EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

FACULTY OF
SCIENCE
Computer Graphics

Computer Graphics
(Graphische Datenverarbeitung)

- Light Transport 2 -

WS 2021/2022

Corona

- Regular random lookup of the 3G certificates

- Contact tracing: We need to know who is in the class room
  - New ILIAS group for every lecture slot
  - Register via ILIAS or this QR code (only if you are present in this room)

Overview

- Previous lecture
  - Physics behind ray tracing
  - Physical light quantities
  - Perception of light
  - Light sources

- Today
  - Repetition
  - Light transport simulation
  - Radiosity
  - Path Tracing

- Next Lectures
  - Reflection models
  - Texturing

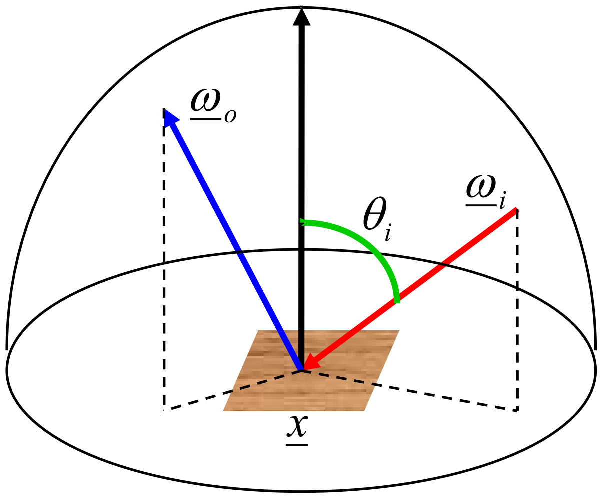
The Rendering Equation

How to express the nature of global illumination?
(The single, most important formula)

Surface Reflectance

$$L(x,\omega_o) = L_e(x,\omega_o) + \int_{\Omega} f_r(\omega_o,x,\omega_i)L\,(x,\omega_i)\ \cos\theta_i\ \ d\omega_i$$

- Visible surface radiance $L(\underline{x},\underline{\omega}_o)$
  - Surface position $\underline{x}$
  - Outgoing direction $\underline{\omega}_o$
  - Incoming illumination direction $\underline{\omega}_i$
- Self-emission $L_e(\underline{x},\underline{\omega}_o)$
- Reflected light
  - Incoming radiance from all directions $L_i(\underline{x},\underline{\omega}_i)$
  - Direction-dependent reflectance (BRDF: bidirectional reflectance distribution function) $f_r(\underline{\omega}_i,\underline{x},\underline{\omega}_o)$

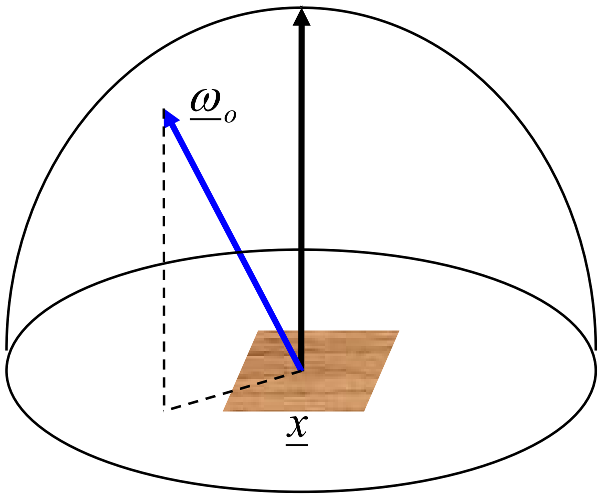


Surface Reflectance

- emittance only

$$L(x, \omega_o) = L_e(x, \omega_o)$$

- Visible surface radiance $L(\underline{x}, \underline{\omega}_o)$
  - Surface position $\underline{x}$
  - Outgoing direction $\underline{\omega}_o$
  - Incoming illumination direction $\underline{\omega}_i$
- Self-emission $L_e(\underline{x}, \underline{\omega}_o)$
- Reflected light
  - Incoming radiance from all directions $L_i(\underline{x}, \underline{\omega}_i)$
  - Direction-dependent reflectance (BRDF: bidirectional reflectance distribution function) $f_r(\underline{\omega}_i, \underline{x}, \underline{\omega}_o)$



Surface Reflectance

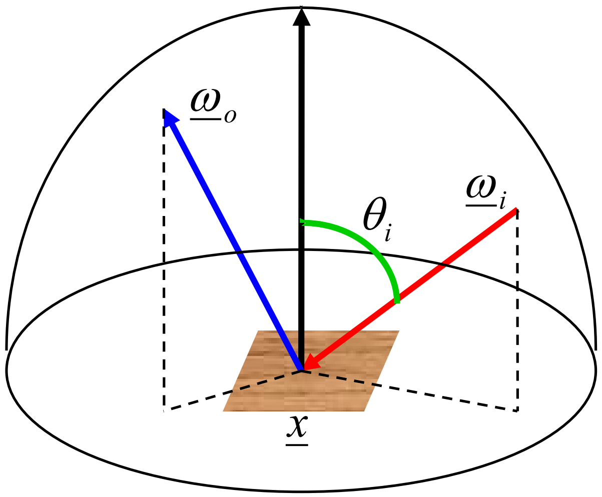
- No emittance, single light source

$$L(x,\omega_o) = \qquad f_r(\omega_o,x,\omega_i)L(x,\omega_i)\cos\theta_i$$

- Visible surface radiance $L(\underline{x},\underline{\omega}_o)$
  - Surface position $\underline{x}$
  - Outgoing direction $\underline{\omega}_o$
  - Incoming illumination direction $\underline{\omega}_i$

- Self-emission $L_e(\underline{x},\underline{\omega}_o)$

- Reflected light
  - Incoming radiance from all directions $L_i(\underline{x},\underline{\omega}_i)$
  - Direction-dependent reflectance (BRDF: bidirectional reflectance distribution function) $f_r(\underline{\omega}_i,\underline{x},\underline{\omega}_o)$

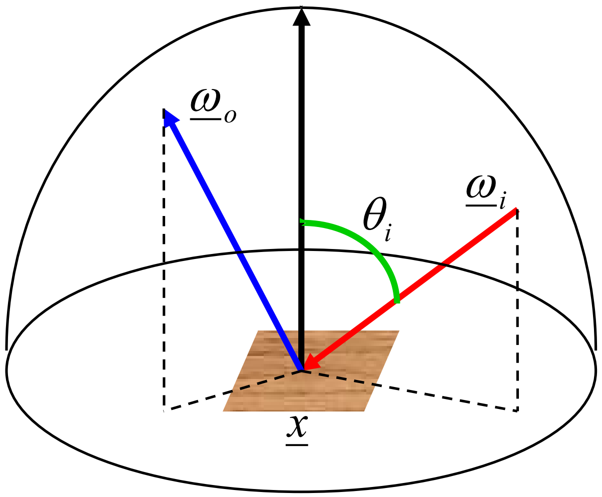


Surface Reflectance

- No emittance, single light source

$$L(x,\omega_o) = \qquad f_r(\omega_o, x, \omega_i) L(x, \omega_i) \cos\theta_i$$

- Visible surface radiance $L(\underline{x},\underline{\omega}_o)$
  - Surface position $\underline{x}$
  - Outgoing direction $\underline{\omega}_o$
  - Incoming illumination direction $\underline{\omega}_i$
- Self-emission $L_e(\underline{x},\underline{\omega}_o)$
- Reflected light
  - Incoming radiance from all directions $L_i(\underline{x},\underline{\omega}_i)$
  - Direction-dependent reflectance (BRDF: bidirectional reflectance distribution function) $f_r(\underline{\omega}_i,\underline{x},\underline{\omega}_o)$

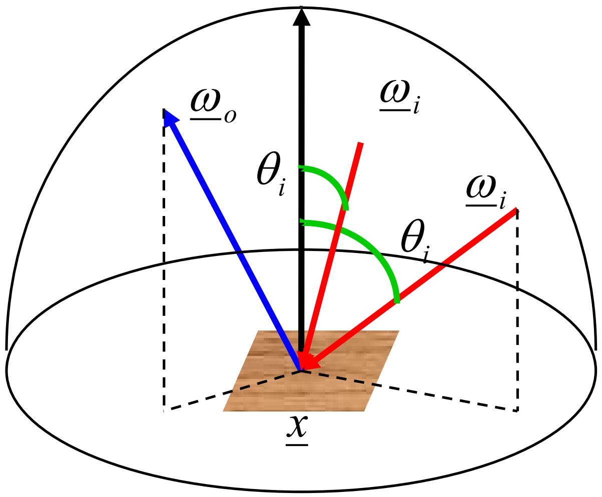


Surface Reflectance

- No emittance, two light sources

$$L(x,\omega_o) = \sum^{2} f_r(\omega_o, x, \omega_i) L(x,\omega_i) \cos\theta_i$$

- Visible surface radiance $L(\underline{x},\underline{\omega}_o)$
  - Surface position $\underline{x}$
  - Outgoing direction $\underline{\omega}_o$
  - Incoming illumination direction $\underline{\omega}_i$
- Self-emission $L_e(\underline{x},\underline{\omega}_o)$
- Reflected light
  - Incoming radiance from all directions $L_i(\underline{x},\underline{\omega}_i)$
  - Direction-dependent reflectance (BRDF: bidirectional reflectance distribution function) $f_r(\underline{\omega}_i,\underline{x},\underline{\omega}_o)$



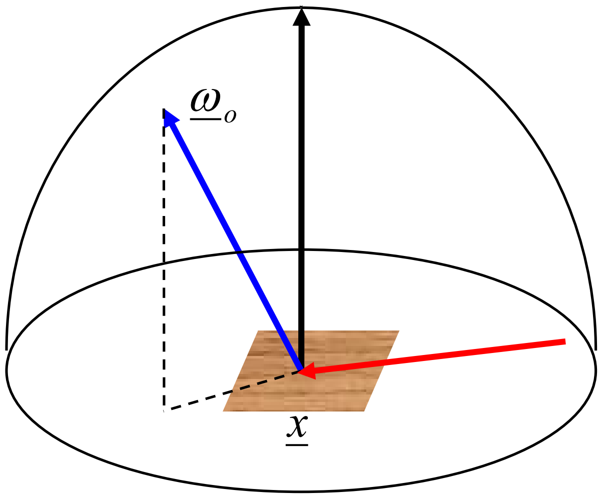
Surface Reflectance

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

- No emittance, all possible incident light directions

$$L(x, \omega_o) = \int_\Omega f_r(\omega_o, x, \omega_i) L(x, \omega_i)\ \cos\theta_i\ \ d\omega_i$$

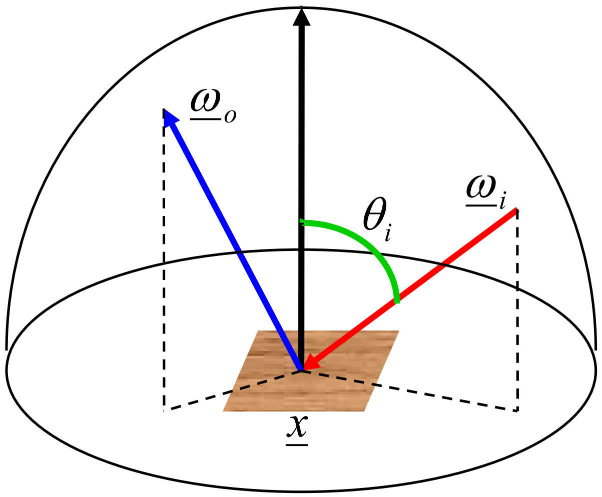
- Visible surface radiance $L(\underline{x}, \underline{\omega}_o)$
  - Surface position $\underline{x}$
  - Outgoing direction $\underline{\omega}_o$
  - Incoming illumination direction $\underline{\omega}_i$
- Self-emission $L_e(\underline{x}, \underline{\omega}_o)$
- Reflected light
  - Incoming radiance from all directions $L_i(\underline{x}, \underline{\omega}_i)$
  - Direction-dependent reflectance (BRDF: bidirectional reflectance distribution function) $f_r(\underline{\omega}_i, \underline{x}, \underline{\omega}_o)$



Surface Reflectance

$$L(x,\omega_o) = L_e(x,\omega_o) + \int_{\Omega} f_r(\omega_o, x, \omega_i) L\,(x,\omega_i)\ \cos\theta_i\ \ d\omega_i$$

- Visible surface radiance $L(\underline{x},\underline{\omega}_o)$
  - Surface position $\underline{x}$
  - Outgoing direction $\underline{\omega}_o$
  - Incoming illumination direction $\underline{\omega}_i$
- Self-emission $L_e(\underline{x},\underline{\omega}_o)$
- Reflected light
  - Incoming radiance from all directions $L_i(\underline{x},\underline{\omega}_i)$
  - Direction-dependent reflectance (BRDF: bidirectional reflectance distribution function) $f_r(\underline{\omega}_i,\underline{x},\underline{\omega}_o)$

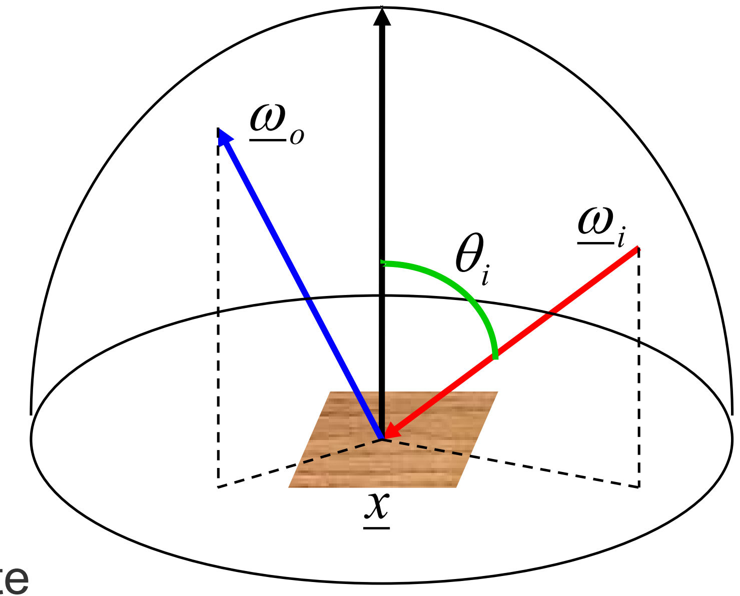


(Surface) Rendering Equation

- In Physics: Radiative Transport Equation
- Expresses energy equilibrium in scene

$$L(\underline{x},\underline{\omega}_o) = L_e(\underline{x},\underline{\omega}_o) + \int_\Omega f_r(\underline{\omega}_i,\underline{x},\underline{\omega}_o)\, L(\underline{x},\underline{\omega}_i) \cos\theta_i \, d\underline{\omega}_i$$

- total radiance = emitted radiance + reflected radiance
- First term: emissivity of the surface
  - non-zero only for light sources
- Second term: reflected radiance
  - integral over all possible incoming directions of irradiance times angle-dependent surface reflection function
- Fredholm integral equation of 2nd kind
  - unknown radiance appears on lhs and inside the integral
  - Numerical methods necessary to compute approximate solution

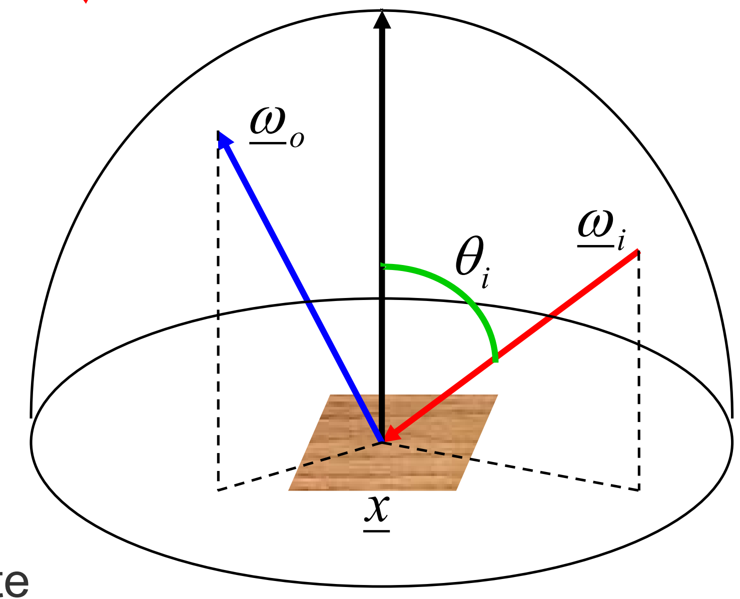


(Surface) Rendering Equation

- In Physics: Radiative Transport Equation
- Expresses energy equilibrium in scene

$$L(\underline{x},\underline{\omega}_o) = L_e(\underline{x},\underline{\omega}_o) + \int_{\Omega} f_r(\underline{\omega}_i,\underline{x},\underline{\omega}_o)\, L(\underline{x},\underline{\omega}_i) \cos\theta_i \, d\underline{\omega}_i$$

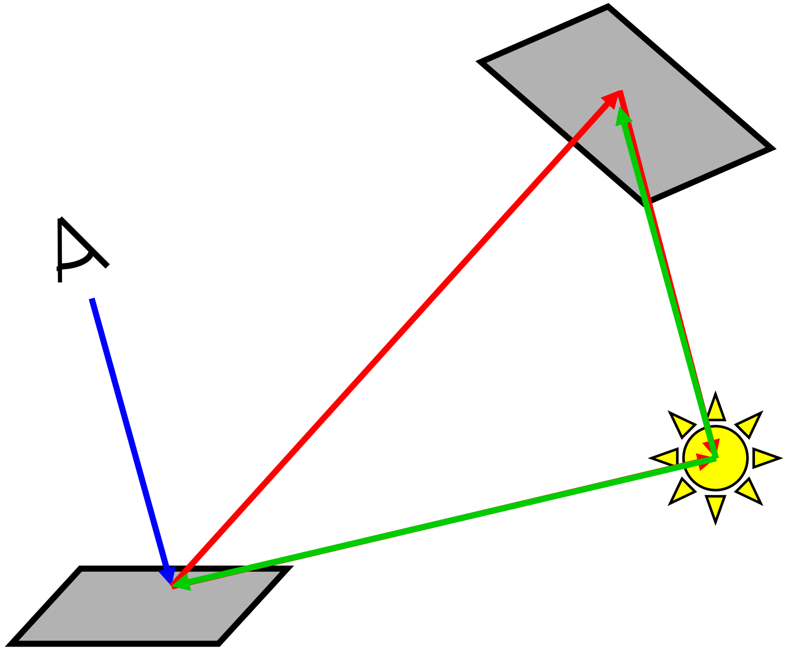
- total radiance = emitted radiance + reflected radiance
- First term: emissivity of the surface
  - non-zero only for light sources
- Second term: reflected radiance
  - integral over all possible incoming directions of irradiance times angle-dependent surface reflection function
- Fredholm integral equation of 2nd kind
  - unknown radiance appears on lhs and inside the integral
  - Numerical methods necessary to compute approximate solution



Ray Tracing

$$L(\underline{x},\underline{\omega}_o) = L_e(\underline{x},\underline{\omega}_o) + \int_\Omega f_r(\underline{\omega}_i,\underline{x},\underline{\omega}_o)\, L(\underline{x},\underline{\omega}_i) \cos\theta_i \, d\underline{\omega}_i$$

- Simple ray tracing
  - Illumination from light sources only - **local illumination (integral → sum)**
  - Evaluates angle-dependent reflectance function - **shading**
- Advanced Techniques
  - Distribution ray tracing
    - Multiple reflections/refractions (for specular surfaces)
  - Forward/Backward ray tracing
    - Stochastic sampling (Monte Carlo methods)
  - Photon mapping
  - ...



Rendering Equation II

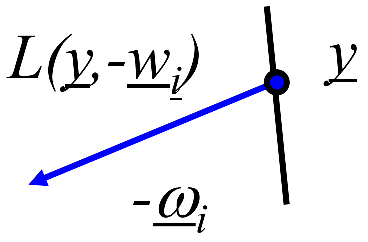
- Outgoing illumination at a point

$$L(\underline{x},\underline{\omega}_o) = L_e(\underline{x},\underline{\omega}_o) + L_r(\underline{x},\underline{\omega}_o)$$

$$= L_e(\underline{x},\underline{\omega}_o) + \int_{\Omega_+} f_r(\underline{\omega}_i,\underline{x},\underline{\omega}_o) L(\underline{x},\underline{\omega}_i) \cos\theta_i \, d\underline{\omega}_i$$

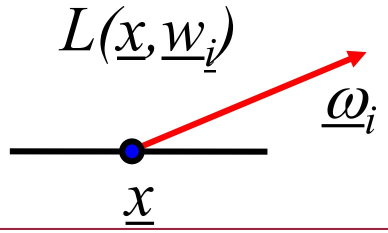
- Linking with other surface points
  - Incoming radiance at x is outgoing radiance at y

$$L_i(\underline{x},\underline{\omega}_i) = L(\underline{y},-\underline{\omega}_i) = L(RT(\underline{x},\underline{\omega}_i),-\underline{\omega}_i)$$



  - Ray-Tracing operator

$$\underline{y} = RT(\underline{x},\underline{\omega}_i)$$



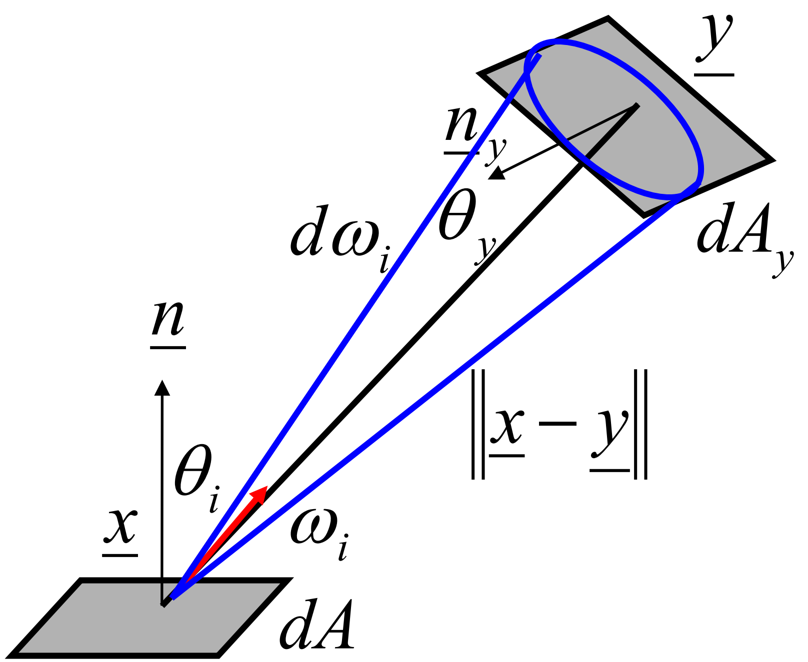
Rendering Equation III

- Directional parameterization

$$L(\underline{x},\underline{\omega}_o) = L_e(\underline{x},\underline{\omega}_o) + \int_{\Omega_+} f_r(\underline{\omega}_i,\underline{x},\underline{\omega}_o) L(\underline{y}(\underline{x},\underline{\omega}_i),-\underline{\omega}_i)\cos\theta_i \, d\omega_i$$

- Re-parameterization over surfaces *S*

$$d\omega_i = \frac{\cos\theta_y}{\|\underline{x}-\underline{y}\|^2} dA_y$$



$$L(\underline{x},\underline{\omega}_o) = L_e(\underline{x},\underline{\omega}_o) + \int_{\underline{y}\in S} f_r(\underline{\omega}_i,\underline{x},\underline{\omega}_o)\, L(\underline{y},\underline{\omega}_i(\underline{x},\underline{y})) V(\underline{x},\underline{y}) \frac{\cos\theta_i \cos\theta_y}{\|\underline{\mathrm{x}}-\underline{y}\|^2} dA_y$$

Rendering Equation IV

$$L(\underline{x},\underline{\omega}_o) = L_e(\underline{x},\underline{\omega}_o) + \int_{\underline{y}\in S} f_r(\underline{\omega}_i,\underline{x},\underline{\omega}_o)\, L(\underline{y},\underline{\omega}_i(\underline{x},\underline{y}))\, \mathrm{V}(\underline{\mathrm{x}},\underline{\mathrm{y}}) \frac{\cos\theta_i \cos\theta_y}{\left\|\underline{\mathrm{x}}-\underline{y}\right\|^2}\, dA_y$$

- Geometry term

$$G(\underline{x},\underline{y}) = V\left(\underline{x},\underline{y}\right) \frac{\cos\theta_i \cos\theta_y}{\|\,\underline{x}-\underline{y}\,\|^2}$$

- Visibility term

$$V(\underline{x},\underline{y}) = \begin{cases} 1 & \text{if visible} \\ 0 & \text{if not visible} \end{cases}$$

- Integration over all surfaces

$$L(\underline{x},\underline{\omega}_o) = L_e(\underline{x},\underline{\omega}_o) + \int_{\underline{y}\in S} f_r(\underline{\omega}_i,\underline{x},\underline{\omega}_o)\, L(\underline{y},\underline{\omega}_i(\underline{x},\underline{y}))\, G(\underline{x},\underline{y})\, dA_y$$

Global Light Transport

- Rendering equation models outgoing radiance at one point dependent all directions

$$L(x_0, \omega_o) = L_e(x, \omega_o) + \int_{\Omega} f_r(\omega_o, x, \omega_i) L\,(x_0, \omega_i)\ \cos\theta_i\ \ d\omega_i$$

Global Light Transport

- Rendering equation models outgoing radiance at one point dependent all directions
- Rendering equation models outgoing radiance at all points dependent all directions

$$L(x_0, \omega_o) = L_e(x, \omega_o) + \int_\Omega f_r(\omega_o, x, \omega_i) L\,(x_0, \omega_i)\ \cos\theta_i\ \ d\omega_i$$

$$L(x_1, \omega_o) = L_e(x, \omega_o) + \int_\Omega f_r(\omega_o, x, \omega_i) L\,(x_1, \omega_i)\ \cos\theta_i\ \ d\omega_i$$

$$\vdots$$

$$L(x_n, \omega_o) = L_e(x, \omega_o) + \int_\Omega f_r(\omega_o, x, \omega_i) L\,(x_n, \omega_i)\ \cos\theta_i\ \ d\omega_i$$

Global Light Transport

- Rendering equation models outgoing radiance at one point dependent all directions
- Rendering equation models outgoing radiance at all points dependent on all other points

$$L(x_0, \omega_o) = L_e(x, \omega_o) + \int_{y \in S} f_r(\omega_o, x, \omega_i) L\,(y, -\omega_i) G(x_0, y) dA$$

$$L(x_1, \omega_o) = L_e(x, \omega_o) + \int_{y \in S} f_r(\omega_o, x, \omega_i) L\,(y, -\omega_i) G(x_1, y) dA$$

$$\vdots$$

$$L(x_n, \omega_o) = L_e(x, \omega_o) + \int_{y \in S} f_r(\omega_o, x, \omega_i) L\,(y, -\omega_i) G(x_n, y) dA$$

Global Light Transport (discrete)

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

- Rendering equation models outgoing radiance at one point dependent all directions
- Rendering equation models outgoing radiance at all points dependent on all other points

$$L(x_0,\omega_o) = L_e(x,\omega_o) + \sum_{y\in S} f_r(\omega_o,x,\omega_i)L(y,\omega_i)G(x_0,y)dA$$

$$L(x_1,\omega_o) = L_e(x,\omega_o) + \sum_{y\in S} f_r(\omega_o,x,\omega_i)L(y,\omega_i)G(x_1,y)dA$$

$$\vdots$$

$$L(x_n,\omega_o) = L_e(x,\omega_o) + \sum_{y\in S} f_r(\omega_o,x,\omega_i)L(y,\omega_i)G(x_n,y)dA$$

Rendering Equation: Approximations

- Using RGB instead of full spectrum
  - follows roughly the eye's sensitivity
- Dividing scene surfaces into small patches
  - Assumes locally constant reflection, visibility, geometry terms
- Sampling hemisphere along finite, discrete directions
  - simplifies integration to summation
- Reflection function model
  - Parameterized function
    - ambient: constant, non-directional, background light
    - diffuse: light reflected uniformly in all directions
    - specular: light of higher intensity in mirror-reflection direction
  - Lambertian surface (only diffuse reflection) → Radiosity
- Approximations based on empirical foundations
  - An example: polygon rendering in OpenGL

Radiosity

Radiosity: Idea

- Simplification of the Rendering Equation
  - Assumption: no directional dependence (only diffuse reflections, Lambertian)
- Use Radiosity instead of Radiance:
  - Radiosity:

$$B(x) = \int_{\Omega} L_o(x,\omega)\cos\theta \, d\omega$$

- Use Reflectance instead of (angular-dependent) BRDF:

$$\rho(\underline{x}) = \int_0^{\pi/2}\int_0^{2\pi} f_r(\underline{\omega_i},\underline{x},\underline{\omega_o})\cos\theta \, d\omega = \pi \cdot f_r(\underline{x})$$

Radiosity Equation

- Rendering Equation (surface parameterization):

$$L(\underline{x},\underline{\omega}_o)=L_e(\underline{x},\underline{\omega}_o)+\int\limits_{\underline{y}\in S} f_r(\underline{\omega}_i,\underline{x},\underline{\omega}_o)\,L(\underline{y},\underline{\omega}_i(x,y))\,G(\underline{x},\underline{y})\,d\quad{}_{y}$$

- Step 1: constant reflectance

Def. of Irradiance

$$L(\underline{x},\underline{\omega}_o)=L_e(\underline{x},\underline{\omega}_o)+\rho(\underline{x})\boxed{\int\limits_{\underline{y}\in S} L(\underline{y},\underline{\omega}_i(x,y))\,G(\underline{x},\underline{y})\,dA_y}$$

- Step 2: Radiosity instead of Radiance:

$$B(\underline{x})=B_e(\underline{x})+\rho(\underline{x})\boxed{E(\underline{x})}$$

Radiosity Equation

- Rendering Equation (surface parameterization):

$$L(\underline{x},\underline{\omega}_o) = L_e(\underline{x},\underline{\omega}_o) + \int_{\underline{y}\in S} f_r(\underline{\omega}_i,\underline{x},\underline{\omega}_o)\, L(\underline{y},\underline{\omega}_i(x,y))\, G(\underline{x},\underline{y})\, d\;{}_{y}$$

- Step 1: constant reflectance

Def. of Irradiance

$$L(\underline{x},\underline{\omega}_o) = L_e(\underline{x},\underline{\omega}_o) + \rho(\underline{x}) \boxed{\int_{\underline{y}\in S} L(\underline{y},\underline{\omega}_i(x,y))\, G(\underline{x},\underline{y})\, dA_y}$$

- Step 2: Radiosity instead of Radiance:

$$B(\underline{x}) = B_e(\underline{x}) + \rho(\underline{x}) \int_{\underline{y}\in S} F(\underline{x},\underline{y})\, B(\underline{y})\, dA_y$$

using *form factor*

$$F(\underline{x},\underline{y}) = \frac{G(\underline{x},\underline{y})}{\pi} =$$ percentage of light leaving dA_y that arrives at dA

Solving the Radiosity Equation

- Step 1: represent integral equation by linear operators

$$B(\underline{x}) = B_e(\underline{x}) + \rho(\underline{x}) \int_{\underline{y} \in S} F(\underline{x}, \underline{y})\, B(\underline{y})\, dA_y$$

- Properties
  - Fredholm integral of 2^{nd} kind
  - Global linking
    - Potentially each point with each other
    - Often sparse systems (occlusions)
  - No consideration of volume effects!!

Solving the Radiosity Equation

- Step 1: represent integral equation by linear operators

$$B(\underline{x}) = B_e(\underline{x}) + \rho(\underline{x}) \int_{\underline{y} \in S} F(\underline{x}, \underline{y})\, B(\underline{y})\, dA_y$$

- Defining linear operator "$\circ$":

$$(F \circ B)(x) \equiv \int F(x, y) B(y)\, dy$$

  - acts on functions like matrices act on vectors
  - superposition principle
  - scaling and addition: $F \circ (\alpha f + \beta g) = \alpha (F \circ f) + \beta (F \circ g)$

- Radiosity Equation with linear operators:

$$B(x) = B_e(x) + (F \circ B)(x)$$

Global Light Transport

- Rendering equation models outgoing radiance at one point dependent all directions
- Rendering equation models outgoing radiance at all points dependent on all other points

$$L(x_0, \omega_o) = L_e(x, \omega_o) + \int_{y \in S} f_r(\omega_o, x, \omega_i) L\,(y, -\omega_i) G(x_0, y) dA$$

$$L(x_1, \omega_o) = L_e(x, \omega_o) + \int_{y \in S} f_r(\omega_o, x, \omega_i) L\,(y, -\omega_i) G(x_1, y) dA$$

$$\vdots$$

$$L(x_n, \omega_o) = L_e(x, \omega_o) + \int_{y \in S} f_r(\omega_o, x, \omega_i) L\,(y, -\omega_i) G(x_n, y) dA$$

Global Light Transport (Radiosity)

- Rendering equation models outgoing radiance at one point dependent all directions
- Rendering equation models outgoing radiance at all points dependent on all other points

$$L(x_0, \omega_o) = L_e(x, \omega_o) + \sum_{y \in S} \rho(x) L(y, -\omega_i) G(x_0, y) dA$$

$$L(x_1, \omega_o) = L_e(x, \omega_o) + \sum_{y \in S} \rho(x) L(y, -\omega_i) G(x_1, y) dA$$

$$\vdots$$

$$L(x_n, \omega_o) = L_e(x, \omega_o) + \sum_{y \in S} \rho(x) L(y, -\omega_i) G(x_n, y) dA$$

Global Light Transport (Radiosity)

- Rendering equation models outgoing radiance at one point dependent all directions
- Rendering equation models outgoing radiance at all points dependent on all other points

$$B(x_0) = B_e(x_0) + \sum_{x_j} F(x_0, x_j) B(x_j)$$

$$B(x_1) = B_e(x_1) + \sum_{x_j} F(x_1, x_j) B(x_j)$$

$$\vdots$$

$$B(x_n) = B_e(x_n) + \sum_{x_j} F(x_n, x_n) B(x_j)$$

Global Light Transport (Radiosity)

- Rendering equation models outgoing radiance at one point dependent all directions
- Rendering equation models outgoing radiance at all points dependent on all other points
- Discretizing and representing by vectors B and B_e and matrix F

$$B = B_e + FB$$

Solving the Radiosity Equation

- Step 2: apply Neumann Series

$$B(x) = B_e(x) + (F \circ B)(x)$$

$$(I \circ B) = B_e + (F \circ B)$$

$$\Rightarrow (I - F) \circ B = B_e$$

$$B = (I - F)^{-1} \circ B_e$$

$$\boxed{B = (I + F + F^2 + F^3 + \ldots) \circ B_e}$$

- Neumann series:

$$\frac{1}{1-x} = 1 + x + x^2 + \ldots$$

Solving the Radiosity Equation

- Step 2: apply Neumann Series

$$B(x) = B_e(x) + (F \circ B)(x)$$

$$(I \circ B) = B_e + (F \circ B)$$

$$\Rightarrow (I - F) \circ B = B_e$$

$$B = (I - F)^{-1} \circ B_e$$

$$\boxed{B = (I + F + F^2 + F^3 + \ldots) \circ B_e}$$

- Interpretation:
  Global Illumination as emitted light + once reflected light + twice reflected light + …

Solving the Radiosity Equation

- Successive approximation

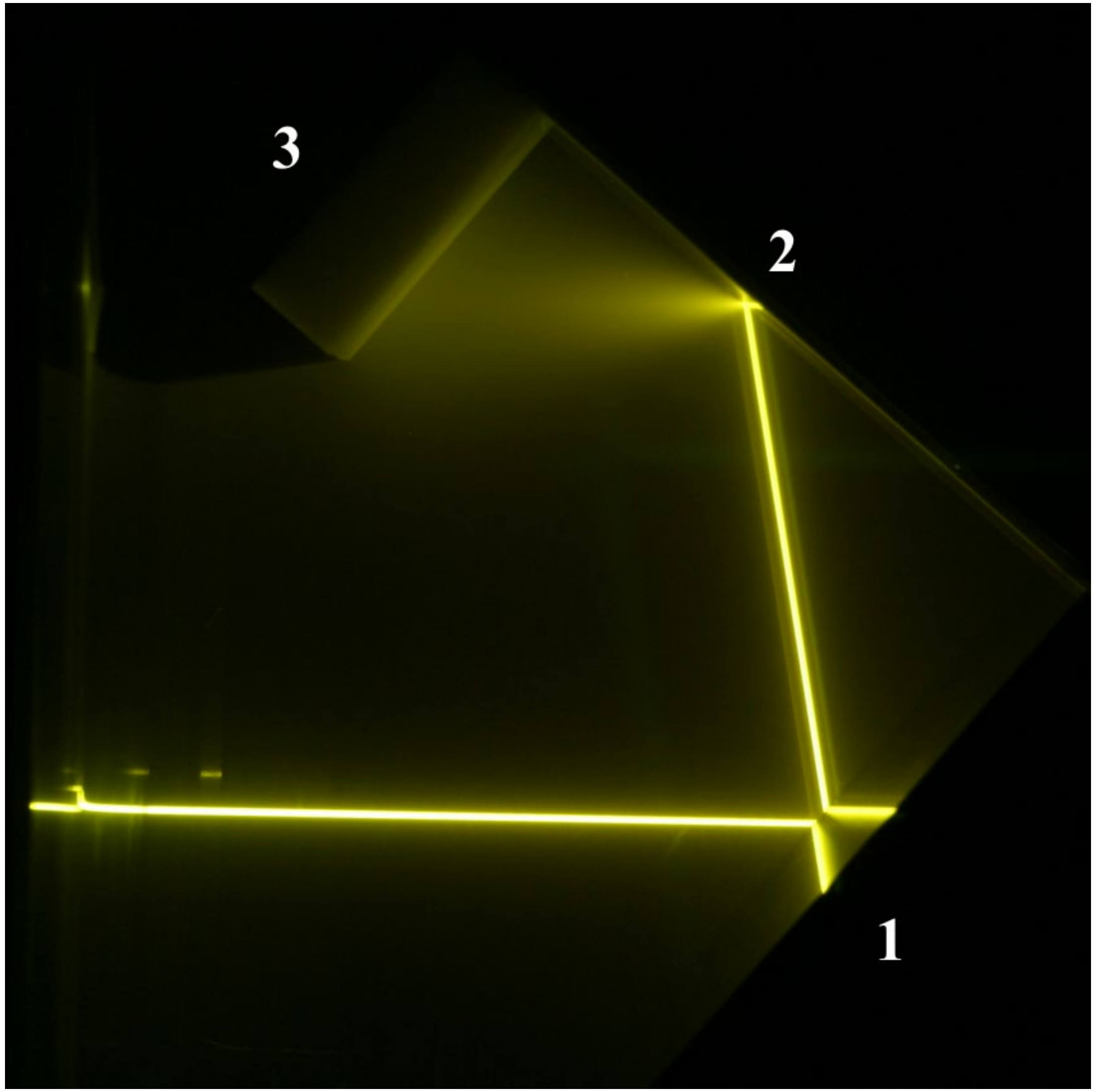
$$(I - F)^{-1} \circ B_e = B_e + F \circ B_e + F^2 \circ B_e + \ldots$$
$$= (B_e + F \circ (B_e + F \circ (B_e + \ldots$$

- Direct light from the light source: $B_1 = B_e$
- Light which is reflected and transported one time: $B_2 = B_e + F \circ B_1$

$\vdots$ $\qquad$ $\vdots$

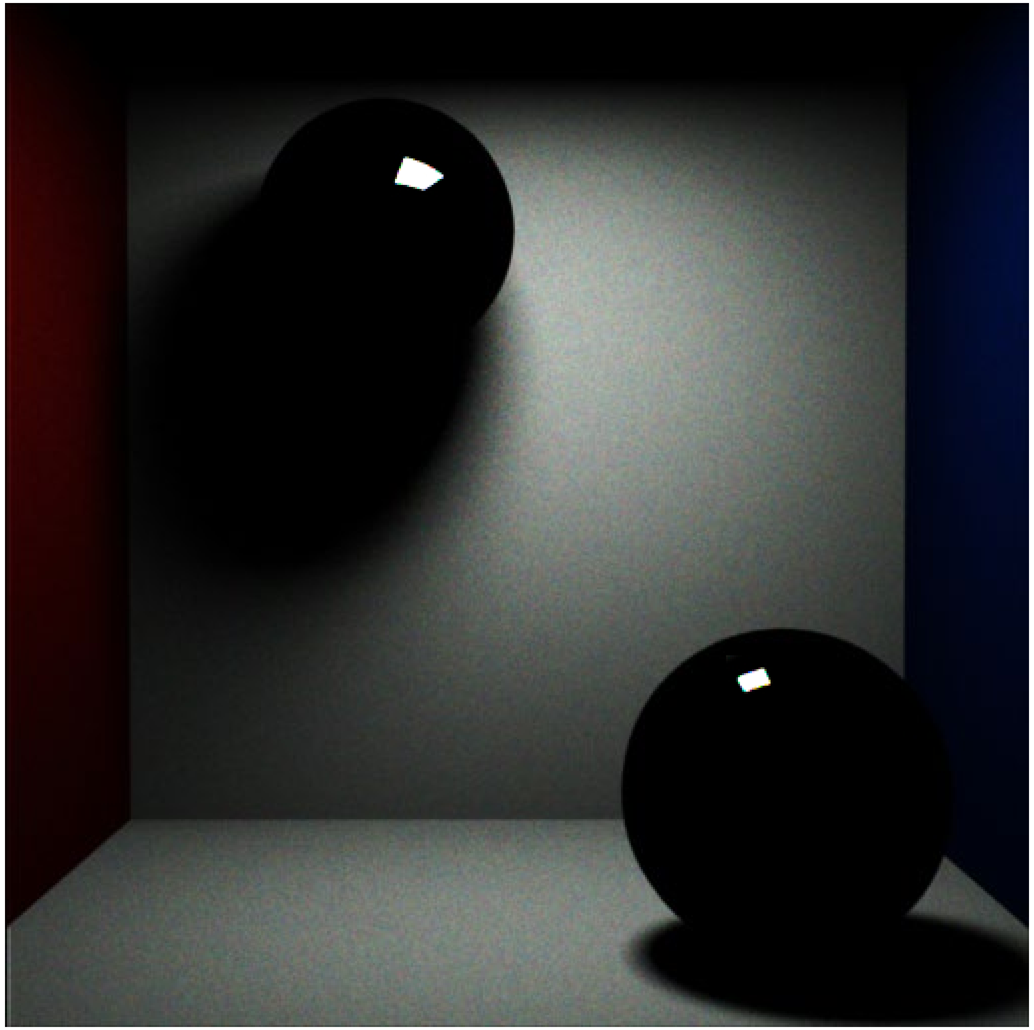
- Light which is reflected and transported n-times: $B_n = B_e + F \circ B_{n-1}$

Solving the Radiosity Equation

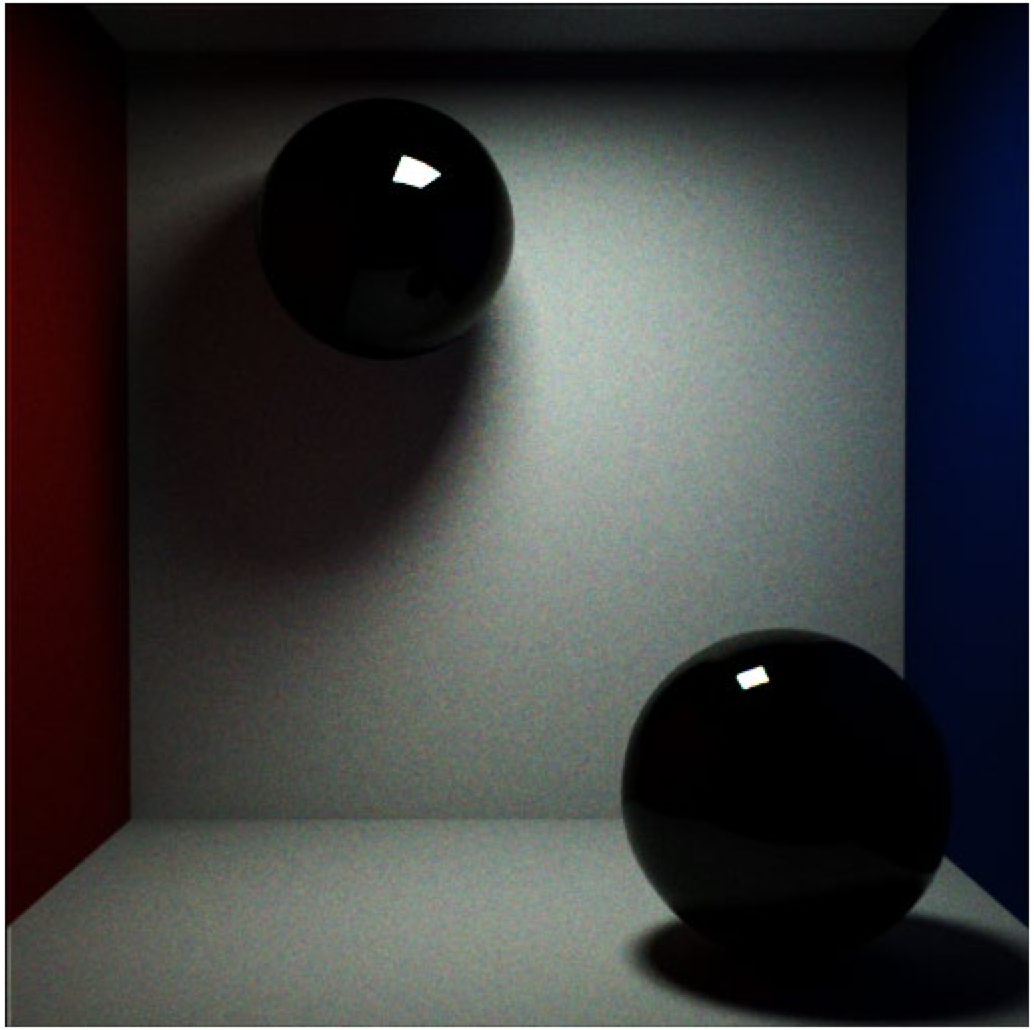


Path Tracing

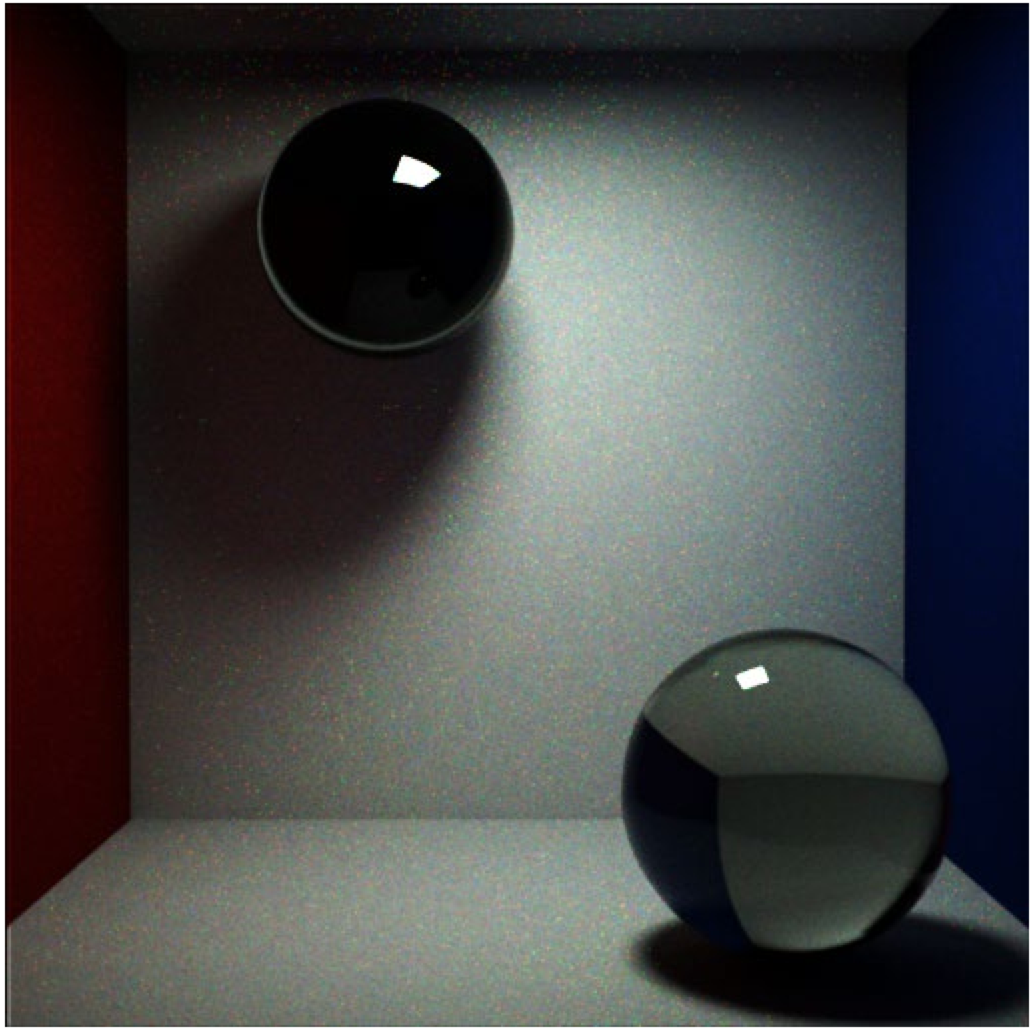
Second Example: Bounce 1



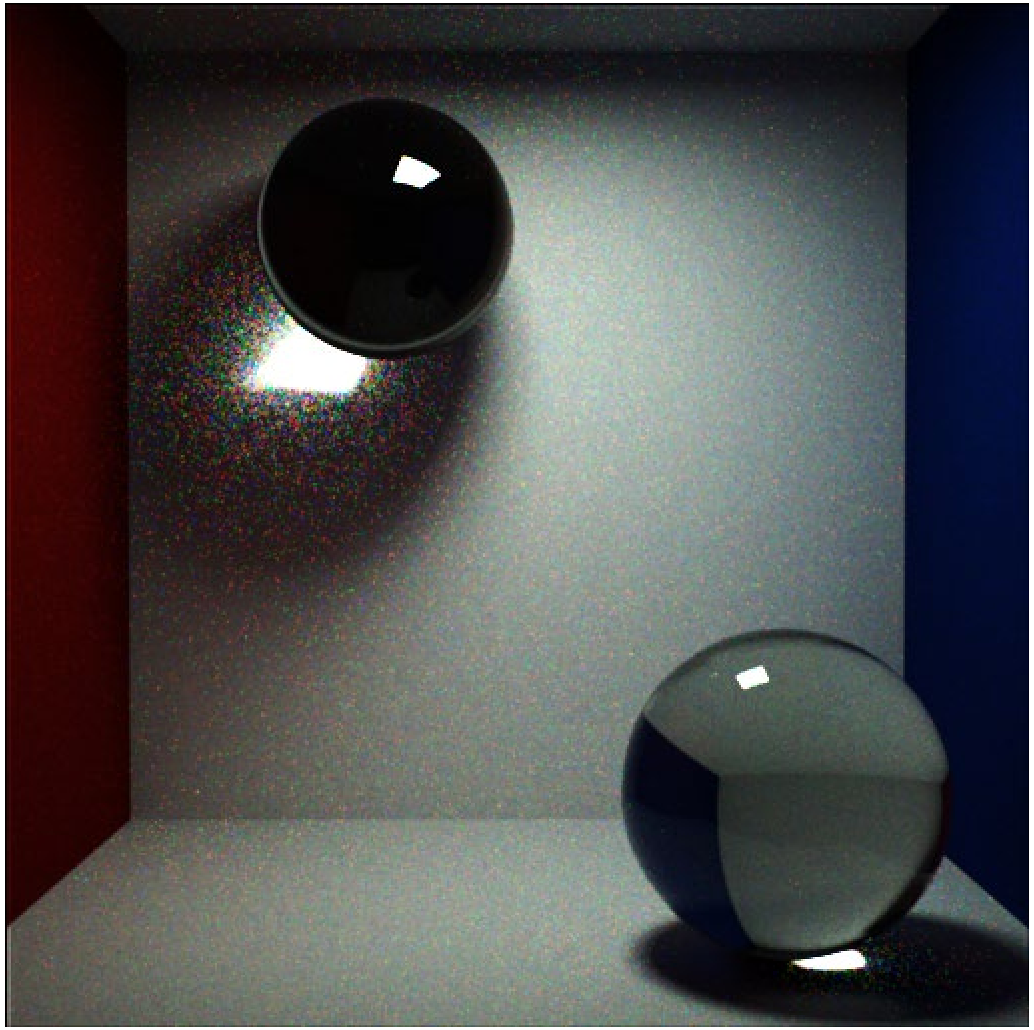
Second Example: Bounce 2



Second Example: Bounce 3



Second Example: Bounce 4

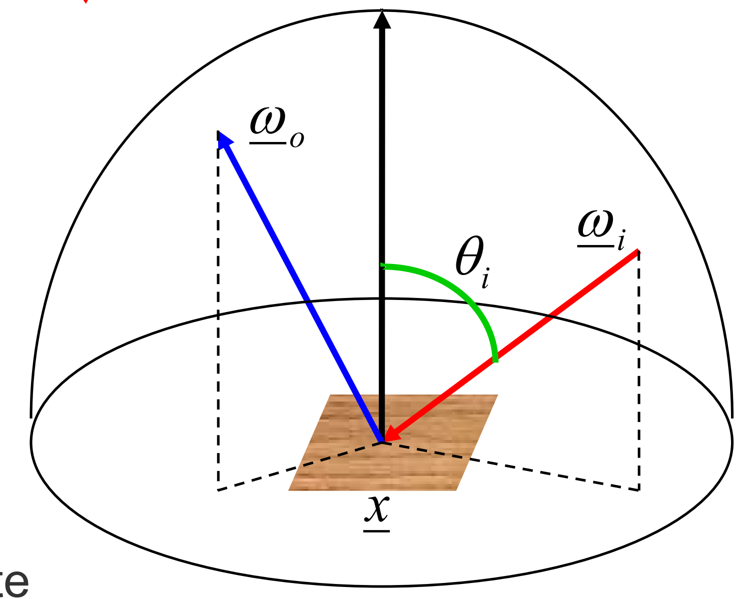


(Surface) Rendering Equation

- In Physics: Radiative Transport Equation
- Expresses energy equilibrium in scene

$$L(\underline{x},\underline{\omega}_o) = L_e(\underline{x},\underline{\omega}_o) + \int_{\Omega} f_r(\underline{\omega}_i,\underline{x},\underline{\omega}_o)\, L(\underline{x},\underline{\omega}_i) \cos\theta_i \, d\underline{\omega}_i$$

- total radiance = emitted radiance + reflected radiance
- First term: emissivity of the surface
  - non-zero only for light sources
- Second term: reflected radiance
  - integral over all possible incoming directions of irradiance times angle-dependent surface reflection function
- Fredholm integral equation of 2nd kind
  - unknown radiance appears on lhs and inside the integral
  - Numerical methods necessary to compute approximate solution



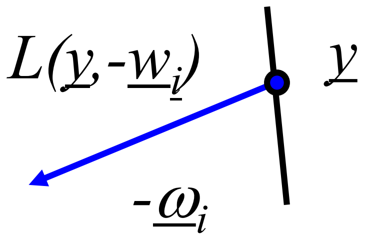
Rendering Equation II

- Outgoing illumination at a point

$$L(\underline{x},\underline{\omega}_o) = L_e(\underline{x},\underline{\omega}_o) + L_r(\underline{x},\underline{\omega}_o)$$
$$= L_e(\underline{x},\underline{\omega}_o) + \int_{\Omega_+} f_r(\underline{\omega}_i,\underline{x},\underline{\omega}_o) L(\underline{x},\underline{\omega}_i)\cos\theta_i \, d\underline{\omega}_i$$

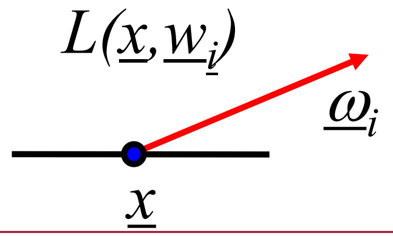
- Linking with other surface points
  - Incoming radiance at x is outgoing radiance at y

$$L_i(\underline{x},\underline{\omega}_i) = L(\underline{y},-\underline{\omega}_i) = L(RT(\underline{x},\underline{\omega}_i),-\underline{\omega}_i)$$



  - Ray-Tracing operator

$$\underline{y} = RT(\underline{x},\underline{\omega}_i)$$



Monte Carlo Integration

- Estimating an integral of $f(x)$ where x is distributed according to $X \sim p(x)$

$$E(f(x)) = \int_{x \in S} f(x)p(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

- or substituting

$$g(x) = f(x)p(x) \qquad \int_{x \in S} g(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} \frac{g(x_i)}{p(x)} \qquad \begin{matrix} p(x) > 0 \\ g(x) \neq 0 \end{matrix}$$

- for uniformly distributed samples $p(x) = 1/S$

$$\int_{x \in S} g(x)dx \approx \frac{S}{N}\sum_{i=1}^{N} g(x_i)$$

Approximation by Sampling

- Monte-Carlo Approximation

$$L(\underline{x},\underline{\omega}_o) = L_e(\underline{x},\underline{\omega}_o) + L_r(\underline{x},\underline{\omega}_o)$$
$$= L_e(\underline{x},\underline{\omega}_o) + \int_{\Omega_+} f_r(\underline{\omega}_i,\underline{x},\underline{\omega}_o) L_i(\underline{x},\underline{\omega}_i)\cos\theta_i \, d\underline{\omega}_i$$

- for uniformly distributed samples $\underline{\omega}_i$ on the hemisphere (lecture „Rendering“):

$$L(\underline{x},\underline{\omega}_o) \approx L_e(\underline{x},\underline{\omega}_o) + \frac{2\pi}{N}\sum_{i=1}^{N} \left(f_r(\underline{\omega}_i,\underline{x},\underline{\omega}_o) L_i(\underline{x},\underline{\omega}_i)\cos\theta_i \, d\underline{\omega} \right)$$

- Evaluate $L_i(\underline{x},\underline{\omega}_i)$ recursively/iteratively

Path Tracing

$$"N = 1"$$

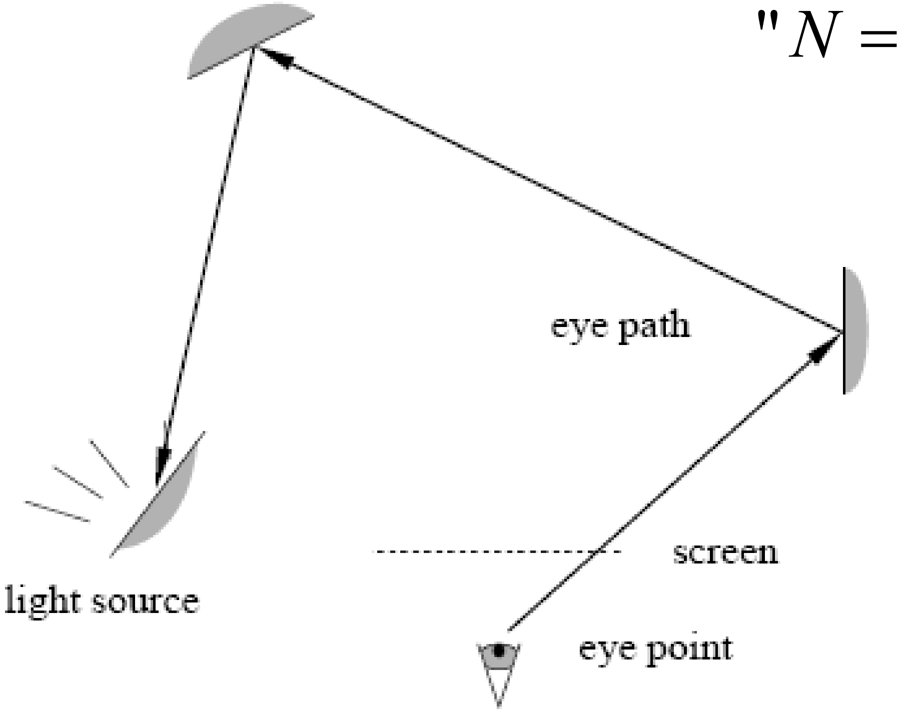


Figure 4.2: Schematic overview of the path tracing algorithm. The radiant flux through a pixel has to be estimated. The tracing of a primary ray from the virtual eye point through a pixel corresponds to sampling the expression for the flux. The subsequent random walk through the scene corresponds to recursively estimating the radiance values. Each time a light source is hit a contribution is added to the estimate.

PT Pseudocode – pixel sampling

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

```
computeImage()
    foreach pixel (i,j)
        estimatedRadiance[i,j] = 0
        for s = 1 to #viewSamples
            generate Q in pixel (i,j)
            theta = (Q - E)/|Q-E|
            x = trace(E,theta)
            estimatedRadiance [i,j] +=
                    computeRadiance(x,-theta)
        estimatedRadiance [i,j] /= #viewSamples
```

[Dutré, Bekaert, Bala]

PT Pseudocode – radiance estimation

```
computeRadiance(x,theta)
    estimatedRadiance = Le(x,theta)
    estimatedRadiance += directIllumination(x, theta)
    estimatedRadiance += indirectIllumination(x, theta)
    return estimatedRadiance
```

[Dutré, Bekaert, Bala]

PT Pseudocode - direct illumination

```
directIllumination(x,theta)
    estimatedRadiance = 0
    for s = 1 to #shadowRays
        k = pick random light
        y = generate random point on light k
        psi = (x-y) /  |x-y|
        estimatedRadiance += Le_k(y,-psi) *
                             BRDF(x,psi,tetha) *
                             G(x,y) * V(x,y) /
                             (p(k)*p(y|k))
    estimateRadiance /= #shadowRays
    return estimatedRadiance
```

[Dutré, Bekaert, Bala]

PT Pseudocode - indirect illumination

```
indirectIllumination(x,theta)
    estimatedRadiance = 0
    if(not absorbed) // russian roulette
        for s = 1 to #indirectSamples
            psi = generate random dir on hemisphere
            y = trace(x, psi)
            estimatedRadiance +=
                    computeRadiance(y,-psi) *
                    BRDF(x,psi,theta) *
                    cos(Nx,psi) / pdf(psi)
         estimatedRadiance /= #indirectSamples
    return estimatedRadiance /(1-absorption)
```

[Dutré, Bekaert, Bala]

Path Tracing

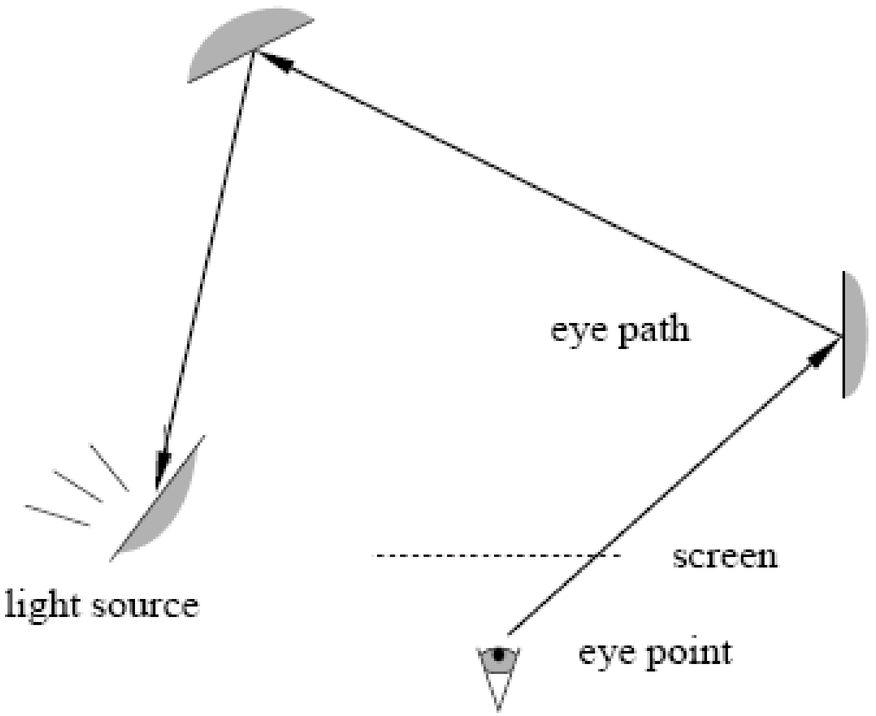


Figure 4.2: Schematic overview of the path tracing algorithm. The radiant flux through a pixel has to be estimated. The tracing of a primary ray from the virtual eye point through a pixel corresponds to sampling the expression for the flux. The subsequent random walk through the scene corresponds to recursively estimating the radiance values. Each time a light source is hit a contribution is added to the estimate.

Path Tracing with Next Event Estimate

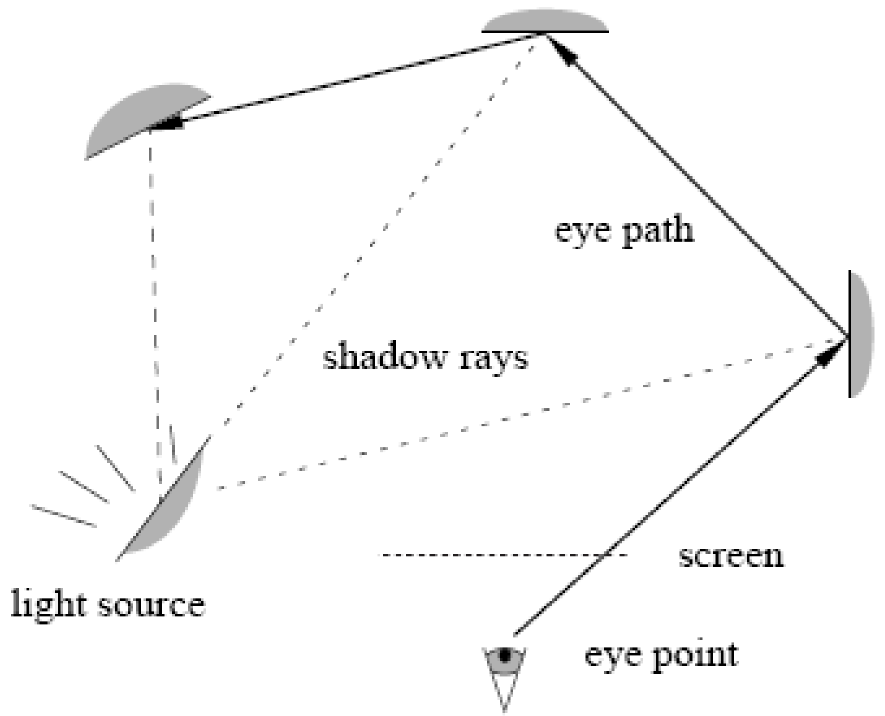


Figure 4.7: Schematic overview of the path tracing algorithm with next event estimation. Direct illumination is now computed explicitly at each point on the random walk by sampling the light sources.

Light Tracing

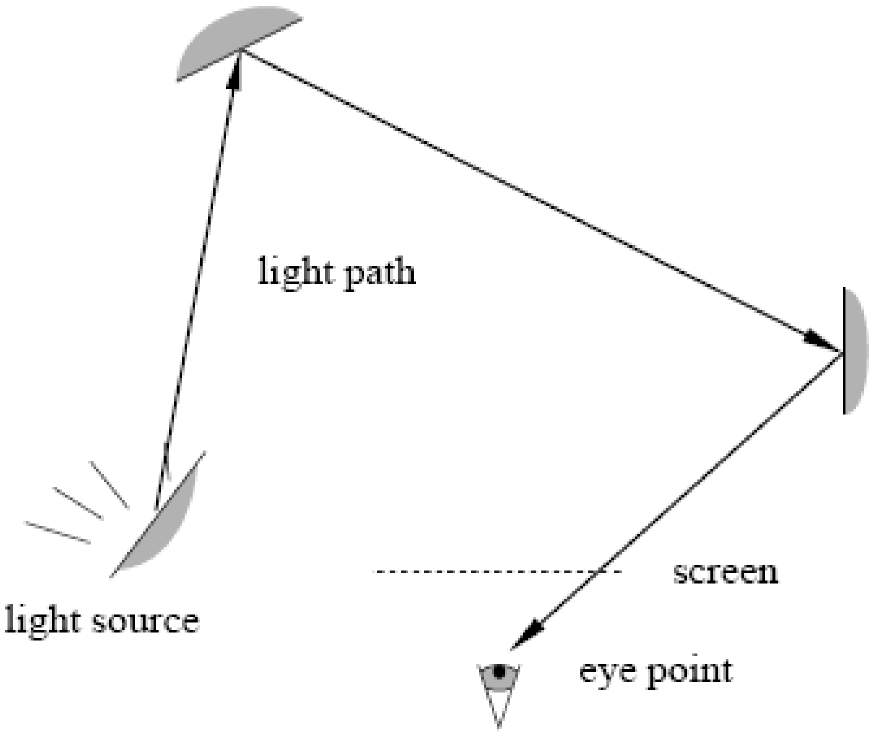


Figure 4.12: Schematic overview of the light tracing algorithm. Again the radiant flux through each pixel has to be estimated. The tracing of a primary ray from a light source corresponds to sampling the expression for the flux in terms of potential. The subsequent random walk through the scene corresponds to recursively estimating the potential values. Each time a ray passes through a pixel a contribution is added to the estimate.

Light Tracing with next event estimate

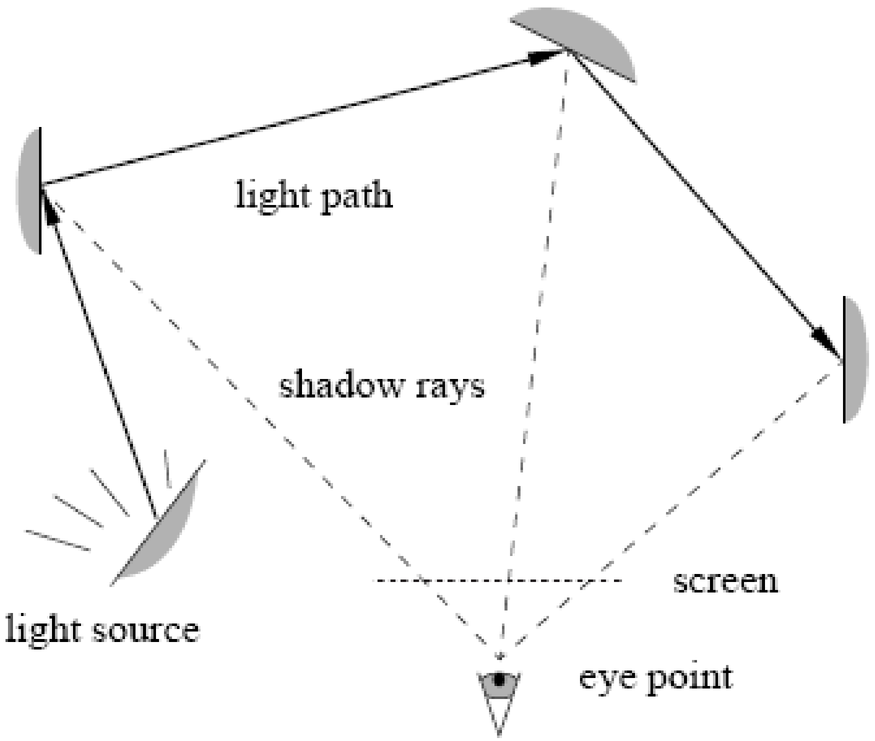
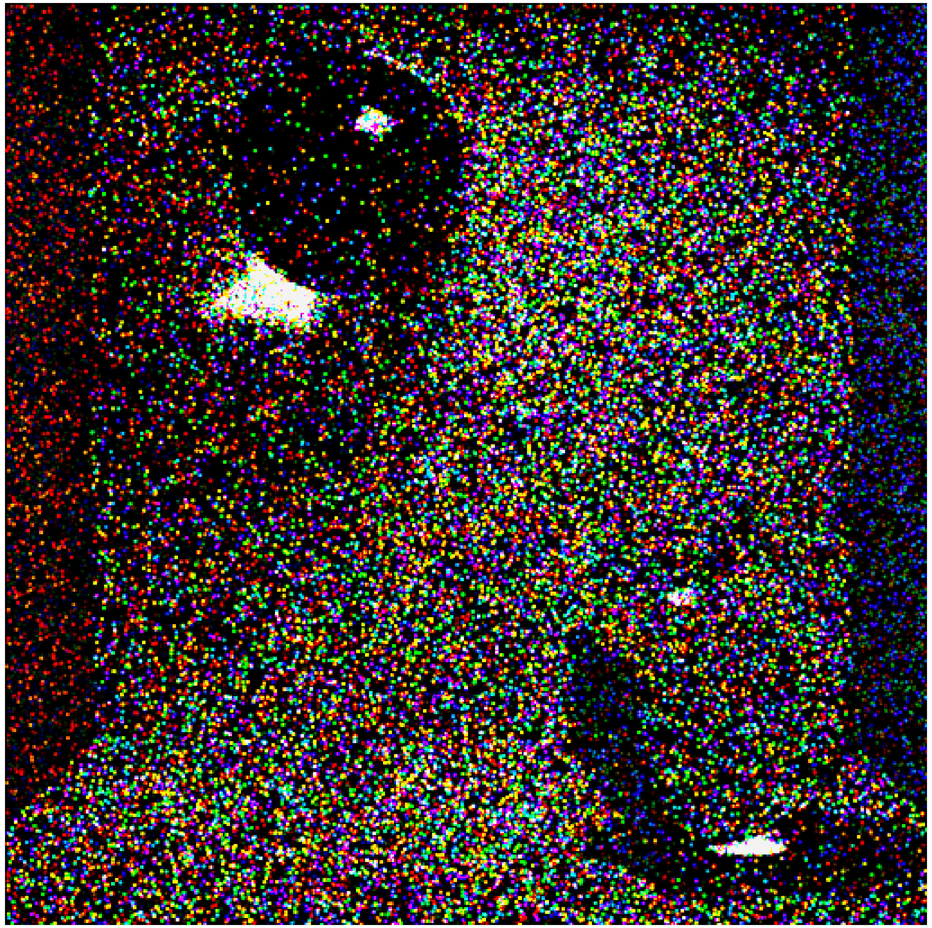


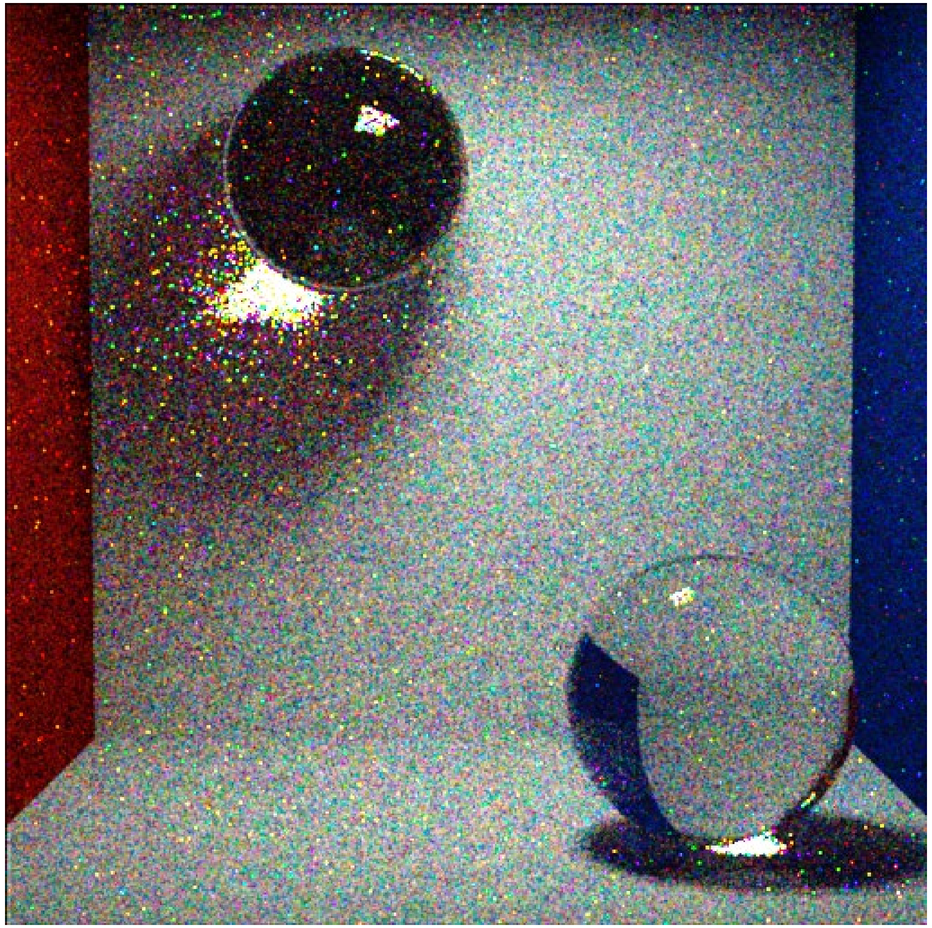
Figure 4.14: Schematic overview of the light tracing algorithm with next event estimation. Direct contributions of potential are now computed at each point on the random walk by sampling the relevant pixels, if any.

Path Tracing vs. Light Tracing

- Dual methods
- Performance depends on
  - Image size vs. scene size
  - Nature of light transport paths
  - Number of sources

Examples





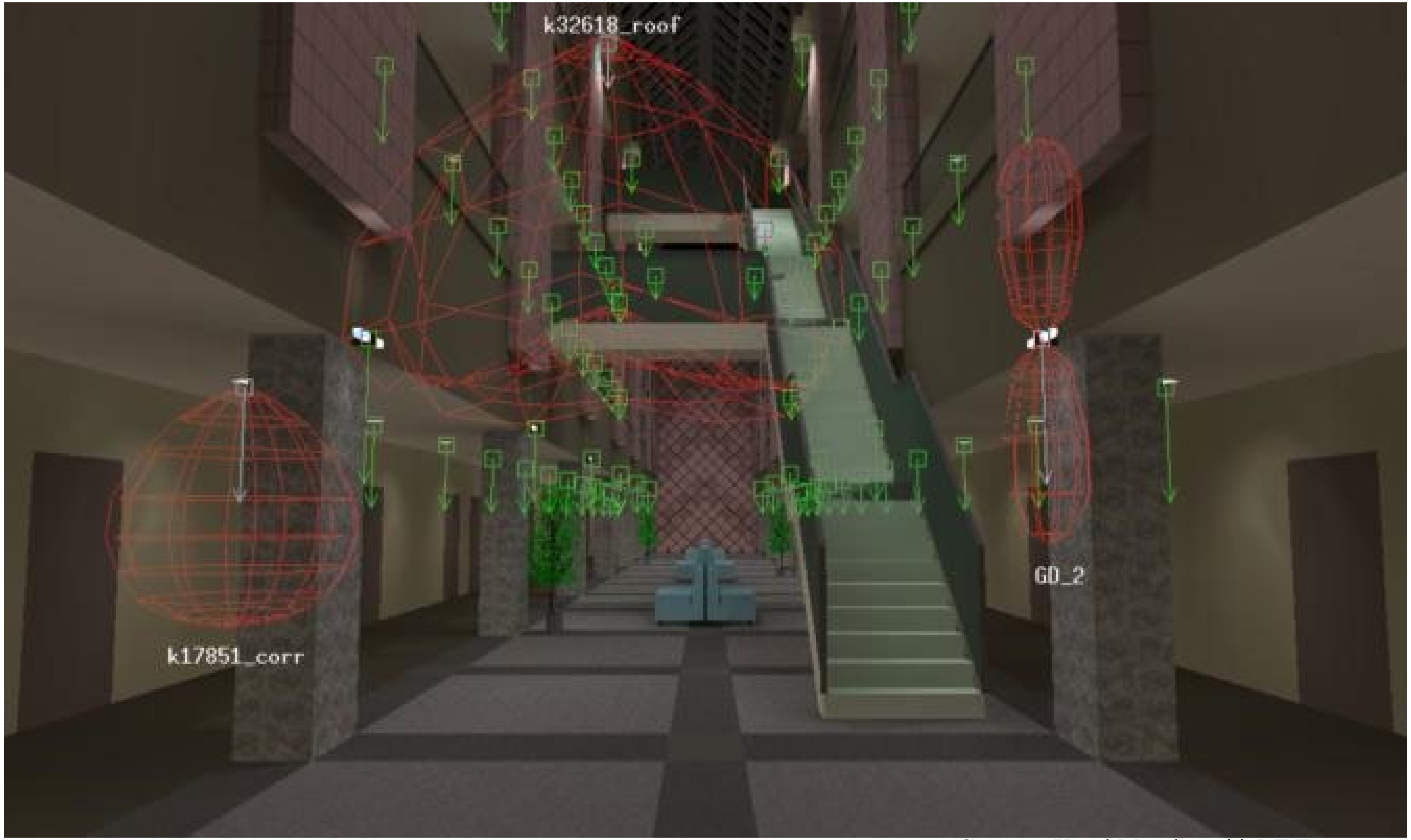
path tracing

path tracing + direct light

Lighting Simulation

Courtesy Karol Myszkowski, MPII

Lighting Simulation



Courtesy Karol Myszkowski, MPII

Lighting Simulation

Courtesy Karol Myszkowski, MPII

Lighting Simulation

Courtesy Karol Myszkowski, MPII

Wrap-up

- Rendering Equation
  - Integral equation
  - Balance of radiance
- Radiosity
  - Diffuse reflectance function
  - Radiative equilibrium between emission and absorption, escape
  - System of linear equations
  - Iterative solution
- Path Tracing
  - Monte Carlo Approximation of the rendering equation