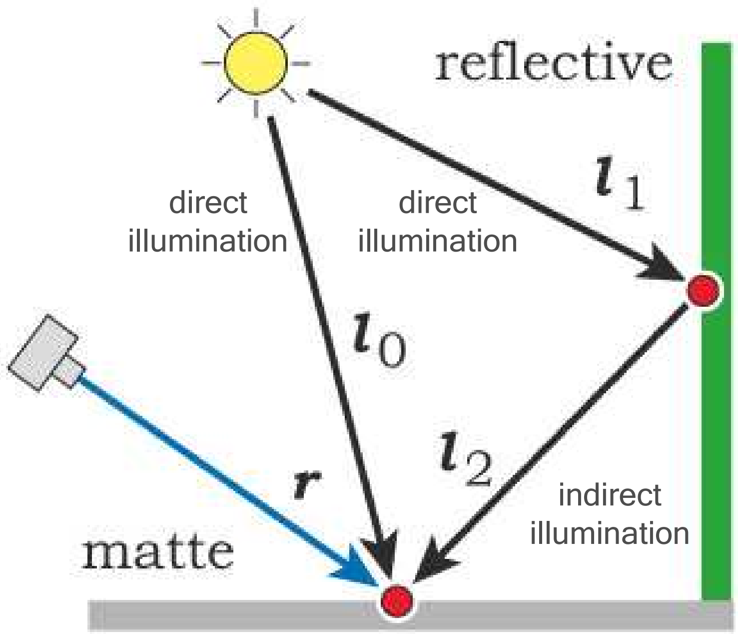
Advanced Computer Graphics
Path Tracing

Computer Graphics
Computer Science Department
University of Freiburg

SS 11

Motivation

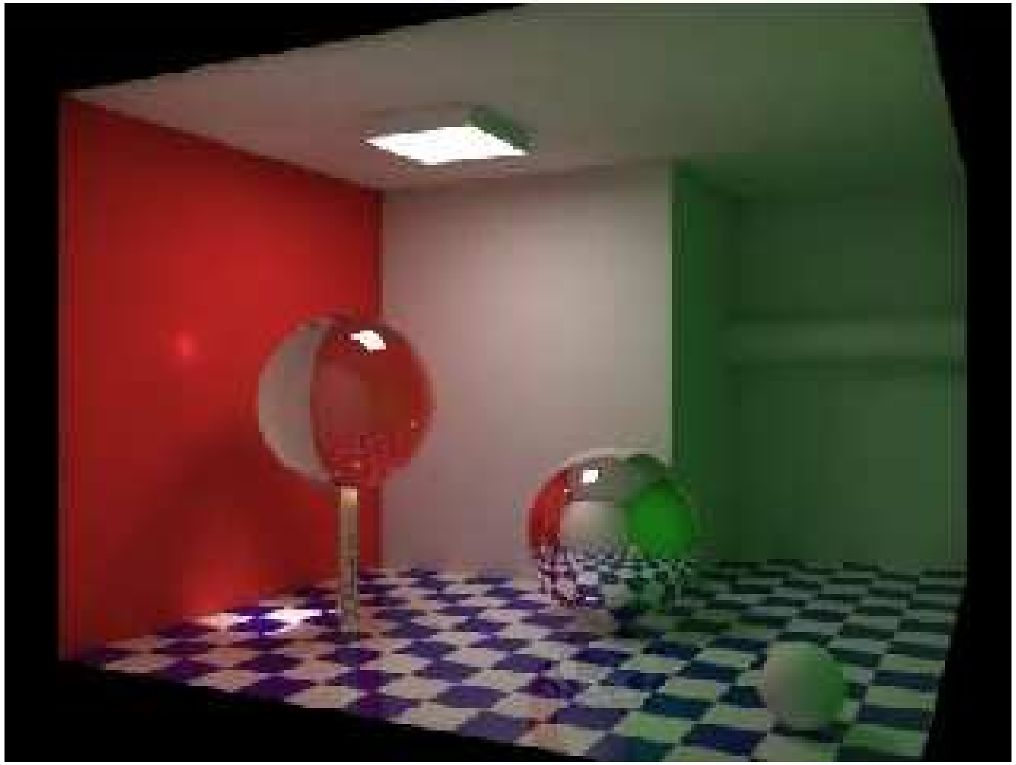
- global illumination tries to account for all light-transport mechanisms in a scene
  - considers direct and indirect illumination (emitted and reflected radiance)
  - allows for effects such as, e. g.,
    - interreflections (surfaces illuminate each other, potentially changing their colors)
    - caustics (reflected radiance from surfaces is focused at a scene point)



[Suffern]

Motivation

- caustics and color transfer / bleeding from red and green side walls, i. e. interreflections



http://www.cse.iitb.ac.in/~rhushabh/

[Rhushabh Goradia]

Outline

- path tracing
- brute-force path tracing
- direct illumination
- indirect illumination

Path Tracing - Concept

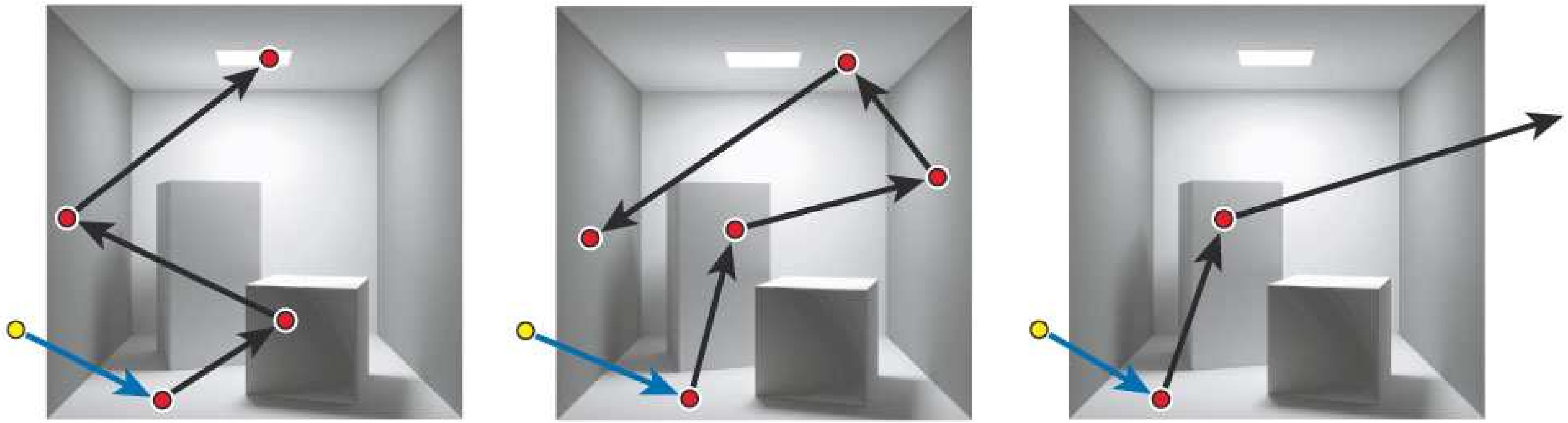
- generate light transport paths (a chain of rays) from visible surface points to light sources
  - rays are traced recursively until hitting a light source
- recursively evaluate the rendering equation along a path
- ray generation in a path is governed by
  - light sources
  - BRDFs
- recursion depth is generally governed by the amount of radiance along a ray
- can distinguish direct and indirect illumination
  - direct: emitted radiance from light sources
  - indirect: reflected radiance from surfaces (and light sources)

Path Tracing - Algorithm

- for a pixel p, a ray is cast from the eye through p into the scene
- at the hit point, the rendering equation is evaluated using Monte Carlo integration
- to approximate the incident radiance, rays are traced into random sample directions
- sample directions are chosen according to
  - cosine weighting of the incident radiance
  - BRDF
  - light sources
- this scheme is recursively applied as long as there is a significant amount of radiance transported along a ray

Path Tracing - Illustration

- rays are recursively cast along a path to estimate the transported radiance
- recursion stops if
  - a light source is hit
  - a maximum depth / minimum radiance is reached
  - the ray leaves the scene / hits the background



[Suffern]

Outline

- path tracing
- brute-force path tracing
- direct illumination
- indirect illumination

Path Tracing - Brute Force

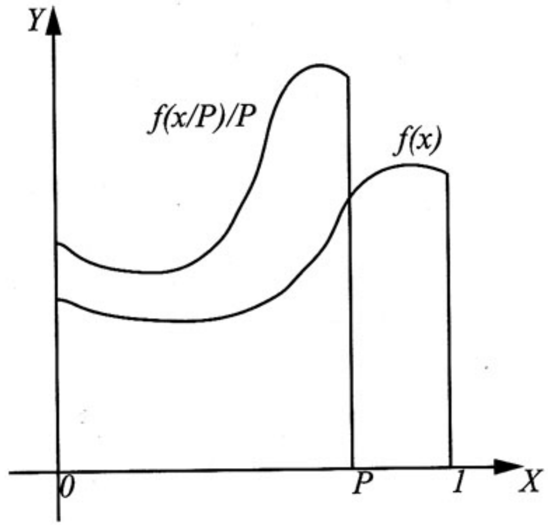
- emitted radiance is returned or reflected radiance at point p is computed
  $$L_o(p,\omega_o) = L_e(p,\omega_o) + \int_{2\pi^+} f_r(p,\omega_i,\omega_o) L_i(p,\omega_i) \cos\theta_i \mathrm{d}\omega_i$$
- therefore, rays are cast into the scene to compute the emitted radiance from point $r_c(p,\omega_i)$
  $$L_o(p,\omega_o) = L_e(p,\omega_o) + \int_{2\pi^+} f_r(p,\omega_i,\omega_o) L_o(r_c(p,\omega_i),-\omega_i) \cos\theta_i \mathrm{d}\omega_i$$
  using Monte Carlo
  $$F_N = \frac{1}{N}\sum_{i=1}^{N} \frac{f_r(p,\omega_i,\omega_o) L_o(r_c(p,\omega_i),-\omega_i)\cos\theta_i}{p(\omega_i)}$$
- if a path does not hit a light source, zero radiance is returned
- if a path hits a light source, the emitted radiance is transported along the ray weighted with $\frac{f_r(p,\omega_i,\omega_o)\cos\theta_i}{p(\omega_i)}$ at each intersection point

Path Tracing - Recursion Depth

- depending on material properties
  - larger paths for specular surfaces / shorter paths for diffuse material
- depending on hemispherical-hemispherical reflectance
  - total reflection over the hemisphere due to illumination from the hemisphere
  - $\rho_{hh}(2\pi^+, 2\pi^+) = \frac{1}{\pi} \int_{2\pi^+} \int_{2\pi^+} f_r(\omega_i, \omega_o) \cos\theta_i \cos\theta_o \mathrm{d}\omega_i \mathrm{d}\omega_o$
- depending on the weight of the returned radiance
  - at each intersection point, returned radiance is weighted with $\frac{f_r(p,\omega_i,\omega_o)\cos\theta_i}{p(\omega_i)}$
  - this coefficient can be accumulated at intersection points along a path
  - stop if the accumulated coefficient is below a threshold value

Path Tracing - Russian Roulette

- allows to choose
  - fewer (and longer) paths or
  - more (shorter) paths
- f is scaled horizontally by $P \leq 1$ and vertically by $1/P$
  $I = \int_0^1 f(x)\mathrm{d}x = \int_0^P \frac{1}{P} f\left(\frac{x}{P}\right) \mathrm{d}x$
- for uniform random samples $X_i \in [0, 1)$ the estimator
  $F_N = \frac{1}{N} \sum_{i=1}^N \frac{1}{P} f\left(\frac{X_i}{P}\right) \quad \forall X_i \leq P$
  approximates the integral



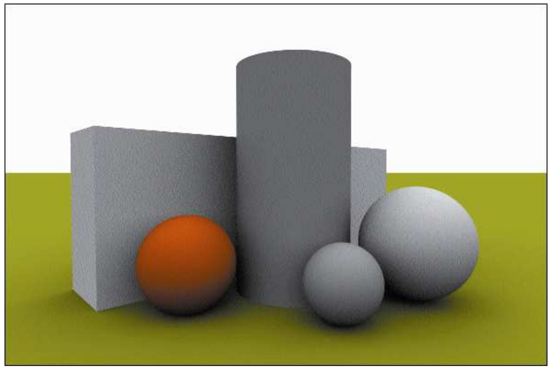
[Dutre, Bala, Bekaert]

Path Tracing - Russian Roulette

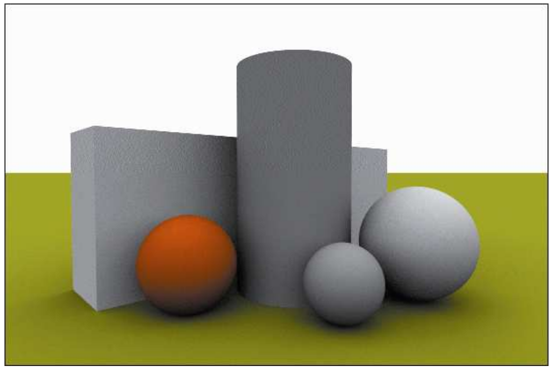
- absorption probability $\alpha = 1 - P$
  - describes the probability that a path stops at an intersection
- samples generated in the interval $(P, 1)$ return zero radiance compensated by the factor $1/P \geq 1$ for the samples in $[0, P]$
- absorption probability is a parameter
  - e. g. $P \propto \rho_{hh}$
  - dark surfaces absorb a path with higher probability

Path Tracing - Properties

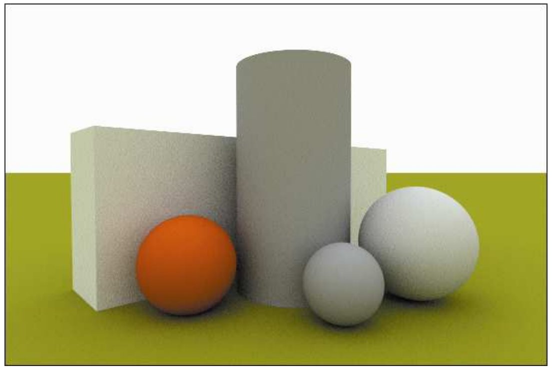
- works with environment lights
- rather inefficient for area light sources as many paths return zero radiance
- does not work with point lights or directional light



direct illumination



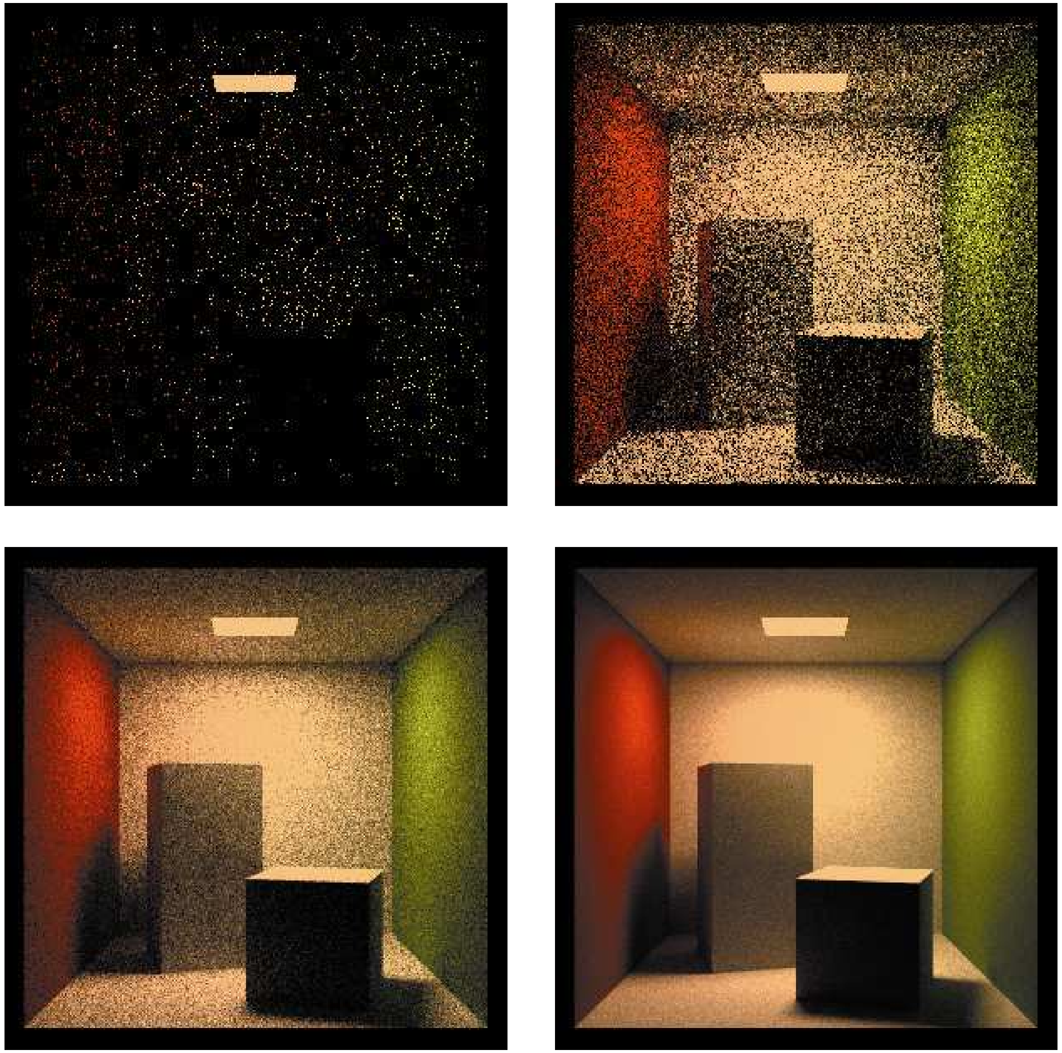
path tracing
one bounce



path tracing
five bounces

[Suffern]

Path Tracing - Results



1 sample per pixel

100 samples per pixel

1000 samples per pixel

10000 samples per pixel

[Suffern]

Outline

- path tracing
- brute-force path tracing
- direct illumination
- indirect illumination

Motivation

- light sources dominate the illumination of surfaces and are rather small
- improved efficiency by explicitly sending rays to light sources

$$L_o(p,\omega_o) = \underbrace{L_e(p,\omega_o)}_{\text{emitted radiance}} + \underbrace{\int_{2\pi^+} f_r(p,\omega_i,\omega_o) L_i(p,\omega_i) \cos\theta_i \mathrm{d}\omega_i}_{\text{reflected radiance}}$$

$$L_o(p,\omega_o) = L_e(p,\omega_o) + L_r(p,\omega_o)$$

- assumption
  - no emitted radiance for objects $\quad L_e(p,\omega_o) = 0 \quad L_r(p,\omega_o) > 0$
  - for light sources $\quad L_e(p,\omega_o) > 0 \quad L_r(p,\omega_o) > 0$

Motivation

- for objects

$$L_o(p,\omega_o) = \int_{2\pi^+} f_r(p,\omega_i,\omega_o)(L_e(r_c(p,\omega_i),-\omega_i)+L_r(r_c(p,\omega_i),-\omega_i))\cos\theta_i \mathrm{d}\omega_i$$

emitted radiance
direct illumination

reflected radiance
indirect illumination

$$L_o(p,\omega_o) = \int_{2\pi^+} f_r(p,\omega_i,\omega_o)L_r(r_c(p,\omega_i),-\omega_i)\cos\theta_i \mathrm{d}\omega_i$$
$$+\int_{2\pi^+} f_r(p,\omega_i,\omega_o)L_e(r_c(p,\omega_i),-\omega_i)\cos\theta_i \mathrm{d}\omega_i$$

zero for objects

this second term significantly contributes to L_o

- contribution of direct illumination is computed using the area form of the integral

$$L_o(p,\omega_o) = \int_{2\pi^+} f_r(p,\omega_i,\omega_o)L_r(r_c(p,\omega_i),-\omega_i)\cos\theta_i \mathrm{d}\omega_i$$
$$+\int_{2\pi^+} f_r(p,\omega_i,\omega_o)L_e(p',-\omega_i)\frac{\cos\theta_i\cos\theta'}{\|p'-p\|^2}V(p,p')\mathrm{d}A$$

V represents the visibility of the light source, estimated by a shadow ray

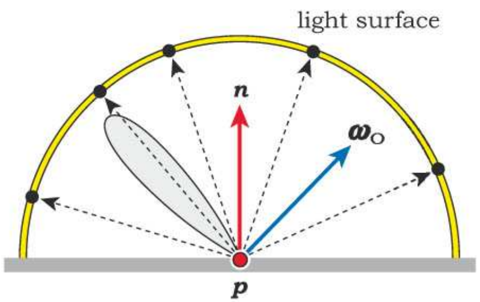
- requires a sampling of the light sources

Hemisphere vs. Area Form

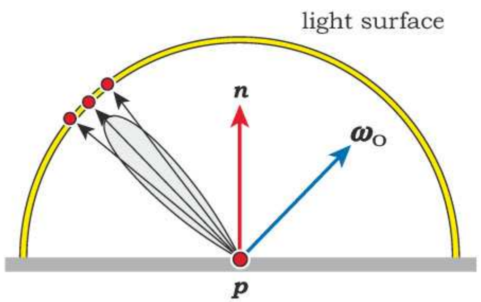
- preferred form depends on BRDF and light source

light-source sampling

sampling according to BRDF

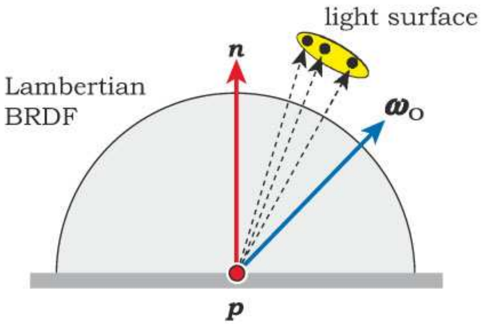


BRDF is small for many samples

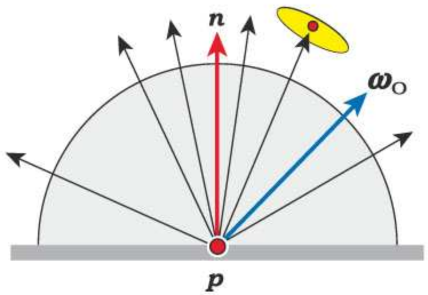


appropriate sampling

glossy surface



appropriate sampling



incident radiance is small for many samples

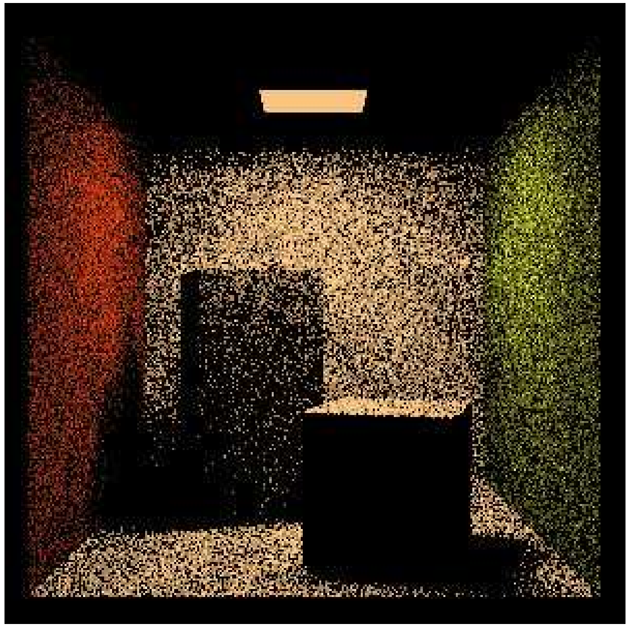
diffuse surface

[Suffern]

Hemisphere vs. Area Form

- both forms can be combined
  - area form with light-source sampling plus hemisphere form ignoring the emitted radiance
  - $L_o(p, \omega_o) = L_e(p, \omega_o) + L_r(p, \omega_o)$
  - area form considers $L_e(p, \omega_o)$
    - light source sampling excludes directions with $L_e(p, \omega_o) = 0$
  - hemisphere form considers $L_r(p, \omega_o)$
    - $L_e(p, \omega_o)$ is ignored to guarantee that direct illumination is not considered twice

Result



path tracing with one bounce
and 100 samples per pixel
- for some pixels, no path hits
the light source
- almost no soft shadows (requires
a minimum number of rays into
the direction of the light source)

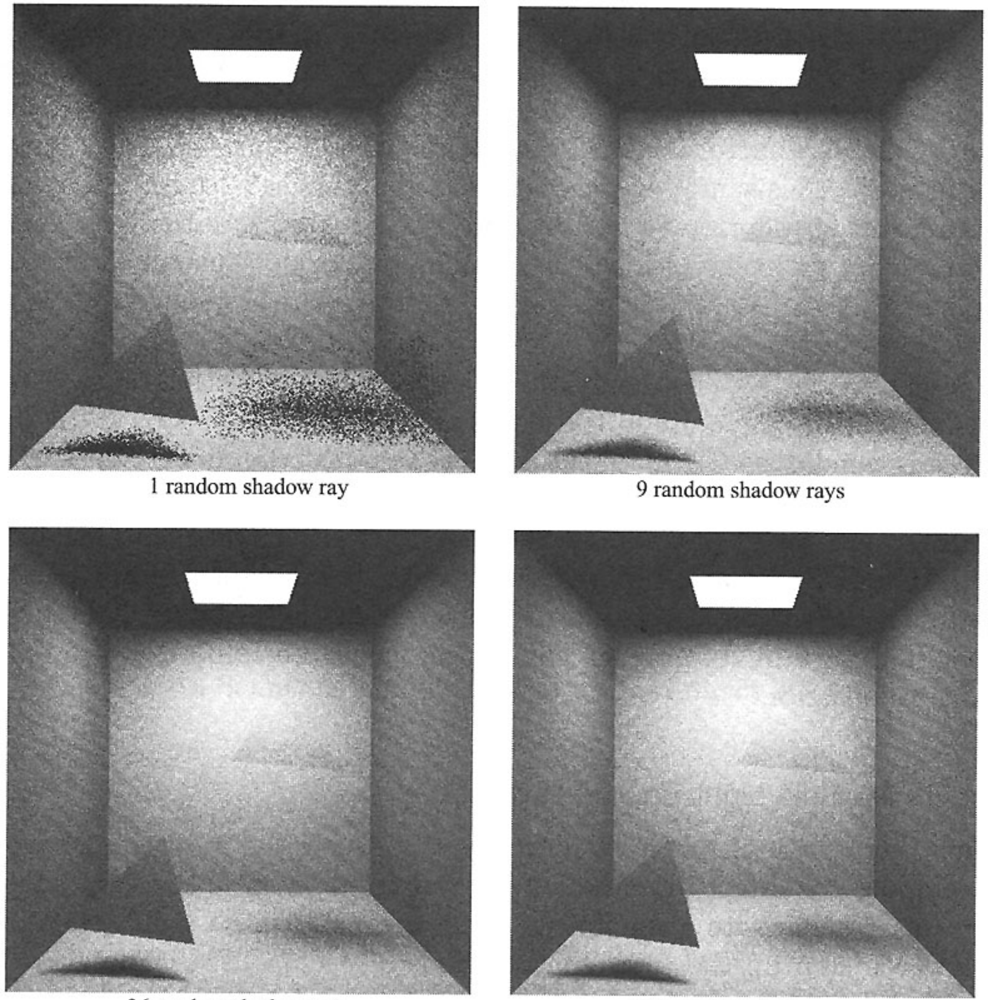
path tracing with one bounce
and 100 light source samples
per pixel

[Suffern]

Light-Source Sampling

- point lights, spot lights, directional light
  - one sample
- area lights
  - estimator
    $L_{\text{direct}}(p, \omega_i) = \frac{1}{N_s} \sum_{i=1}^{N_s} \frac{1}{p(p')} f_r(p, \omega_i, \omega_o) L_e(p', -\omega_i) G(p, p') V(p, p')$
  - uniform PDF $p(p') = \frac{1}{A}$
    $L_{\text{direct}}(p, \omega_i) = \frac{A}{N_s} \sum_{i=1}^{N_s} f_r(p, \omega_i, \omega_o) L_e(p', -\omega_i) G(p, p') V(p, p')$
  - number of samples essential for the quality of soft shadows
  - for points in penumbra regions, $V(p, p') = 0$ or $V(p, p') = 1$
    - too few shadow rays (i.e. light source samples) cause noise in these regions
  - for points outside shadow, variations in $G(p, p')$ can cause noise

Light-Source Sampling



1 random shadow ray
9 random shadow rays
36 random shadow rays
100 random shadow rays

[Dutre, Bala, Bekaert]

Sampling of Multiple Area Light Sources

- can be handled separately
  - add all contributions of shadow rays
- can be handled in a combined way
  - discrete PDF $p_L(k)$ selects a light source from N_L sources
  - $p_L(k)$ depends on position *p*
  - conditional PDF $p(y|k)$ is used to generate a sample point given light source *k*

    $$L_{\text{direct}}(p, \omega_i) = \frac{1}{N_s} \sum_{i=1}^{N_s} \frac{1}{p_L(k) \cdot p(y|k)} f_r(p, \omega_i, \omega_o) L_e(p', -\omega_i) G(p, p') V(p, p')$$
  - theoretically, only one shadow ray can be used

Combined Sampling of Multiple Area Light Sources

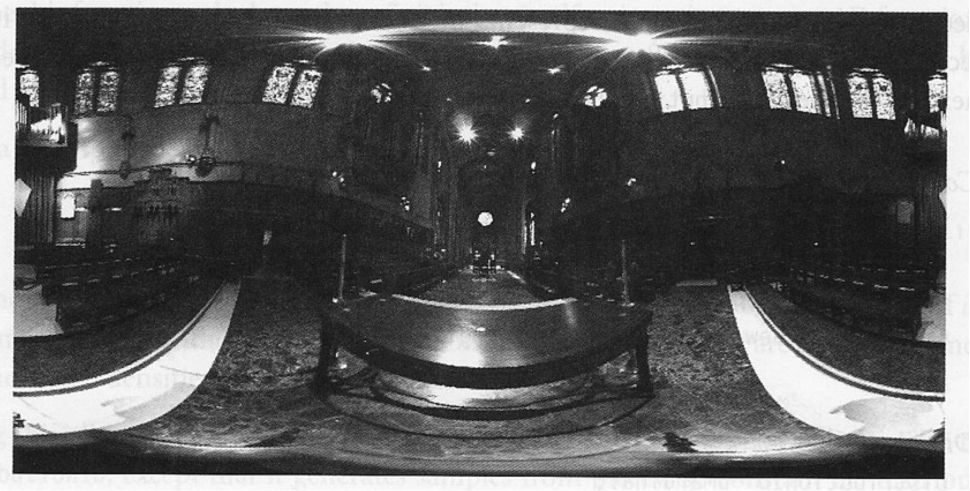
- uniform source selection, uniform area sampling
  - $p_L(k) = \frac{1}{N_L} \quad p(y|k) = \frac{1}{A_{L_k}}$
  - $L_{\text{direct}}(p, \omega_i) = \frac{N_L}{N_s} \sum_{i=1}^{N_s} A_{L_k} f_r(p, \omega_i, \omega_o) L_e(p', -\omega_i) G(p, p') V(p, p')$
- power-proportional source selection
  - $p_L(k) = \frac{P_k}{P_{\text{total}}} \quad p(y|k) = \frac{1}{A_{L_k}}$
  - $L_{\text{direct}}(p, \omega_i) = \frac{P_{\text{total}}}{N_s} \sum_{i=1}^{N_s} \frac{A_{L_k}}{P_k} f_r(p, \omega_i, \omega_o) L_e(p', -\omega_i) G(p, p') V(p, p')$
  - gives importance to bright sources
  - is less efficient if bright sources are occluded
  - better than just ignoring dark, small or far away sources (which would result in bias)

Optimizations

- efficient computation of $V(p, p')$
  - light buffer keeps track of occluding geometry
  - for a light source, it stores occluding geometry in all directions
  - does not work for area lights, less efficient for a large number of light sources
- importance-proportional light source sampling per surface point

Environment Map Illumination

- encodes the total illumination from the hemisphere of directions around a point
- 2D parameterization, e. g. latitude, longitude
- hemisphere form of the rendering equation can be used



Paul Debevec, Grace Cathedral

 [Pharr, Humphreys]

Piecewise-Constant 2D Distribution

- $n_u \times n_v$ samples defined over $(u,v) \in [0,1]^2$
  - a parameterization of the environment map
- $f(u,v)$ is defined by a set of $n_u \times n_v$ values $f[u_i, v_i]$
  - $u_i \in [0,\ldots,n_u - 1] \quad v_i \in [0,\ldots,n_v - 1]$
  - $f[u_i, v_i]$ is the value of $f(u,v)$ in the range $\left[\frac{i}{n_u}, \frac{i+1}{n_u}\right) \times \left[\frac{j}{n_v}, \frac{j+1}{n_v}\right)$
  - $f(u,v) = f[\tilde{u}, \tilde{v}]$ with $\tilde{u} = \lfloor n_u u \rfloor$ and $\tilde{v} = \lfloor n_v v \rfloor$
- integral over the domain
  - $I_f = \int \int f(u,v)\ \mathrm{d}u\ \mathrm{d}v = \frac{1}{n_u n_v} \sum_i \sum_j f[u_i, v_j]$
- PDF
  - $p(u,v) = \frac{1}{I_f} f(u,v) = \frac{1}{I_f} f[\tilde{u}, \tilde{v}]$

Piecewise-Constant 2D Distribution

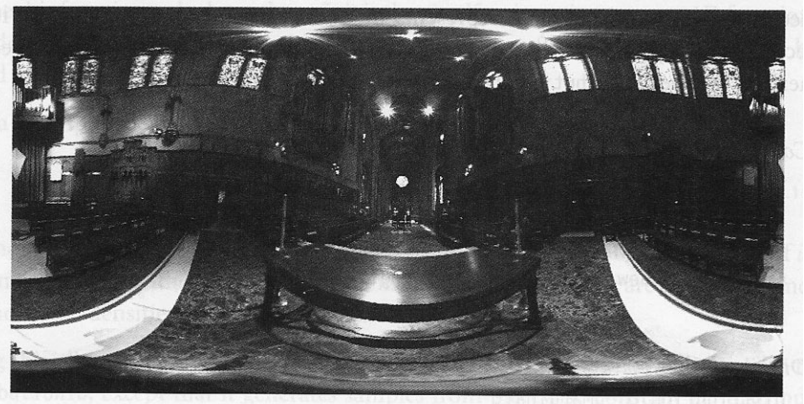
- marginal density function

  $$p_v(v) = \int p(u,v)\ \mathrm{d}u = \frac{1}{I_f}\frac{1}{n_u}\sum_i f[u_i, \tilde{v}]$$

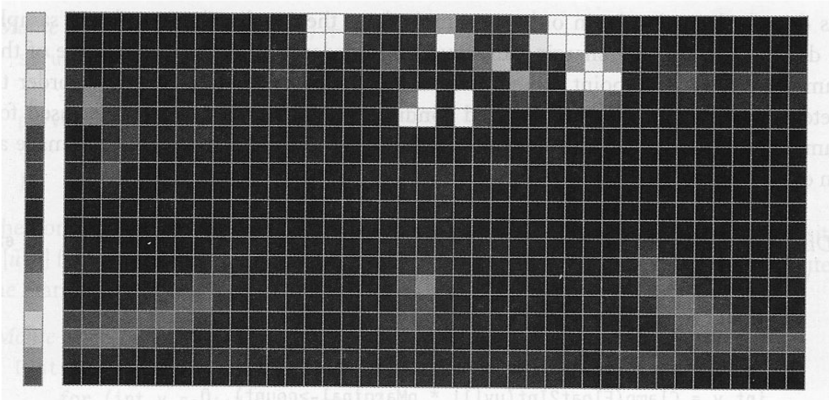
  - piecewise-constant 1D function
  - defined by n_v values $p_v[\tilde{v}]$
- conditional density
  - $p_u(u|v) = \frac{p(u,v)}{p_v(v)} = \frac{1}{I_f}\frac{f[\tilde{u},\tilde{v}]}{p[\tilde{v}]}$
  - piecewise-constant 1D function
- sample generation
  - see example 3 of the inversion method

Piecewise-Constant 2D Distribution

- environment map



- low-resolution of the marginal density function and the con-ditional distributions for rows
- first, a row is selected according to the marginal density function
- then, a column is selected from the row's 1D conditional distribution



Paul Debevec, Grace Cathedral

[Pharr, Humphreys]

Outline

- path tracing
- brute-force path tracing
- direct illumination
- indirect illumination

Introduction

- reflected light is indirect illumination
- indirect illumination is difficult / expensive to capture
- indirect illumination is important for photo-realistic effects
  - e. g., caustics, diffuse interreflections

Sampling

- Monte-Carlo estimator for indirect illumination

  $L_o(p,\omega_o) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{p(\omega_i)} f_r(p,\omega_i,\omega_o) L_r(r_c(p,\omega_i),-\omega_i)\cos\theta_i$
- importance sampling
  - uniform sampling (less efficient)
  - proportional to the cosine factor (useful for diffuse surfaces)
  - proportional to the BRDF (useful for glossy surfaces)
  - proportional to the incident radiance (usually unknown, but can be determined by other techniques, e. g. photon mapping)
  - a combination of these factors

Summary

- generate light transport paths (a chain of rays) from visible surface points to light sources
  - rays are traced recursively until hitting a light source
- recursively evaluate the rendering equation along a path
- ray generation in a path is governed by
  - light sources
  - BRDFs
- recursion depth is generally governed by the amount of radiance along a ray
- distinguishes direct and indirect illumination
  - direct: emitted radiance from light sources
  - indirect: reflected radiance from surfaces (and light sources)