Advanced Computer Graphics
Stochastic Raytracing

Matthias Teschner
Outline

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

- Radiosity equation governs light transport for diffuse surfaces. \( \Rightarrow \) 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
  - \(L(p \rightarrow \omega_o) = L_e(p \rightarrow \omega_o) + \int_S f_r(p, \omega_i \leftrightarrow \omega_o) L(x \rightarrow -\omega_i)V(p, x) \frac{\cos(\omega_i, n_p) \cos(-\omega_i, n_x)}{r^2_{px}} dA_x\)

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

- Finite Element Method (FEM)
- Start with a continuous form / function
  \[ B(p) = B_e(p) + \frac{\rho(p)}{\pi} \int_S B(x) V(p, x) \frac{\cos(\omega_i, n_p) \cos(-\omega_i, n_x)}{r^{2 \rho x}} dA_x \]
- 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

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

– Solving for a function with unknown radiances \( L(p \to \omega_o) \)
  – 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
  - \((\mathbf{K} h)(p \rightarrow \omega_o) = \int_{\Omega} f_r(p, \omega_i \leftrightarrow \omega_o) h(p \leftarrow \omega_i) \cos(\omega_i, n_p) d\omega_i\)
  - Applied to an incident radiance function \(L(p \leftarrow \omega_i)\), exitant radiance resulting from one scattering step is returned
  - \(L(p \rightarrow \omega_o) = (\mathbf{K} L)(p \leftarrow \omega_i)\)
  - \(\mathbf{K}\) 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
  - \((\mathbf{G}h)(p \leftarrow \omega_i) = h(p' \rightarrow -\omega_i)\) \(p'\) indicates the raycast operator applied to \(p\)
  - Applied to an exitant radiance function \(L(p' \rightarrow -\omega_i)\), incident radiance at \(p\) from direction \(\omega_i\) is returned
  - \(L(p \leftarrow \omega_i) = (\mathbf{GL})(p' \rightarrow -\omega_i)\)
  - Radiance is preserved / propagated along the line between \(p\) and \(p'\)
  - \(p\) and \(p'\) can be reversed, i.e. \(L(p' \leftarrow -\omega_o) = (\mathbf{GL})(p \rightarrow \omega_o)\)

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
- \(L = \sum_{k=0}^{\infty} (T^k L_e)\)
  \(\approx L_e + T L_e + T^2 L_e + T^3 L_e + \ldots\)
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 \( T^2L_e \)
    - ...
  - \( L \approx L_e + TL_e + T^2L_e + T^3L_e + \ldots \)
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
  
  $$L_1(p \rightarrow \omega_o) = \int_\Omega f_r(p, \omega_i \leftrightarrow \omega_o) GL_e \cos(\omega_i, n_p) d\omega_i$$

  for all direction $\omega_o \Rightarrow TL_e$
- Send rays to propagate $TL_e$
- At all intersection points $p$, solve the integral
  
  $$L_2(p \rightarrow \omega_o) = \int_\Omega f_r(p, \omega_i \leftrightarrow \omega_o) GL_1 \cos(\omega_i, n_p) d\omega_i$$

  for all direction $\omega_o \Rightarrow TTTL_e$
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 the relation to $L_e \leftarrow \omega_o$.
Setting at First-Level Intersections

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

- \(\int_{\text{Light Sources}} \) \(\ldots \) is the part of TL\(_e\) visible to the sensor
- Computation of \(\int_{\text{Scene}} \) \(\ldots \) requires \(L(p_2 \rightarrow -\omega_i)\)
Setting at Second-Level Intersections

\[
- \quad L(p_2 \rightarrow \omega_o) = \int_S f_r(p_2, \omega_i \leftrightarrow \omega_o)L(p_3 \rightarrow -\omega_i)G(p_2, p_3)dA_{p_3} \\
\quad = \int_{\text{Light Sources}} f_r(p_2, \omega_i \leftrightarrow \omega_o)L_\epsilon(p_3 \rightarrow -\omega_i)G(p_2, p_3)dA_{p_3} \\
+ \int_{\text{Scene}} f_r(p_2, \omega_i \leftrightarrow \omega_o)L(p_3 \rightarrow -\omega_i)G(p_2, p_3)dA_{p_3}
\]

- \int_{\text{Light Sources}} \ldots \text{ is the part of} \ TT L_\epsilon \text{ visible to the sensor}

- Computation of \int_{\text{Scene}} \ldots \text{requires} \ L(p_3 \rightarrow -\omega_i)

\[
L(p_2 \leftrightarrow \omega_i) = \frac{L(p_2 \rightarrow \omega_o)}{L(p_3 \rightarrow -\omega_i)}
\]
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
  – 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