Lecture 8: Monte Carlo Rendering

Chapters 4 and 5 in Advanced GI

Fall 2004

Kavita Bala
Computer Science
Cornell University

Homework

- HW 1out, due Oct 5
- Assignments done separately
  – Might revisit this policy for later assignments

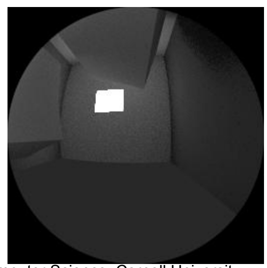
Rendering Equation

$$L(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + \int_{\Omega_x} f_r(\Psi \leftrightarrow \Theta) \cdot L(x \leftarrow \Psi) \cdot \cos(\Psi, n_x) \cdot d\omega_\Psi$$

function to integrate over all incoming directions over the hemisphere around x

Value we want

$= L_e + \int_{\Omega_x}$



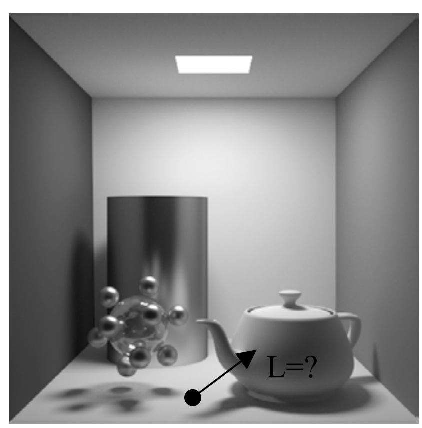
$\cdot f_r \cdot \cos$

How to compute?

$L(x \rightarrow \Theta) = ?$

Check for $L_e(x \rightarrow \Theta)$

Now add $L_r(x \rightarrow \Theta) =$



$$\int_{\Omega_x} f_r(\Psi \leftrightarrow \Theta) \cdot L(x \leftarrow \Psi) \cdot \cos(\Psi, n_x) \cdot d\omega_\Psi$$

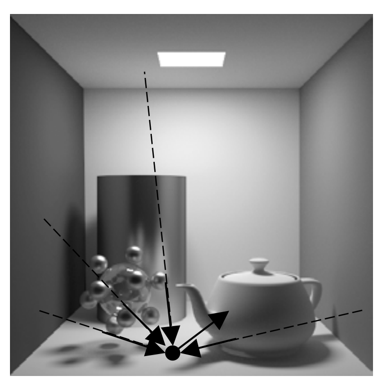
How to compute?

- Monte Carlo!
- Generate random directions on hemisphere Ω_x, using pdf $p(\Psi)$

$$L(x \to \Theta) = \int_{\Omega_x} f_r(\Psi \leftrightarrow \Theta) \cdot L(x \leftarrow \Psi) \cdot \cos(\Psi, n_x) \cdot d\omega_\Psi$$

$$\langle L(x \to \Theta) \rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{f_r(\Psi_i \leftrightarrow \Theta) \cdot L(x \leftarrow \Psi_i) \cdot \cos(\Psi_i, n_x)}{p(\Psi_i)}$$

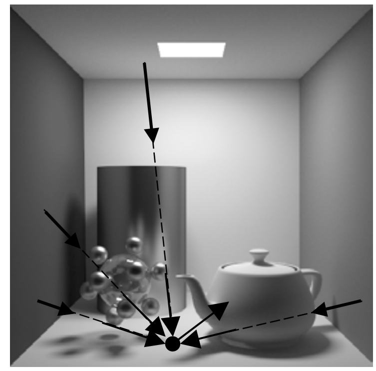
How to compute?



- evaluate $L(x \leftarrow \Psi_i)$?
- Radiance is invariant along straight paths
- $vp(x, \Psi_i)$ = first visible point
- $L(x \leftarrow \Psi_i) = L(vp(x, \Psi_i) \to \Psi_i)$

How to compute? Recursion ...

- Recursion ….
- Each additional bounce adds one more level of indirect light
- Handles ALL light transport
- “Stochastic Ray Tracing”



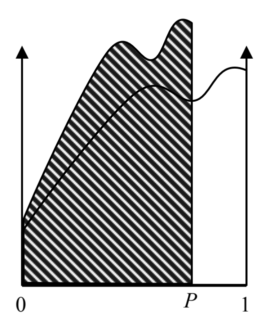
Russian Roulette

Integral

$$I = \int_0^1 f(x)dx = \int_0^1 \frac{f(x)}{P} P dx = \int_0^P \frac{f(y/P)}{P} dy$$

Estimator

$$\langle I_{roulette} \rangle = \begin{cases} \frac{f(x_i)}{P} & \textbf{if } x_i \le P, \\ 0 & \textbf{if } x_i > P. \end{cases}$$



Variance $\sigma_{roulette} > \sigma$

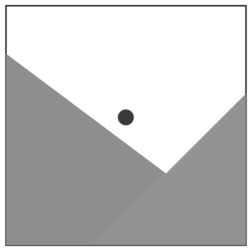
Russian Roulette

- Pick some 'absorption probability' α
  - probability $1-\alpha$ that ray will bounce
  - estimated radiance becomes $L/(1-\alpha)$
- E.g. $\alpha = 0.9$
  - only 1 chance in 10 that ray is reflected
  - estimated radiance of that ray is multiplied by 10
  - instead of shooting 10 rays, we shoot only 1, but count the contribution of this one 10 times

Algorithm so far ...

- Shoot viewing ray through each pixel
- Shoot # indirect rays, sampled over hemisphere
- Terminate recursion using Russian Roulette

Pixel Anti-Aliasing



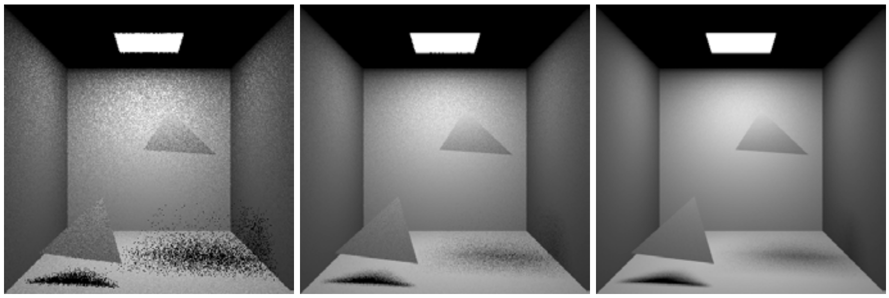
- Compute radiance only at center of pixel: jaggies
- Simple box filter:
- … evaluate using MC

Stochastic Ray Tracing

- Parameters?
  - # starting rays per pixel
  - # random rays for each surface point (branching factor)
- Path Tracing
  - Branching factor == 1

© Kavita Bala, Computer Science, Cornell University

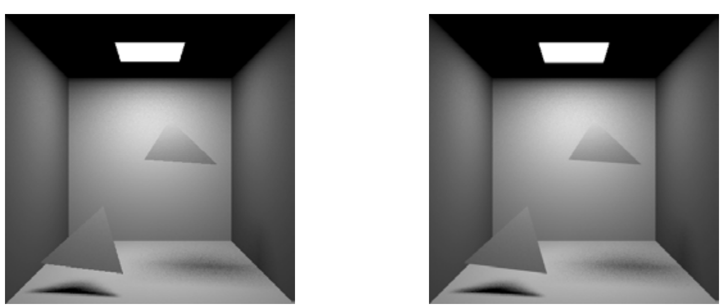
Path tracing



1 ray / pixel 10 rays / pixel 100 rays / pixel

- Pixel sampling + light source sampling folded into one method

Comparison



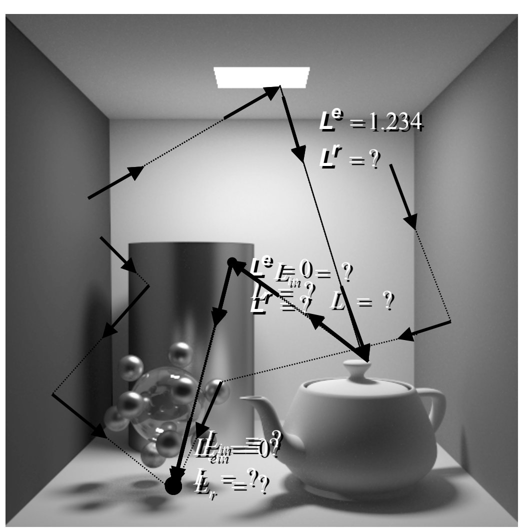
1 centered viewing ray
100 random shadow rays per viewing ray

100 random viewing rays
1 random shadow ray per viewing ray

Algorithm so far ...

- Shoot # viewing rays through each pixel
- Shoot # indirect rays, sampled over hemisphere
  - Path tracing shoots only 1 indirect ray
- Terminate recursion using Russian Roulette

Algorithm



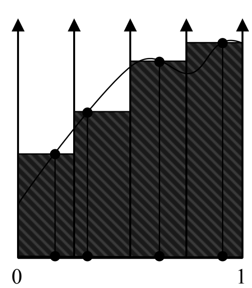
Performance/Error

- Want better quality with smaller number of samples
  - – Fewer samples/better performance
  - – Stratified sampling
  - – Quasi Monte Carlo: well-distributed samples

- Faster convergence
  - – Importance sampling: next-event estimation

Stratified Sampling

- Samples could be arbitrarily close



- Split integral in subparts

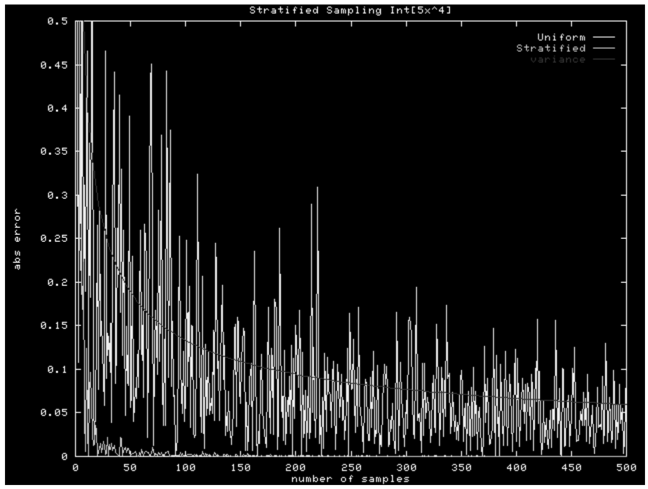
$$I = \int_{X_1} f(x)dx + \ldots + \int_{X_N} f(x)dx$$

- Estimator

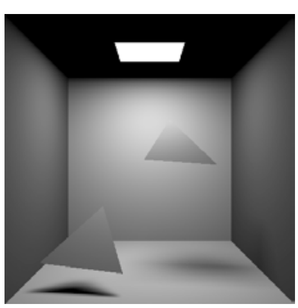
$$\bar{I}_{strat} = \frac{1}{N}\sum_{i=1}^{N}\frac{f(\bar{x}_i)}{p(\bar{x}_i)}$$

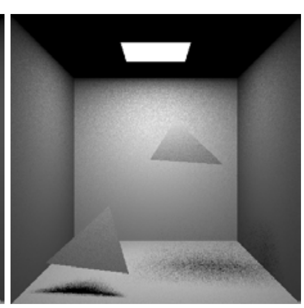
- Variance: $\sigma_{strat} \leq \sigma_{\sec}$

Numerical example

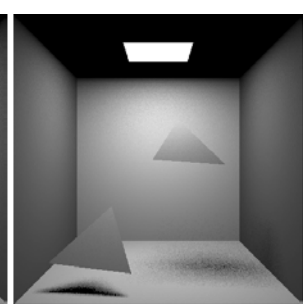


Stratified Sampling



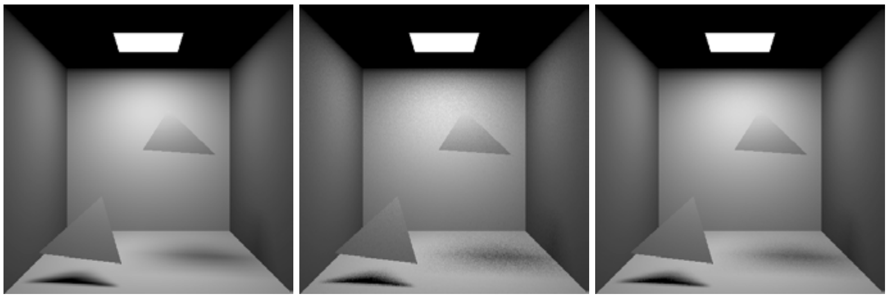


9 shadow rays
not stratified



9 shadow rays
stratified

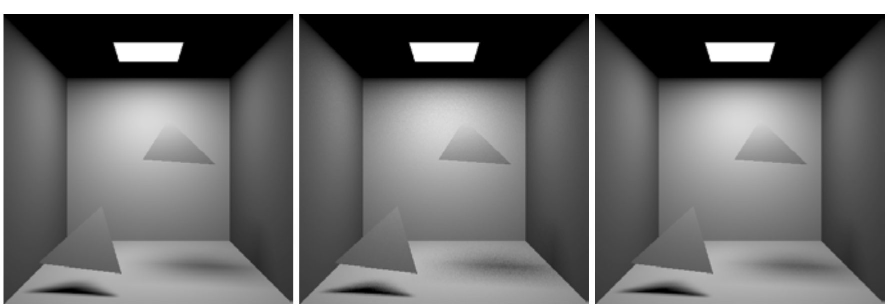
Stratified Sampling



36 shadow rays
not stratified

36 shadow rays
stratified

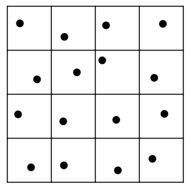
Stratified Sampling



100 shadow rays
not stratified

100 shadow rays
stratified

2 Dimensions

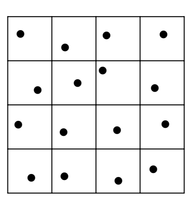


$\rightarrow N^2$ samples

- Problem for higher dimensions
- Sample points can still be arbitrarily close to each other

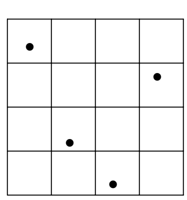
Higher Dimensions

- Stratified grid sampling:



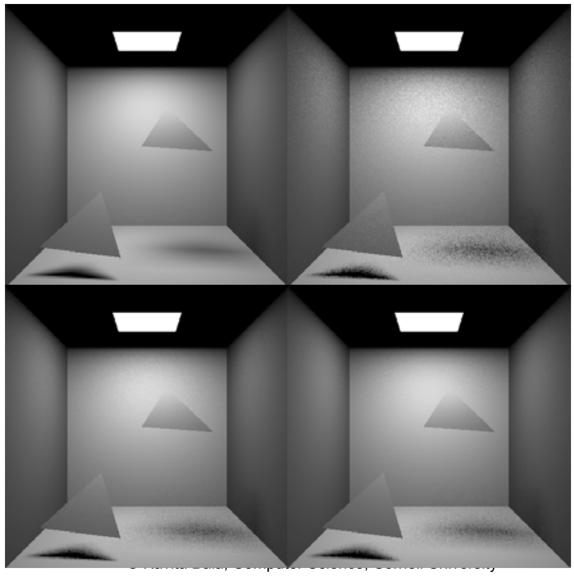
$\rightarrow N^d$ samples

- N-rooks sampling:



$\rightarrow N$ samples

N-Rooks Sampling - 9 rays

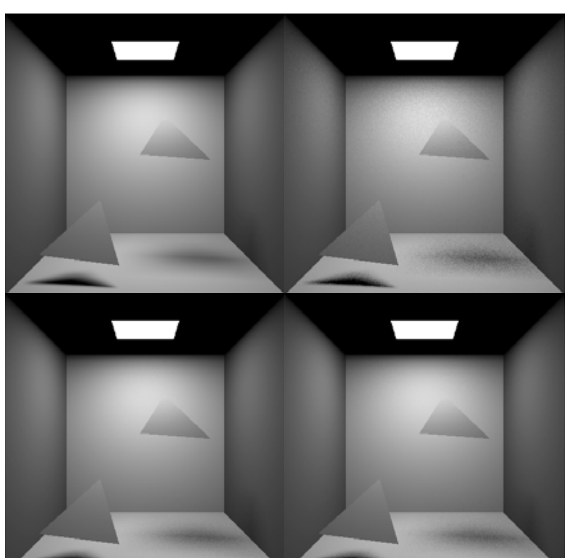


not stratified

stratified

N-Rooks

N-Rooks Sampling - 36 rays



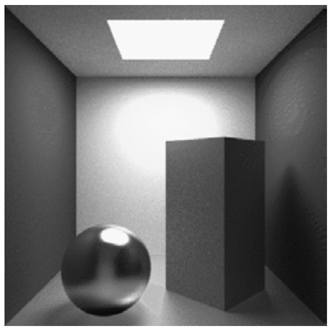
not stratified

stratified

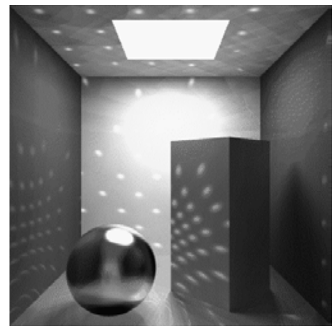
N-Rooks

Other types of Sampling

- How does it relate to regular sampling



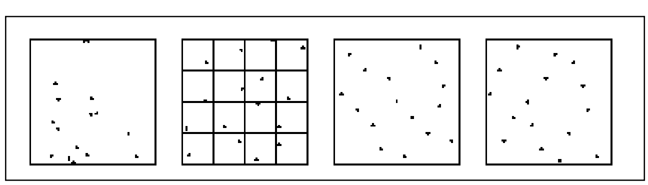
Random sampling



Regular sampling

Quasi Monte Carlo

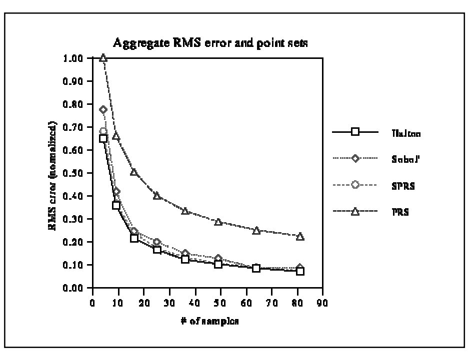
- Eliminates randomness to find well-distributed samples
- Samples are determinisitic but “appear” random



Quasi-Monte Carlo (QMC)

- Notions of variance, expected value don't apply
- Introduce the notion of discrepancy
  - Discrepancy mimics variance
  - E.g., subset of unit interval [0,x]
    - Of N samples, n are in subset
    - Discrepancy: |x-n/N|
  - Mainly: "it looks random"

Quasi Monte Carlo



Example: Hammersley

- Radical inverse $\phi_p(i)$ for primes *p*
- Reflect digits (base p) about decimal point
  - $\phi_2(i)$: $111010_2 \rightarrow 0.010111$
- For N samples, a Hammersley point
  - $(i/N, \phi_2(i))$
- For more dimensions:
  - $X_i = (i/N, \phi_2(i), \phi_3(i), \phi_5(i), \phi_7(i), \phi_{11}(i), \ldots)$

Halton

- Radical inverse function
  - Inverts bits around decimal point

$$i = \sum_j a_j(i) b^j$$

$$\Phi_b(i) = \sum_j a_j(i) b^{-j-1}$$

- Sample: $x_i = (\Phi_{b_1}(i), \Phi_{b_2}(i), \Phi_{b_3}(i), \ldots)$
  - Where b_1, b_2, b_3 are primes

Quasi Monte Carlo

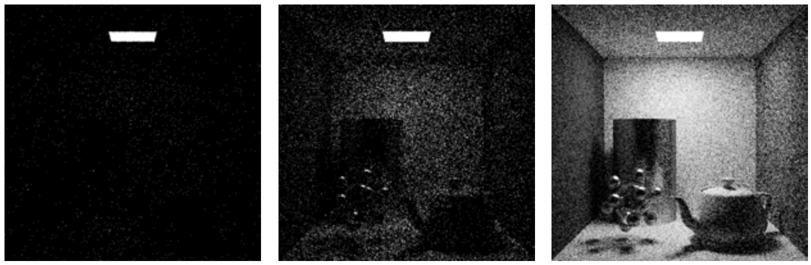
- Converges as fast as stratified sampling
  - Does not require knowledge about how many samples will be used
- Using QMC, directions evenly spaced no matter how many samples are used
- Samples properly stratified-> better than pure MC

Performance/Error

- Want better quality with smaller number of samples
  - Fewer samples/better performance
  - Stratified sampling
  - Quasi Monte Carlo: well-distributed samples
- Faster convergence
  - Importance sampling: next-event estimation

Path Tracing

Sample hemisphere

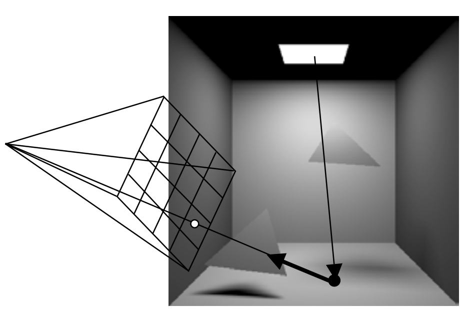


1 sample/pixel 16 samples/pixel 256 samples/pixel

- Importance Sampling: compute direct illumination separately!

Direct Illumination

- Paths of length 1 only, between receiver and light source



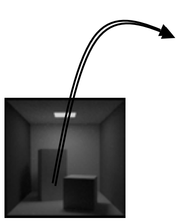
Direct Illumination

Global Illumination

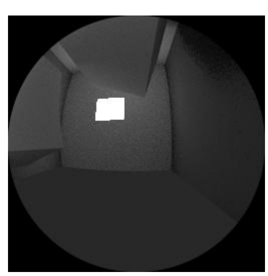
Next Event Estimation

$$L(x \to \Theta) = L_e(x \to \Theta) + \int_{\Omega_x} f_r(\Psi \leftrightarrow \Theta) \cdot L(x \leftarrow \Psi) \cdot \cos(\Psi, n_x) \cdot d\omega_\Psi$$

Radiance from light sources + radiance from other surfaces



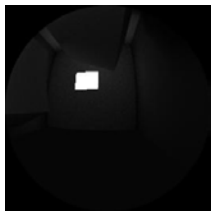
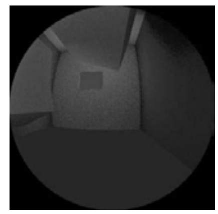
$$= L_e + \int_{\Omega_x}$$



$$\cdot f_r \cdot \cos$$

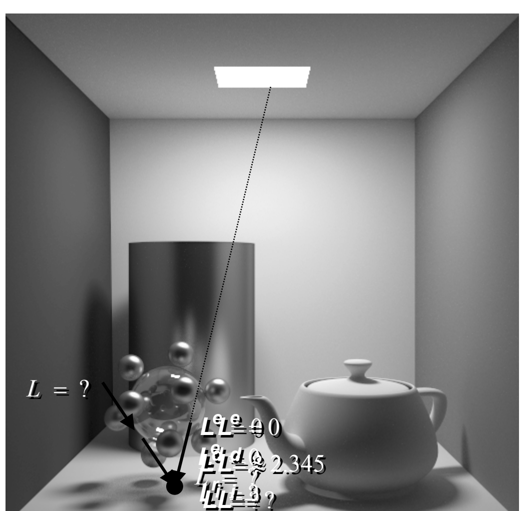
Next Event Estimation

$$L(x \rightarrow \Theta) = L_e + L_{direct} + L_{indirect}$$

$$= L_e + \int_{\Omega_x}$$  $$\cdot f_r \cdot \cos + \int_{\Omega_x}$$  $$\cdot f_r \cdot \cos$$

- So … sample direct and indirect with separate MC integration

Algorithm

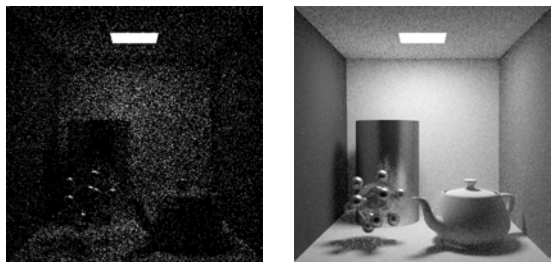


Algorithm

Algorithm

→a variant of path tracing

Comparison



Without N.E.E. With N.E.E.

16 samples/pixel

Rays per pixel

1 sample/ pixel

4 samples/ pixel

16 samples/ pixel

256 samples/ pixel

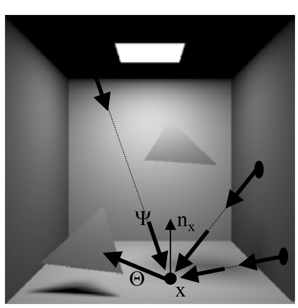
How to sample direct illumination

- Sampling a single light source
- Sampling for many lights

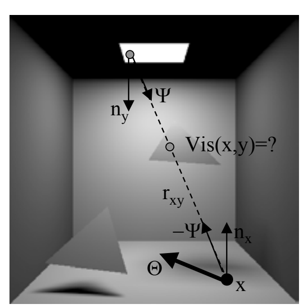
Direct Illumination

$$L(x \to \Theta) = \int_{A_{source}} f_r(x, -\Psi \leftrightarrow \Theta) \cdot L(y \to \Psi) \cdot G(x,y) \cdot dA_y$$

$$G(x,y) = \frac{\cos(n_x, \Theta)\cos(n_y, \Psi) Vis(x,y)}{r_{xy}^2}$$



hemisphere integration



area integration

Two forms of the RE

- Hemisphere integration

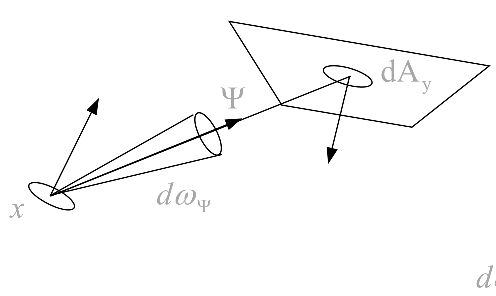
$$L(x \to \Theta) = L_e(x \to \Theta) + \int_{\Omega_x} f_r(\Psi \leftrightarrow \Theta) \cdot L(x \leftarrow \Psi) \cdot \cos\theta_x \cdot d\omega_\Psi$$

- Area integration (over polygons from set A)

$$L(x \to \Theta) = L_e(x \to \Theta) + \int_A f_r(\Psi \leftrightarrow \Theta) \cdot L(y \to -\Psi) \cdot \frac{\cos\theta_x \cdot \cos\theta_y}{r_{xy}^2} \cdot V(x,y) \cdot dA_y$$

Rendering Equation

$$L_{direct}(x \to \Theta) = \int_{\Omega_x} f_r(\Psi \leftrightarrow \Theta) \cdot L(x \leftarrow \Psi) \cdot \cos\theta_x \cdot d\omega_\Psi$$



$$y = vp(x, \Psi)$$

$$L(x \leftarrow \Psi) = L(vp(x,\Psi) \to -\Psi)$$

$$d\omega_\Psi = \frac{dA_y \cos\theta_y}{r_{xy}^2}$$

Rendering Equation: visible surfaces

$$L_{direct}(x \to \Theta) = \int_{\Omega_x} f_r(\Psi \leftrightarrow \Theta) \cdot L(x \leftarrow \Psi) \cdot \cos\theta_x \cdot d\omega_\Psi$$

Coordinate transform ↓

$$L_{direct}(x \to \Theta) = \int_{\substack{y \text{ on} \\ \text{all lights}}} f_r(\Psi \leftrightarrow \Theta) \cdot L(y \to -\Psi) V(x,y) \cos\theta_x \cdot \frac{\cos\theta_y}{r_{xy}^2} \cdot dA_y$$

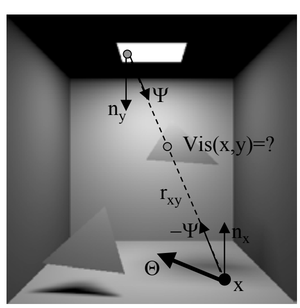
$$y = vp(x, \Psi)$$

Integration domain = surface points y on lights

RE over light sources

$$L(x \to \Theta) = \int_{A_{source}} f_r(x, -\Psi \leftrightarrow \Theta) \cdot L_{source}(y \to -\Psi) \cdot G(x,y) \cdot dA_y$$

$$G(x,y) = \frac{\cos\theta_x \cos\theta_y Vis(x,y)}{r_{xy}^2}$$



area integration

Estimator for direct lighting

- Pick a point on the light's surface with pdf

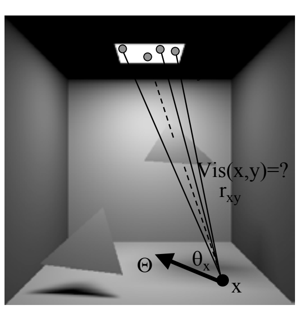
$p(y)$

- For N samples, direct light at point x is:

$$E(x) = \frac{1}{N}\sum_{i=1}^{N}\frac{f_r L_{source}\dfrac{\cos\theta_x \cos\theta_{\bar{y}_i}}{r_{x\bar{y}_i}^2} Vis(x,\bar{y}_i)}{p(\bar{y}_i)}$$

Generating direct paths

- Pick surface points y_i on light source
- Evaluate direct illumination integral



$$\langle L(x \to \Theta)\rangle = \frac{1}{N}\sum_{i=1}^{N}\frac{f_r(\ldots)L(\ldots)G(x,y_i)}{p(y_i)}$$

PDF for sampling light

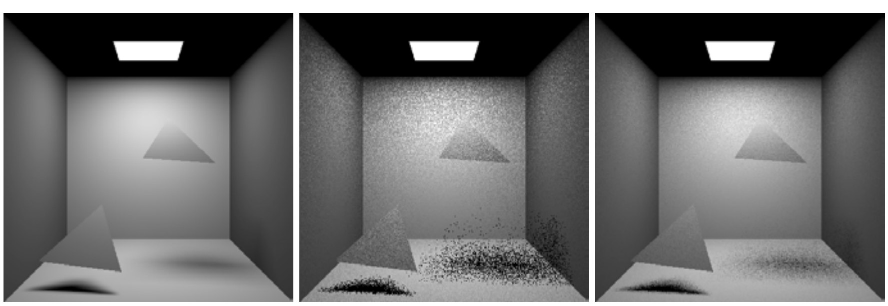
- Uniform $$p(y) = \frac{1}{Area_{source}}$$

- Pick a point uniformly over light's area
  – Can stratify samples

- Estimator:

$$E(x) = \frac{Area_{source}}{N} \sum_{i=1}^{N} f_r L_{source} \frac{\cos\theta_x \cos\theta_{\bar{y}_i}}{r_{x\bar{y}_i}^2} Vis(x, \bar{y}_i)$$

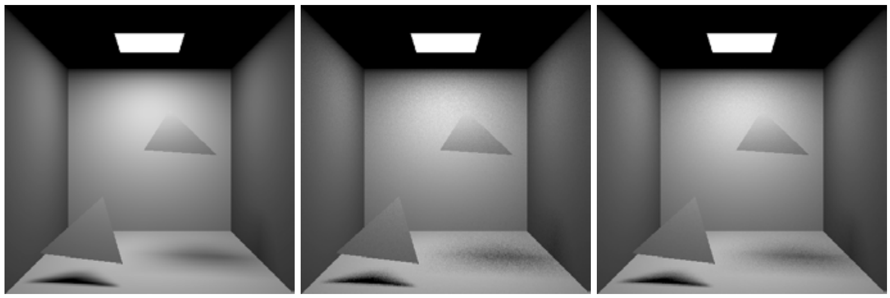
More points ...



1 shadow ray 9 shadow rays

$$E(x) = \frac{Area_{source}}{N} \sum_{i=1}^{N} f_r L_{source} \frac{\cos\theta_x \cos\theta_{\bar{y}_i}}{r_{x\bar{y}_i}^2} Vis(x, \bar{y}_i)$$

Even more points ...



36 shadow rays 100 shadow rays

$$E(x) = \frac{Area_{source}}{N} \sum_{i=1}^{N} f_r L_{source} \frac{\cos\theta_x \cos\theta_{\bar{y}_i}}{r_{x\bar{y}_i}^2} Vis(x, \bar{y}_i)$$

Different pdfs

- Uniform

$$p(y) = \frac{1}{Area_{source}}$$

- Solid angle sampling
  - Removes cosine and distance from integrand
  - Better when significant foreshortening

$$E(x) = \frac{1}{N} \sum_{i=1}^{N} \frac{f_r L_{source} \frac{\cos\theta_x \cos\theta_{\bar{y}_i}}{r_{x\bar{y}_i}^2} Vis(x, \bar{y}_i)}{p(\bar{y}_i)}$$

Parameters

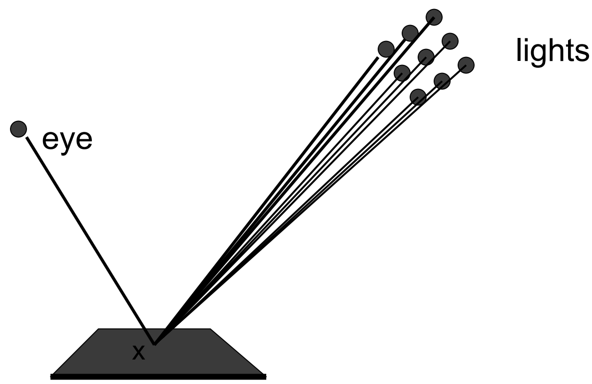
- How to distribute paths within light source?
  - Uniform
  - Solid angle
  - What about light distribution?
- How many paths ("shadow-rays")?
  - Total?
  - Per light source? (~intensity, importance, …)

Scenes with many lights

- Many lights in scenes: M lights
- How to handle many lights?
- Formulation 1: M integrals, one per light
  - Same solution technique as earlier for each light

$$L(x \to \Theta) = \sum_{i=1}^{M} \int_{A_{source}} f_r(x, -\Psi \leftrightarrow \Theta) \cdot L_{source}(y \to -\Psi) \cdot G(x,y) \cdot dA_y$$

Lighting: point sources



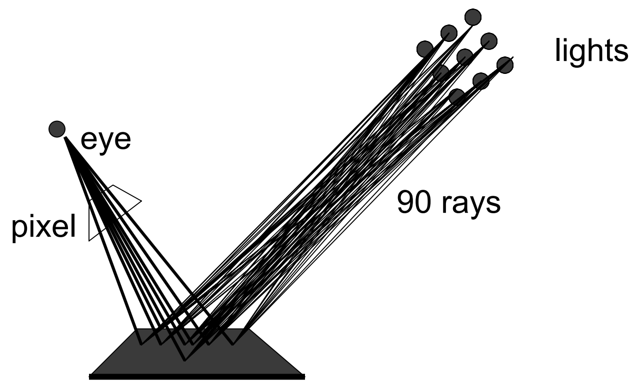
Scenes with many lights

- Various choices:
  - Shadow rays per light source
  - Distribution of shadow rays within a light source
- Total # rays = M N
  - Where, M = #lights, N = #rays per source

© Kavita Bala, Computer Science, Cornell University

Antialiasing: pixel

- Anti-aliasing: k M N



Formulation over all lights

- When M is large, each direct lighting sample is very expensive
- We would like to importance sample the lights
- Instead of M integrals

$$L(x \to \Theta) = \sum_{i=1}^{M} \int_{A_{source}} f_r(x, -\Psi \leftrightarrow \Theta) \cdot L_{source}(y \to -\Psi) \cdot G(x,y) \cdot dA_y$$

- Formulation over 1 integration domain

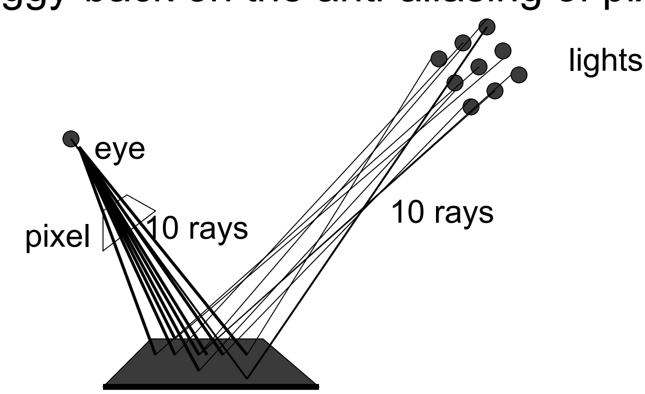
$$L(x \to \Theta) = \int_{A_{all\ lights}} f_r(x, -\Psi \leftrightarrow \Theta) \cdot L_{source}(y \to -\Psi) \cdot G(x,y) \cdot dA_y$$

Why?

- Do not need a minimum of M rays/sample
- Can use only one ray/sample
- Still need N samples, but 1 ray/sample
- Ray is distributed over the whole integration domain
  – Can importance sample the lights

Anti-aliasing

- Can piggy-back on the anti-aliasing of pixel



How to sample the lights?

- A discrete pdf $p_L(k_i)$ picks the light k_i
- A surface point is then picked with pdf $p(y_i|k_i)$
- Estimator with N samples:

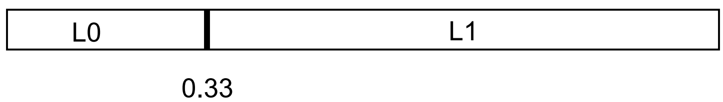
$$E(x) = \frac{1}{N}\sum_{i=1}^{N}\frac{f_r L_{source} G(x,\bar{y}_i)}{p_L(k_i)\,p(y_i \mid k_i)}$$

Strategies for picking light

- Uniform $p_L(k) = \frac{1}{M}$
- Area $p_L(k) = \frac{A_k}{\sum A_k}$
- Power $p_L(k) = \frac{P_k}{\sum P_k}$

Example for 2 lights

- Light 0 has power 1, Light 1 has power 2
- Using power for pdf:
  - $p_L(L_0) = 1/3$, $p_L(L_1) = 2/3$



- Overall pdf $p(y) = \frac{1}{3} p_{L_0}(y) + \frac{2}{3} p_{L_1}(y)$

Example for 2 lights

- Pick a random value: ξ_0
- If $\xi_0 < \frac{1}{3}$
- Sample Light 0 and compute estimate e0
- Overall estimate is $\frac{e_0}{\frac{1}{3}}$

Example for 2 lights

- If

$$\frac{1}{3} \le \xi_0 < 1$$

- Sample Light 1 and compute estimate e1
- Overall estimate is $\frac{e_1}{\frac{2}{3}}$

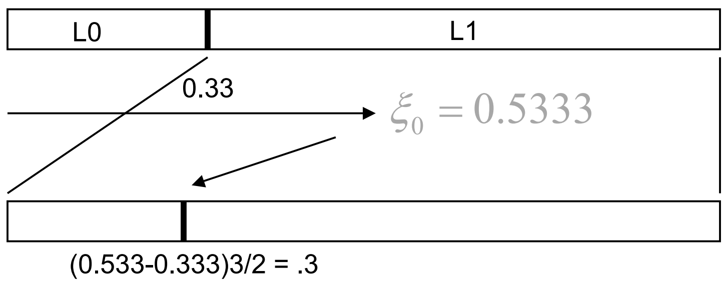
How to sample light?

- Once light is picked, can pick two random numbers ξ_1, ξ_2 according to $p_{L0}(y)$, $p_{L1}(y)$
- To decrease variance we should reuse ξ_0
- But, already used information in ξ_0 to pick the light

Example for 2 lights

- Rescale ξ_0

$$\xi'_0 = \frac{3}{2}(\xi_0 - \frac{1}{3})$$



- Use (ξ'_0, ξ_1) to pick samples on light 1