Computer Graphics 2018 Cheat Sheet

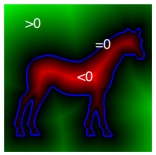
Lecture 1

Parametric representation

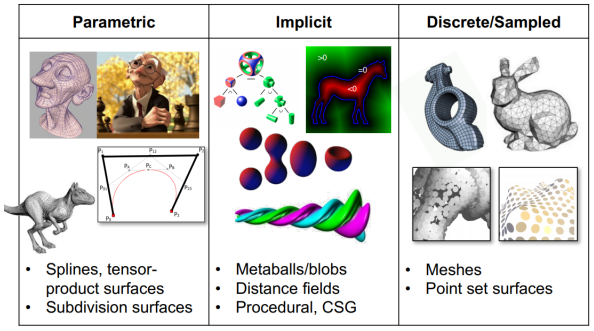
$s : \mathbb{R}^m \to \mathbb{R}^n$ with $s : \mathbb{R}^2 \to \mathbb{R}^3$ then
$s(u, v) = (x(u, v), y(u, v), y(u, v))$
Advantages: Easy to generate points on the curve / survace. Analytic formulas for derivatives. **Disadvantages:** Hard to determine inside outside, Hard to determine if a point is on the curve/survace.

Implicit curves and surfaces

Surface and Courve defined by kernel function: $f : \mathbb{R}^m \to \mathbb{R}$ **Curve in 2D:** $S = \{x \in \mathbb{R}^2 | f(x) = 0\}$
Surface in 3D: $S = \{x \in \mathbb{R}^3 | f(x) = 0\}$
Space partitioning: $\{x \in \mathbb{R}^m | f(x) > 0\} \to Outside$ then with $f(x) < 0, inside$ and lastly with $f(x) = 0$ on courve / surface. **Boolean set operations:** Can be applied like Union, Intersection ethc. $\bigcup_i f_i(x) = min f_i(x)$ and intersection: $\bigcap_i f_i(x) = max f_i(x)$
Advantages: Easy to determine inside/outside, Easy to determine if a point is on the curve/surface. **Disadvantages:** Hard to generate points on the curve / surface. Does not lend itself to (real-time) rendering.



Surface Representation Zoo



2 - Polygonal Meshes

Piecewise linear boundary representations of objects. Since linear approx we have $O(h^2)$ error. The more faces the less error obviously.

Polygon

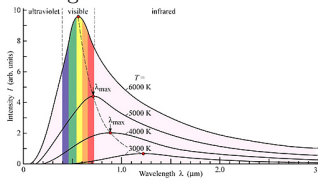
Verices: $v_0, v_1, ...v_{n-1}$ with Edges: $(v_0, v_1), ..., (v_{n-2}, v_{n-1})$ and closed if $v_0 = v_{n-1}$. **Planar:** All vertices on a plane.
Simple: Not self intersecting.
Polygonal Mesh: Set of M closed, simple polygons Q_i. The intersection of two polygons in M is either empty, vertex or an edge (so no overlap!).
$M = \langle V, E, F \rangle = \langle Vertices, Edges, Faces(Polygon) \rangle$
Boundary: Set of all edges that belong to only one polygon.
Vertex degree:

Texture Mapping

Mapping between the surface and the image. Each point (x, y, z) on the surface has mapped coordinates (u, v) in the texture image: $P : M \to [0, 1] \times [0, 1]$. $P(x, y, z) = (u, v)$. And this (u, v) then map to a color rgb defined in the image.
$T(u, v) = r, g, b$. This means that
$Color(x, y, z) = T(P(x, y, z))$..
Mesh parametererization desiderata: Minimal distortion: preserve 2D angles, distances and areas. No stretch.

Lecture 3 - Light and Matter

Incandescence: Visible light produced from heat
Black body: Completly absorbs all wavelengths of thermal radiation incident on it. Appear black and temperatur low enough to not be self-luminous. In a perfect blackbody the color spectrum of the emission is defined purely by the temperature of the material. **Planck's Law:** Defines color spectrum of a black body at specific temperature. Planck's law accurately describes black body radiation. Shown here are a family of curves for different temperatures. The classical (black) curve diverges from observed intensity at high frequencies.



Luminescence: Emission of light by a substance not resulting from heat. (chemical electrical subatomic etc)
Atomic Emission: When an atom changes its energy level (in does that in discrete steps - meaning as soon as it is fully loaded it jumps possibly multiple levels to the next layer) saving energy and then it jumps to lower level of layers and releases energy by releasing wavelengths which is what we see.
Fluorescence: occurs when light striking a surface is briefly absorbed and then re-emitted at a lower frequency. (e.g. Blacklight)
Radiometry studies the measurement of electromagnetic radiation. Radiometry assumes that light consists of photons.
State of a Photon: x: Position, $\vec{w}$: Direction of travel, λ: Wavelength. Each Photon has a method to calculate its Energy level: $E = \frac{hc}{\lambda}$ where h is Planck constant and c is speed of light.

Flux $\Phi(A)$: Total amount of radiant energy (photons) passing through a surface or space per unit of time!! $\Phi(A) = \frac{J}{s} = W$.
Irradiance $E(x)$: Flux per unit area. So we just measure the flux over unit area. $E(x) = \frac{d\Phi(A)}{dA(x)} = \frac{W}{m^2}$. Imaging just if we have a wall then average flux on that wall. Just divide through area of wall.
Radiosity: Is just the flux leaving the surface per unit area.
Radiant Intensity: Is the directional flux - flux per solid angle. We want to now exactly how much light hits our point from a given direction for this we need radiant intensity. (2 vectors imagine 3d). $I(\vec{w}) = \frac{d\Phi}{d\vec{w}} = \frac{W}{sr}$. Solid angle can be written in integral form as follows: $\int_0^{2\pi} d\phi \int_0^{\pi} sin\theta d\theta = \int d\Omega$
Radiance: Radiance is useful because it indicates how much of the power emitted, reflected, transmitted or received by a surface will be received by an optical system looking at that surface from a specified angle of view. $L_{e,\Omega} = \frac{\partial^2 \Phi_e}{\partial\Omega \, \partial A \cos\theta}$,

Overview of Quantities

• flux:	$\Phi(A)$	$\left[\frac{J}{s} = W\right]$	
• irradiance:	$E(\mathbf{x}) = \frac{d\Phi(A)}{dA(\mathbf{x})}$	$\left[\frac{W}{m^2}\right]$	
• radiosity:	$B(\mathbf{x}) = \frac{d\Phi(A)}{dA(\mathbf{x})}$	$\left[\frac{W}{m^2}\right]$	
• intensity:	$I(\vec{\omega}) = \frac{d\Phi}{d\vec{\omega}}$	$\left[\frac{W}{sr}\right]$	
• radiance:	$L(\mathbf{x}, \vec{\omega}) = \frac{d^2\Phi(A)}{\cos\theta dA(\mathbf{x}) d\vec{\omega}}$	$\left[\frac{W}{m^2 sr}\right]$	

Some real life examples:
Point light source irradiance:
$E = \frac{\phi}{4\pi r^2 = SurfaceArea}$, $\phi = \int_A E(x) dA$
Radiant Intensity: Is better described as $\phi = \int_\Omega I(w) dw$ over the unit sphere we have $I = \phi / 4\pi$
Radiance: $\phi = \int_A \int_\Omega L(x, w) cos\theta dw dA$
E(p): $E(p) = \int_\Omega L_i(p, \omega) |cost\theta| d\omega =$
$\int_0^{2\pi} \int_0^{\pi/2} L_i(p, \theta, \phi) cos\theta sin\theta d\theta d\phi = \pi L_i$ if same radiance from all directions.

Lecture 4 - Ray Tracing

Local Coordinate Frame

- Goal: create a local frame defined by normal $\vec{n}$
  - Step 1: Compute tangent $\vec{t}$
    - Zero one component of $\vec{n}$, swap the other two negating one of them and normalize
  - Step 2: Compute bi-tangent $\vec{b} = \vec{n} \times \vec{t}$
  - Step 3: Construct a "TBN" matrix: $\mathbf{M} = \begin{bmatrix} \vec{t}_x & \vec{b}_x & \vec{n}_x \\ \vec{t}_y & \vec{b}_y & \vec{n}_y \\ \vec{t}_z & \vec{b}_z & \vec{n}_z \end{bmatrix}$
- Transforming between local $\vec{a}_l$ and global $\vec{a}_g$: $\vec{a}_g = \mathbf{M} \cdot \vec{a}_l \qquad \vec{a}_l = \mathbf{M}^{-1} \cdot \vec{a}_g = \mathbf{M}^T \cdot \vec{a}_g$

Ray:
$r(t) = \vec{o} + t\vec{d}$

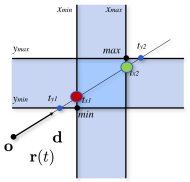
Forward ray tracng (light tracing) Trace all lights from light source and wait for a light to hit the camera / eye.
Backward ray tracing (camera tracing): Shoot light from eye (primary ray) and wait for ray to hit light source. If we shoot a ray to the lightsource directly from an intersection then it is called shadow ray. Any other next ray after intersection is called secondary ray.
Sphere equation: (implicit) $||x - c||^2 - r^2 = 0$. To intersect with ray just insert $r(t) = o + td$ in position of x and solve for t.
Plane equation (implicit): $ax + by + cz + d = 0$ (algebraic), $(x - p) \cdot n = 0$ (geometric).
Barycententric coordinates conversions: Given bary coord: $\lambda_1, \lambda_2, \lambda_3$, $x = \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3$ where x_1, x_2, x_3 are the corresponding triangle vertices.
Given cartesian coordinates rearrange the above equation by rewriting: $\lambda_3 = 1 - \lambda_2 - \lambda_1$ and then rarranging this for for the unknown factors.
Triangle equation: Where triangle is just a plane where the normal gets constructed from the 3 points (use cross product (p2 - p1) x (p3 - p1)). To test if point is inside the triangle use the barycentric coordinates and check if they sum to one and are between 0 and 1.
BRDF (bidirectional reflectance distribution function:) Ratio of outgoing light to incident light.
Diffuse Shading: Depends on surface orientation ($\vec{n}$), light position $\vec{w}_l$, material parameter (albedo param), and is independent of camera position.
$L_d(x, \vec{w}) = k_d I(\vec{w}) cos(\theta) = k_d I(\vec{w}) \vec{n} \cdot \vec{w}_l$ mind the lambertian cosine law that states that the bigger the angle between the surface and the light source the further out spread is the light and therefore the less light falls on a single spot.

Lecture 5 - Ray Tracing Acceleration

Ray-AABB Intersection

- Intersection of slabs

$\mathbf{o}_x + t_{x1}\mathbf{d}_x = x_{min}$
$\mathbf{o}_x + t_{x2}\mathbf{d}_x = x_{max}$
↓
x slabs: solve for t_{x1}, t_{x2}
$t_{x1} = \frac{x_{min} - \mathbf{o}_x}{\mathbf{d}_x}, t_{x2} = \frac{x_{max} - \mathbf{o}_x}{\mathbf{d}_x}$
if $t_{x1} > t_{x2}$: swap(t_{x1}, t_{x2})
repeat for : $t_{y1}, t_{y2}, t_{z1}, t_{z2}$
$t_{min} = \max(t_{x1}, t_{y1}, t_{z1})$
$t_{max} = \min(t_{x2}, t_{y2}, t_{z2})$
hit if: $t_{min} < t_{max}$

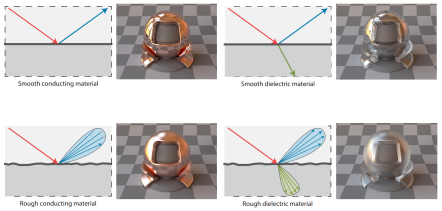


Uniform Grids: Cut scene into uniform grid and shoot ray through the grid. We can stop at first intersection. + Easy to code. , - Uniform cells do not adapt to non uniform scenes, - Hierarchical grids
KD Tree: 1) Compute bounding box and then recursively split ells using axis aligned plane. Until max depth or min num of objects. 2) When
Key Points:

1. Ray-surface intersections dominate computation in ray tracing
2. spatial pre-sorting significantly reduces ray-surface intersection (O(N) -¿ O(log(N)))
3. How to decide which is best? Uniform grids, hierarchical grids, kd-trees, bsp-trees, bounding volume hieararchies.

Lecture 7 - Appareance Models

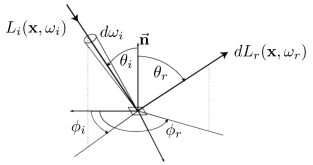
Conductors vs. Dielectrics



BRDF

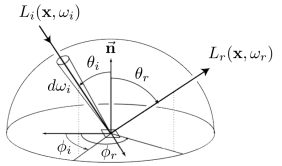
- **B**idirectional **R**eflectance **D**istribution **F**unction

$$f_r(\mathbf{x}, \vec{\omega}_i, \vec{\omega}_r) = \frac{dL_r(\mathbf{x}, \vec{\omega}_r)}{dE_i(\mathbf{x}, \vec{\omega}_i)} = \frac{dL_r(\mathbf{x}, \vec{\omega}_r)}{L_i(\mathbf{x}, \vec{\omega}_i) \cos\theta_i \, d\vec{\omega}_i} \quad [1/sr]$$



Reflection Equation

- The Reflection Equation describes a *local illumination* model
  - reflected radiance due to incident illumination from all directions



$$L_r(\mathbf{x}, \vec{\omega}_r) = \int_{H^2} f_r(\mathbf{x}, \vec{\omega}_i, \vec{\omega}_r) L_i(\mathbf{x}, \vec{\omega}_i) \cos\theta_i \, d\vec{\omega}_i$$

BRDFs Properties

- Physically-based BRDFs:
  - Helmholtz reciprocity

$$f_r(\mathbf{x}, \vec{\omega}_i, \vec{\omega}_r) = f_r(\mathbf{x}, \vec{\omega}_r, \vec{\omega}_i)$$
$$f_r(\mathbf{x}, \vec{\omega}_r \leftrightarrow \vec{\omega}_i)$$

  - Energy conservation

$$\int_{H^2} f_r(\vec{\omega}_i, \vec{\omega}_r) \cos\theta_i \, d\omega_i \leq 1, \quad \forall \, \vec{\omega}_r$$

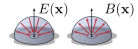
Isotropic material: If the brdf is unchanged as we rotate the material around the normal. (else anisotropic). Isotropic brdf are functions of 3 variables, incoming angle, outcoming angle and azimuth.

Ideal Diffuse BRDF

- For Lambertian reflection, the BRDF is a constant:

$$L_r(\mathbf{x}, \vec{\omega}_r) = \int_{H^2} f_r(\mathbf{x}, \vec{\omega}_i, \vec{\omega}_r) L_i(\mathbf{x}, \vec{\omega}_i) \cos\theta_i \, d\vec{\omega}_i$$
$$L_r(\mathbf{x}) = f_r \int_{H^2} L_i(\mathbf{x}, \vec{\omega}_i) \cos\theta_i \, d\vec{\omega}_i$$
$$L_r(\mathbf{x}) = f_r \, E(\mathbf{x})$$

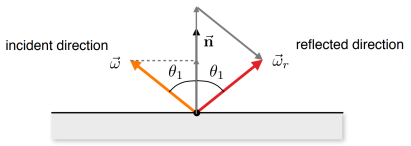
- If *all* incoming light is reflected:



$$E(\mathbf{x}) = B(\mathbf{x})$$
$$E(\mathbf{x}) = \int_{H^2} L_r(\mathbf{x}) \cos\theta \, d\vec{\omega}$$
$$E(\mathbf{x}) = L_r(\mathbf{x}) \int_{H^2} \cos\theta \, d\vec{\omega}$$
$$E(\mathbf{x}) = L_r(\mathbf{x}) \, \pi$$

Lambertian reflection: We just have a constant brdf (albedo) denoted ρ:
$L_r(x, \vec{w_r} = \int_{H2} f_r(x, \vec{w_i}, \vec{w_r}) L_i(x, \vec{w_i}) cos\theta_i d\vec{w_i}) = L_r(x) = \frac{\rho}{\pi} \int_{H2} L_i(x, \vec{w_i}) cos\theta_i d\vec{w_i}$

Reflected Direction



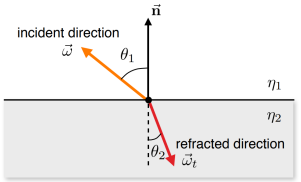
$$\vec{\omega}_r = 2\vec{\mathbf{n}} \cos\theta - \vec{\omega} = 2(\vec{\mathbf{n}} \cdot \vec{\omega})\vec{\mathbf{n}} - \vec{\omega}$$

Index of

Refaction: $\frac{speedoflightinvacuum}{speedoflightinmedium}$

Ideal Specular Refraction

- Snell's law



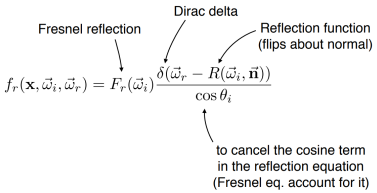
$$\eta_1 \sin\theta_1 = \eta_2 \sin\theta_2$$

Fresnel equation for Dielectrics:

$$\rho_{||} = \frac{\eta_2 cos\theta_1 - \eta_1 cos\theta_2}{\eta_2 cos\theta_1 + \eta_1 cos\theta_2},\ \rho_{\perp} = \frac{\eta_1 cos\theta_1 - \eta_2 cos\theta_2}{\eta_1 cos\theta_1 + \eta_2 cos\theta_2}$$

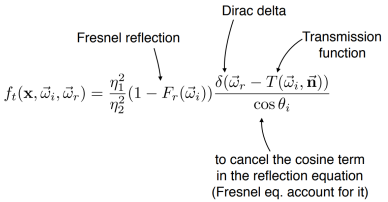
BRDF of Specular Reflection

- Ideal specular reflection:



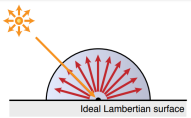
BTDF of Specular Refraction

- Ideal specular refraction (transmission):

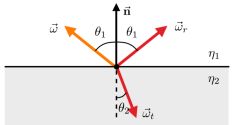


Recap: Basic BRDF Models

- Diffuse



- Specular Reflection and Refraction



Lecture 8 - Microfacet Theory

Normalized Phong: Give a sennse of roughness by blurring the reflected rays in a cone about the mirror direction.

$f_r(\vec{w_o}, \vec{w_i}) = \frac{e+2}{2\pi}(\vec{w_r} \cdot \vec{w_o})^e$ with $\vec{w_r} = (2\vec{n}(\vec{n} \cdot \vec{w_i}) - \vec{w_i})$ which is just normal reflected direction as we know it.

Blinn-Phong: Blurr in normal domain instead of reflection directions: $f_r(\vec{w_o}, \vec{w_i}) = \frac{e+2}{2\pi}(\vec{w_h} \cdot \vec{n})^e$ with $w_h = \frac{\vec{w_i} + \vec{w_o}}{||\vec{w_i} + \vec{w_o}}$ is the half-way vector.

General Microfacet Model

Fresnel coefficient — Microfacet distribution — Shadowing/ masking

$$f(\vec{\omega}_i, \vec{\omega}_o) = \frac{F(\vec{\omega}_h, \vec{\omega}_o) \cdot D(\vec{\omega}_h) \cdot G(\vec{\omega}_i, \vec{\omega}_o)}{4|(\vec{\omega}_i \cdot \vec{\mathbf{n}})(\vec{\omega}_o \cdot \vec{\mathbf{n}})|}$$

$$\vec{\omega}_h = \frac{\vec{\omega}_i + \vec{\omega}_o}{\|\vec{\omega}_i + \vec{\omega}_o\|}$$

Fresnel Reflectance

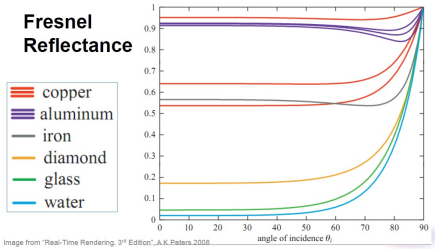


Image from "Real-Time Rendering, 3rd Edition", A.K.Peters 2008

Microfacet distribution: What fraction of faces participates in the reflection ? Prob. distr must be normalized over projected solid angle $\int_{H^2} D(\vec{w_h}) cos\theta_h d\vec{w_h} = 1$

Beckmann Distribution: Follows gaussian:

$$D(\vec{w_h}) = \frac{1}{\pi\alpha^2 cos^4\theta_h} e^{-\frac{tan^2\theta_h}{\alpha^2}}$$ The bigger α the rougher (milkier) the surfaces is gonna be.

Shadow / Masking: Since microfacet can shadow each other this term does account for that.

$$G(\vec{w}) = \frac{2}{1 + erf(s) + \frac{1}{s\sqrt{\pi}e^{-s^2}}},\ \text{with } s = \frac{1}{\alpha tan\theta},$$

$G(\vec{w_i}, \vec{w_o}) = G(\vec{w_i}) \cdot G(\vec{w_o})$.

Denominator: Correction term coming from energy conversation, jacobians.

Oren - Nayar Model: Same concept as the microfacet models but assumes that the facets are diffuse! (No analytic solution only fitted approximation)

Lecture 9 - Monte Carlo Integration

Condition:

$\int_a^b f(x) = \int_a^b p(x)\frac{f(x)}{p(x)} = \mathbb{E}[\frac{f(x)}{p(x)}] \approx \frac{1}{N}\sum_{i=1}^N \frac{f(x_i)}{p(x_i)}$ where $x_i \sim p(x)$ with the conditions that 1) $\int_a^b p(x) = 1$ and obviosuyl $p(x_i)$ can not be 0 for any of the values where $f(x) > 0$.

$$F = \int_a^b e^{\sin(3x^2)}\,dx \approx F_N = \frac{1}{N}\sum_{i=1}^N \frac{f(x_i)}{p(x_i)} \Rightarrow \frac{b-a}{N}\sum_{i=1}^N f(x_i)$$

Reducing Variance: Importance Sampling

- Importance sampling

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

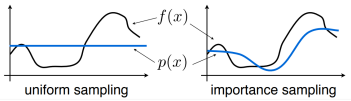
- assume $p(x) = cf(x)$

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

- estimator $\frac{f(X_i)}{p(X_i)} = \frac{1}{c} = \int f(x)dx$ zero variance!

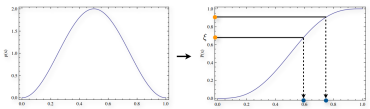
Reducing Variance: Importance Sampling

- $p(x) = cf(x)$ requires knowledge of integral, which is what we are trying to solve!
- But: If pdf is similar to integrand, variance can be significantly reduced
- **Common strategy: sample according to part of the integrand**



Sampling arbitrary distributions

- The inversion method:
  1. Compute the CDF $P(x) = \int_0^x p(x')\,dx'$
  2. Compute its inverse $P^{-1}(x)$
  3. Obtain a uniformly distributed random number ξ
  4. Compute $X_i = P^{-1}(\xi)$



Transforming Between Distributions

- Given an n-dimensional random variable $X_i \sim p_x(x)$
- Lets say we have a one-to-one (bijective) transformation T
- What is the distribution of $Y_i = T(X_i)$? But why?
- New density is:

$$p_y(y) = p_y(T(x)) = \frac{p_x(x)}{|J_T(x)|}$$

- where $|J_T(x)|$ is the absolute value of the determinant of the Jacobian matrix of T

Recipe for Area-preserving Sampling

1. Define the desired probability density of samples in a convenient coordinate system
2. Find (another) coordinate system for a convenient parameterization of the samples
3. Relate the PDFs in the two systems
   - Requires computing the determinant of the Jacobian
4. Compute marginal and conditional 1D PDFs
5. Sample 1D PDFs using the inversion method

Example sample from disk:

1. Defined desired probability: $p_c(x,y) = \frac{1}{\pi}$ if $x^2+y^2<1$, 0 otherwise.
2. Find other coordinate system for convenient parametrization of the samples: $x = rcos(\theta), y = rsin\theta$
3. relate the 2 pdfs in the two systems. $T(r,\theta) = (x,y)$, $p_c(x,y) = p_c(T(r,\theta)) = \frac{p_p(r,\theta)}{|J_T(r,\theta)|}$

   Jacobian matrix:

   $$J_T(r,\theta) = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \theta} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{bmatrix}$$

   Determinant is: $|J_T| = r(\cos^2\theta + \sin^2\theta) = r$

   Since: $p_c(x,y) = \begin{cases} \frac{1}{\pi} & x^2+y^2<1 \\ 0 & \text{otherwise} \end{cases}$ $p_c(x,y) = \frac{p_p(r,\theta)}{|J_T(r,\theta)|}$

   Therefore: $p_p(r,\theta) = rp_c(x,y) = r/\pi$
4. Computer marginal and conditional 1D Pdfs: Marginal $p(r) = \int_0^{2\pi} p_p(r,\theta)d\theta = 2r$, Conditional PDF: $p(\theta|r) = \frac{p_p(r,\theta)}{p(r)} = \frac{1}{2\pi}$
5. Compute CDFS: $P(r), P(\theta)$ and inverting yields: $r = \sqrt{\xi_1}, \theta = 2\pi\xi_2$, $P(r) = P(R <= r) = \int_{-\infty}^{r} p(k)dk = \int_{-\infty}^{r} 2kdk = [k^2]_0^r = r^2 = y \implies \sqrt{\xi} = r$

Lecture 10 - Direct Illumination

Energy equilibrium: $L_o(x,\vec{w_o}) = L_e(x,\vec{w_o}) + L_r(x,\vec{w_o}) =$ Outgoing light at point x in direction w = Emitted + reflected light

Rendering equation:
$L_o(x,\vec{w_o}) = L_e(x,\vec{w_o}) + \int_{H^2} f_r(x,\vec{w_i},\vec{w_o})L_i(x,\vec{w_i})cos\theta_i d\vec{w_i}$

Direct illumination: If L_i directly comes from an emitter we call id direct illumination. (Recursion end). If it comes from a surface of another object we call it **indirect illumination**.

Direct illumination: $L_o(x,\vec{w_o}) = L_e(x,\vec{w_o}) + \int_{H^2} f_r(x,\vec{w_i},\vec{w_o})L_e(r(x,\vec{w_i}),-\vec{w_i})cos\theta_i d\vec{w_i}$ So we just care about reflection actually. Note that we should not integrate over the hole hemisphere but just over the angle that actually could hit our eye via reflection. (solid angle Ω). Instead to sample over direction we can just sample from the lightsource which leads to a much more efficiency in many cases:

$L_r(x,z) = \int_{A_e} f_r(x,y,z)L_e(y,x)V(x,y)\frac{|cos\theta_i||cos\theta_o|}{||x-y||^2}dA(y)$

Point light: Omnidirectional emission from a single point. Intensity: $I = \frac{\Phi}{4\pi}$

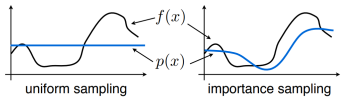
Sphere light: Irradiance is independent of radius! (Assuming it always emits the same power)

Lecture 11 - Importance sampling / MIS

Importance Sampling

- Placing samples intelligently reduces variance

$$\langle L_r(\mathbf{x},\vec{\omega}_r)^N\rangle = \frac{1}{N}\sum_{k=1}^{N}\frac{f_r(\mathbf{x},\vec{\omega}_{i,k},\vec{\omega}_r)L_i(\mathbf{x},\vec{\omega}_{i,k})\cos\theta_{i,k}\,d\vec{\omega}_{i,k}}{p_\Omega(\vec{\omega}_{i,k})}$$



Importance Sampling Emissive Surfaces:

$$L_r(\mathbf{x},\mathbf{z}) = \int_{A_e} f_r(\mathbf{x},\mathbf{y},\mathbf{z})L_e(\mathbf{y},\mathbf{x})V(\mathbf{x},\mathbf{y})\frac{|\cos\theta_i|\,|\cos\theta_o|}{\|\mathbf{x}-\mathbf{y}\|^2}dA(\mathbf{y})$$

Integrate over emissive surfaces only

Average Sampling: Sample from multiple distributions by averaging their PDF: $\frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{0.5(p_1(x_i)+p_2(x_i))}$ when sampling later from the average pdf just sample one of the pdfs first uniformly and then specifically from that distribution.

MIS: So of course just averaging is not the best solution. So weighted average would be good for M possible strategies.

$\mathbb{E}[F] = \sum_{s=1}^{M}\frac{1}{N_s}\sum_{i=1}^{N_s} w_s(x_i)\frac{f(x_i)}{p_s(x_i)}$, with $\sum_{s=1}^{M} w_s(x) = 1$

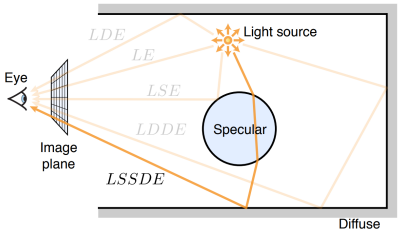
Balance Heuristic: $w_s(x) = \frac{N_s p_s(x)}{\sum_j N_j p_j(x)}$

Power heuristic: $w_s(x) = \frac{(N_s p_s(x))^\beta}{\sum_j (N_j p_j(x))^\beta}$

Fireflies: We have very high variance when the pdf is not proportional to the integrand. If we have rare samples with huge contribution: $F^N = \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i) = big}{p(x_i) = small}$ So strategy is that every pdf(strategy) should be proportional to a part of the integral.

Lecture 12 - Global illumination

Heckbert's Classification



More Heckbert:

1. Direct Illumination: $L(D|S)E$
2. Indirect Illumination: $L(D|S)(D|S)+E$
3. Classical Ray tracing: $LDS*E$
4. Full Global Illumination: $L(D|S)*E$
5. Diffuse inter-reflections: $LDD+E$
6. Caustics: $LS+DE$

Russian Roulette: Termination of recursive algorithm: $E[F'] = (1-q)\cdot(\frac{E[F]}{1-q}) + q\cdot 0 = E[F]$ where $F = \frac{F}{1-q}$ if $\xi > q$ for the termination probability q and 0 otherwise. But be careful, it always increases variance but it always reduces time per sample and can improve efficiency (if small contribution, samples tend to be terminated).

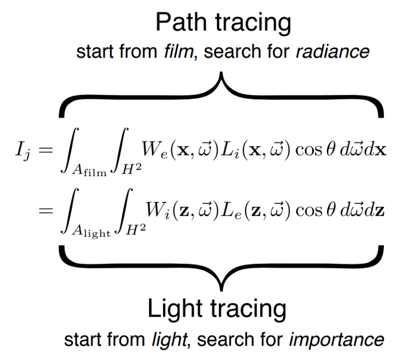
Path Tracing

$$L(\mathbf{x},\vec{\omega}) = L_e(\mathbf{x},\vec{\omega}) + L_d(\mathbf{x},\vec{\omega}) + L_i(\mathbf{x},\vec{\omega})$$

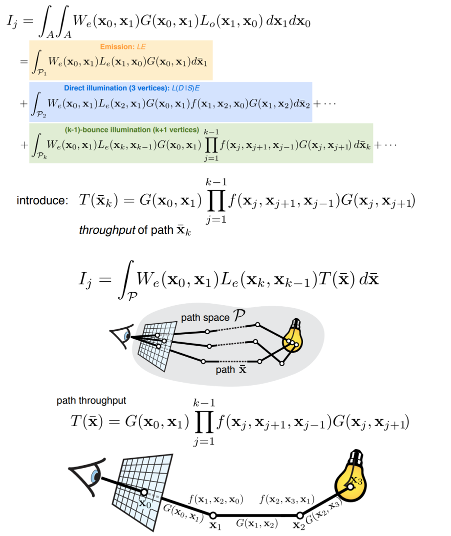
```
color shade (point x, normal n)
{
    for all lights  // direct illumination
        L_d += contribution from light;

    if rand() > q  // indirect illumination
        ω' = random direction in hemisphere above n;
        L_i += brdf * shade(trace(x,ω')) * dot(n,ω') / (p(ω'));

    if not last bounce specular  // prevent double-counting

        return L_e + L_d + L_i / (1-q);

    return L_e + L_d + L_i / (1-q);
}
```

Path Tracing - Summary:

1. \+ Full solution to the rendering equation, simple to implement
2. \- Slow converge,ce (4x more samples to half error), Robustness issues, (caustics (LS+DE) is problematic), No reuse or caching of computation, General sampling issues.



Path Integral Form of Measurement Eq



Monte Carlo Path Estimator:
$\frac{1}{N}\sum_{i=1}^{N}\frac{W_e(x_{i,0},x_{i,1})L_e(x_{i,k},x_{i,k-1})T(\overline{x_i})}{p(\overline{x_i})}$ with
$p(\overline{x_i}) = p(x_0, x_1, ..., x_{k-1}, x_k)$

Lecture 13 - Sampling and Antialising

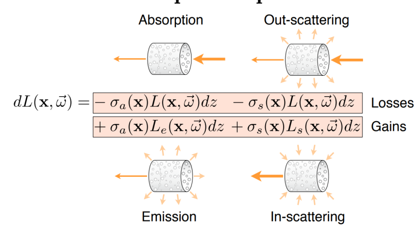
Sampling Algorithms:

1. Random sampling: Iterate: Randomly pick a point
2. Stratified sampling: Lay grid over area to sample, then For each startum / grid area, Randomly pick a point in that grid
3. Stratified sampling - correlated: For each strata randomly pick the same offset
4. Dart throwing: Iterate: Randomly pick a point, if it is not within the region of others, add the point

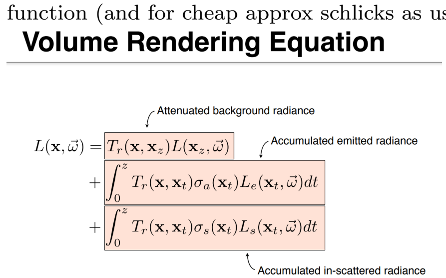
Low discrepancy patterns, discrepancy: Ratio of space vs Ratio of points

Lecture 14 - Participating Media

Properties: Given:
Absorbtion coefficient: $\sigma_a(x)$,
Scattering coefficient: $\sigma_s(x)$,
Phase function: $f_p(x, \vec{w}', \vec{w})$.
Derived:
Extinction coefficient: $\sigma_t(x) = \sigma_a(x) + \sigma_s(x)$
Albedo: $\alpha(x) = \frac{\sigma_s(x)}{\sigma_t(x)}$
Mean free path: $\mathbb{E}[1/\sigma_t(x)]$
Transmittance: $T_r(x, y) = e^{-\int_0^{||x-y||}\sigma_t(t)dt}$
Radiative Transport Equation:



Beer-Lambert Law: Expresses remaining radiance after traveling finite distance trough a medium with constant extinction coefficient. The fraction is referred to as **transmittance**:
$\frac{L_z = (radiance\, after\, travelling\, z)}{L_0 = (Radiance\, at\, beginning\, of\, beam)} = e^{-\sigma_t z}$ **In**
Homogenous volume: $T_r(x, y) = e^{-\sigma_t||x-y||}$
Heteroeneous (spatially varying σ_t):
$T_r(x, y) = e^{-\int_0^{||x-y||}\sigma_t(t)dt}$
Transmittance is multiplicative.
Phase function f_p Describes distribution of scattered light analog of BRDF but for scattering in media.
Unfiorm scattering: $f_p(\vec{w}', \vec{w}) = \frac{1}{4\pi}$
For anistropic scattering: use henyey greenstein phase function (and for cheap approx schlicks as usual xD)



Delta Trackgin:

1. Unbiased technique for free-path sampling
2. inspired by rejection sampling
3. Idea: Add a fictitious volume, combined volume (real + fictitious) is homogeneous, generate tentative free-paths analytically, probabilitcally reject / accept collisions based on local concentrations of real vs fictitious volumes.

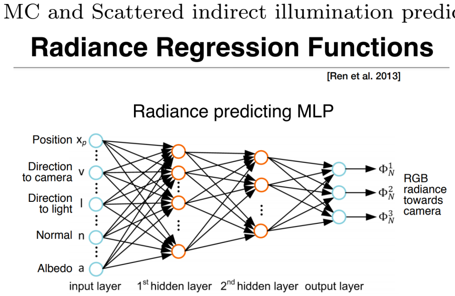
Volumetric Photon Mapping (same as regular photon mapping)

1. Shoot photons from light sources
2. Construct a blanced kD-tree for the photons
3. Assign a radius for each photon (photon-discs)
4. Create acceleration structure of photon spheres.
5. Render: For each ray through the medium accumulate all photon discs that intersect ray. (Beam density estimation)

There are some differences between tracing photons and tracing rays. The main difference is that when a photon undergoes refraction, the power carried by the photon does not change. In contrast, the radiance of a ray must be weighted by the square of the relative indices of refraction **Russian roulette photon:** 300 photons with power 1.0 W hit a surface with reflectance 50% instead of 300 photons with power 0.5 W RR will make 150 photons continue with power 1.0 Very important! $p = 1 - min(1, \phi'/\phi)$ if rand() ¡ p terminate else $\phi' = \phi'/(1-p)$
Radiance estimation:
$L_r(x, \vec{w}) \approx \sum_{p=1}^{k} f_r(x, \vec{w_p}, \vec{w})\frac{\phi_p}{A = \pi r_k^2}$ where r is radius to k nearest photon. Also possible is to just define radius r and search for all photons in radius r.
Convergence conditions: Infinitely small radius, infinite number of nearby photons (infinite storage requirement!)
Progressive Photon Map: Shrink the kernel from image to image and then at the end average all images together. The radius for image is given by: $r_{i+1}^2 = \frac{i+\alpha}{i+1}r_i^2$ with α low is fast shrinking of kernel and high α is slow shrinking. convergence need alpha in 0,1. Converges withour requiring infinite memory storage!

Lecture 15 - ML in rendering

General approach: Scattered direct illumination estimated via MC and Scattered indirect illumination predicted using



NN.
Path Guiding: Idea Learn a coarse approximiation of incident radiance $L_i(x, \vec{w_i})$ and use it for directional sampling. Learning is performed in a preprocess or progressively as we render the scene.

Lecture 16 - Subsurface Scattering

Common in all non metals

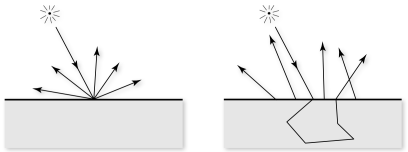
BRDF vs BSSRDF

The BRDF

$$\frac{dL_o(\mathbf{x},\vec{\omega}_o)}{d\Phi(\mathbf{x},\vec{\omega}_i)} = f_r(\mathbf{x},\vec{\omega}_i,\vec{\omega}_o)$$

The BSSRDF

$$\frac{dL_o(\mathbf{x}_o,\vec{\omega}_o)}{d\Phi(\mathbf{x}_i,\vec{\omega}_i)} = S(\mathbf{x}_i,\vec{\omega}_i;\mathbf{x}_o,\vec{\omega}_o)$$



Highly - Scattering Materials (e.g. Milk) te distribution of light approaches uniformity. g = 0.9

Diffuse approximation:

Derivation of DA from RTE

1. Radiative Transfer Equation:

$$(\vec{\omega}\cdot\nabla)L(\mathbf{x},\vec{\omega}) = -\sigma_t L(\mathbf{x},\vec{\omega}) \quad \text{Extinction}$$
$$+\sigma_s\int_{S^2} f_p(\mathbf{x},\vec{\omega}',\vec{\omega})L(\mathbf{x},\vec{\omega}')d\vec{\omega}' \quad \text{Scattering}$$
$$+Q(\mathbf{x},\vec{\omega}) \quad \text{Emission}$$

2. First-order approximation:

$$L(\mathbf{x},\vec{\omega}) \approx \frac{1}{4\pi}\phi(\mathbf{x}) + \frac{3}{4\pi}\vec{\omega}\cdot\vec{E}(\mathbf{x})$$

3. By requiring identity of the first two moments derive the following diffusion approximation:

$$-D\nabla^2\phi(\mathbf{x}) + \sigma_a\phi(\mathbf{x}) = Q(\mathbf{x}) \qquad D = \frac{1}{3\sigma_t'}$$

4. Step by step derivation: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.357.5122&rep=rep1&type=pdf

More on Subsurface scattering

Lecture 17 - Bidirectional Rendering

Radiance vs Importance: **Radiance**: Emitted from light sources, describes amount of light traveling within differential beam. **Importance** "emitted" from sensors, describes the response of the sensor to radiance traveling within a differential beam.

Path tracing
start from ***film***, search for ***radiance*** at light

$$I_j = \int_{A_{\text{film}}}\int_{H^2} W_e(\mathbf{x},\vec{\omega})L_i(\mathbf{x},\vec{\omega})\cos\theta\, d\vec{\omega}d\mathbf{x}$$
$$= \int_{A_{\text{light}}}\int_{H^2} W_i(\mathbf{z},\vec{\omega})L_e(\mathbf{z},\vec{\omega})\cos\theta\, d\vec{\omega}d\mathbf{z}$$

Light tracing
start from ***light***, search for ***importance*** at sensor

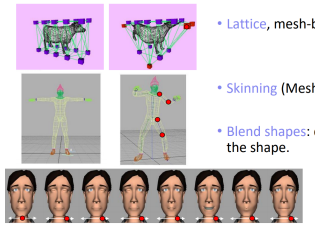
Still not robust enough for: LSDSE (difficult for any unbiased method)

Lecture 18 - Animation

Embeddings: Offest handles, lower dimensionality controls. e.g. bounding box with 4 vertices with which we can change the appearance of the underlying model.

Mesh/lattice-based: Parametric curves, surface, volumes.

Mesh-free Hierarchical deformations (Skinning - via skeleton). **Blend shapes**.



Lattice based embedding: Weights are distances to the nodes p_k at rest state. Given any basis-function: $x(u,v) = \sum_k p_k B_k(u,v)$ where B_k is bilinear function in 2D.

Skinning: Transfer the linear transformations of the joints to the vertices. Start with rest shape $V_i(0)$ and assign each vertex to joint. Deform vertices in the joint local frame of reference. $A_k = T_k R_k$ (4x4 Affine Transformation of joint k)

Blend shapes: $V_i(0)$ Neutral shape nesg vertuces u. Deformed Mesh $V_i' = V_i(0) + \sum_k \sum_i w^k V_i^k$

Lecture 19 - Image based rendering

Overview: In computer graphics and computer vision, image-based modeling and rendering (IBMR) methods rely on a set of two-dimensional images of a scene to generate a three-dimensional model and then render some novel views of this scene. **Image based modeling:** 1) Select building blocks, 2) Align them in each image 3) Solve for camera pose and block parameters.

	Acquisition	Reconstruction	Rendering	FVV range
Model-based	Few cameras	Full 3D geometry and reflectance reconstruction, difficult for complete scenes, often applied for objects of interest, exploiting a-priori knowledge or interactive operation	Classical computer graphics	Wide
Depth-based	Few cameras	Depth estimation, error prone	Depth-based view interpolation	Medium
Image-based	Dense sampling of the scene, many cameras necessary to enable navigation	Data size / sampling issues	View interpolation, light field rendering	Limited

Lecture 20 - Light field

An image is a set of rays collected at a location within a certain field of view. Whereas a **light field** is a set of rays collected at all locations to all directions. The light field is a vector function that describes the amount of light flowing in every direction through every point in space. The space of all possible light rays is given by the five-dimensional plenoptic function, and the magnitude of each ray is given by the radiance $https://en.wikipedia.org/wiki/Light\_field$

Lecture 21 - Denoising

Non Local Means

Non-local means is an algorithm in image processing for image denoising. Unlike "local mean" filters, which take the mean value of a group of pixels surrounding a target pixel to smooth the image, non-local means filtering takes a mean of all pixels in the image, weighted by how similar these pixels are to the target pixel. This results in much greater post-filtering clarity, and less loss of detail in the image compared with local mean algorithms

$u(p) = \frac{1}{C(p)}\sum_{q\in\Omega} v(q)f(p,q)$, with

$u(p) bzw. v(p) = value at that specific pixel.$, and

$f(p,q) = e^{-\frac{|B(q)-B(p)|^2}{h^2}}$ is the gaussian kernel but the difference is over the spatial difference but over the difference of the neighborhoods: $B(p) = \frac{1}{|R(p)|}\sum_{i\in R(p)} v(i)$ where Ω is the

Bilinear Filtering

A bilateral filter is a non-linear, edge-preserving, and noise-reducing smoothing filter for images. It replaces the intensity of each pixel with a weighted average of intensity values from nearby pixels. This weight can be based on a Gaussian distribution. Crucially, the weights depend not only on Euclidean distance of pixels, but also on the radiometric differences (e.g., range differences, such as color intensity, depth distance, etc.). This preserves sharp edges.

Filter:

$$I^{\text{filtered}}(x) = \frac{1}{W_p}\sum_{x_i\in\Omega} I(x_i)f_r(\|I(x_i)-I(x)\|)g_s(\|x_i-x\|),$$

with **normalization term:**

$$W_p = \sum_{x_i\in\Omega} f_r(\|I(x_i)-I(x)\|)g_s(\|x_i-x\|)$$

$I^{filtered}$ is filtered image, I is original image to be filtered, x are coordinates of the current pixel to be filtered, Ω is the window centered at x, f_r is the range kernel for smoothing differences in intensities, g_s is the spatial kernel for smoothing differences in coordinates.

alt =

$$"w(i,j,k,l) = \exp\left(-\frac{(i-k)^2+(j-l)^2}{2\sigma_d^2} - \frac{\|I(i,j)-I(k,l)\|^2}{2\sigma_r^2}\right),"$$

with the sigmas as smoothing parameters. And after calculating the weights, normalization is important.

Appendix A - Rewriting the Light integral

$L(x\to x') = L(x,w)$ where x, x' are both points on scene surfaces and $w = \hat{x'-x}$ is the unit vector pointing from x to x'. Then the bsdf can be rewritten as:

$f_s(x\to x'\to x'') = f_s(x', w_i\to w_o)$ where $w_i = x - x'$ and

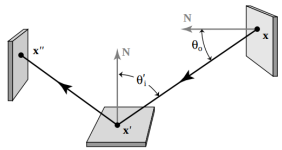
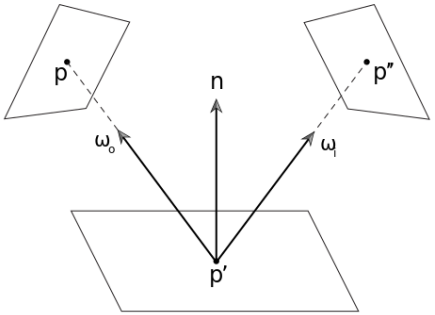


Figure 8.1: Geometry for the light transport equation in three-point form.

$w_o = x'' - x'$

Then the three-point form of the light transport can be rewritten as: $L(x'\to x'') = L_e(x'\to x'') + \int_{\mathcal{M}} L(x\to x')f_s(x\to x'\to x'')G(x\leftrightarrow x')dA(x)$

Area Integral Exitant radiance from a point p' to a point p: $L(p' \to p) = L(p', \omega)$ with $\hat{w} = p - p'$ (if p, p' visible). THEN BSDF goes to: $f(p'' \to p' \to p) = f(p', \omega_o, \omega_i)$ with $\omega_o = x - x'$ and $\omega_i = x'' - x'$



The three-point form of the light transport equation converts the integral to be over the domain of points on surfaces in the scene, rather than over directions over the sphere. It is a key transformation for deriving the path integral form of the light transport equation.
Transforming the integral: Multiply by the jacobian that relates solid angle to area in order to transform the light from integral over direction to one over surface area which is: $|cos\theta'|/r^2$. This and visibility and source $cost\theta$ gets combined into a single G term: $G(p \leftrightarrow p') = V(p \leftrightarrow p')\frac{|cos\theta||cos\theta'|}{||p-p'||^2}$.
Substituting this into the light transport equation leads to: $L(p' \to p) = L_e(p' \to p) + \int_A f(p'' \to p' \to p)L(p'' \to p')G(p'' \leftrightarrow p')dA(p'')$

The Integral over Paths

One of the main motivations for using path space is that it provides an expression for the value of a measurement as an explicit integral over paths, as opposed to the unwieldy recursive definition resulting from the energy balance equation. So we define it by:
$L(p_1 \to p_0) = L_e(p_1 \to p_0) + \int_A L_e(p_2 \to p_1)f(p_2 \to p_1 \to p_0)G(p_2 \leftrightarrow p_1)dA(p_2) + \int_A \int_A L_e(p_3 \to p_2)f(p_3 \to p_2 \to p_1)G(p_3 \leftrightarrow p_2) \times f(p_2 \to p_1 \to p_0)G(p_2 \leftrightarrow p_1)dA(p3)dA(p2)+...$
Not that we only multiply the path with no further light multiplication.

This infinite sum can be written compactly as

$$L(\mathrm{p}_1 \to \mathrm{p}_0) = \sum_{n=1}^{\infty} P(\bar{\mathrm{p}}_n).$$

$P(\bar{\mathrm{p}}_n)$ gives the amount of radiance scattered over a path $\bar{\mathrm{p}}_n$ with $n+1$ vertices,

$$\bar{\mathrm{p}}_n = \mathrm{p}_0, \mathrm{p}_1, \ldots, \mathrm{p}_n,$$

where p_0 is on the film plane or front lens element and p_n is on a light source, and

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

Before we move on, we will define one additional term that will be helpful in the subsequent discussion. The product of a path's BSDF and geometry terms is called the *throughput* of the path; it describes the fraction of radiance from the light source that arrives at the camera after all of the scattering at vertices between them. We will denote it by

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

so

$$P(\mathrm{p}_n) = \underbrace{\int_A \int_A \cdots \int_A}_{n-1} L_e(\mathrm{p}_n \to \mathrm{p}_{n-1}) T(\bar{\mathrm{p}}_n) \, \mathrm{d}A(\mathrm{p}_2) \cdots \mathrm{d}A(\mathrm{p}_n).$$

Appendix B - Transforming between Distributions

Given: $X_i \sim p_X(x)$, and $Y_i = y(X_i)$
Question: What is distribution of new variable Y_i?
Derivation: $\implies P(Y \leq y(x)) = P(X \leq x)$ this implies that $P_y(y) = P_y(y(x)) = P_X(x)$ then
$P_Y(y)\frac{dy}{dx} = p_X(x) \implies p_Y(y) = |\frac{dy}{dx}|^{-1} p_X(x)$ for multidimensional: $p_Y(y) = p_Y(T(x)) = \frac{p_X(x)}{|J_T(x)|}$

Appendix C - Some mathy definitions

Solid angle: Area of a set of points on the unit sphere. (This points can be specified by 2 angles. Azimuth and Zenith.)
Integral from direction into spherical: $dw = sin(\theta)d\theta d\phi$
Integral from directions into area: $d\omega = \frac{dAcos\theta}{r^2}$ where dA is some area and θ is angle between area-normal and r is the distance between the point p from where we look at the surface and the surface.