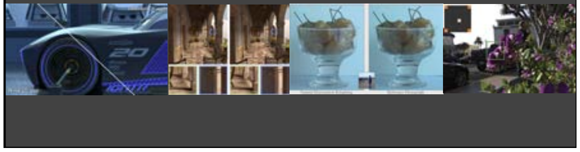
Sampling and Reconstruction of Visual Appearance

CSE 274 [Winter 2018], Lecture 3

Ravi Ramamoorthi

http://www.cs.ucsd.edu/~ravir



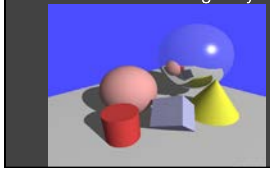
Motivation: Monte Carlo Rendering

- Key application area for sampling/reconstruction
- Modern methods for denoising now popular
- 1-3 order of magnitude speedups in mature area
- Denoising now standard in production rendering
- This week: Basic background in rendering
  - Reflection and Rendering Equations
  - Monte Carlo Integration
  - Path Tracing (Basic Monte Carlo rendering method)
  - A review for those who have taken CSE 168

Illumination Models

Local Illumination
- Light directly from light sources to surface
- No shadows (cast shadows are a global effect)

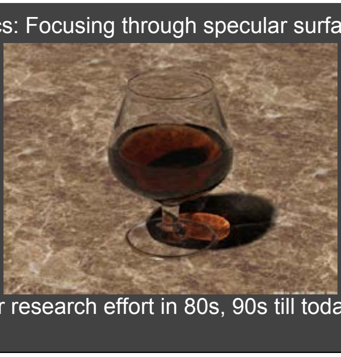
Global Illumination: multiple bounces (indirect light)
- Hard and soft shadows
- Reflections/refractions (already seen in ray tracing)
- Diffuse and glossy interreflections (radiosity, caustics)



Some images courtesy Henrik Wann Jensen

Caustics

Caustics: Focusing through specular surface



- Major research effort in 80s, 90s till today

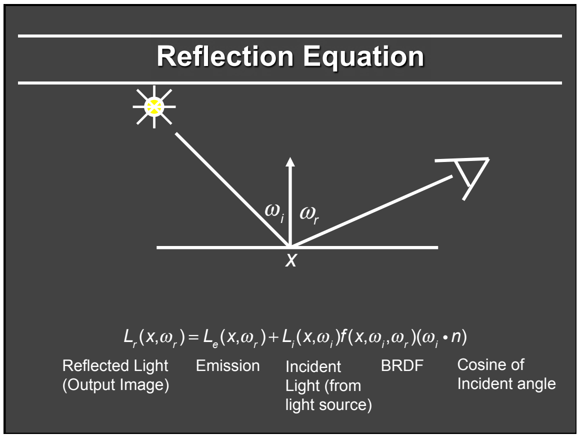
Overview of lecture

- ***Theory*** for all global illumination methods (ray tracing, *path tracing*, radiosity)
- We derive ***Rendering Equation*** [Kajiya 86]
  - Major theoretical development in field
  - Unifying framework for all global illumination
  - Introduced Path Tracing: core rendering method
- Discuss existing approaches as special cases

Fairly theoretical lecture (but important). Not well covered in textbooks (though see Eric Veach's thesis). See reading if you are interested.

Outline

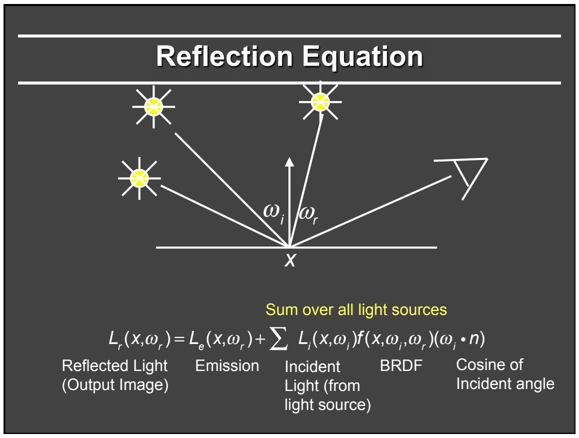
- *Reflectance Equation*
- *Global Illumination*
- *Rendering Equation*
- As a general Integral Equation and Operator
- Approximations (Ray Tracing, Radiosity)
- Surface Parameterization (Standard Form)



Reflection Equation

$$L_r(x,\omega_r) = L_e(x,\omega_r) + L_i(x,\omega_i) f(x,\omega_i,\omega_r)(\omega_i \bullet n)$$

Reflected Light (Output Image)	Emission	Incident Light (from light source)	BRDF	Cosine of Incident angle

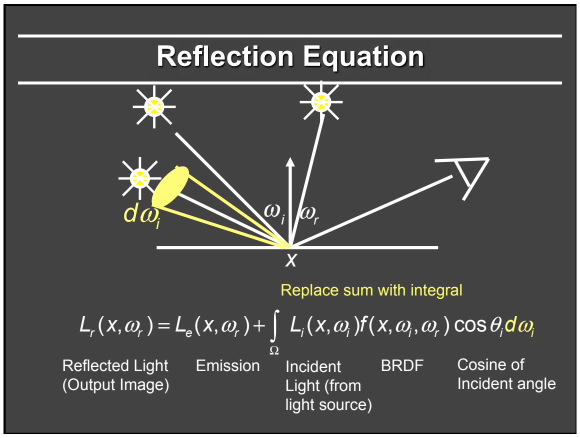


Reflection Equation

Sum over all light sources

$$L_r(x,\omega_r) = L_e(x,\omega_r) + \sum L_i(x,\omega_i) f(x,\omega_i,\omega_r)(\omega_i \bullet n)$$

Reflected Light (Output Image)	Emission	Incident Light (from light source)	BRDF	Cosine of Incident angle

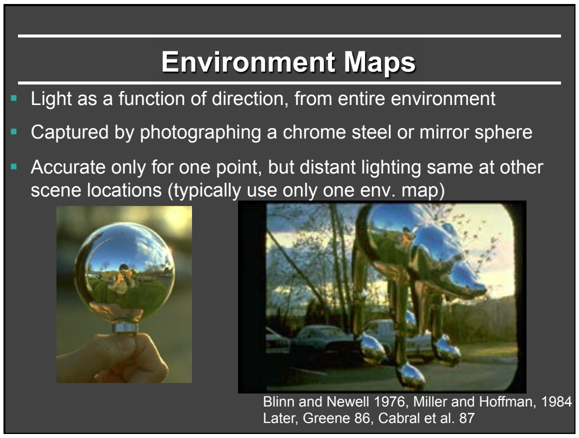


Reflection Equation

Replace sum with integral

$$L_r(x,\omega_r) = L_e(x,\omega_r) + \int_\Omega L_i(x,\omega_i) f(x,\omega_i,\omega_r) \cos\theta_i d\omega_i$$

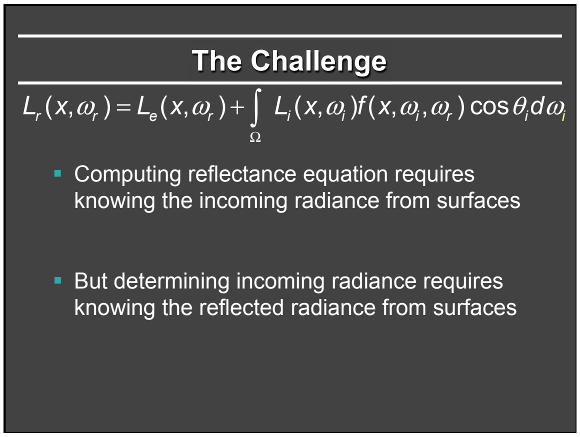
Reflected Light (Output Image)	Emission	Incident Light (from light source)	BRDF	Cosine of Incident angle



Environment Maps

- Light as a function of direction, from entire environment
- Captured by photographing a chrome steel or mirror sphere
- Accurate only for one point, but distant lighting same at other scene locations (typically use only one env. map)

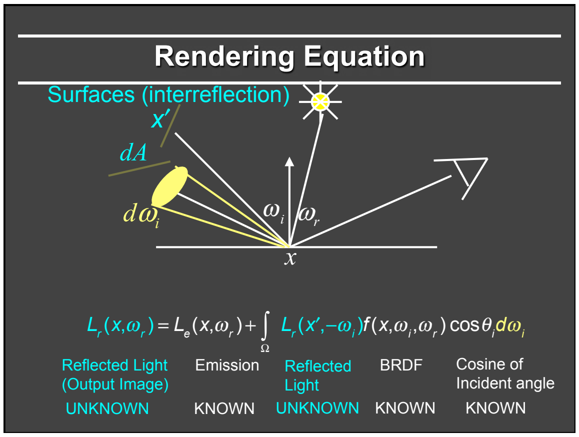
Blinn and Newell 1976, Miller and Hoffman, 1984
Later, Greene 86, Cabral et al. 87



The Challenge

$$L_r(x,\omega_r) = L_e(x,\omega_r) + \int_\Omega L_i(x,\omega_i) f(x,\omega_i,\omega_r) \cos\theta_i d\omega_i$$

- Computing reflectance equation requires knowing the incoming radiance from surfaces
- But determining incoming radiance requires knowing the reflected radiance from surfaces



Rendering Equation

$$L_r(x,\omega_r) = L_e(x,\omega_r) + \int_\Omega L_r(x',-\omega_i) f(x,\omega_i,\omega_r) \cos\theta_i d\omega_i$$

Reflected Light (Output Image)	Emission	Reflected Light	BRDF	Cosine of Incident angle
UNKNOWN	KNOWN	UNKNOWN	KNOWN	KNOWN

Outline

- Reflectance Equation (review)
- Global Illumination
- Rendering Equation
- *As a general Integral Equation and Operator*
- *Approximations (Ray Tracing, Radiosity)*
- Surface Parameterization (Standard Form)

Rendering Equation (Kajiya 86)

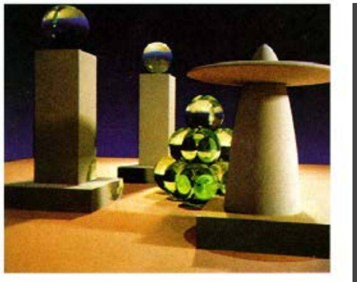


Figure 6. A sample image. All objects are neutral grey. Color on the objects is due to caustics from the green glass balls and color bleeding from the base polygon.

Rendering Equation as Integral Equation

$$L_r(x,\omega_r) = L_e(x,\omega_r) + \int_\Omega L_r(x',-\omega_i)\boxed{f(x,\omega_i,\omega_r)\cos\theta_i d\omega_i}$$

Reflected Light (Output Image)	Emission	Reflected Light	BRDF	Cosine of Incident angle
UNKNOWN	KNOWN	UNKNOWN	KNOWN	KNOWN

Is a Fredholm Integral Equation of second kind [extensively studied numerically] with canonical form

$$l(u) = e(u) + \int l(v)\boxed{K(u,v)dv}$$

Kernel of equation

Linear Operator Theory

- Linear operators act on functions like matrices act on vectors or discrete representations

$$h(u) = (M \circ f)(u)$$

M is a linear operator.
f and h are functions of u

- Basic linearity relations hold

a and b are scalars
f and g are functions

$$M \circ (af + bg) = a(M \circ f) + b(M \circ g)$$

- Examples include integration and differentiation

$$(K \circ f)(u) = \int k(u,v) f(v)\, dv$$

$$(D \circ f)(u) = \frac{\partial f}{\partial u}(u)$$

Linear Operator Equation

$$l(u) = e(u) + \int l(v)\boxed{K(u,v)dv}$$

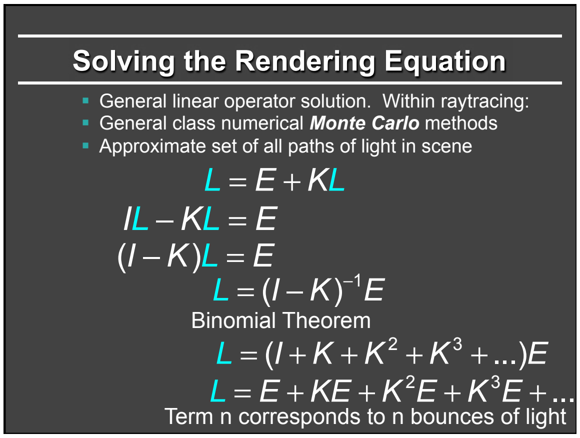
Kernel of equation
Light Transport Operator

$$L = E + KL$$

Can be discretized to a simple matrix equation [or system of simultaneous linear equations] (L, E are vectors, K is the light transport matrix)

Solving the Rendering Equation

- Too hard for analytic solution, numerical methods
- Approximations, that compute different terms, accuracies of the rendering equation
- Two basic approaches are ray tracing, radiosity. More formally, Monte Carlo and Finite Element. Today Monte Carlo path tracing is core rendering method
- Monte Carlo techniques sample light paths, form statistical estimate (example, path tracing)
- Finite Element methods discretize to matrix equation



Solving the Rendering Equation

- General linear operator solution. Within raytracing:
- General class numerical ***Monte Carlo*** methods
- Approximate set of all paths of light in scene

$$L = E + KL$$

$$IL - KL = E$$

$$(I - K)L = E$$

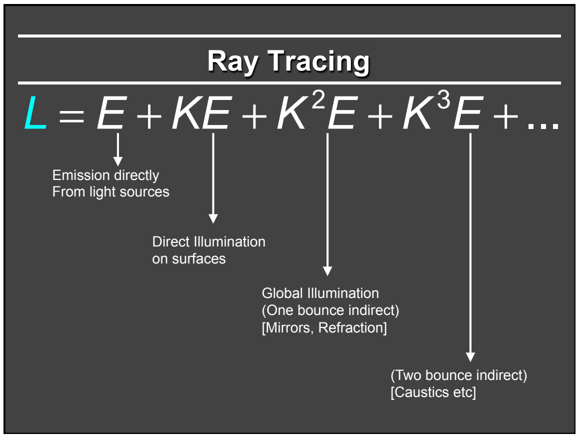
$$L = (I - K)^{-1}E$$

Binomial Theorem

$$L = (I + K + K^2 + K^3 + ...)E$$

$$L = E + KE + K^2E + K^3E + ...$$

Term n corresponds to n bounces of light



Ray Tracing

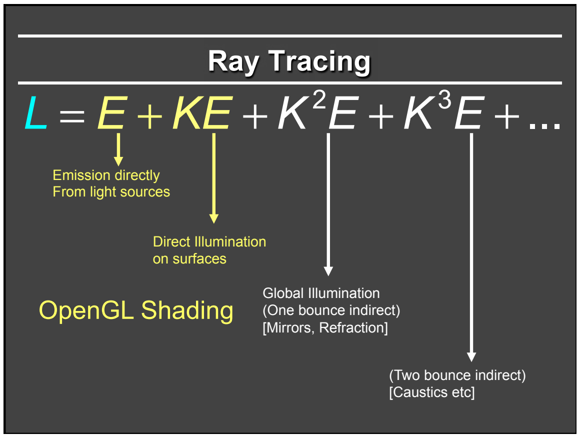
$$L = E + KE + K^2E + K^3E + ...$$

Emission directly
From light sources

Direct Illumination
on surfaces

Global Illumination
(One bounce indirect)
[Mirrors, Refraction]

(Two bounce indirect)
[Caustics etc]



Ray Tracing

$$L = E + KE + K^2E + K^3E + ...$$

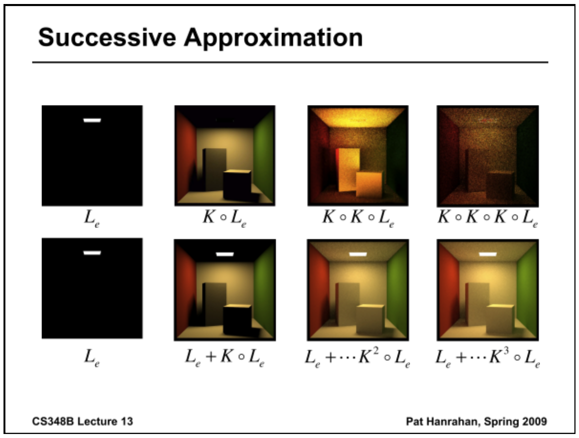
Emission directly
From light sources

Direct Illumination
on surfaces

OpenGL Shading

Global Illumination
(One bounce indirect)
[Mirrors, Refraction]

(Two bounce indirect)
[Caustics etc]

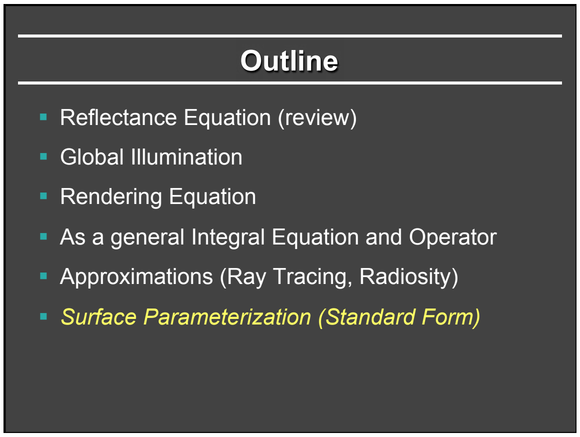


Successive Approximation

L_e
$K \circ L_e$
$K \circ K \circ L_e$
$K \circ K \circ K \circ L_e$
L_e
$L_e + K \circ L_e$
$L_e + \cdots K^2 \circ L_e$
$L_e + \cdots K^3 \circ L_e$

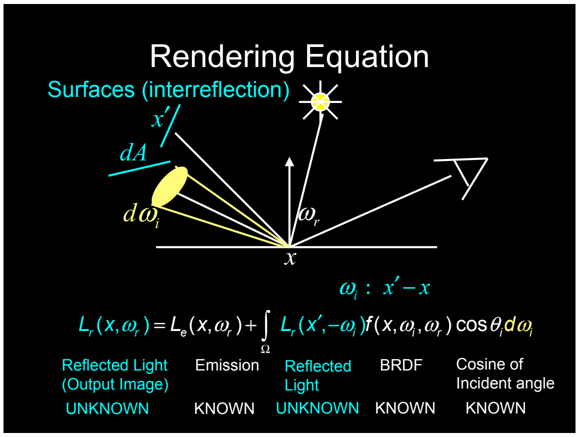
CS348B Lecture 13

Pat Hanrahan, Spring 2009



Outline

- Reflectance Equation (review)
- Global Illumination
- Rendering Equation
- As a general Integral Equation and Operator
- Approximations (Ray Tracing, Radiosity)
- *Surface Parameterization (Standard Form)*



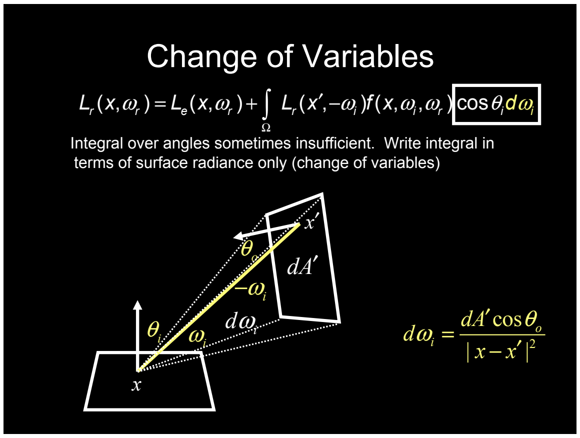
Rendering Equation

Surfaces (interreflection)

$$\omega_i : x' - x$$

$$L_r(x,\omega_r) = L_e(x,\omega_r) + \int_\Omega L_r(x',-\omega_i) f(x,\omega_i,\omega_r)\cos\theta_i d\omega_i$$

Reflected Light (Output Image)	Emission	Reflected Light	BRDF	Cosine of Incident angle
UNKNOWN	KNOWN	UNKNOWN	KNOWN	KNOWN

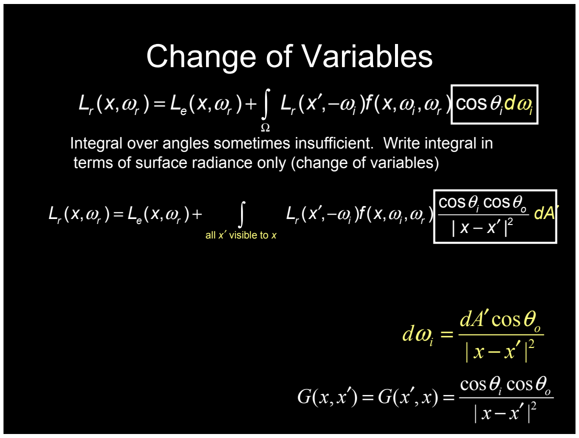


Change of Variables

$$L_r(x,\omega_r) = L_e(x,\omega_r) + \int_{\Omega} L_r(x',-\omega_i) f(x,\omega_i,\omega_r) \boxed{\cos\theta_i d\omega_i}$$

Integral over angles sometimes insufficient. Write integral in terms of surface radiance only (change of variables)

$$d\omega_i = \frac{dA'\cos\theta_o}{|x-x'|^2}$$



Change of Variables

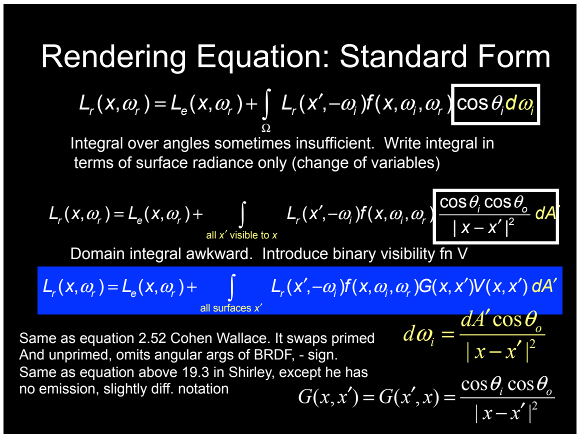
$$L_r(x,\omega_r) = L_e(x,\omega_r) + \int_{\Omega} L_r(x',-\omega_i) f(x,\omega_i,\omega_r) \boxed{\cos\theta_i d\omega_i}$$

Integral over angles sometimes insufficient. Write integral in terms of surface radiance only (change of variables)

$$L_r(x,\omega_r) = L_e(x,\omega_r) + \int_{\text{all } x' \text{ visible to } x} L_r(x',-\omega_i) f(x,\omega_i,\omega_r) \boxed{\frac{\cos\theta_i \cos\theta_o}{|x-x'|^2} dA'}$$

$$d\omega_i = \frac{dA'\cos\theta_o}{|x-x'|^2}$$

$$G(x,x') = G(x',x) = \frac{\cos\theta_i\cos\theta_o}{|x-x'|^2}$$



Rendering Equation: Standard Form

$$L_r(x,\omega_r) = L_e(x,\omega_r) + \int_{\Omega} L_r(x',-\omega_i) f(x,\omega_i,\omega_r) \boxed{\cos\theta_i d\omega_i}$$

Integral over angles sometimes insufficient. Write integral in terms of surface radiance only (change of variables)

$$L_r(x,\omega_r) = L_e(x,\omega_r) + \int_{\text{all } x' \text{ visible to } x} L_r(x',-\omega_i) f(x,\omega_i,\omega_r) \boxed{\frac{\cos\theta_i \cos\theta_o}{|x-x'|^2} dA'}$$

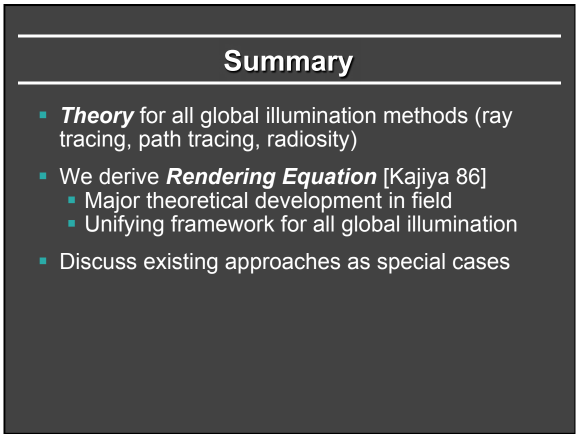
Domain integral awkward. Introduce binary visibility fn V

$$L_r(x,\omega_r) = L_e(x,\omega_r) + \int_{\text{all surfaces } x'} L_r(x',-\omega_i) f(x,\omega_i,\omega_r) G(x,x') V(x,x')\, dA'$$

Same as equation 2.52 Cohen Wallace. It swaps primed
And unprimed, omits angular args of BRDF, - sign.
Same as equation above 19.3 in Shirley, except he has
no emission, slightly diff. notation

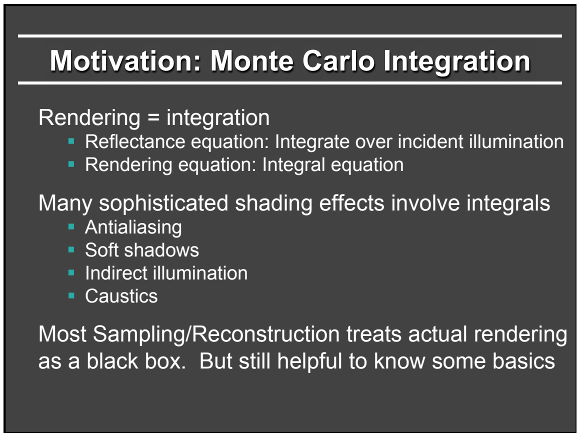
$$d\omega_i = \frac{dA'\cos\theta_o}{|x-x'|^2}$$

$$G(x,x') = G(x',x) = \frac{\cos\theta_i\cos\theta_o}{|x-x'|^2}$$



Summary

- ***Theory*** for all global illumination methods (ray tracing, path tracing, radiosity)
- We derive ***Rendering Equation*** [Kajiya 86]
  - Major theoretical development in field
  - Unifying framework for all global illumination
- Discuss existing approaches as special cases

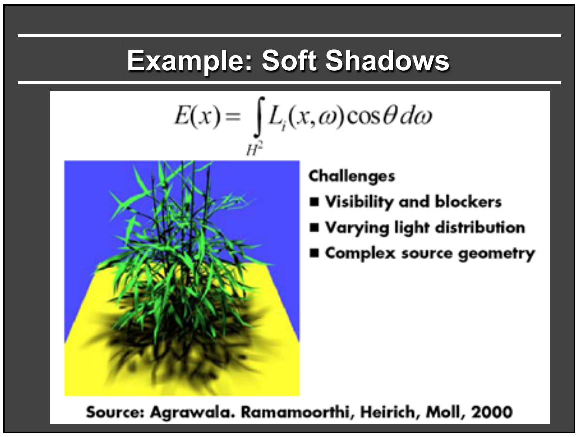


Motivation: Monte Carlo Integration

Rendering = integration
- Reflectance equation: Integrate over incident illumination
- Rendering equation: Integral equation

Many sophisticated shading effects involve integrals
- Antialiasing
- Soft shadows
- Indirect illumination
- Caustics

Most Sampling/Reconstruction treats actual rendering as a black box. But still helpful to know some basics

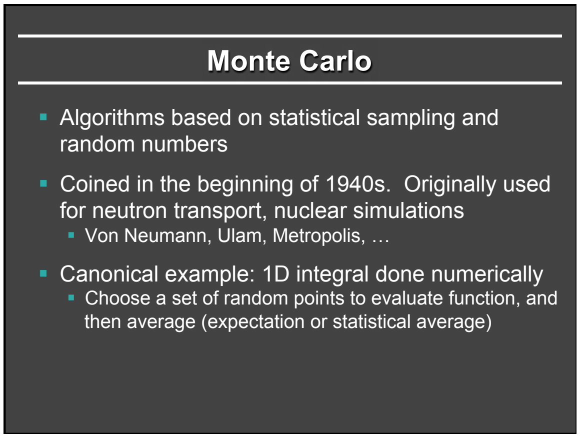


Example: Soft Shadows

$$E(x) = \int_{H^2} L_i(x,\omega)\cos\theta\, d\omega$$

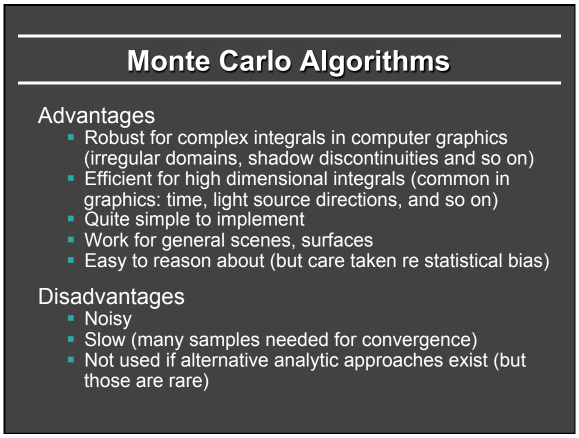
Challenges
- **Visibility and blockers**
- **Varying light distribution**
- **Complex source geometry**

Source: Agrawala. Ramamoorthi, Heirich, Moll, 2000



Monte Carlo

- Algorithms based on statistical sampling and random numbers
- Coined in the beginning of 1940s. Originally used for neutron transport, nuclear simulations
  - Von Neumann, Ulam, Metropolis, …
- Canonical example: 1D integral done numerically
  - Choose a set of random points to evaluate function, and then average (expectation or statistical average)



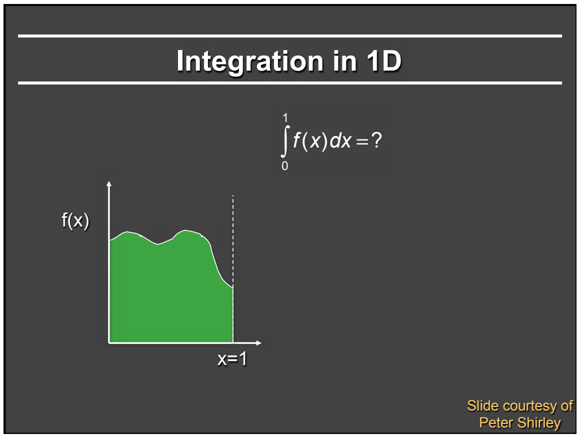
Monte Carlo Algorithms

Advantages

- Robust for complex integrals in computer graphics (irregular domains, shadow discontinuities and so on)
- Efficient for high dimensional integrals (common in graphics: time, light source directions, and so on)
- Quite simple to implement
- Work for general scenes, surfaces
- Easy to reason about (but care taken re statistical bias)

Disadvantages

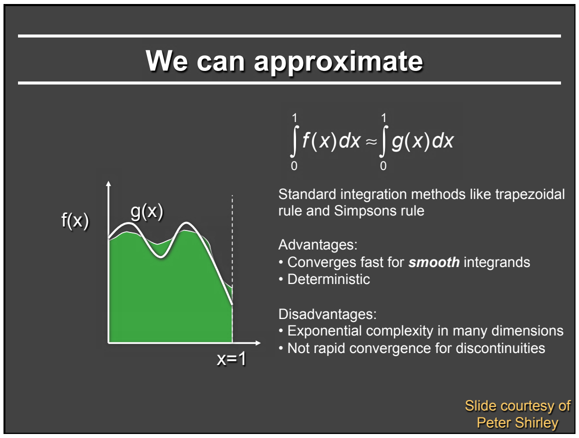
- Noisy
- Slow (many samples needed for convergence)
- Not used if alternative analytic approaches exist (but those are rare)



Integration in 1D

$$\int_0^1 f(x)dx = ?$$

Slide courtesy of Peter Shirley



We can approximate

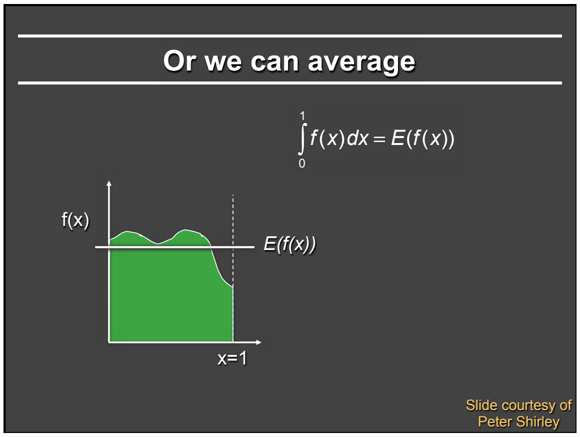
$$\int_0^1 f(x)dx \approx \int_0^1 g(x)dx$$

Standard integration methods like trapezoidal rule and Simpsons rule

Advantages:
• Converges fast for ***smooth*** integrands
• Deterministic

Disadvantages:
• Exponential complexity in many dimensions
• Not rapid convergence for discontinuities

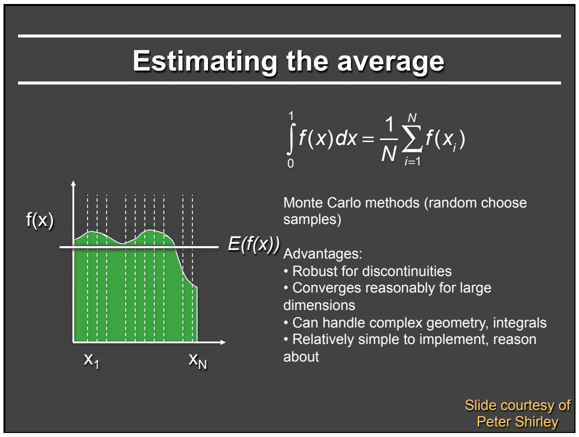
Slide courtesy of Peter Shirley



Or we can average

$$\int_0^1 f(x)dx = E(f(x))$$

Slide courtesy of Peter Shirley



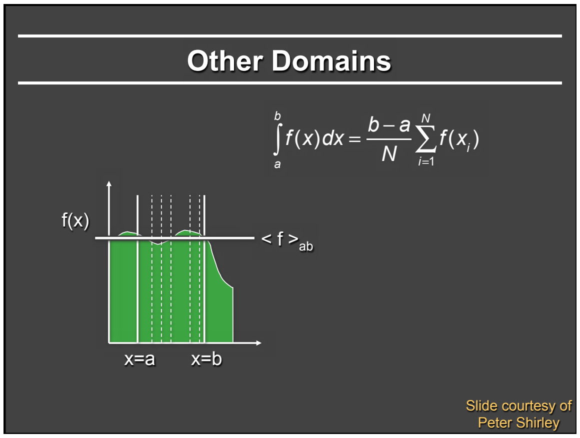
Estimating the average

$$\int_0^1 f(x)dx = \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

Monte Carlo methods (random choose samples)

Advantages:
• Robust for discontinuities
• Converges reasonably for large dimensions
• Can handle complex geometry, integrals
• Relatively simple to implement, reason about

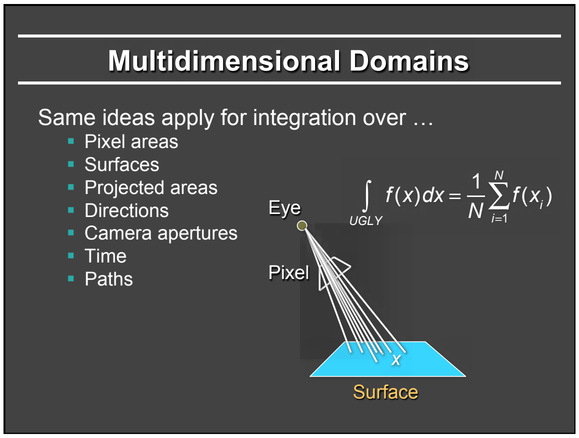
Slide courtesy of Peter Shirley



Other Domains

$$\int_a^b f(x)dx = \frac{b-a}{N}\sum_{i=1}^{N} f(x_i)$$

Slide courtesy of Peter Shirley

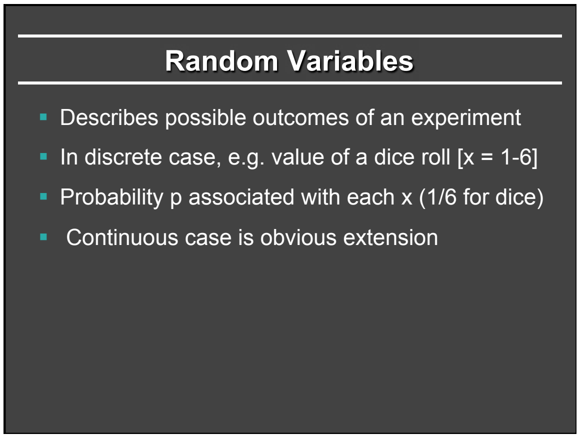


Multidimensional Domains

Same ideas apply for integration over …

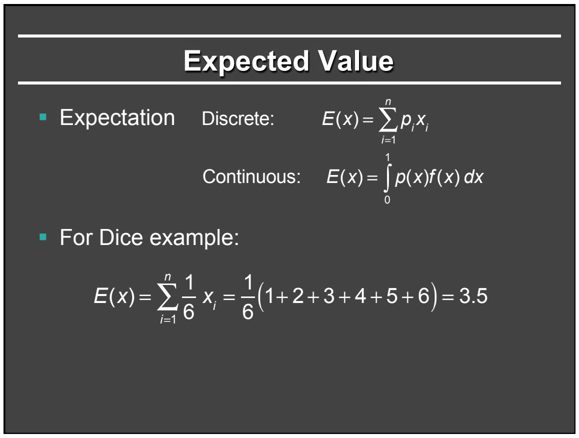
- Pixel areas
- Surfaces
- Projected areas
- Directions
- Camera apertures
- Time
- Paths

$$\int_{UGLY} f(x)dx = \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$



Random Variables

- Describes possible outcomes of an experiment
- In discrete case, e.g. value of a dice roll [x = 1-6]
- Probability p associated with each x (1/6 for dice)
- Continuous case is obvious extension



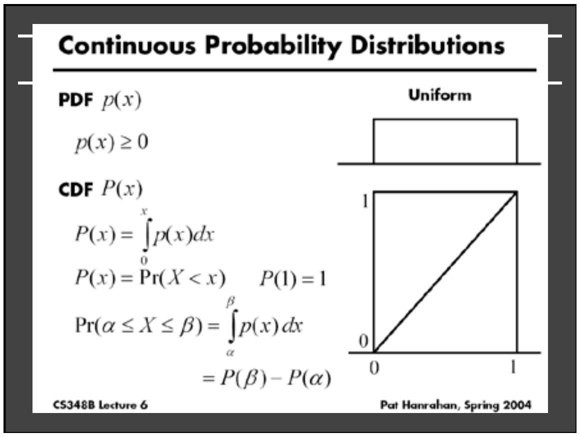
Expected Value

- Expectation Discrete: $E(x) = \sum_{i=1}^{n} p_i x_i$

  Continuous: $E(x) = \int_0^1 p(x)f(x)\,dx$

- For Dice example:

$$E(x) = \sum_{i=1}^{n} \frac{1}{6} x_i = \frac{1}{6}(1+2+3+4+5+6) = 3.5$$



Continuous Probability Distributions

PDF $p(x)$

$p(x) \ge 0$

CDF $P(x)$

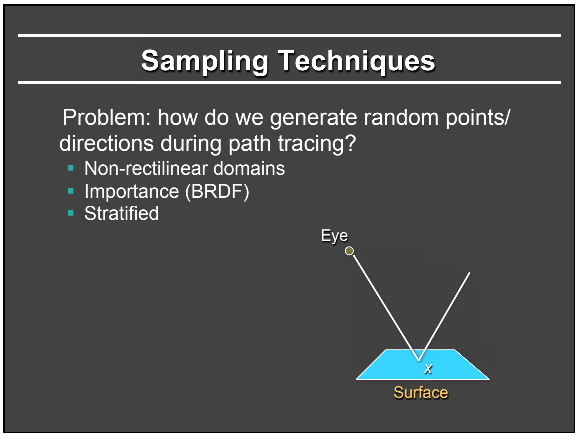
$P(x) = \int_0^x p(x)dx$

$P(x) = \Pr(X < x) \qquad P(1) = 1$

$\Pr(\alpha \le X \le \beta) = \int_\alpha^\beta p(x)dx$

$= P(\beta) - P(\alpha)$

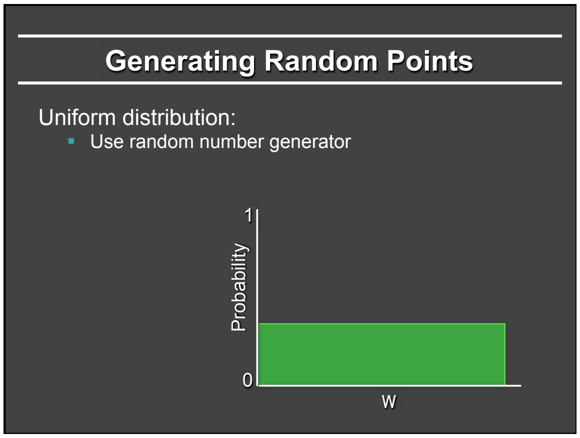
CS348B Lecture 6 Pat Hanrahan, Spring 2004



Sampling Techniques

Problem: how do we generate random points/ directions during path tracing?

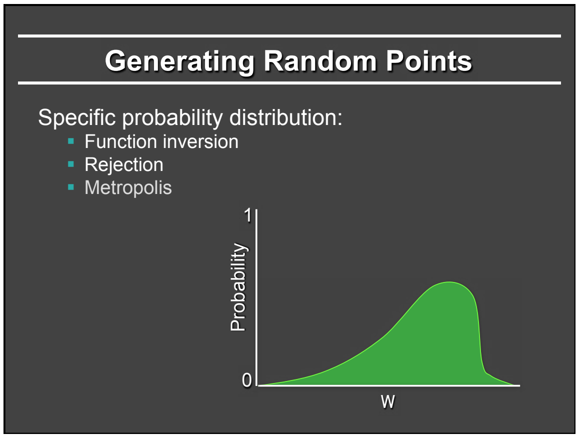
- Non-rectilinear domains
- Importance (BRDF)
- Stratified



Generating Random Points

Uniform distribution:

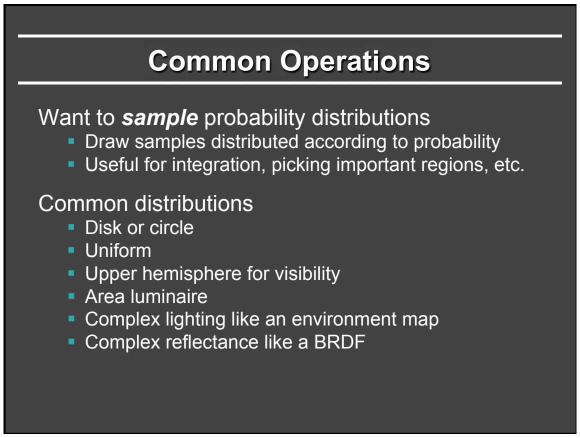
- Use random number generator



Generating Random Points

Specific probability distribution:

- Function inversion
- Rejection
- Metropolis



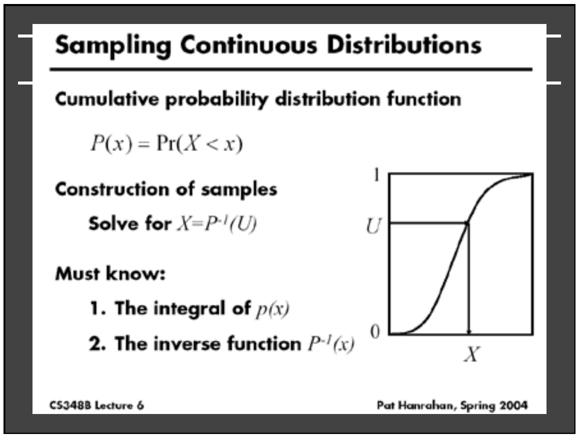
Common Operations

Want to ***sample*** probability distributions

- Draw samples distributed according to probability
- Useful for integration, picking important regions, etc.

Common distributions

- Disk or circle
- Uniform
- Upper hemisphere for visibility
- Area luminaire
- Complex lighting like an environment map
- Complex reflectance like a BRDF



Sampling Continuous Distributions

Cumulative probability distribution function

$$P(x) = \Pr(X < x)$$

Construction of samples

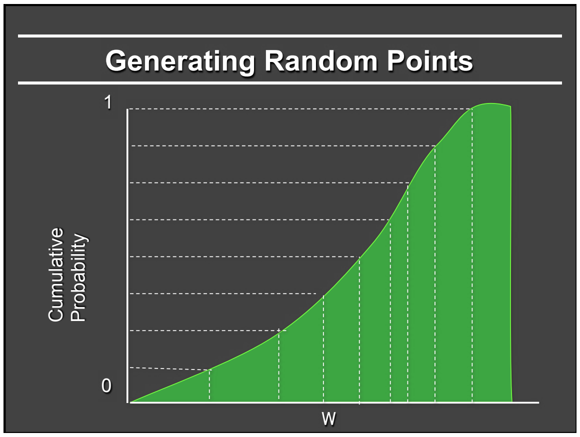
Solve for $X = P^{-1}(U)$

Must know:

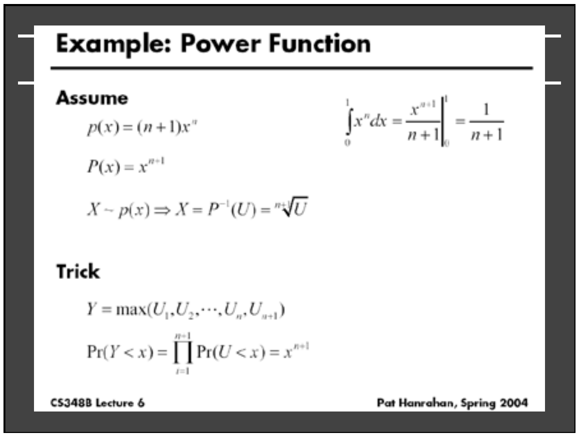
1. **The integral of** $p(x)$
2. **The inverse function** $P^{-1}(x)$

CS348B Lecture 6

Pat Hanrahan, Spring 2004



Generating Random Points



Example: Power Function

Assume

$$p(x) = (n+1)x^n$$

$$\int_0^1 x^n dx = \left. \frac{x^{n+1}}{n+1} \right|_0^1 = \frac{1}{n+1}$$

$$P(x) = x^{n+1}$$

$$X \sim p(x) \Rightarrow X = P^{-1}(U) = \sqrt[n+1]{U}$$

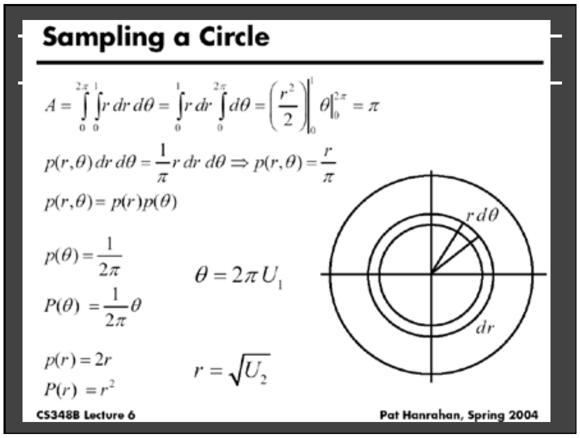
Trick

$$Y = \max(U_1, U_2, \cdots, U_n, U_{n+1})$$

$$\Pr(Y < x) = \prod_{i=1}^{n+1} \Pr(U < x) = x^{n+1}$$

CS348B Lecture 6

Pat Hanrahan, Spring 2004



Sampling a Circle

$$A = \int_0^{2\pi}\int_0^1 r\,dr\,d\theta = \int_0^1 r\,dr\int_0^{2\pi} d\theta = \left(\frac{r^2}{2}\right)\Bigg|_0^1 \theta\Big|_0^{2\pi} = \pi$$

$$p(r,\theta)\,dr\,d\theta = \frac{1}{\pi} r\,dr\,d\theta \Rightarrow p(r,\theta) = \frac{r}{\pi}$$

$$p(r,\theta) = p(r)p(\theta)$$

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

$$P(\theta) = \frac{1}{2\pi}\theta$$

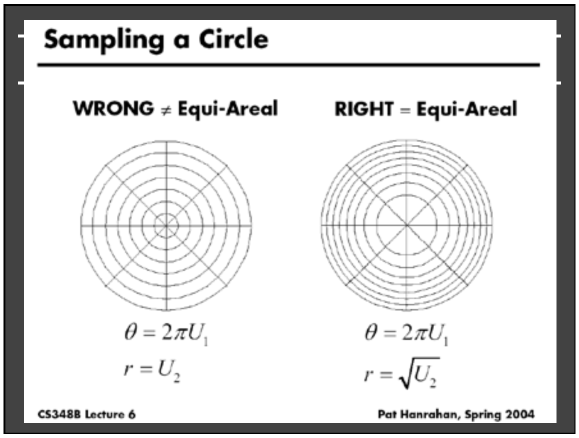
$$\theta = 2\pi U_1$$

$$p(r) = 2r$$

$$P(r) = r^2$$

$$r = \sqrt{U_2}$$

CS348B Lecture 6 Pat Hanrahan, Spring 2004



Sampling a Circle

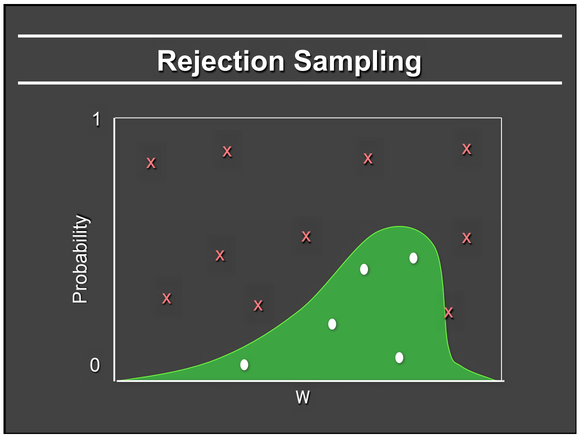
WRONG ≠ Equi-Areal

$$\theta = 2\pi U_1$$
$$r = U_2$$

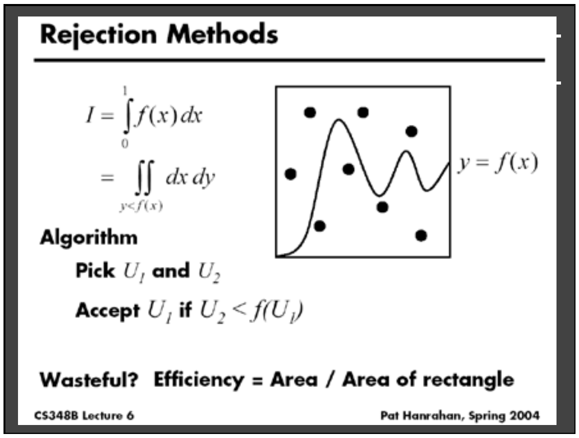
RIGHT = Equi-Areal

$$\theta = 2\pi U_1$$
$$r = \sqrt{U_2}$$

CS348B Lecture 6 Pat Hanrahan, Spring 2004



Rejection Sampling



Rejection Methods

$$I = \int_0^1 f(x)\,dx$$
$$= \iint_{y<f(x)} dx\,dy$$

$y = f(x)$

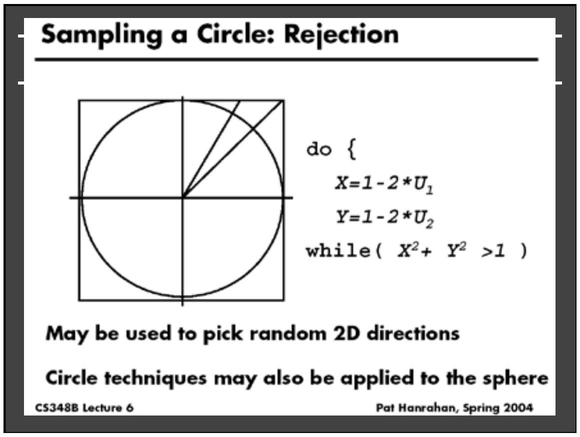
Algorithm

Pick U_1 and U_2

Accept U_1 if $U_2 < f(U_1)$

Wasteful? Efficiency = Area / Area of rectangle

CS348B Lecture 6 Pat Hanrahan, Spring 2004



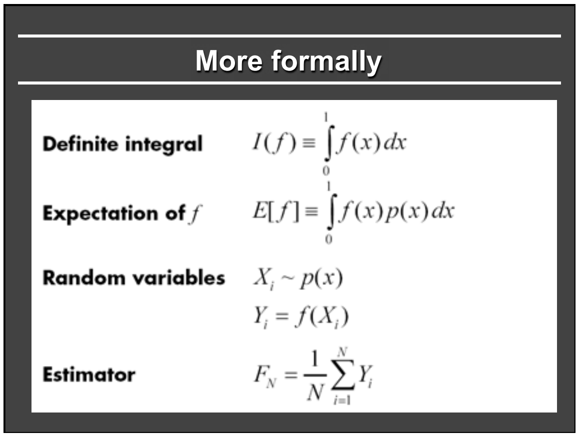
Sampling a Circle: Rejection

```
do {
   X=1-2*U1
   Y=1-2*U2
while( X²+ Y² >1 )
```

May be used to pick random 2D directions

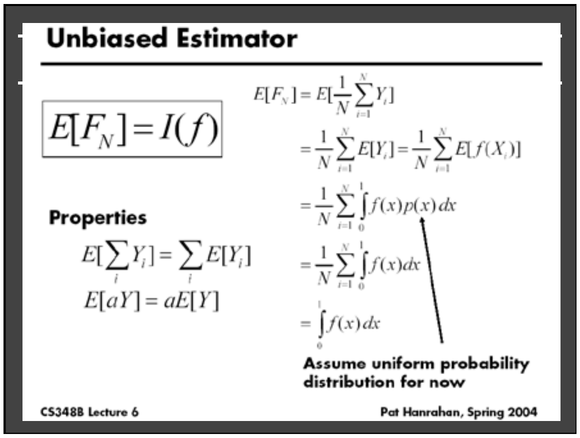
Circle techniques may also be applied to the sphere

CS348B Lecture 6 Pat Hanrahan, Spring 2004



More formally

Definite integral	$I(f) \equiv \int_0^1 f(x)\,dx$
Expectation of f	$E[f] \equiv \int_0^1 f(x)p(x)\,dx$
Random variables	$X_i \sim p(x)$ $Y_i = f(X_i)$
Estimator	$F_N = \frac{1}{N}\sum_{i=1}^{N} Y_i$



Unbiased Estimator

$$\boxed{E[F_N] = I(f)}$$

Properties

$$E[\sum_i Y_i] = \sum_i E[Y_i]$$

$$E[aY] = aE[Y]$$

$$E[F_N] = E[\frac{1}{N}\sum_{i=1}^{N} Y_i]$$

$$= \frac{1}{N}\sum_{i=1}^{N} E[Y_i] = \frac{1}{N}\sum_{i=1}^{N} E[f(X_i)]$$

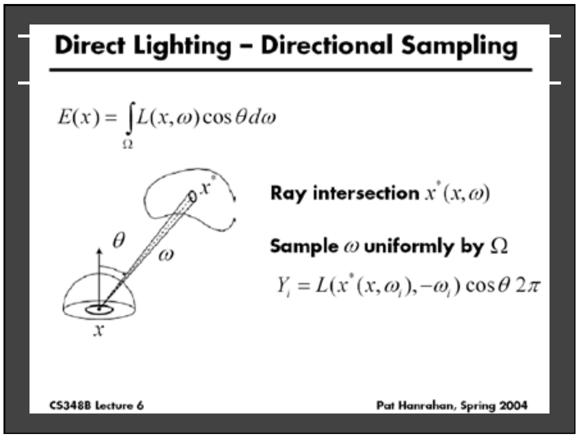
$$= \frac{1}{N}\sum_{i=1}^{N}\int_0^1 f(x)p(x)\,dx$$

$$= \frac{1}{N}\sum_{i=1}^{N}\int_0^1 f(x)dx$$

$$= \int_0^1 f(x)\,dx$$

Assume uniform probability distribution for now

CS348B Lecture 6 Pat Hanrahan, Spring 2004



Direct Lighting – Directional Sampling

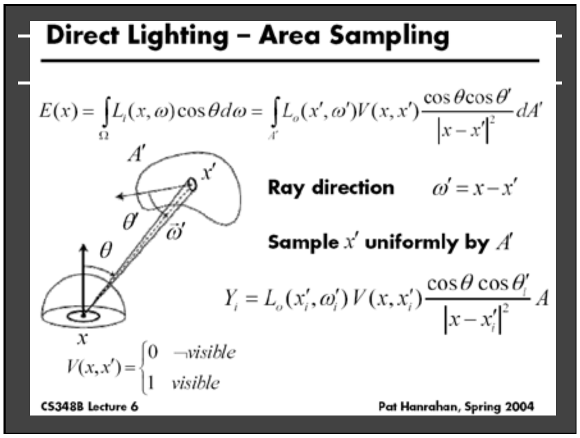
$$E(x) = \int_{\Omega} L(x,\omega)\cos\theta\, d\omega$$

Ray intersection $x^*(x,\omega)$

Sample ω **uniformly by** Ω

$$Y_i = L(x^*(x,\omega_i),-\omega_i)\cos\theta\, 2\pi$$

CS348B Lecture 6 Pat Hanrahan, Spring 2004



Direct Lighting – Area Sampling

$$E(x) = \int_{\Omega} L_i(x,\omega)\cos\theta\, d\omega = \int_{A'} L_o(x',\omega')V(x,x')\frac{\cos\theta\cos\theta'}{|x-x'|^2}dA'$$

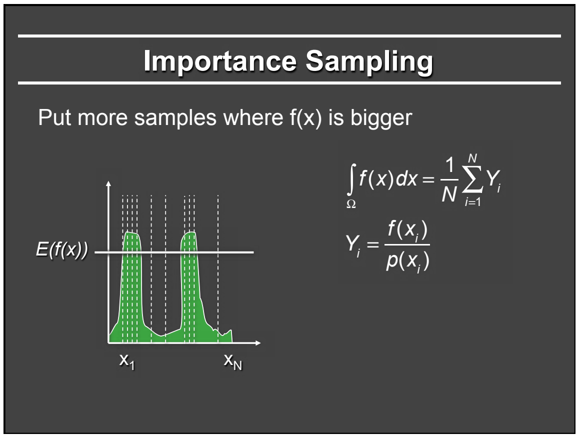
Ray direction $\omega' = x - x'$

Sample x' **uniformly by** A'

$$Y_i = L_o(x_i',\omega_i')\,V(x,x_i')\frac{\cos\theta\,\cos\theta_i'}{|x-x_i'|^2}A$$

$$V(x,x') = \begin{cases} 0 & \neg visible \\ 1 & visible \end{cases}$$

CS348B Lecture 6 Pat Hanrahan, Spring 2004

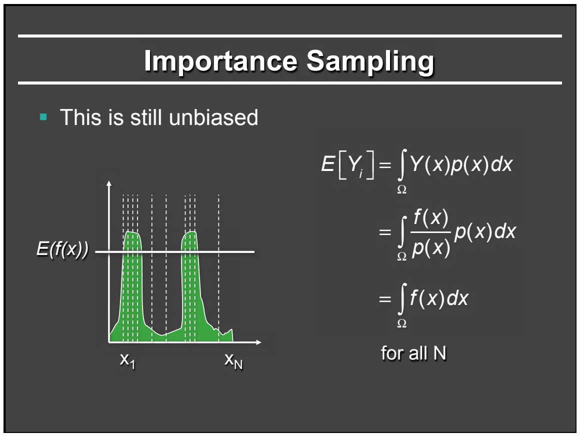


Importance Sampling

Put more samples where f(x) is bigger

$$\int_{\Omega} f(x)dx = \frac{1}{N}\sum_{i=1}^{N} Y_i$$

$$Y_i = \frac{f(x_i)}{p(x_i)}$$



Importance Sampling

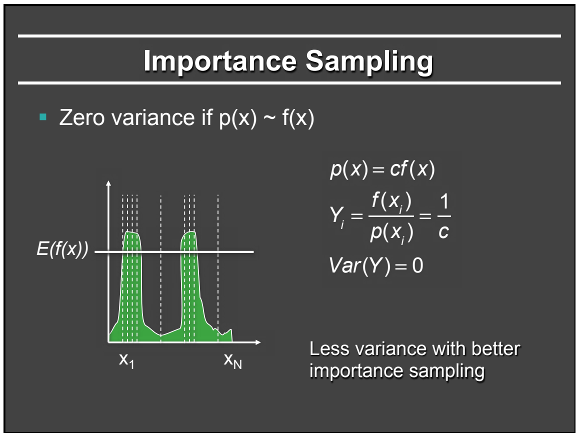
- This is still unbiased

$$E[Y_i] = \int_{\Omega} Y(x)p(x)dx$$

$$= \int_{\Omega}\frac{f(x)}{p(x)}p(x)dx$$

$$= \int_{\Omega} f(x)dx$$

for all N



Importance Sampling

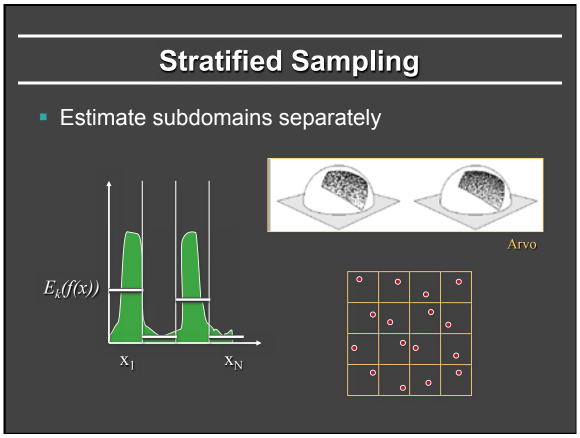
- Zero variance if p(x) ~ f(x)

$$p(x) = cf(x)$$

$$Y_i = \frac{f(x_i)}{p(x_i)} = \frac{1}{c}$$

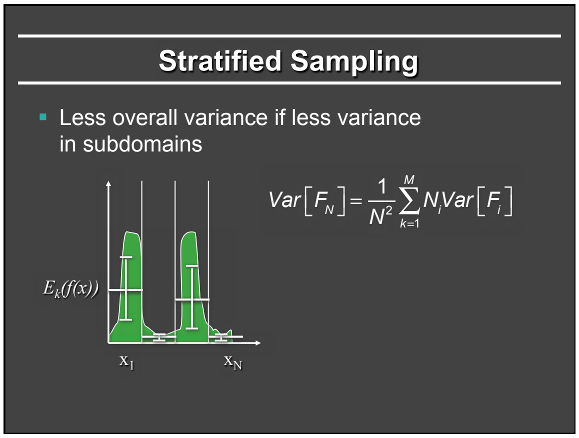
$$Var(Y) = 0$$

Less variance with better importance sampling



Stratified Sampling

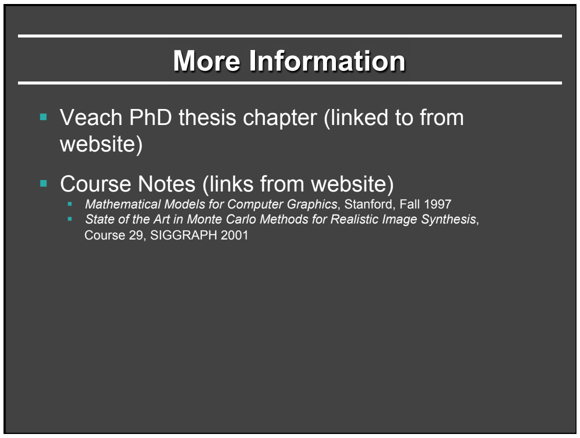
- Estimate subdomains separately



Stratified Sampling

- Less overall variance if less variance in subdomains

$$Var\left[F_N\right] = \frac{1}{N^2}\sum_{k=1}^{M} N_i Var\left[F_i\right]$$



More Information

- Veach PhD thesis chapter (linked to from website)
- Course Notes (links from website)
  - *Mathematical Models for Computer Graphics*, Stanford, Fall 1997
  - *State of the Art in Monte Carlo Methods for Realistic Image Synthesis*, Course 29, SIGGRAPH 2001