Advanced Computer Graphics
Stochastic Raytracing

Matthias Teschner
Outline

– Context
– Diffuse vs. general global illumination
– Monte Carlo integration
– Sampling of random variables
- Radiosity equation governs light transport for diffuse surfaces. ⇒ How to describe light transport for general surfaces?
- How to solve for the light transport?
- How to compute the relevant part of the light transport towards a sensor?
Context

– Light transport towards the sensor requires to solve

\[ L(p \rightarrow \omega_o) = \int_S f_r(p, \omega_i \leftrightarrow \omega_o) L(p' \rightarrow -\omega_i) G(p, p') dA_{p'} \]

– Monte Carlo integration approximates this integral
  – E.g., \( L(p \rightarrow \omega_o) \approx \sum_i f_r(p, \omega_i \leftrightarrow \omega_o) L(p' \rightarrow -\omega_i) G(p, p') A_{p'} \)
  – Send rays into the hemisphere
  – Associate an area / solid angle with each ray
  – Compute radiance along this ray
  – Sum up all contributions
Outline

– Context
– Diffuse vs. general global illumination
– Monte Carlo integration
– Sampling of random variables
Governing Equations

- Rendering equation
  - Governing equation for general global illumination methods
    \[ L(p \rightarrow \omega_o) = L_e(p \rightarrow \omega_o) + \int_S f_r(p, \omega_i \leftrightarrow \omega_o) L(x \rightarrow -\omega_i) V(p, x) \frac{\cos(\omega_i, n_p) \cos(-\omega_i, n_x)}{r_{px}^2} \, dA_x \]
  - Radiosity equation
    - Governing equation for diffuse global illumination methods
      \[ L(p \rightarrow \omega_o) = \frac{B(p)}{\pi} \quad f_r(p, \omega_i \leftrightarrow \omega_o) = \frac{\rho(p)}{\pi} \]
      \[ B(p) = B_e(p) + \frac{\rho(p)}{\pi} \int_S B(x)V(p, x) \frac{\cos(\omega_i, n_p) \cos(-\omega_i, n_x)}{r_{px}^2} \, dA_x \]
A Solution Strategy (Radiosity)

- Finite Element Method (FEM)
- Start with a continuous form / function
  \[ B(p) = B_e(p) + \frac{\rho(p)}{\pi} \int_S B(x) V(p, x) \frac{\cos(\omega_i, n_p) \cos(-\omega_i, n_x)}{r^2_{px}} dA_x \]
- Discretization
  \[ B = B_e + FB \]
  \[ B = (I - F)^{-1} B_e \]
- Solving for a vector with unknown radiosities
  \[ (I - F)^{-1} = \sum_{k=0}^{\infty} F^k \]
  \[ B = B_e + FB_e + FFB_e + FFFB_e + \ldots \]
An Alternative Strategy

– Start with the general form of the rendering equation, e.g. in hemispherical form

\[ L(p \rightarrow \omega_o) = L_e(p \rightarrow \omega_o) + \int_{\Omega} f_r(p, \omega_i \leftrightarrow \omega_o) L(p \leftrightarrow \omega_i) \cos(\omega_i, n_p) d\omega_i \]

– Solving for a function with unknown radiances \( L(p \rightarrow \omega_o) \)
  – I.e., radiance at all surface positions into all directions
Operator Form of the Rendering Equation

- A linear operator transforms a function into another one
- Scattering operator
  - \((K h)(p \rightarrow \omega_o) = \int_{\Omega} f_r(p, \omega_i \leftrightarrow \omega_o) h(p \leftarrow \omega_i) \cos(\omega_i, n_p) d\omega_i\)
  - Applied to an incident radiance function \(L(p \leftarrow \omega_i)\), exitant radiance resulting from one scattering step is returned
  - \(L(p \rightarrow \omega_o) = (KL)(p \leftarrow \omega_i)\)
  - \(K\) operates on an entire function, i.e. on all incident radiances for all positions \(p\) and direction \(\omega_i\)

Operator Form of the Rendering Equation

- Propagation operator
  
  - \((Gh)(p \leftarrow \omega_i) = h(p' \rightarrow -\omega_i)\) \(p'\) indicates the raycast operator applied to \(p\)
  
  - Applied to an existant radiance function \(L(p' \rightarrow -\omega_i)\), incident radiance at \(p\) from direction \(\omega_i\) is returned
  
  - \(L(p \leftarrow \omega_i) = (GL)(p' \rightarrow -\omega_i)\)
  
  - Radiance is preserved / propagated along the line between \(p\) and \(p'\)
  
  - \(p\) and \(p'\) can be reversed, i.e. \(L(p' \leftarrow -\omega_o) = (GL)(p \rightarrow \omega_o)\)

Operator Form of the Rendering Equation

- Light transport operator
  - \( T = KG \)
  - Composition of scattering and propagation
  - Maps an exitant radiance function the exitant radiance function after one scattering step
  - Remember: \( G \) maps exitant radiance to incident radiance propagated along a direction. Then, \( K \) maps incident radiance to exitant radiance after scattering

Operator Form of the Rendering Equation

- \( L(p \rightarrow \omega_o) = L_e(p \rightarrow \omega_o) + \int_{\Omega} f_r(p, \omega_i \leftrightarrow \omega_o) L(p \leftarrow \omega_i) \cos(\omega_i, n_p) \, d\omega_i \)
- Can be written as
  \[
  L(p \rightarrow \omega_o) = L_e(p \rightarrow \omega_o) + (KGL)(p \rightarrow \omega_o)
  \]
- Light transport equation
  - \( L = L_e + TL \) Infinite number of equations with an infinite number of unknown exitant radiances
  - Relates exitant radiance functions
  - Represents the light propagation equilibrium
Light Transport Equation

- \( L = L_e + TL \)
- Solving for the unknown radiance function
  - \((I - T)L = L_e\)
  - \(L = (I - T)^{-1}L_e\)
  - Neumann series
  - \(L = \sum_{k=0}^{\infty} (T^k L_e)\)
    \(\approx L_e + TL_e + T^2L_e + T^3L_e + \ldots\)
Light Transport Equation

– Discussion
  – Radiance function is linear with respect to emission
    \[ L = (I - T)^{-1}(L_{e,1} + L_{e,2}) = (I - T)^{-1}L_{e,1} + (I - T)^{-1}L_{e,2} \]
  – Radiance function is a sum of
    – Emitted radiance \( L_e \)
    – Emitted radiance after one reflection \( TL_e \)
    – Emitted radiance after two reflections \( TTL_e \)
    – ...
  – \( L \approx L_e + TL_e + TTL_e + TTTL_e + \ldots \)
Terms in the Neumann Series

– Example contributions to terms
Forward Raytracing

– Send rays / propagate radiance from all light source positions into all directions $\Rightarrow L_e$
– At all intersection points $p$, solve the integral
  \[ L_1(p \to \omega_o) = \int_{\Omega} f_r(p, \omega_i \leftrightarrow \omega_o) GL_e \cos(\omega_i, n_p) d\omega_i \]
  for all direction $\omega_o \Rightarrow TL_e$
– Send rays to propagate $TL_e$
– At all intersection points $p$, solve the integral
  \[ L_2(p \to \omega_o) = \int_{\Omega} f_r(p, \omega_i \leftrightarrow \omega_o) GL_1 \cos(\omega_i, n_p) d\omega_i \]
  for all direction $\omega_o \Rightarrow TTL_e$
Forward Raytracing

- At a sensor: Accumulate radiance contributions of rays after n scattering steps, i.e. compute $L_e + TL_e + T^2L_e + \ldots$
Backward Raytracing

- Send rays from the sensor into the scene
- Propagate radiance from visible light sources
- $\Rightarrow$ part of $L_e$ visible to the sensor
- At intersection points $p$ with the scene, compute radiance $L(p \rightarrow \omega_o) = \int_{\Omega} f_r(p, \omega_i \leftrightarrow \omega_o) L_e(p \leftarrow \omega_i) \cos(\omega_i, n_p) d\omega_i$ that is propagated in direction $\omega_o$ towards the sensor
- $\Rightarrow$ part of $TL_e$ visible to the sensor
- ...
Backward Raytracing

– Trace rays from the sensor into the scene
Setting at Sensor

- How to compute $L(p_1 \leftarrow \omega_o)$ and what is its relation to $L_e + TL_e + TTL_e + \ldots$
Setting at First-Level Intersections

\[ L(p_1 \rightarrow \omega_o) = \int_S f_r(p_1, \omega_i \leftrightarrow \omega_o)L(p_2 \rightarrow -\omega_i)G(p_1, p_2)dA_{p_2} \]

\[ = \int_{\text{Light Sources}} f_r(p_1, \omega_i \leftrightarrow \omega_o)L_e(p_2 \rightarrow -\omega_i)G(p_1, p_2)dA_{p_2} \]

\[ + \int_{\text{Scene}} f_r(p_1, \omega_i \leftrightarrow \omega_o)L(p_2 \rightarrow -\omega_i)G(p_1, p_2)dA_{p_2} \]

\[ \int_{\text{Light Sources}} \ldots \text{ is the part of } TL_e \text{ visible to the sensor} \]

\[ \text{Computation of } \int_{\text{Scene}} \ldots \text{ requires } L(p_2 \rightarrow -\omega_i) \]
Setting at Second-Level Intersections

\[ L(p_2 \rightarrow \omega_o) = \int_S f_r(p_2, \omega_i \leftrightarrow \omega_o) L(p_3 \rightarrow -\omega_i) G(p_2, p_3) dA_{p_3} \]

\[ = \int_{\text{Light Sources}} f_r(p_2, \omega_i \leftrightarrow \omega_o) L_e(p_3 \rightarrow -\omega_i) G(p_2, p_3) dA_{p_3} \]

\[ + \int_{\text{Scene}} f_r(p_2, \omega_i \leftrightarrow \omega_o) L(p_3 \rightarrow -\omega_i) G(p_2, p_3) dA_{p_3} \]

- \( \int_{\text{Light Sources}} \cdots \) is the part of \( L_e \) visible to the sensor
- Computation of \( \int_{\text{Scene}} \cdots \) requires \( L(p_3 \rightarrow -\omega_i) \)

\[ L(p_2 \leftarrow \omega_i) = \]

\[ L(p_3 \rightarrow -\omega_i) \]

towards \( p_1 \)
Summary

- Recursive evaluation of

\[ L(p \rightarrow \omega_o) = \int_S f_r(p, \omega_i \leftrightarrow \omega_o) L(p' \rightarrow -\omega_i) G(p, p') dA_{p'} \]

\[ = \int_{\text{Light Sources}} f_r(p, \omega_i \leftrightarrow \omega_o) L_e(p' \rightarrow -\omega_i) G(p, p') dA_{p'} \]

\[ + \int_{\text{Scene}} f_r(p, \omega_i \leftrightarrow \omega_o) L(p' \rightarrow -\omega_i) G(p, p') dA_{p'} \]

- Each recursion level computes parts of the functions

\[ L_e, TL_e, T^2L_e, \ldots \] that are visible to the sensor
Numerical Integration

– The integral $\int_s \ldots$ is approximately computed with a sum of samples $\sum_i \ldots$
– For each sample $i$,
  – A ray is cast into the scene
  – Intersection with the scene is computed
  – Radiance along the ray is computed
Numerical Integration

– Typically, $\int S \cdots = \int_{\text{Scene}} \cdots + \int_{\text{Light Sources}} \cdots \approx \sum_{\text{Scene}_i} \cdots + \sum_{\text{Light Source}_i} \cdots$ is considered
– For $\sum_{\text{Light Source}_i} \cdots$, light source areas are sampled and rays towards those positions are processed
– For $\sum_{\text{Scene}_i} \cdots$, the respective solid angle is sampled and rays towards those directions are processed
Numerical Integration

- Due to the recursive nature, the number of processed rays grows exponentially with the recursion level
- ⇒ Monte Carlo integration
  - Efficient for multidimensional integral
  - Very flexible in terms of the number of used samples
  - Adaptive sample distribution
  - Even one sample can be used to approximate an integral
  - ⇒ e.g., Path tracing
    - At each recursion level, send a fixed number of rays to light sources and one ray into the scene (which generates a ray path)
Outline

– Context
– Diffuse vs. general global illumination
– Monte Carlo integration
– Sampling of random variables
Goal

- Approximating the solution of the light transport equation \( L = \sum_{k=0}^{\infty} (T^k L_e) \)
- Recursive evaluation of

\[
L(p \rightarrow \omega_o) = \int_{S} f_r(p, \omega_i \leftrightarrow \omega_o) L(p' \rightarrow -\omega_i) G(p, p') dA_{p'}
\]

\[
= \int_{\text{Light Sources}} f_r(p, \omega_i \leftrightarrow \omega_o) L_e(p' \rightarrow -\omega_i) G(p, p') dA_{p'}
\]

\[
+ \int_{\text{Scene}} f_r(p, \omega_i \leftrightarrow \omega_o) L(p' \rightarrow -\omega_i) G(p, p') dA_{p'}
\]

- Each recursion level computes parts of the functions \( L_e, TL_e, T^2L_e, \ldots \) that are visible to the sensor
Numerical Integration – Fixed Sample Size

- E.g. Riemann sum
  \[ \int_a^b f(x)\,dx \approx \sum_i f(x_i)\Delta x \quad \Delta x = \frac{b-a}{N} \]
- More / smaller samples ⇒ better accuracy
- \( d \) dimensional integrals require \( N^d \) samples
Numerical Integration – Adaptive Sample Size

- E.g., Monte Carlo integration
  \[ \int_a^b f(x) \, dx \approx \sum_i f(x_i) \Delta x_i, \]  
  adaptive sample size \( \Delta x_i \)
- More / smaller samples \( \Rightarrow \) better accuracy
- \( d \)-dimensional integrals work with arbitrary sample numbers
- Sample size is only approximated \( \Rightarrow \) noise
Stochastic Raytracing - Concept

- Approximately evaluate the integral
  \[ \int_{\Omega} f_r(p, \omega_i \leftrightarrow \omega_o)L(p \leftarrow \omega_i) \cos(\omega_i, n_p) d\omega_i \]
  by
  - Tracing rays into randomly sampled 2D directions
  - Computing the incoming radiances

- Integral is approximated with
  \[ \sum_{i} f_r(p, \omega_i \leftrightarrow \omega_o)L(p \leftarrow \omega_i) \cos(\omega_i, n_p) \Delta \Omega_i \]
  - 2 dimensional sample directions \( \omega_i = (\theta_i, \phi_i) \)
  - \( \Delta \Omega_i \) is an approximation of the solid angle of sample direction \( \omega_i = (\theta_i, \phi_i) \)
Introduction

– Challenges
  – Approximate the integral as exact as possible
  – Trace as few rays as possible / use as few samples as possible
  – Trace relevant rays / use relevant samples
    – Rays / samples to light sources are very relevant
    – For diffuse surfaces, rays / samples in normal direction are more relevant than rays / samples perpendicular to the normal
    – For specular surfaces, rays / samples in reflection direction are relevant
Properties

– Benefits
  – Processes only evaluations of the integrand at arbitrary surface points in the domain
  – Appropriate for integrals of arbitrary dimensions
  – Allows for non-uniform sample patterns / adaptive sample sizes
  – Works for a large variety of integrands, e.g., it handles discontinuities
Properties

- Drawbacks
  - Using $n$ samples, the scheme converges to the correct result with $O(n^{1/2})$
  - I.e., to half the error, $4n$ samples are required
  - Errors are perceived as noise, i.e. pixels are arbitrarily too bright or dark (due to the erroneous approximation of the sample size)
  - Evaluation of the integrand at a point is expensive (ray intersections tests)
Continuous Random Variables

- Motivation: Random sampling of directions
- Continuous random variables $X$
  - In contrast to discrete random variables, infinite number of possible values
- Canonical uniform random variable $0 \leq \xi < 1$
  - Sample sets with arbitrary distributions can be computed from $\xi$
Probability Density Function PDF \( p(x) \)

- Motivation: PDF governs the size / solid angle of a sample / sample direction
- Probability of a random variable taking certain value ranges
  - \( p(x) \geq 0 \quad \forall x \in [a, b] \)
  - \( \int_{a}^{b} p(x) \, dx = 1 \)
  - \( Pr(x_0 \leq X \leq x_1) = \int_{x_0}^{x_1} p(x) \, dx \)
- Example
  - Uniform PDF for \( 0 \leq X \leq 5 \)
  - \( 1 = \int_{0}^{5} p(x) \, dx = p(x) \int_{0}^{5} \, dx = 5 \, p(x) \)
  - \( p(x) = \frac{1}{5} \)
Cumulative Distribution Function CDF $P(x)$

- Motivation: CDFs are required to generate sample sets for arbitrary PDFs from uniform sample sets.
- Probability of a random variable to be less or equal to $x$:
  
  $P(x) = Pr(X \leq x) = \int_{a}^{x} p(x)dx$

- $P(a) = 0 \leq P(x) \leq 1 = P(b)$

- $Pr(x_0 \leq X \leq x_1) = P(x_1) - P(x_0)$
Expected Value

- Motivation: expected value of an estimator function is equal to the integral in the reflectance equation
- Expected value $E_p[f(x)]$ of a function $f(x)$ is defined as the weighted average value of the function over a domain $D$
  \[ E_p[f(x)] = \int_D f(x) \, p(x) \, dx \text{ with } \int_D p(x) \, dx = 1 \]
- Properties
  - $E[af(x)] = aE[f(x)]$
  - $E[\sum_i f(X_i)] = \sum_i E[f(X_i)]$ For independent random variables $X_i$
Expected Value

- Examples for uniform PDF \( p(x) \)
  
  - \( f(x) = \cos(x) \quad D = [0, \pi] \quad p(x) = \frac{1}{\pi} \)
    
    \[ E_p[\cos(x)] = \int_0^\pi \cos(x) \frac{1}{\pi} \, dx = \frac{1}{\pi}(-\sin \pi + \sin 0) = 0 \]
  
  - \( f(x) = x \quad D = [0, 6] \quad p(x) = \frac{1}{6} \)
    
    \[ E_p[x] = \int_0^6 x \frac{1}{6} \, dx = \frac{1}{6}(\frac{6^2}{2} - 0) = 3 \]
  
  - \( f(x) \)

\[
E_p[f(x)] = \frac{1}{b-a} \int_a^b f(x) \, dx
\]

\[
\int_a^b f(x) \, dx = E_p[f(x)](b-a)
\]
Monte Carlo Estimator - Uniform Random Variables

- Motivation: approximation of the integral in the reflectance equation
- Goal: computation of \( \int_{a}^{b} f(x) \, dx \)
- Uniformly distributed random variables \( X_i \in [a, b] \)
- Probability density function \( p(x) = \frac{1}{b-a} \) Constant and integration to one
- Monte Carlo estimator \( F_N = \frac{b-a}{N} \sum_{i=1}^{N} f(X_i) \)
- Expected value of \( F_N \) is equal to the integral \( \int_{a}^{b} f(x) \, dx \)
  \[- E[F_N] = \int_{a}^{b} f(x) \, dx \]
Monte Carlo Estimator - Uniform Random Variables

\[ E[F_N] = E\left[\frac{b-a}{N} \sum_{i=1}^{N} f(X_i)\right] \]

\[ = \frac{b-a}{N} \sum_{i=1}^{N} E[f(X_i)] \]

\[ = \frac{b-a}{N} \sum_{i=1}^{N} \int_{a}^{b} f(x)p(x)dx \]

\[ = \frac{b-a}{N} \sum_{i=1}^{N} \int_{a}^{b} f(x) \frac{1}{b-a} dx \]

\[ = \frac{1}{N} \sum_{i=1}^{N} \int_{a}^{b} f(x)dx \]

\[ = \int_{a}^{b} f(x)dx \]
Monte Carlo Estimator - Uniform Random Variables

- PDF \( p(x) = \frac{1}{b-a} \)
- Estimator \( F_N = \frac{b-a}{N} \sum_{i=1}^{N} f(X_i) \)
- Integral
  - \( \int_{a}^{b} f(x)dx \approx \frac{b-a}{N} \sum_{i=1}^{N} f(X_i) = \sum_{i=1}^{N} f(X_i) \frac{b-a}{N} = \sum_{i=1}^{N} f(X_i) \frac{1}{N \cdot p(X_i)} \)
- Function value \( f(X_i) \)
- Approximate sample size \( \frac{1}{N \cdot p(X_i)} \)
Examples - Uniform Random Variables

- Integral \( \int_0^1 5x^4 \, dx = 1 \)
- Estimator \( F_N = \frac{1-0}{N} \sum_{i=1}^{N} 5X_i^4 \) Sample size approx. 1/N
- For an increasing number of uniformly distributed random variables \( x_i \), the estimator converges to one

\[
F_N = \frac{b-a}{N} \sum_{i=1}^{N} f(X_i) \\
F_N = (b-a) \frac{1}{N} \sum_{i=1}^{N} f(X_i) \\
F_N = (b-a)f(x) \\
E[F_N] = \int_a^b f(x) \, dx
\]

Uniformly distributed random samples [Suffern]
Monte Carlo Estimator - Non-uniform Random Variables

- Monte Carlo estimator  \( F_N = \frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{p(X_i)} \quad p(X_i) \neq 0 \)

- \( E[F_N] = E \left[ \frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{p(X_i)} \right] \)
  \[ = \frac{1}{N} \sum_{i=1}^{N} \int_{a}^{b} \frac{f(x)}{p(x)} p(x) \, dx \]
  \[ = \frac{1}{N} \sum_{i=1}^{N} \int_{a}^{b} f(x) \, dx \]
  \[ = \int_{a}^{b} f(x) \, dx \]
Monte Carlo Estimator - Non-uniform Random Variables

- PDF \( p(x) \)
- Estimator \( F_N = \frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{p(X_i)} \)
- Integral
  - \( \int_{a}^{b} f(x) \, dx \approx \frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{p(X_i)} = \sum_{i=1}^{N} f(X_i) \frac{1}{N \, p(X_i)} \)
  - Function value \( f(X_i) \)
  - Approximate sample size \( \frac{1}{N \, p(X_i)} \)
Sample Size

- Sample size / distance for uniform PDF: \( \approx \frac{b-a}{N} = \frac{1}{Np(X_i)} \)

- Sample size for non-uniform PDF: \( \approx \frac{1}{Np(X_i)} \)
Monte Carlo Estimator - Multiple Dimensions

- Samples $X_i$ are multidimensional
- E.g., $\int_{x_0}^{x_1} \int_{y_0}^{y_1} \int_{z_0}^{z_1} f(x, y, z) dx dy dz$
- Uniformly distributed random samples $(x_0, y_0, z_0) \leq X_i = (x_i, y_i, z_i) \leq (x_1, y_1, z_1)$
- Probability density function $p(X_i) = \frac{1}{x_1-x_0} \frac{1}{y_1-y_0} \frac{1}{z_1-z_0}$
- Monte Carlo estimator $F_N = \frac{(x_1-x_0)(y_1-y_0)(z_1-z_0)}{N} \sum_{i=1}^{N} f(X_i)$
- Approximate sample volume is $\frac{(x_1-x_0)(y_1-y_0)(z_1-z_0)}{N}$
Monte Carlo Estimator - Integration over a Hemisphere

- Approximate computation of the irradiance at a point

\[ E_i(p) = \int_{2\pi} L_i(p, \omega) \cos \theta d\omega \]

\[ = \int_0^{2\pi} \int_0^{\pi / 2} L_i(p, \theta, \phi) \cos \theta \sin \theta d\theta d\phi \]

- Estimator

\[ F_N = \frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{p(X_i)} = \frac{1}{N} \sum_{i=1}^{N} \frac{L_i(p, \theta_i, \phi_i) \cos \theta_i \sin \theta_i}{p(\theta_i, \phi_i)} \]

- Choosing a PDF

  - Should be similar to the shape of the integrand
  - As incident radiance is weighted with \( \cos \theta \), it is appropriate to generate more samples close to the top of the hemisphere
  - \( p(\theta, \phi) \propto \cos \theta \)

This flexibility is an important aspect of Monte Carlo integration.
Monte Carlo Estimator - Integration over a Hemisphere

- Probability distribution

\[ \int_{2\pi}^{2\pi} c \, p(\omega) d\omega = 1 \]
\[ \int_{0}^{2\pi} \int_{0}^{\frac{\pi}{2}} c \, \cos \theta \sin \theta \, d\theta \, d\phi = 1 \]
\[ c \, \frac{2\pi}{1+1} = 1 \]
\[ c = \frac{1}{\pi} \]
\[ p(\theta, \phi) = \frac{\cos \theta \sin \theta}{\pi} \]

- Estimator

\[ F_N = \frac{1}{N} \sum_{i=1}^{N} \frac{L_i(p, \theta_i, \phi_i) \cos \theta_i \sin \theta_i}{p(\theta_i, \phi_i)} \]
\[ = \frac{\pi}{N} \sum_{i=1}^{N} L_i(p, \theta_i, \phi_i) \approx \int_{0}^{2\pi} \int_{0}^{\frac{\pi}{2}} L_i(p, \theta, \phi) \cos \theta \sin \theta \, d\theta \, d\phi \]

If \( \theta \) and \( \phi \) are sampled according to PDF \( p(\theta, \phi) \)
Monte Carlo Estimator - Integration over a Hemisphere

- Integral \( \int_0^{2\pi} \int_0^{\frac{\pi}{2}} L_i(p, \theta, \phi) \cos \theta \sin \theta \, d\theta \, d\phi \)
- PDF \( p(\theta, \phi) = \frac{\cos \theta \sin \theta}{\pi} \)
- Estimator \( \frac{\pi}{N} \sum_{i=1}^{N} L_i(p, \theta_i, \phi_i) \)
  \[ = \sum_{i=1}^{N} L_i(p, \theta_i, \phi_i) \cos \theta_i \sin \theta_i \frac{\pi}{N \cos \theta_i \sin \theta_i} \]
- Function value \( L_i(p, \theta_i, \phi_i) \cos \theta_i \sin \theta_i \) for direction \((\theta_i, \phi_i)\)
- Approximate sample size / solid angle \( \frac{\pi}{N \cos \theta_i \sin \theta_i} \)
Monte Carlo Integration - Steps

- Choose an appropriate probability density function
- Generate random samples according to the PDF
- Evaluate the function for all samples
- Accumulate sample values weighted with their approximate sample size
Monte Carlo Estimator - Variance / Error Reduction

- Importance sampling
  - Motivation: contributions of larger sample values are more important
  - PDF should be similar to the shape of the function
  - Optimal PDF \( p(x) = \frac{f(x)}{\int f(x)dx} \)
  - E.g., if incident radiance is weighted with \( \cos \theta \), the PDF should choose more samples close to the normal direction

[Suffern]
Monte Carlo Estimator - Variance / Error Reduction

- Stratified sampling
  - Domain subdivision into strata
  - E.g., handling direct and indirect illumination differently

\[
L(p \rightarrow \omega_o) = \int_S f_r(p, \omega_i \leftrightarrow \omega_o) L(p' \rightarrow -\omega_i) G(p, p') dA_{p'}
\]

\[
= \int_{\text{Light Sources}} f_r(p, \omega_i \leftrightarrow \omega_o) L_e(p' \rightarrow -\omega_i) G(p, p') dA_{p'}
+ \int_{\text{Scene}} f_r(p, \omega_i \leftrightarrow \omega_o) L(p' \rightarrow -\omega_i) G(p, p') dA_{p'}
\]
Outline

- Context
- Diffuse vs. general global illumination
- Monte Carlo integration
- Sampling of random variables
  - Inversion method
  - Rejection method
  - Transforming between distributions
  - 2D sampling
  - Examples
Inversion Method

– Mapping of a uniform random variable to a goal distribution

– Discrete example
  – Four outcomes with probabilities $p_1, p_2, p_3, p_4$ and $\sum_i p_i = 1$
  – Computation of the cumulative distribution function $P(i) = \sum_{j=1}^i p_j$

[Pharr, Humphreys]
Inversion Method

– Discrete example cont.
  – Take a uniform random variable $\xi$
  – $P^{-1}(\xi)$ has the desired distribution

– Continuous case
  – $P$ and $P^{-1}$ are continuous functions
  – Start with the desired PDF $p(x)$
  – Compute $P(x) = \int_{0}^{x} p(x')dx'$
  – Compute the inverse $P^{-1}(x)$
  – Obtain a uniformly distributed variable
  – Compute $X_i = P^{-1}(\xi)$ which adheres to $p(x)$

[Pharr, Humphreys]
Inversion Method - Example 1

- Power distribution \( p(x) \propto x^n \)
  - E.g., for sampling the Blinn microfacet model
- Computation of the PDF
  - \( \int_0^1 c \, x^n \, dx = 1 \Rightarrow c \left. \frac{x^{n+1}}{n+1} \right|_0^1 = 1 \Rightarrow c = n + 1 \)
- PDF \( p(x) = (n + 1)x^n \)
- CDF \( P(x) = \int_0^x p(x') \, dx' = x^{n+1} \)
- Inverse of the CDF \( P^{-1}(x) = \frac{n+\sqrt{x}}{n+1} \)
- Sample generation
  - Generate uniform random samples \( 0 \leq \xi \leq 1 \)
  - \( X = \frac{n+\sqrt{\xi}}{n+1} \) are samples from the distribution \( p(x) = (n + 1)x^n \)
Inversion Method - Example 2

- Exponential distribution $p(x) \propto e^{-ax}$
  - E.g., for considering participating media
- Computation of the PDF
  - $\int_0^\infty c \ e^{-ax} \, dx = \left. -\frac{c}{a} \ e^{-ax} \right|_0^\infty = \frac{c}{a} = 1$
- PDF $p(x) = a \ e^{-ax}$
- CDF $P(x) = \int_0^x p(x') \, dx' = 1 - e^{-ax}$
- Inverse of the CDF $P^{-1}(x) = -\frac{\ln(1-x)}{a}$
- Sample generation
  - Generate uniform random samples $0 \leq \xi \leq 1$
  - $X = -\frac{\ln(1-\xi)}{a}$ are samples from the distribution $p(x) = a \ e^{-ax}$
Inversion Method - Example 3

- Piecewise-constant distribution
  - E.g., for environment lighting
    \[
    f(x) = \begin{cases} 
    v_0 & x_0 \leq x < x_1 \\
    v_1 & x_1 \leq x < x_2 \\
    \vdots & \vdots 
    \end{cases} \\
    x_i = \Delta \cdot i \\
    \Delta = \frac{1}{N}
    \]

- PDF
  \[ p(x) = \frac{1}{c} f(x) \]
  with
  \[ c = \int_0^1 f(x) \, dx = \sum_{i=0}^{N-1} \Delta \cdot v_i = \sum_{i=0}^{N-1} \frac{v_i}{N} \]

[Pharr, Humphreys]
Inversion Method - Example 3

- **CDF**
  
  \[ P(x_0) = 0 \]
  
  \[ P(x_1) = \int_{x_0}^{x_1} p(x) \, dx = \Delta \cdot \frac{v_0}{c} = \frac{v_0}{N_c} = P(x_0) + \frac{v_0}{N_c} \]
  
  \[ P(x_2) = \int_{x_0}^{x_2} p(x) \, dx = \int_{x_0}^{x_1} p(x) \, dx + \int_{x_1}^{x_2} p(x) \, dx = P(x_1) + \frac{v_1}{N_c} \]
  
  \[ P(x_i) = P(x_{i-1}) + \frac{v_{i-1}}{N_c} \]

- CDF is linear between \( x_i \) and \( x_{i+1} \) with slope \( \frac{v_i}{c} \)

- **Sample generation**
  
  - Generate uniform random samples \( 0 \leq \xi \leq 1 \)
  
  - Compute \( x_i \) with \( P(x_i) \leq \xi \) and \( \xi < P(x_{i+1}) \)
  
  - Compute \( d \) with \( P(x_i) + d(P(x_{i+1}) - P(x_i)) = \xi \)
  
  - \( X = x_i + d(x_{i+1} - x_i) = x_i + \frac{d}{N} \) are samples from \( p(x) = \frac{1}{c}f(x) \)
Outline

- Context
- Diffuse vs. general global illumination
- Monte Carlo integration
- Sampling of random variables
  - Inversion method
  - Rejection method
  - Transforming between distributions
  - 2D sampling
  - Examples
Rejection Method

- Draws samples according to a function $f(x)$
  - Dart-throwing approach
  - Works with a PDF $p(x)$ and a scalar $c$ with $f(x) < c \cdot p(x)$

- Properties
  - $f(x)$ is not necessarily a PDF
  - PDF, CDF and inverse CDF do not have to be computed
  - Simple to implement
  - Useful for debugging purposes

[Pharr, Humphreys]
Rejection Method

- Sample generation
  - Generate a uniform random sample $0 \leq \xi < 1$
  - Generate a sample $X$ according to $p(x)$
  - Accept $X$ if $\xi \cdot c \cdot p(X) \leq f(X)$

[Pharr, Humphreys]
Outline

– Context
– Diffuse vs. general global illumination
– Monte Carlo integration
– Sampling of random variables
  – Inversion method
  – Rejection method
  – Transforming between distributions
  – 2D sampling
  – Examples
Transforming Between Distributions

- Computation of a resulting PDF, when a function is applied to samples from an arbitrary distribution
  - Random variables $X_i$ are drawn from $p_x(x)$
  - Bijective transformation (one-to-one mapping) $Y_i = y(X_i)$
  - How does the distribution $p_y(y)$ look like?
Transforming Between Distributions

- \( Pr\{Y \leq y(x)\} = Pr\{X \leq x\} \)

\[
P_y(y) = P_y(y(x)) = P_x(x)
\]

\[
p_y(y) = \frac{p_x(x)}{|y'(x)|}
\]

- **Example**  
  \( p_x(x) = 2x \quad 0 \leq x \leq 1 \)

  - \( y(x) = \sin x \quad x(y) = \arcsin y \)
  - \( y'(x) = \cos x \)
  - \( p_y(y) = \frac{p_x(x)}{|\cos x|} = \frac{2x}{|\cos x|} = \frac{2 \arcsin y}{|\cos \arcsin(y)|} = \frac{2 \arcsin y}{\sqrt{1-y^2}} \)
Transforming Between Distributions

- Multiple dimensions
  - $X_i$ is an $n$-dimensional random variable
  - $Y_i = T(X_i)$ is a bijective transformation
- Transformation of the PDF

\[ p_y(y) = \frac{p_x(x)}{|J_T(x)|} \quad J_T(x) = \begin{pmatrix} \frac{\partial T_1}{\partial x_1} & \cdots & \frac{\partial T_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial T_n}{\partial x_1} & \cdots & \frac{\partial T_n}{\partial x_n} \end{pmatrix} \]
Transforming Between Distributions

- Example (polar coordinates)
  - Samples \((r, \theta)\) with density \(p(r, \theta)\)
  - Corresponding density \(p(x, y)\) with \(x = r \cos \theta\) and \(y = r \sin \theta\)

\[
J_T(x) = \begin{pmatrix}
\cos \theta & -r \sin \theta \\
\sin \theta & r \cos \theta
\end{pmatrix}
\]

\(|J_T(x)| = r(\cos^2 \theta + \sin^2 \theta) = r\)

- \(p(x, y) = \frac{1}{r} p(r, \theta)\)
  - \(p(r, \theta) = r \cdot p(x, y)\)
Transforming Between Distributions

- Example (spherical coordinates)
  - \( x = r \sin \theta \cos \phi \)
  - \( y = r \sin \theta \sin \phi \)
  - \( z = r \cos \theta \)
  - \( p(r, \theta, \phi) = r^2 \sin \theta \cdot p(x, y, z) \)

- Example (solid angle)
  - \( Pr\{\omega \in \Omega\} = \int_{\Omega} p(\omega)d\omega \)
  - \( d\omega = \sin \theta \ d\theta \ d\phi \)
  - \( p(\theta, \phi) = \sin \theta \cdot p(\omega) \)
Outline

– Context
– Diffuse vs. general global illumination
– Monte Carlo integration
– Sampling of random variables
  – Inversion method
  – Rejection method
  – Transforming between distributions
  – 2D sampling
  – Examples