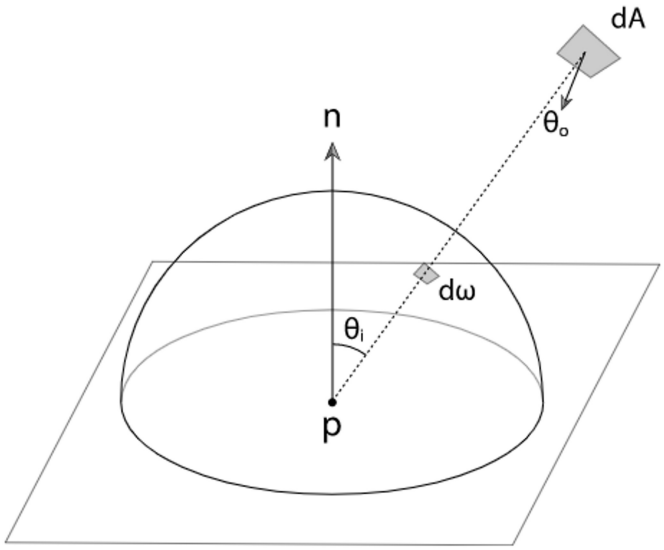
Because light sources may have a surface/shape, we need to be able to get sample points of their surfaces.

An AreaLight has a Shape component. Shape offers methods to sample its surface that AreaLight can call.

A Shape may be sampled with respect to area or with respect to solid angle (the set of directions from which the Shape is visible from some point in the scene). In the base implementation, area-area-sampling is uniform: $PDF = 1/A$. Sampling with respect to solid angle is different, because only a subset of points on the surface of the Shape are visible from the point of reference. However, the PDF can be obtained from the PDF used for area-sampling using the relation between $d\omega$ and dA:

$$d\omega_i = \frac{dA\ cos\theta_o}{r^2}$$

where r^2 is the distance between dA and p:



Given the PDF for sampling a point based on area:

$$p_A = \frac{1}{A}$$

the PDF for sampling a point based on solid angle is:

$$p_\omega = p_A \frac{dA}{d\omega_i} = \frac{r^2}{A \cos\theta_o}$$

Sampling disks

TODO

Sampling cylinders

TODO

Sampling triangles

TODO

Sampling spheres

TODO

14.2.3 Area lights

TODO: Need to implement DiffuseAreaLight too.

14.2.4 Infinite area lights

TODO

14.3 Direct lighting

The DirectLightingIntegrator only accounts for direct lighting: light that has traveled directly from a light source to the point being shaded. Ignores contributions from indirect illumination (light that is reflected by other non-emissive objects).

DirectLightingIntegrator computes an ***unbiased estimate of exitant radiance*** at a point in a given direction.

There are 2 ***strategies*** to sample the light sources:

- UniformSampleAll takes a number of sample from each and every light source and sums the result (the number of samples per light is a member variable of the Light). The DirectLightingIntegrator requests an array-sample. Array-samples have a better distribution than a sequence of single samples generated on demand would have. This results in light samples that are well distributed across light sources and across their individual surfaces.

  - [ ] TODO: Section 13.7 Russian Roulette and Splitting is relevant here.

- UniformSampleOne takes a single samples from only one of the light sources chosen uniformly at random. The DirectLightingIntegrator requests only 2 2D random samples to the Sampler: the 1st 2D sample is for choosing the position on the surface of the light source (see Shape::Sample) and the 2nd 2D sample is for sampling the shaded surface BSDF (see BSDF::Sample_f).

The strategy to choose depends on the number of samples taken per pixel: if many, then UniformSampleOne suffices (because many such single-sample, single-light samples will be taken anyway); if few, then UniformSampleAll to compensate.

The outgoing/exitant radiance equation for direct lighting differs from the general case in that only incident radiance coming directly from light sources are integrated (indirect/bounced radiance is not). We denote such direct lighting incident radiance as $L_{d(j)}$ (or ***direct radiance***), where j is the jth light source:

$$L_o(p, \omega_o) = \sum_{j=1}^{n} \int_{\Omega=S^2} f_r(p, \omega_o, \omega_i)\, L_{d(j)}(p, \omega_i)\, |\cos\theta_i|\, d\omega_i$$

The UniformSampleAll strategy estimates each definite integral individually and then performs the summation. Estimating the integral is explained in the next section, 14.3.1. The EstimateDirect function computes the value of the Monte Carlo estimator for the contribution of a light source chosen at random.

- [ ] TODO: The statement that the expected value of the sum of the scattering equation evaluated at every light source equals the

value of the scattering equation evaluated just at one and multiplied by the number of them; in UniformSampleOneLight. See also Ross (2002, p. 102).

14.3.1 Estimating the direct lighting integral

The Monte Carlo estimator for the direct lighting integral for a given light source is:

$$\frac{1}{N}\int_{j=1}^{N} \frac{f(p.\omega_o,\omega_r)\, L_d(p,\omega_r)\,|\cos\theta_j|}{p(\omega_r)}$$

PBRT doesn't sample the $f * L_d$ product with a distribution that matches it. It uses multiple importance sampling instead. What we need to sample are the directions ω_r. Some of these samples will be obtained with a PDF that matches f, the BSDF, and others with one that matches the light source (this is MIS, this is what reduces variance).

The book shows a scene rendered using only a distribution that matches the BSDF or only a distribution that matches the light sources, and explains the cases in which each one is effective or has high-variance.

14.4 The light transport equation

(The geometric optics assumptions are important to know here.**)**

The ***light transport equation*** describes the equilibrium distribution of radiance in a scene (i.e. this distribution doesn't change over time). It is evaluated at points on surfaces and it gives the ***total reflected radiance*** at the evaluated points. (So far, we have seen how to compute incident and reflected radiance at a point in a given direction only.)

14.4.1 Basic derivation

Conservation of power is a form of ***energy balance***: the difference between the power leaving an object, Φ_o, and the power entering it, Φ_i, is equal to the difference between the power it emits, Φ_e and the power it aborbs, Φ_a:

$$\Phi_o - \Phi_i = \Phi_e - \Phi_a$$

Another instance of energy balance: exitant radiance L_o must be equal to emitted radiance plus scattered incident radiance:

$$L_o(p,\omega_o) = L_e(p,\omega_o) + \int_{S^2} f(p,\omega_o,\omega_i)\, L_i(p,\omega_i)\,|\cos\theta_i|\; d\omega_i$$

$$(Exitant\ radiance = emitted\ radiance + scattered\ incident\ radiance)$$

That is the ***light transport equation*** and we want to ***simplify it***:

- Express L_i at the point in terms of L_o from another corresponding point: in the absence of participating media, the ***radiance carried by a ray is constant***. This is what allows us to say that the outgoing radiance from a light source point is equal to the incident radiance at a surface point when the points are connected by a ray and they are mutually visible (or between 2 surface points, where one reflects light onto the other).

  $$L_i(p,\omega) = L_o(t(p,\omega),-\omega)$$

  where t is a ***ray casting function*** that returns the 1st intersection point found in the ω direction. You see here that radiance arriving at p in the ω direction is the same as the radiance leaving the point $t(p,\omega)$ in the $-\omega$ direction.

  The light transport equation thus becomes:

  $$L_o(p,\omega_o) = L_e(p,\omega_o) + \int_{S^2} f(p,\omega_o,\omega_i)\, L_o(t(p,\omega_i),-\omega_i)\,|\cos\theta_i|\; d\omega_i$$

  which doesn't use L_i, but L_o. This is known as the ***energy balance form*** of the LTE.

14.4.2 Analytic solutions to the LTE

In certain cases, total reflected radiance at all points in all directions can be computed in closed form. For example, it is:

$$L = \frac{L_e}{1 - \rho_{hh}}$$

in the case of the interior of a sphere with a Lambertian BSDF. This expression is the closed form of an ***infinite series***:

$$L = \sum_{i=0}^{\infty} L_e \rho^i{}_{hh}$$

$$= L_e + \rho_{hh}(L_e + \rho_{hh}(L_e + \cdots$$

where the original L_o integral was rewritten as $L = L_e + c\pi L$ (because a Lambertian sphere interior permits it) and the L (L_o actually) on the right-hand side has been expanded recursively: the ray-casting function $t(p, \omega)$ permits us to follow the path of reflection of a ray.

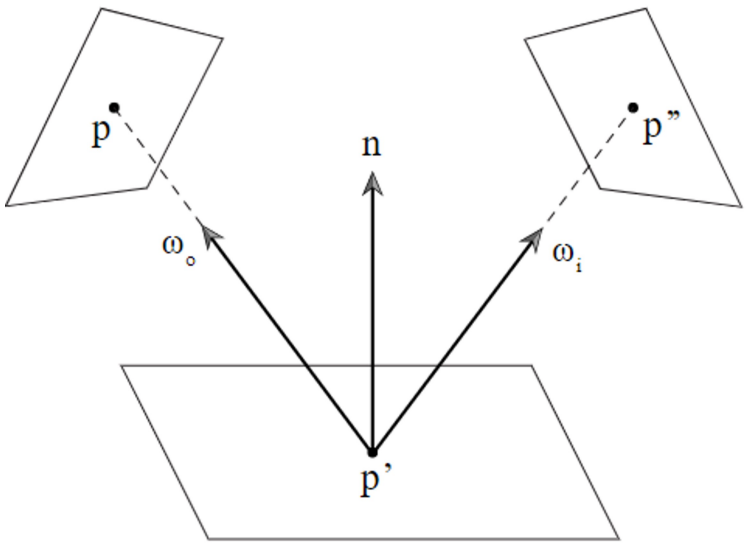
This closed-form expression can't be obtained in general, though. But it helps illustrate that what the integrators and the ray tracing algorithm are doing is compute the value of an infinite series.

14.4.3 The surface form of the LTE

The integral in the light transport equation:

$$L_o(p, \omega_o) = L_e(p, \omega_o) + \int_{S^2} f(p, \omega_o, \omega_i)\, L_o(t(p, \omega_i), -\omega_i)\, |\cos\theta_i|\, d\omega_i$$

is an ***integral over directions*** on the sphere. Ultimately, we want the ***path integral form*** of this equation. And to get there, we first need to rewrite it as an ***integral over area*** instead: the ***three-point form*** or ***surface form*** of the equation. Consider the 3 points p'', p', and p involved in a single ray bounce:



We now introduce notation for L_o and f (the BSDF) in terms of these points:

$$L(p', \omega) = L(p' \rightarrow p)$$

and

$$f(p', \omega_o, \omega_i) = f(p'' \to p' \to p)$$

With these substitutions, the LTE becomes:

$$L_o(p' \to p) = L_e(p' \to p) + \int_{???} f(p'' \to p' \to p)\, L_o(p'' \to p') |\cos\theta_i|\ d\omega$$

Note that $|\cos\theta_i|$ and the solid angle differential $d\omega$ remain, so we can't say we are integrating over area just yet. Transforming differential solid angle to differential area is done with the following relation:

$$d\omega = \frac{cos\theta'}{r^2} dA$$

Note that θ' here is different from θ in the $|\cos\theta_i|$ factor of the equation. Combining:

$$G(p \leftrightarrow p')\, dA(p) = V(p \leftrightarrow p') \frac{|\cos\theta|\ |\cos\theta'|}{||p - p'||^2}\, dA(p)$$

where the factor:

$$\frac{|\cos\theta'|}{||p - p'||^2} dA(p) = \frac{cos\theta'}{r^2} dA = d\omega$$

is the differential solid angle to differential area transformation, $|\cos\theta|$ is the original cosine of the equation, and $V(p \leftrightarrow p')$ determines whether the 2 points are visible to each other.

Substituting in the LTE:

$$L_o(p' \to p) = L_e(p' \to p) + \int_A f(p'' \to p' \to p)\, L_o(p'' \to p')\, G(p'' \leftrightarrow p')\ dA(p'')$$

we can finally say that the integral is over area. The area being integrated over is of the surface where p'' lies on and A is the set of all surfaces of the scene.

14.4.4 Integral over paths

We transformed the energy balance form of the LTE (an integral over directions) into its surface (3-point) form (an integral over area) because the energy balance form has an unwieldy recursive definition. Now we transform the surface form into the ***path integral form*** (the LTE as an integral over paths) because it has an explicit definition.

Paths are points in a high dimensional ***path space***.

The transformation consists in expanding the $L_o(p'' \to p')$ factor in the integral. So if:

$$L_o(p'' \to p') = L_e(p'' \to p') + \int_A f(p''' \to p'' \to p')\, L_o(p''' \to p'')\, G(p''' \leftrightarrow p'')\ dA(p''')$$

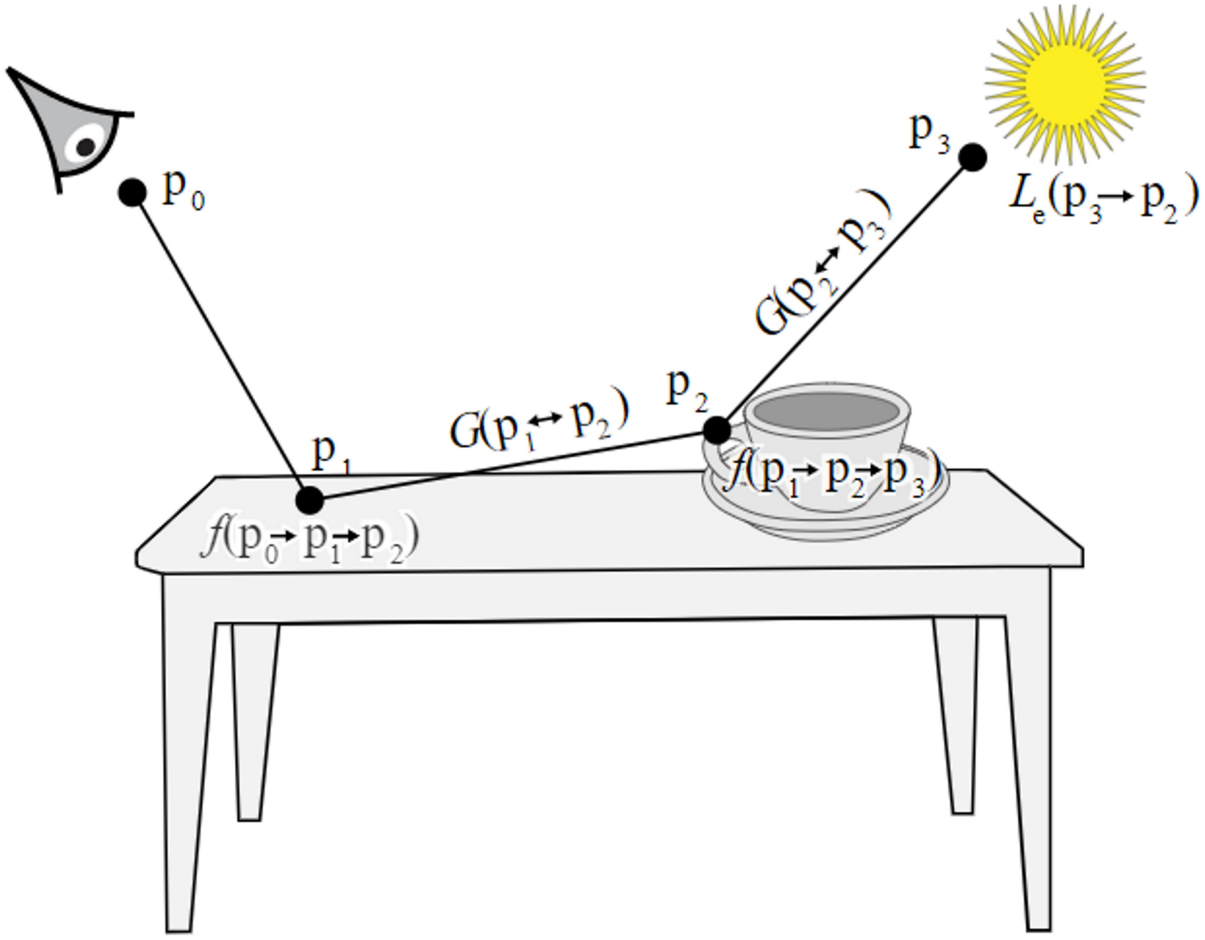
Then:

$$\begin{aligned} &L_o(p' \to p) \\ &= L_e(p' \to p) \\ &+ \int_A f(p'' \to p' \to p)\, [L_e(p'' \to p') + \int_A f(p''' \to p'' \to p')\, L_o(p''' \to p'')\, G(p''' \leftrightarrow p'')\ dA(p''')]\, G(p'' \leftrightarrow p')\ dA(p'') \end{aligned}$$

The expansion can be rewritten as a sum of integrals somehow:

$$
\begin{aligned}
L(\mathrm{p}_1 \to \mathrm{p}_0) = {} & L_\mathrm{e}(\mathrm{p}_1 \to \mathrm{p}_0) \\
& + \int_A L_\mathrm{e}(\mathrm{p}_2 \to \mathrm{p}_1) f\,(\mathrm{p}_2 \to \mathrm{p}_1 \to \mathrm{p}_0)\, G(\mathrm{p}_2 \leftrightarrow \mathrm{p}_1) \mathrm{d}A(\mathrm{p}_2) \\
& + \int_A \int_A L_\mathrm{e}(\mathrm{p}_3 \to \mathrm{p}_2) f\,(\mathrm{p}_3 \to \mathrm{p}_2 \to \mathrm{p}_1)\, G(\mathrm{p}_3 \leftrightarrow \mathrm{p}_2) \\
& \quad \times f\,(\mathrm{p}_2 \to \mathrm{p}_1 \to \mathrm{p}_0)\, G(\mathrm{p}_2 \leftrightarrow \mathrm{p}_1)\, \mathrm{d}A(\mathrm{p}_3)\, \mathrm{d}A(\mathrm{p}_2) + \cdots
\end{aligned}
$$

Each successive term of the sum is a path of increasing length: length 2, length 3, length 4, etc. This is the 3rd term of the above sum, a path of length 3 (4 points):



Since the longest path may be of arbitrary length, the sum is an ***infinite series***:

$$L(p' \to p) = \sum_{n=1}^{\infty} P(\bar{p}_n)$$

where $\bar{p}_n$ is a path of $n + 1$ points and length n, and $P(\bar{p}_n)$ is the radiance scattered over that path.

Can $\bar{p}_n$ be seen as a sequence? Its terms would be:

$\bar{p}_n = p_o, p_1, \ldots p_n$

where p_o is on the ***film plane or lens*** and p_n is on a ***light source***.

The definition of $P(\bar{p}_n)$:

$$P(\bar{\mathrm{p}}_n) = \underbrace{\int_A \int_A \cdots \int_A}_{n-1} L_\mathrm{e}(\mathrm{p}_n \to \mathrm{p}_{n-1}) \times \left(\prod_{i=1}^{n-1} f\left(\mathrm{p}_{i+1} \to \mathrm{p}_i \to \mathrm{p}_{i-1}\right) G(\mathrm{p}_{i+1} \leftrightarrow \mathrm{p}_i) \right) \mathrm{d}A(\mathrm{p}_2) \cdots \mathrm{d}A(\mathrm{p}_n).$$

To write it more compactly, the multiplication of the BRDF and G products is represented with $T(\bar{p}_n)$ and called the ***throughput of the path*** and is the fraction of radiance that arrives finally at p_0 after all the scattering:

$$T(\bar{\mathrm{p}}_n) = \prod_{i=1}^{n-1} f\left(\mathrm{p}_{i+1} \to \mathrm{p}_i \to \mathrm{p}_{i-1}\right) G(\mathrm{p}_{i+1} \leftrightarrow \mathrm{p}_i),$$

The radiance of a path is thus:

$$P(\bar{p}_n) = \int_A \int_A \ldots \int_A L_e(p_n \to p_{n-1})\, T(\bar{p}_n)\, dA(p2) \ldots dA(p_n)$$

We see then that the infinite series formulation of the light transport equation,

$$L(p' \to p) = \sum_{n=1}^{\infty} P(\bar{p}_n)$$

is the sum of ***integral terms*** $P(\bar{p}_n)$. Each of these integral terms $P(\bar{p}_n)$ can be estimated using a ***Monte Carlo estimator***. Each estimator uses as many samples as there are vertices in its own path, that is, n. (A bidirectional method, chapter 16, is one that computes these n samples starting at both the camera and the light source until they meet at an intermediary point in the path. A unidirectional method is one that starts either at the camera or at a light source.) This is called ***path sampling***.

14.4.5 Delta distributions in the integrand

- [ ] TODO

14.4.6 Partitioning the integrand

- [ ] TODO

14.5 Path tracing

The path tracing light transport algorithm computes the path integral form of the light transport equation.

Kajiya's path tracing is an ***unbiased Monte Carlo light transport algorithm***. It is a unidirectional method (as opposed to bidirectional) that samples the points of a path starting at the camera and ending at a light source. (Witthed's method also works this way, but it can only sample delta BSDFs and delta light sources.)

14.5.1 Overview

Let p_0 be a point on the camera and p_1 the first point intersected by a ray. Then, radiance coming from the direction of p_1 is:

$$L(p_1 \to p_0) = \sum_{n=1}^{\infty} P(\bar{p}_n)$$

The path integral formulation of the LTE has an infinite number of terms, so we can't compute them all. Instead, we terminate the sum probabilistically using ***Russian roulette*** without introducing bias:

$$\frac{1}{1-q_1}(P(\bar{p}_1) + \frac{1}{1-q_2}(P(\bar{p}_2) + \frac{1}{1-q_3}(P(\bar{p}_3) + \cdots$$

Each term i (the radiance of a path of length $i+1$) has probability q_i of terminating the sum. The $\frac{1}{1-q_i}$ weight of evaluated terms and the fact that every term has a nonzero probability $1-q$ of being evaluated is what makes the estimation unbiased.

So only a finite number of terms ends up being evaluated.

14.5.2 Path sampling

To compute $P(\bar{p}_i)$, we need to sample the vertices of the path of length $\mathrm{i}+1$. The last vertex $\mathrm{p_i}$ is ***always a point on a light source*** and p_0 is on the camera.

$P(\bar{p}_i)$ is a multiple integral over surface areas of objects in the path.

When sampling the vertices of the path, we want every point on surfaces in the scene to be chosen with equal probability (we'll see that choosing path vertices with equal probability leads to high variance, though). So we define a PDF that assigns equal probability to all points. First, the probability of sampling the ith ***object*** should be proportional to its surface area:

$$p_i = \frac{A_i}{\sum_j A_j}$$

where the denominator is the sum of the surface areas of all the objects in the scene.

Then, we want to sample points on the ith object uniformly:

$$\frac{1}{A_i}$$

The probability of sampling a given point in the scene is thus given by this PMF:

$$p_A(p_i) = \frac{A_i}{\sum_j A_j}\frac{1}{A_i} = \frac{1}{\sum_j A_j}$$

which assigns equal probability to all points in the scene. (Subscript A is for area.)

The first i points of a path of $i+1$ vertices are sampled with this PDF. The last one, though, must be a light source, so its PDF must be defined accordingly:

$$p_A(p_i) = \frac{1}{\sum_j A_{L,j}}$$

where $A_{L,j}$ is the area of the jth light source in the scene. (When the last point of the path is not sampled from among light sources, but among all the objects, the outgoing radiance of the path could be 0 if the last point happens to be on a non-emitter; this sample will introduce ***variance*** and slow down convergence.)

When 2 consecutive points on path p_n are not mutually visible, the outgoing radiance of the path is 0 because a light source can't be reached. Again, this path introduces ***variance*** in the estimate.

Other ***sources of variance*** are delta distributions in the integrand, either in the BSDF (a perfectly specular surface) or in the incident radiance L_i (a point light). When path vertices are sampled randomly with the $p_A(p_i)$ PDF, the probability that light will be reflected by p_k in the direction of p_{k-1} is 0 when the BSDF the surface of p_k is perfectly specular (a perfectly specular surface only reflects light in a single direction out of all the possible directions in the hemisphere; the probability that the $p_{k-1} - p_k$ direction of the sampled points is that single direction is always 0). **(I don't understand very well why the delta distribution of L_i of a point light would cause the same thing.)**

14.5.3 Incremental path construction

As explained earlier, sampling path points with equal probability leads to variance due to occlusion and due to delta BSDFs and light sources.

Incremental path construction samples points by sampling the BSDF at point p_k to obtain the direction that leads to a next point p_{k+1}. This next point p_{k+1} will be the closest intersectation of a ray that follows the sampled direction. This strategy ensures that consecutive vertices of the path are mutually visible.

Since the Monte Carlo estimator of the path integral form of the LTE uses a PDF over area, p_A, and incremental path construction samples the BSDFs using a PDF over solid angle, p_ω, we need to convert p_ω to p_A:

$$p_A(p_i) = p_\omega \frac{|\cos\theta_i|}{||p_{i+1} - p_i||^2}$$

The Monte Carlo estimate of a path becomes:

$$\frac{L_e(p_i \rightarrow p_{i-1}) f(p_i \rightarrow p_{i-1} \rightarrow p_{i-2}) G(p_i \leftrightarrow p_{i-1})}{p_A(p_i)} \times \left(\prod_{j=1}^{i-2} \frac{f(p_{j+1} \rightarrow p_j \rightarrow p_{j-1}) |\cos\theta_j|}{p_\omega(p_{j+1} - p_j)} \right).$$

Note that the path throughput changed: when adding the p_ω-to-p_A PDF conversion factor, some factors got eliminated. This new path throughput expression is referred to in code simply as ***beta*** β. The Monte Carlo estimate of a path is said to be the product of beta and the radiance coming from the light source at the last vertex of the path.

The last surface point in the path, vertex i, receives direct illumination from a light source (if not occluded). The light source is thus sampled the same way the direct lighting is 14.3 Direct lighting, with multiple importance sampling.

14.5.4 Implementation

The implementation differs from the theory in that paths of increasing length $n,\ n+1,\ n+2$ are not computed from scratch starting at the camera, but where the previous shorter one left off. That is, the path sampled by the estimate $P(\bar{p}_n)$ of path $\bar{p}_n$ is reused by the estimate of path $\bar{p}_{n+1}$: just one more ray is shot to determine the next vertex. This introduces correlation among the paths, which has an adverse effect in quality, but computation is more efficient.