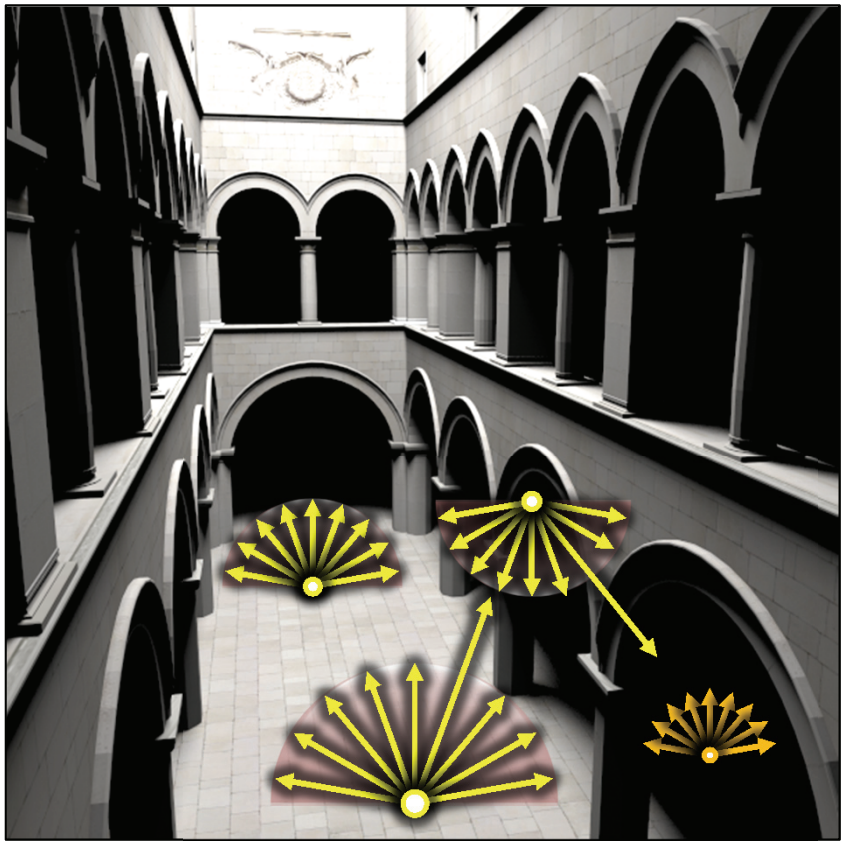
Computergrafik

Vorlesung im Wintersemester 2014/15
Kapitel 2: Ray Tracing
(Teil 3)

Prof. Dr.-Ing. Carsten Dachsbacher
Lehrstuhl für Computergrafik
Karlsruher Institut für Technologie

Lichttransport, Beleuchtungssimulation

$$L(\boldsymbol{x}, \boldsymbol{\omega}) = L_e(\boldsymbol{x}, \boldsymbol{\omega}) + \int_{\Omega^+} f_r(\boldsymbol{\omega_i}, \boldsymbol{x}, \boldsymbol{\omega}) L_i(\boldsymbol{x}, \boldsymbol{\omega_i}) \cos\theta_i \, d\boldsymbol{\omega_i}$$

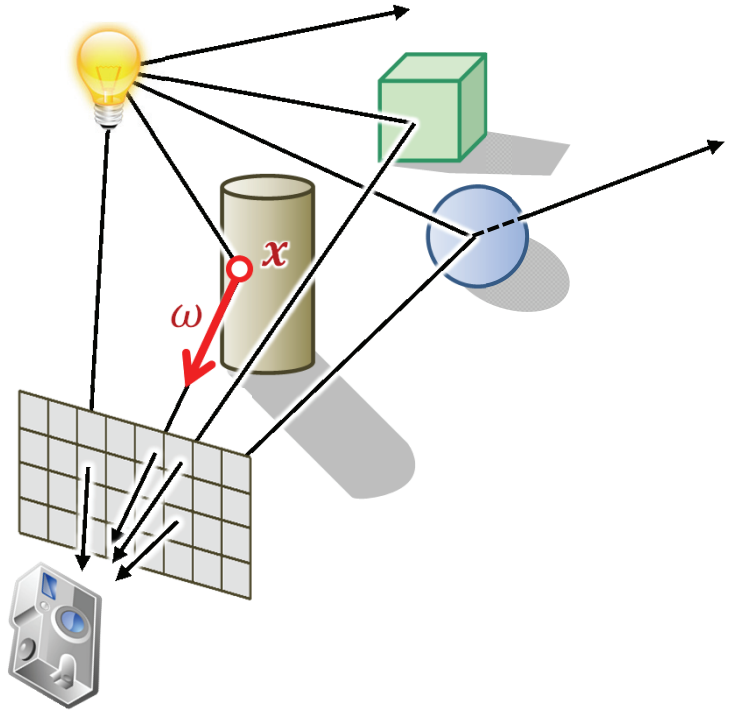


Exkurs: Rendering Gleichung

Lichtverteilung in einer Szene [Kajiya86]

$$L(\boldsymbol{x}, \boldsymbol{\omega}) = L_e(\boldsymbol{x}, \boldsymbol{\omega}) + \int_{\Omega^+} f_r(\boldsymbol{\omega_i}, \boldsymbol{x}, \boldsymbol{\omega}) L_i(\boldsymbol{x}, \boldsymbol{\omega_i}) \cos\theta_i \, d\boldsymbol{\omega_i}$$

- L Strahldichte (engl. Radiance) [W/m²sr]
- E Emissionsterm [W/m²sr]
- Abhängigkeit von der Wellenlänge

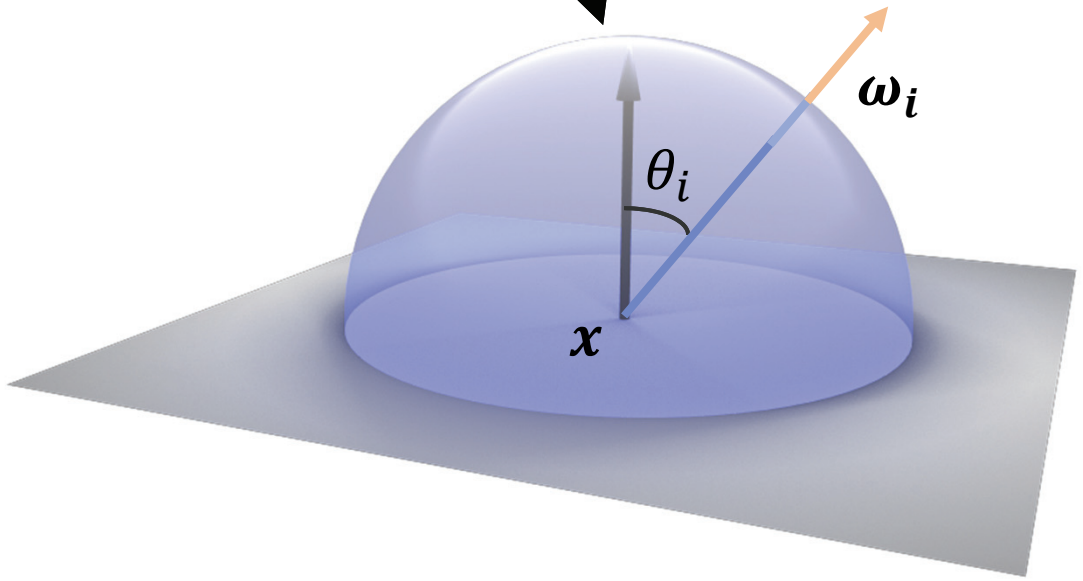


Exkurs: Rendering Gleichung

Einfallsrichtung

$$L(\boldsymbol{x}, \boldsymbol{\omega}) = L_e(\boldsymbol{x}, \boldsymbol{\omega}) + \int_{\Omega^+} f_r(\boldsymbol{\omega_i}, \boldsymbol{x}, \boldsymbol{\omega}) L_i(\boldsymbol{x}, \boldsymbol{\omega_i}) \cos\theta_i \, d\boldsymbol{\omega_i}$$

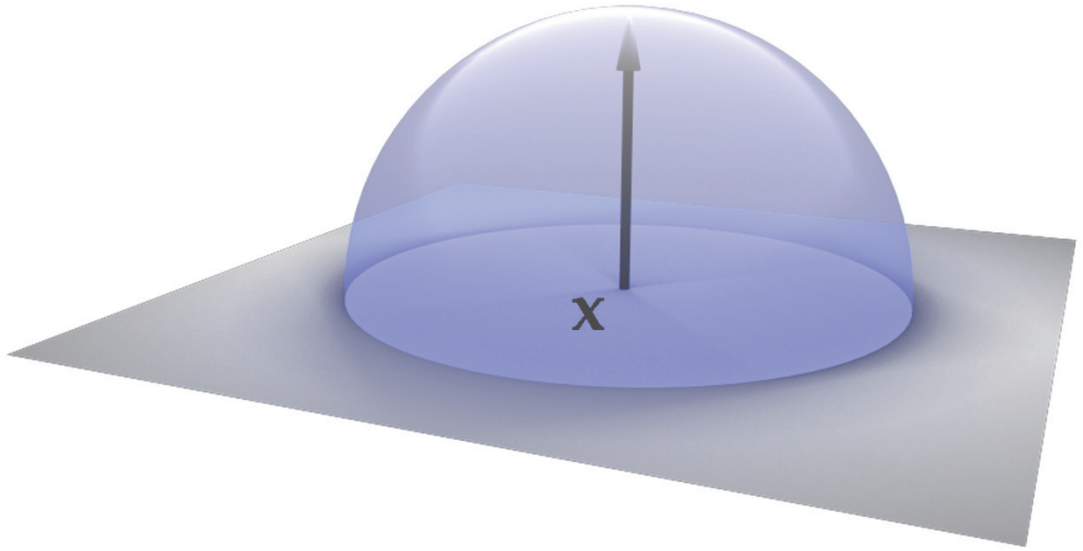
Integration über die positive Hemisphäre:
alle Richtungen aus denen Licht einfällt



Exkurs: Rendering Gleichung

$$L(\boldsymbol{x}, \boldsymbol{\omega}) = L_e(\boldsymbol{x}, \boldsymbol{\omega}) + \int_{\Omega^+} f_r(\boldsymbol{\omega_i}, \boldsymbol{x}, \boldsymbol{\omega}) L_i(\boldsymbol{x}, \boldsymbol{\omega_i}) \cos\theta_i \, d\boldsymbol{\omega_i}$$

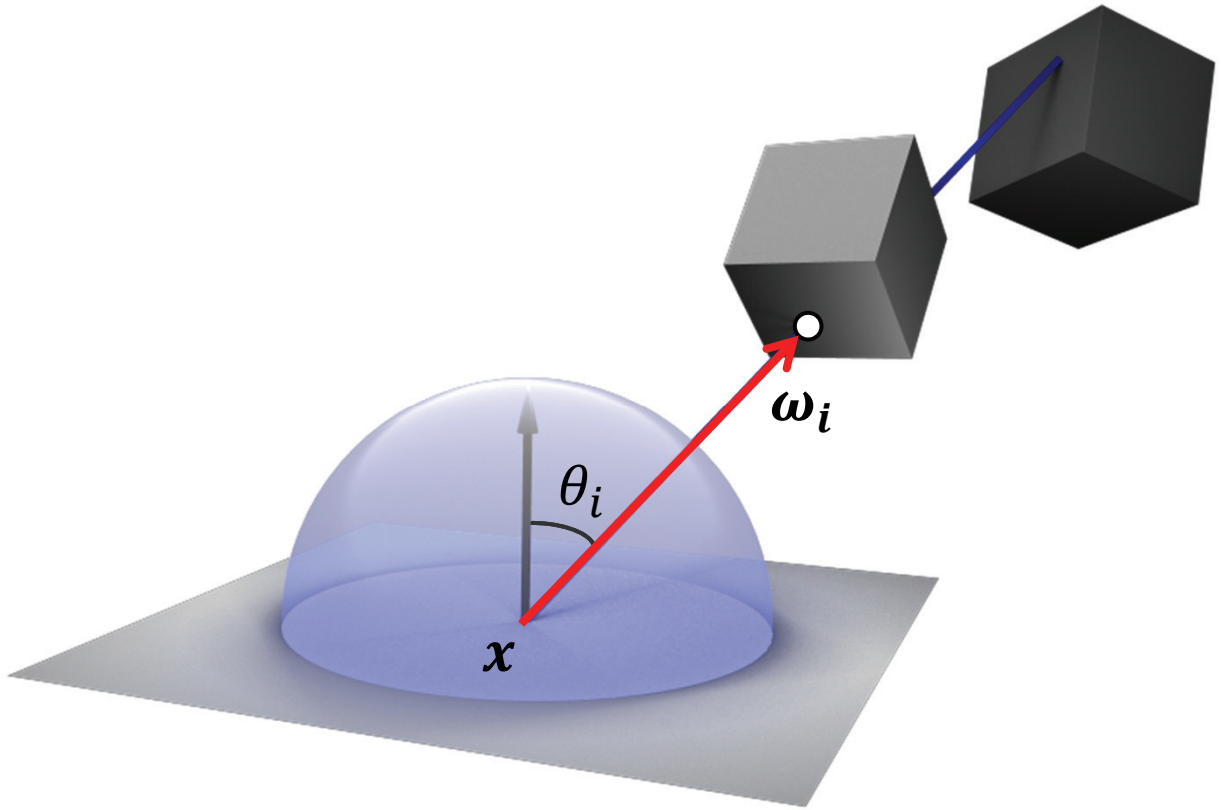
Reflektanzverteilungsfunktion:
Wie viel Licht aus Richtung $\boldsymbol{\omega_i}$
wird in Richtung $\boldsymbol{\omega}$ reflektiert



Exkurs: Rendering Gleichung

$$L(\boldsymbol{x}, \boldsymbol{\omega}) = L_e(\boldsymbol{x}, \boldsymbol{\omega}) + \int_{\Omega^+} f_r(\boldsymbol{\omega_i}, \boldsymbol{x}, \boldsymbol{\omega}) L_i(\boldsymbol{x}, \boldsymbol{\omega_i}) \cos\theta_i \, d\boldsymbol{\omega_i}$$

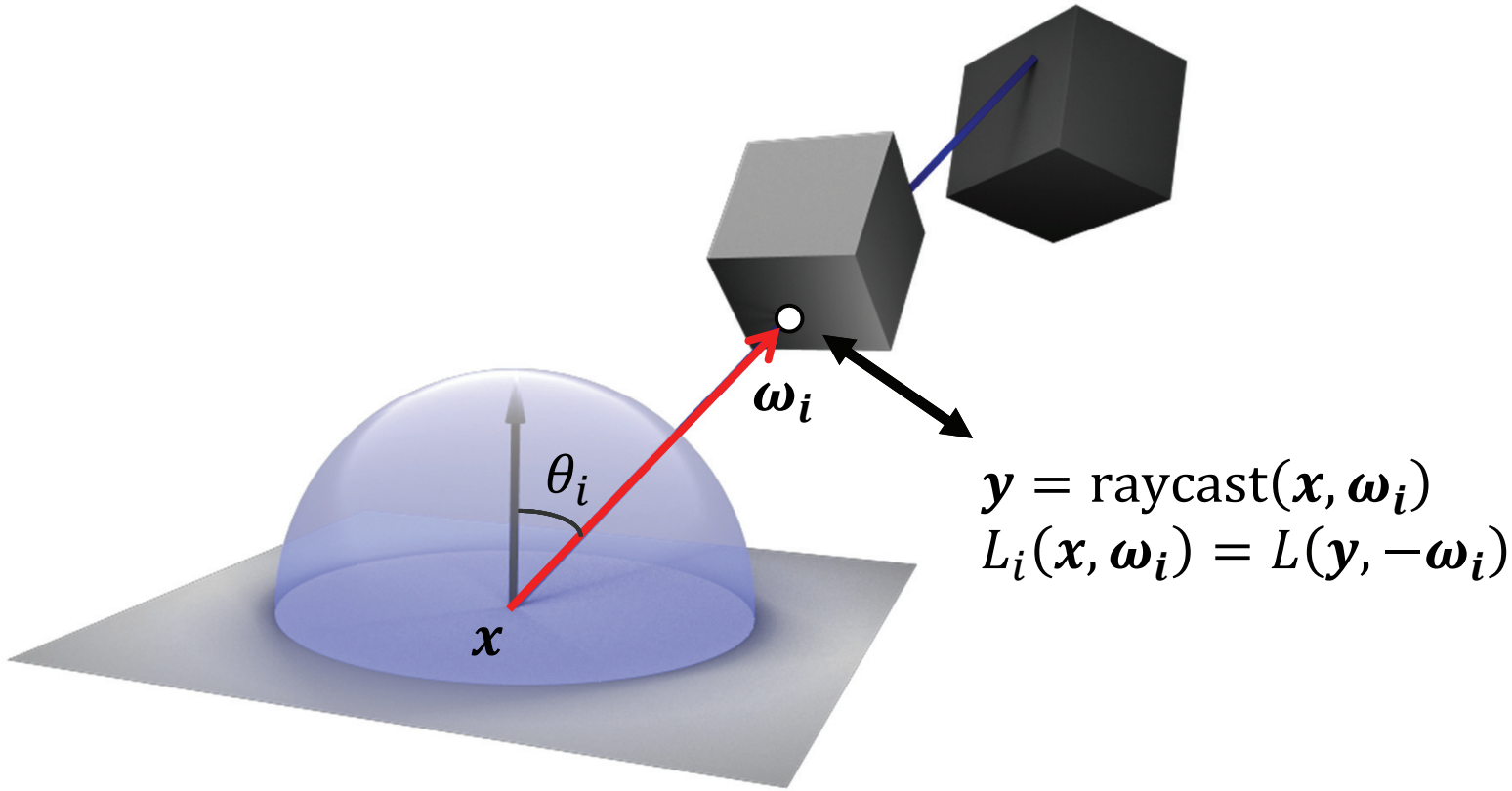
einfallendes Licht



Exkurs: Rendering Gleichung

$$L(\boldsymbol{x}, \boldsymbol{\omega}) = L_e(\boldsymbol{x}, \boldsymbol{\omega}) + \int_{\Omega^+} f_r(\boldsymbol{\omega_i}, \boldsymbol{x}, \boldsymbol{\omega}) L(\boldsymbol{y}, -\boldsymbol{\omega_i}) \cos\theta_i \, d\boldsymbol{\omega_i}$$

ausgehendes Licht



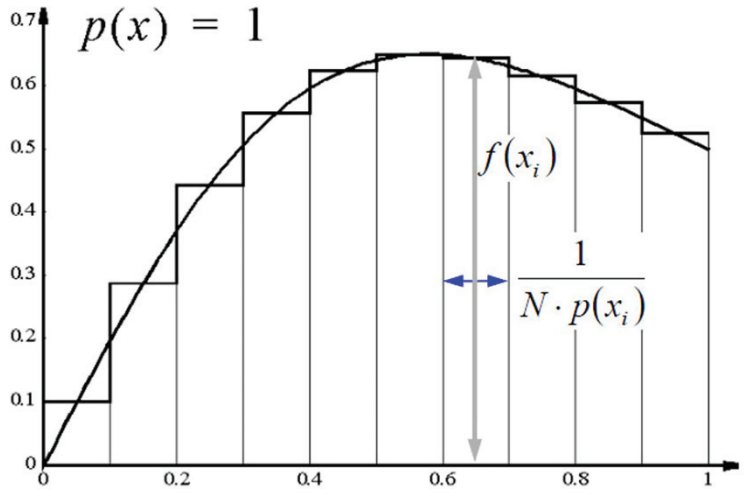
Exkurs: Monte Carlo Integration

- Grundidee der Monte Carlo Integration (vereinfacht!):
  wir können den Wert eines Integrals schätzen in dem wir es an N zufälligen uniform-verteilten Stellen $x_i \in [a; b]$ auswerten

$$\int_a^b f(x)dx \approx \frac{b-a}{N}\sum_{i=1}^{N} f(x_i)$$

  - die Schätzung wird besser, wenn N größer wird

- Vergleiche Riemann-Integral:

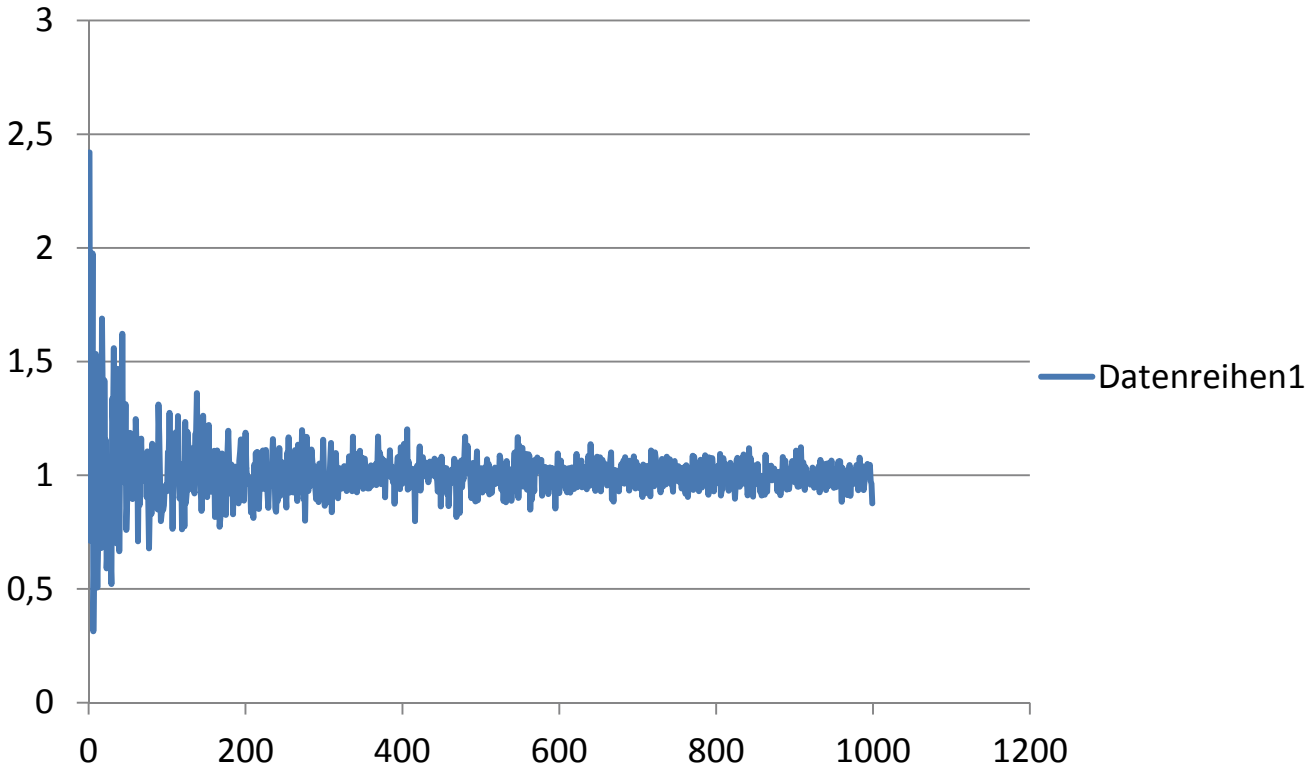


Exkurs: Monte Carlo Integration

Beispiel

- berechne $I = \int_0^1 5x^4 dx = 1$

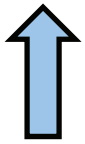
- naïve Monte Carlo Integration mit $N = 1 \ldots 1000$



Exkurs: Rendering Gleichung

- um den Wert des Integrals zu lösen verwendet Distributed Ray Tracing (und verwandte Verfahren) die Monte Carlo Integration
  - im einfachsten Fall: berechne des Wert des Integranden für (viele) zufällige Richtungen und mittle

$$L(\boldsymbol{x}, \boldsymbol{\omega}) = L_e(\boldsymbol{x}, \boldsymbol{\omega}) + \frac{2\pi}{N} \sum_{i=1}^{N} f_r(\boldsymbol{\omega_i}, \boldsymbol{x}, \boldsymbol{\omega}) L_i(\boldsymbol{x}, \boldsymbol{\omega_i}) \cos\theta_i$$

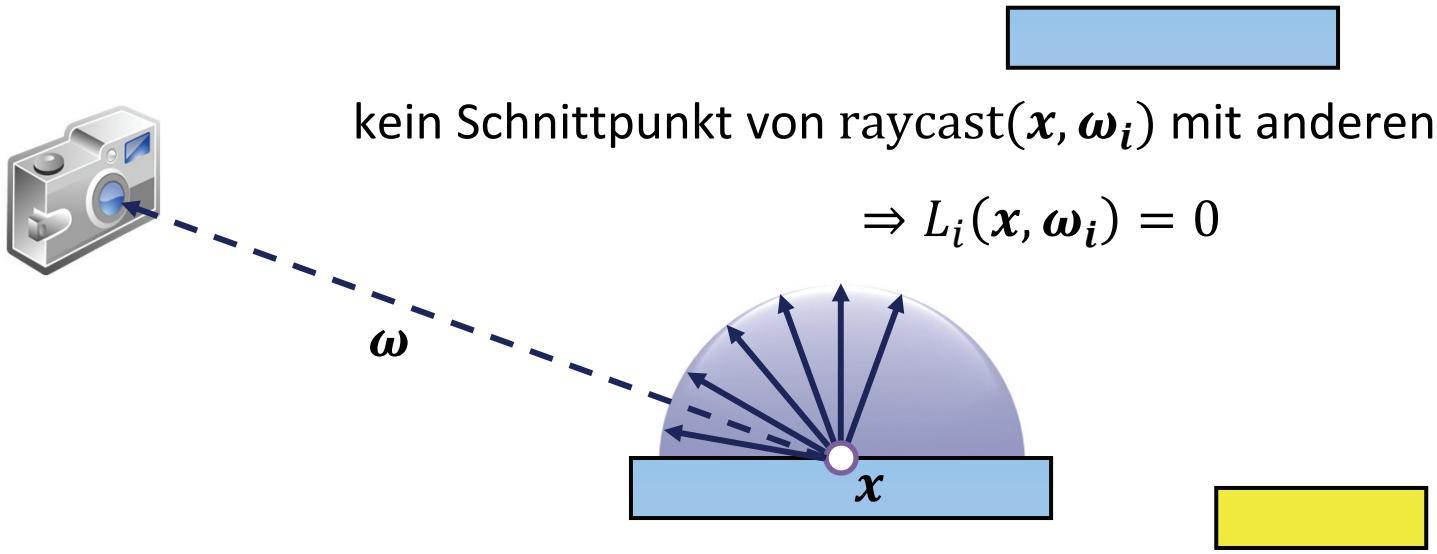


$$L(\boldsymbol{x}, \boldsymbol{\omega}) = L_e(\boldsymbol{x}, \boldsymbol{\omega}) + \int_{\Omega^+} f_r(\boldsymbol{\omega_i}, \boldsymbol{x}, \boldsymbol{\omega}) L_i(\boldsymbol{x}, \boldsymbol{\omega_i}) \cos\theta_i \, d\boldsymbol{\omega_i}$$

Exkurs: Monte Carlo Integration

- gut geeignet zur Lösung hochdimensionaler Integrale

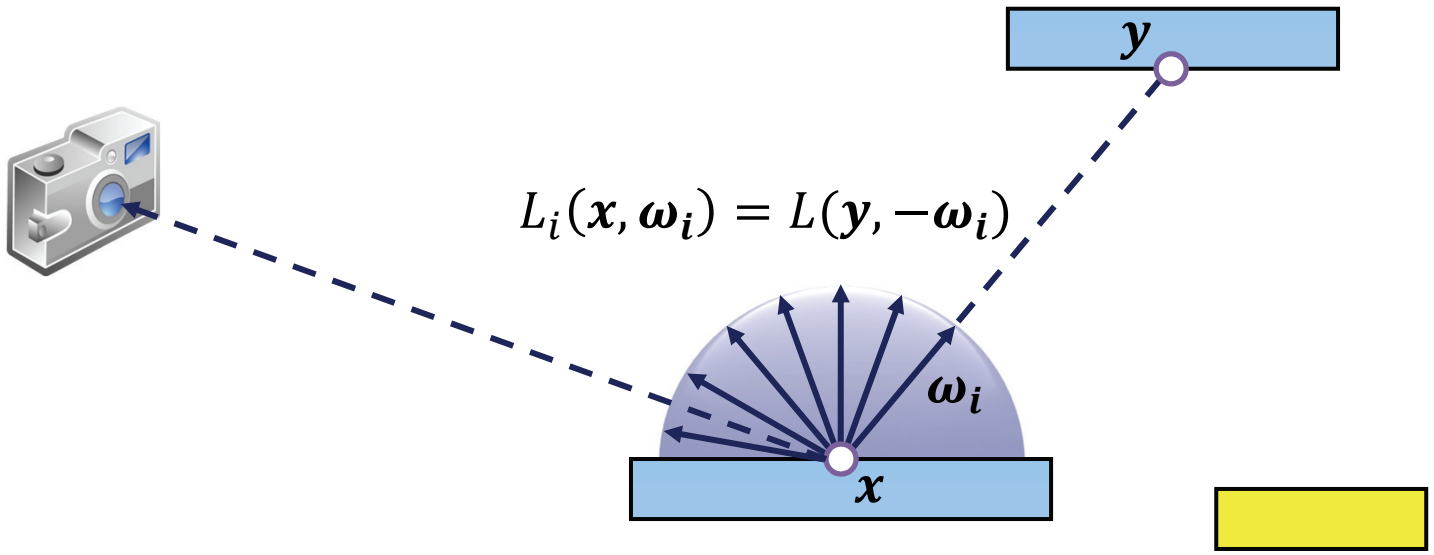
$$L(\boldsymbol{x}, \boldsymbol{\omega}) = L_e(\boldsymbol{x}, \boldsymbol{\omega}) + \frac{2\pi}{N} \sum_{i=1}^{N} f_r(\boldsymbol{\omega_i}, \boldsymbol{x}, \boldsymbol{\omega}) L_i(\boldsymbol{x}, \boldsymbol{\omega_i}) \cos\theta_i$$



Exkurs: Monte Carlo Integration

- gut geeignet zur Lösung hochdimensionaler Integrale

$$L(\boldsymbol{x}, \boldsymbol{\omega}) = L_e(\boldsymbol{x}, \boldsymbol{\omega}) + \frac{2\pi}{N} \sum_{i=1}^{N} f_r(\boldsymbol{\omega_i}, \boldsymbol{x}, \boldsymbol{\omega}) L_i(\boldsymbol{x}, \boldsymbol{\omega_i}) \cos\theta_i$$

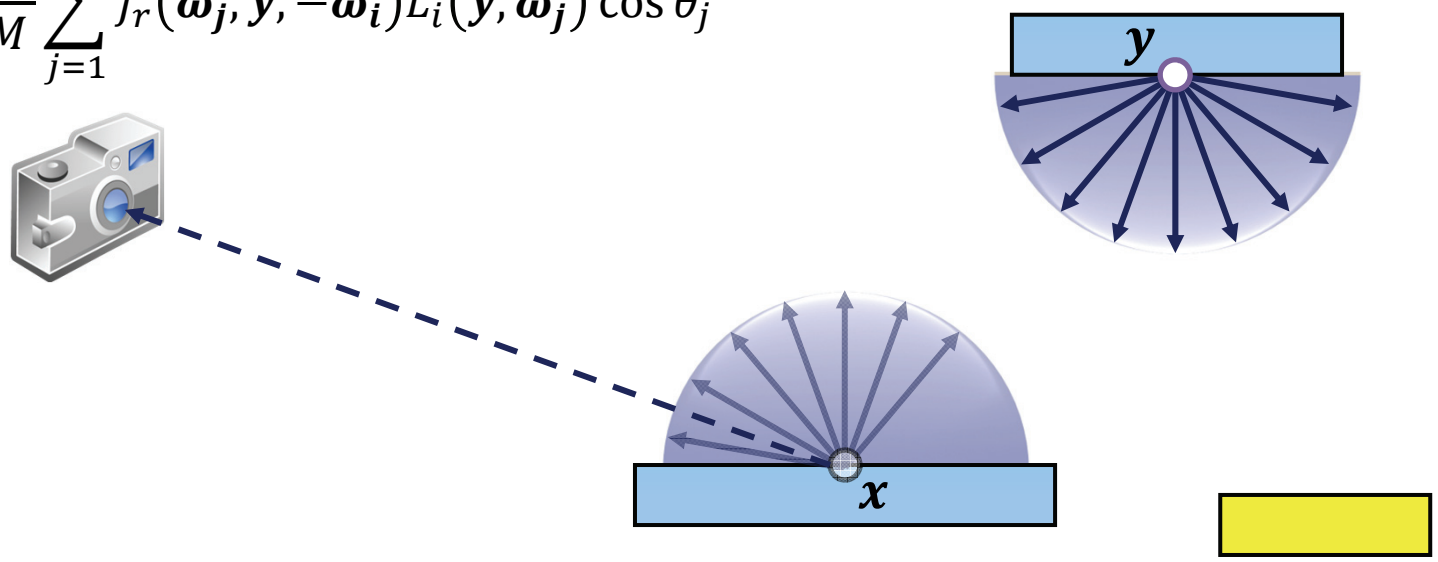


Exkurs: Monte Carlo Integration

- gut geeignet zur Lösung hochdimensionaler Integrale
- rekursive Approximation der Integrale

$$L(\boldsymbol{x}, \boldsymbol{\omega}) = L_e(\boldsymbol{x}, \boldsymbol{\omega}) + \frac{2\pi}{N} \sum_{i=1}^{N} f_r(\boldsymbol{\omega_i}, \boldsymbol{x}, \boldsymbol{\omega}) L_i(\boldsymbol{x}, \boldsymbol{\omega_i}) \cos\theta_i$$

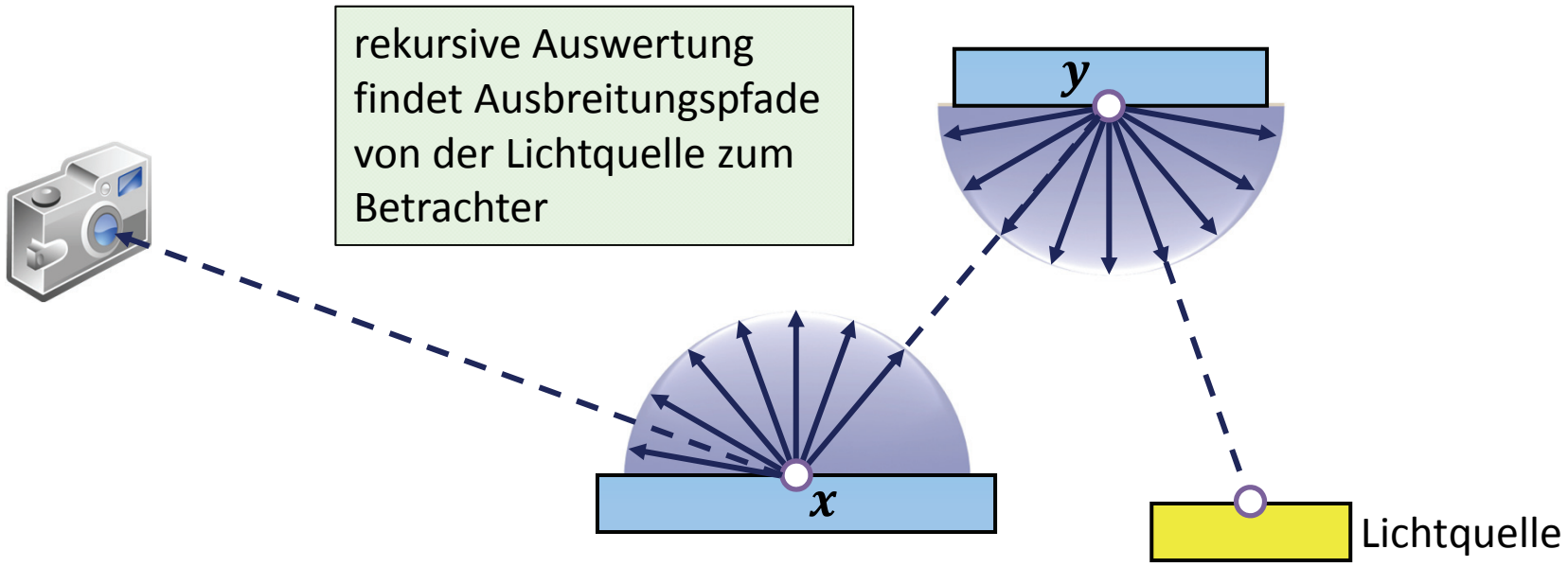
$$L(\boldsymbol{y}, -\boldsymbol{\omega_i}) = L_e(\boldsymbol{y}, -\boldsymbol{\omega_i}) +$$
$$+ \frac{2\pi}{M} \sum_{j=1}^{M} f_r(\boldsymbol{\omega_j}, \boldsymbol{y}, -\boldsymbol{\omega_i}) L_i(\boldsymbol{y}, \boldsymbol{\omega_j}) \cos\theta_j$$



Exkurs: Monte Carlo Integration

- gut geeignet zur Lösung hochdimensionaler Integrale
- rekursive Approximation der Integrale

$$L(\boldsymbol{x}, \boldsymbol{\omega}) = L_e(\boldsymbol{x}, \boldsymbol{\omega}) + \frac{2\pi}{N} \sum_{i=1}^{N} f_r(\boldsymbol{\omega_i}, \boldsymbol{x}, \boldsymbol{\omega}) L_i(\boldsymbol{x}, \boldsymbol{\omega_i}) \cos\theta_i$$



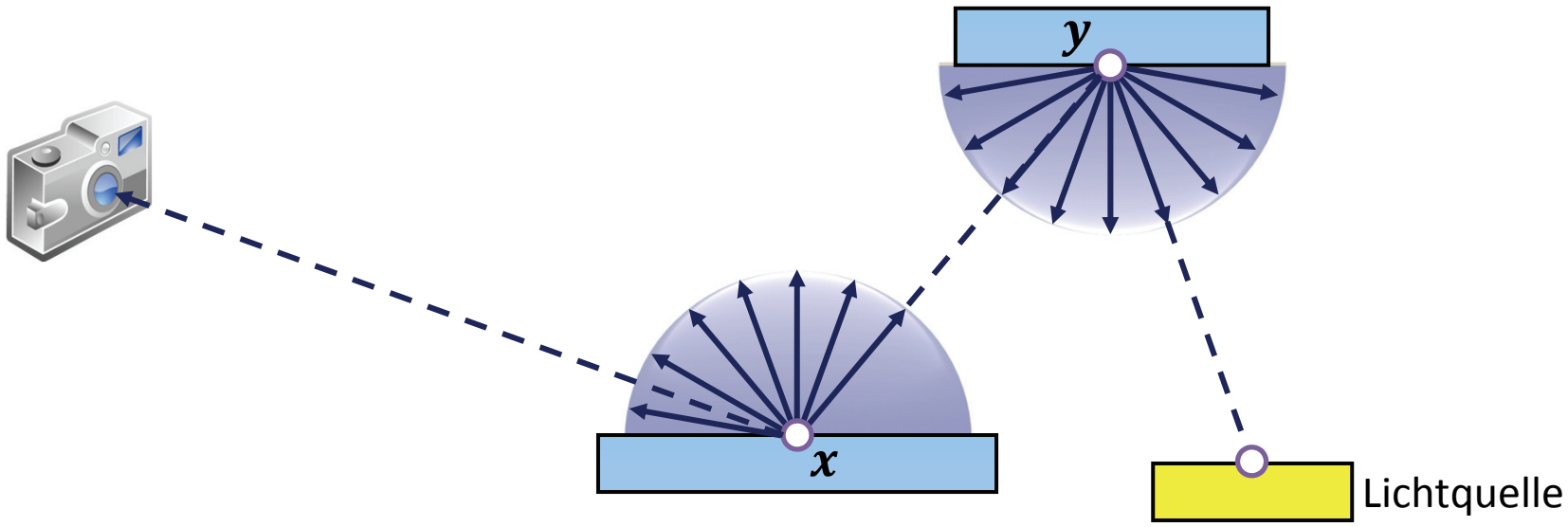
Exkurs: Monte Carlo Integration

Wie kann man Distributed Ray Tracing einfach umsetzen?

- Phong-Beleuchtungsmodell:

$$f_r(\boldsymbol{\omega_i}, \boldsymbol{x}, \boldsymbol{\omega}) = k_d + k_s \left(\left((2\mathbf{N} \cdot (\mathbf{N} \cdot \boldsymbol{\omega_i}) - \boldsymbol{\omega_i}) \cdot \boldsymbol{\omega} \right)^+ \right)^n / (\mathbf{N} \cdot \boldsymbol{\omega_i})^+$$

- Anmerkung: das Skalarprodukt $(\mathbf{N} \cdot \boldsymbol{\omega_i})^+$ ist schon Teil des Integranden, daher taucht es als Faktor bei k_d nicht auf (obwohl wir es dort im Phong-Beleuchtungsmodell kennen) und muß gleichermaßen für den spekularen Teil durch Division entfernt werden



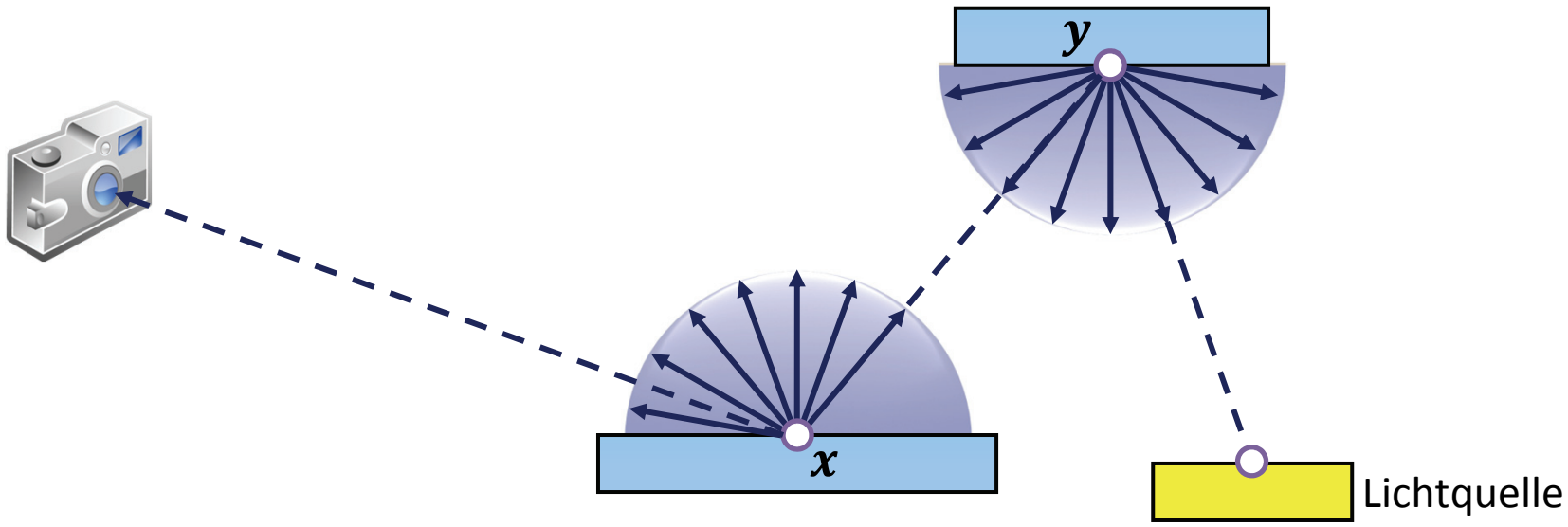
Exkurs: Monte Carlo Integration

Wie kann man Distributed Ray Tracing einfach umsetzen?

- Phong-Beleuchtungsmodell:

$$f_r(\boldsymbol{\omega_i}, \boldsymbol{x}, \boldsymbol{\omega}) = k_d + k_s \left(\left((2\mathbf{N} \cdot (\mathbf{N} \cdot \boldsymbol{\omega_i}) - \boldsymbol{\omega_i}) \cdot \boldsymbol{\omega} \right)^+ \right)^n / (\mathbf{N} \cdot \boldsymbol{\omega_i})^+$$

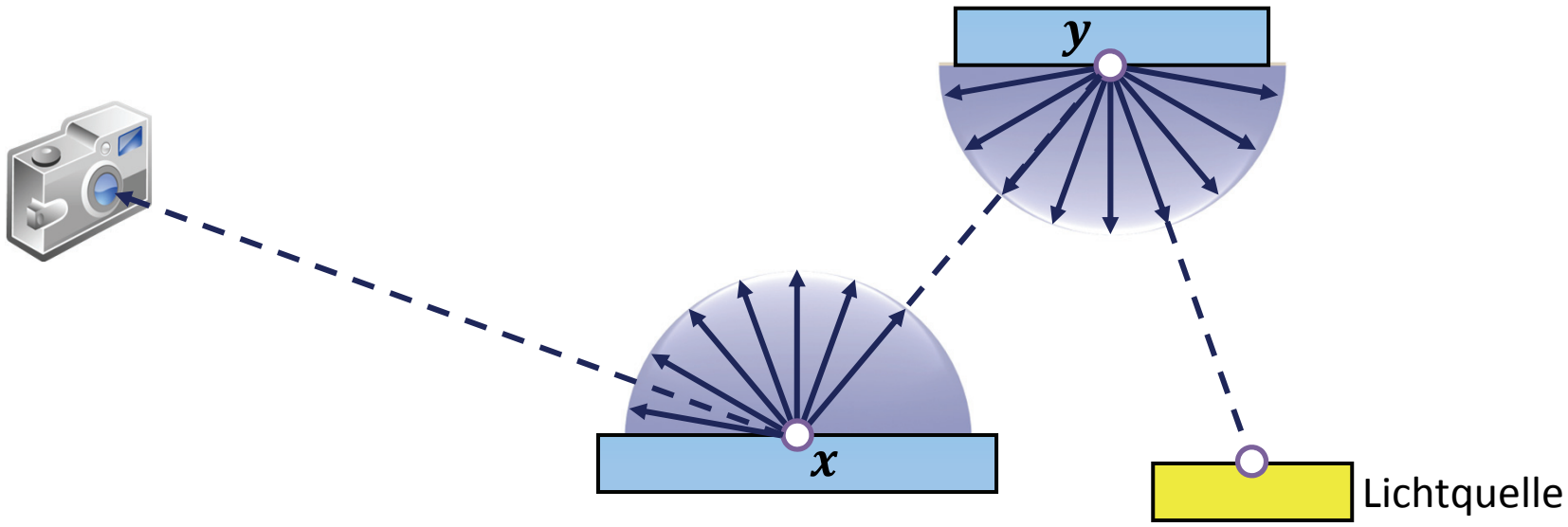
- Flächenlichtquellen
  - füge zur Materialdefinition einen Emissionsterm hinzu
  - $L_e(\boldsymbol{x}, \boldsymbol{\omega}) > 0$ für Lichtquellen (emittierende geometrische Primitive)
  - wenn Sie DR probieren, dann am besten zuerst mit **großen Flächenlichtquellen**



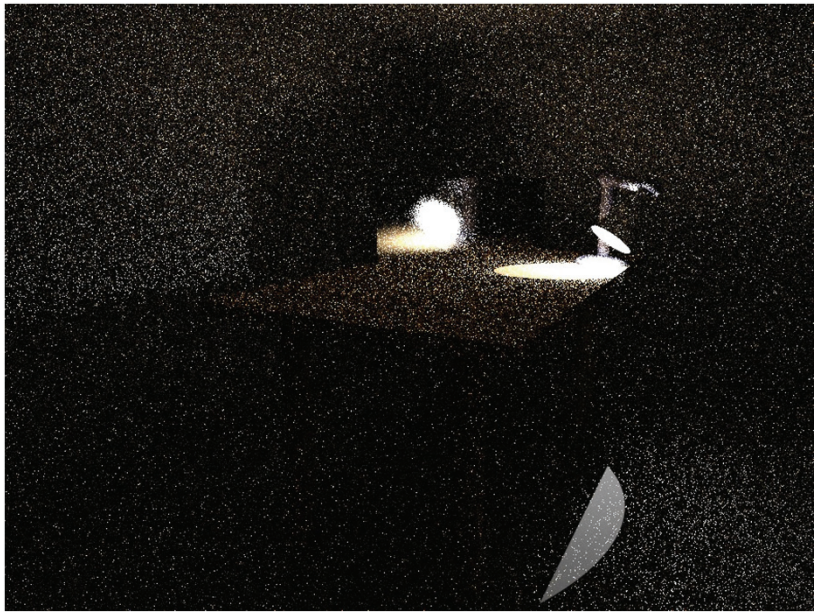
Exkurs: Monte Carlo Integration

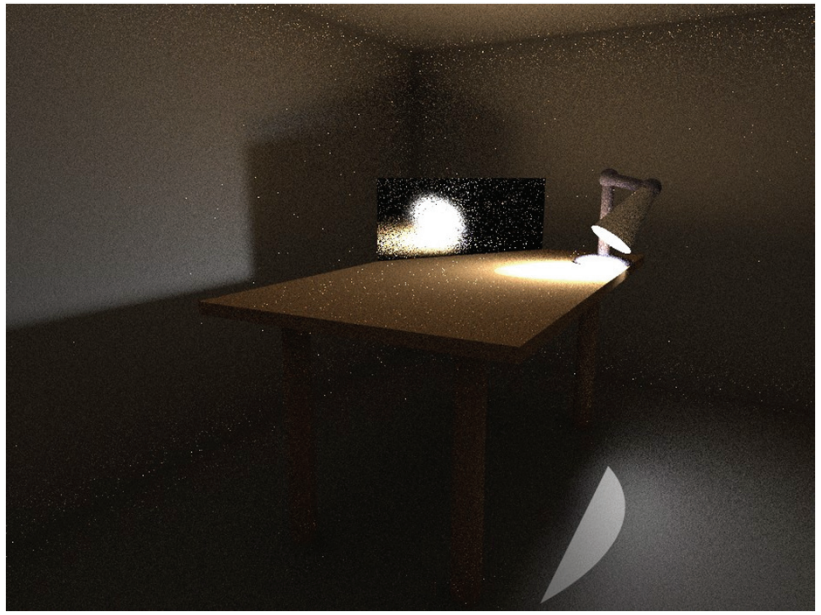
Wie kann man Distributed Ray Tracing einfach umsetzen?

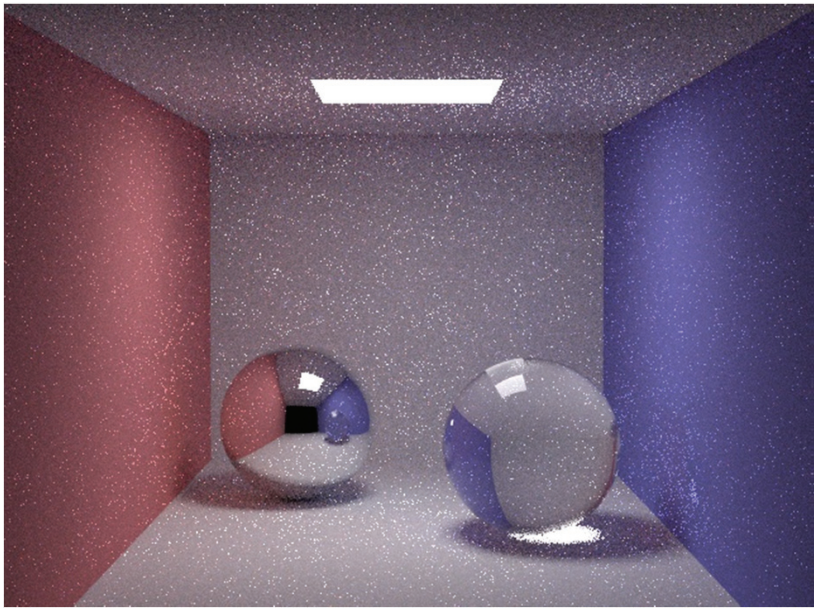
- zufällige Richtungen erzeugen (in Kugelkoordinaten)
  - erzeuge 2 Zufallszahlen $\xi_1, \xi_2 \in [0; 1)$
  - $\theta = \arccos(1 - 2\xi_1)$ und $\phi = 2\pi\xi_2$
  - $\boldsymbol{\omega_i} = (\sin\theta\cos\phi\,, \sin\theta\sin\phi\,, \cos\theta)$
  - verwerfe die Richtung und erzeuge neue, wenn $(\mathbf{N} \cdot \boldsymbol{\omega_i}) < 0$
- damit rekursiv berechnen, z.B. bis zu einer bestimmten Rekursionstiefe
  - z.B. $N = 64$ (für größere Rekursionstiefe: kleinere N wählen)

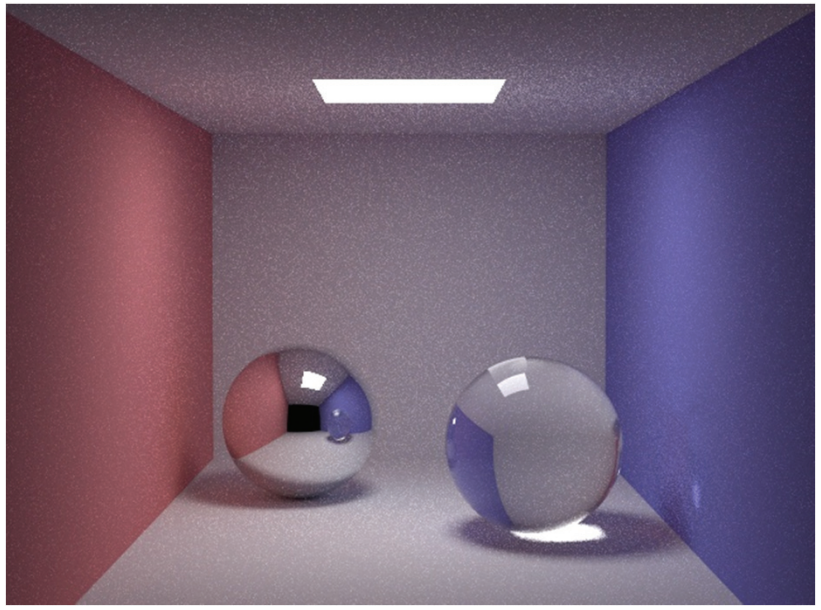


Exkurs: Distributed RT, Path Tracing



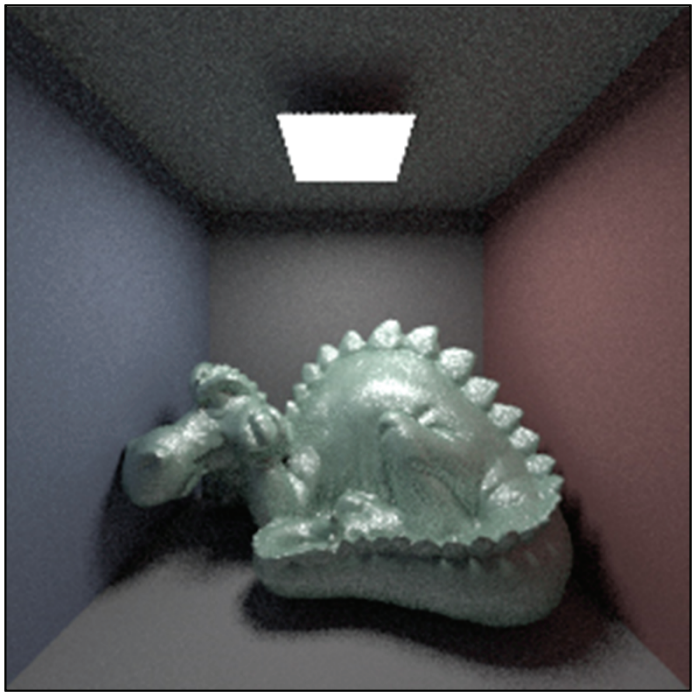






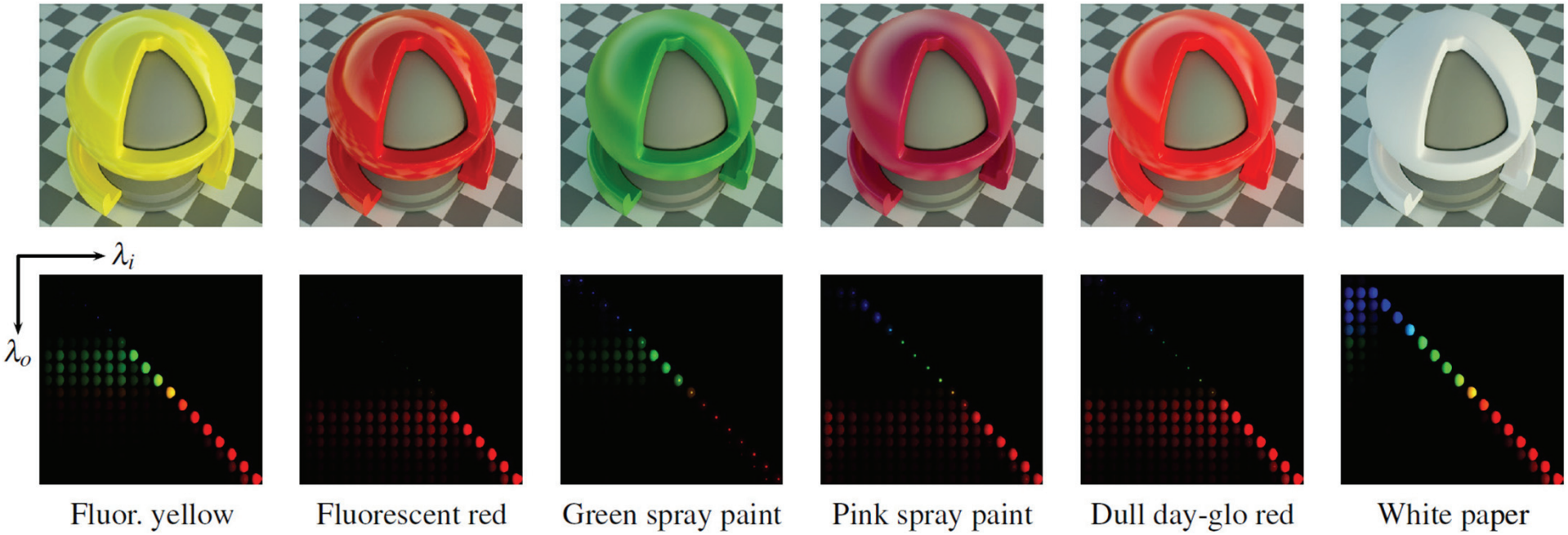
Path Tracing

- verwandt mit Distributed Ray Tracing: nur jeweils ein weiterer Strahl, dafür mehrere „Pfade“ pro Pixel
- und wie lange dauert es ein Bild mit diesem Dreiecksnetz zu berechnen?
  - 32× stochastisches Supersampling
  - 256^2 Pixel, 480000 Dreiecke
  - Brute Force: 176462sec ≈ 49h
    - insgesamt ca. $6.3 \cdot 10^6$ Primär-, Sekundär- und Schattenstrahlen
    - ca. $3 \cdot 10^{12}$ Schnitttests (nur Rekursionstiefe 1!)
- später in dieser Vorlesung: <80sec (BVH 18.3sec, Rendering 59sec) → 2200 mal so schnell!

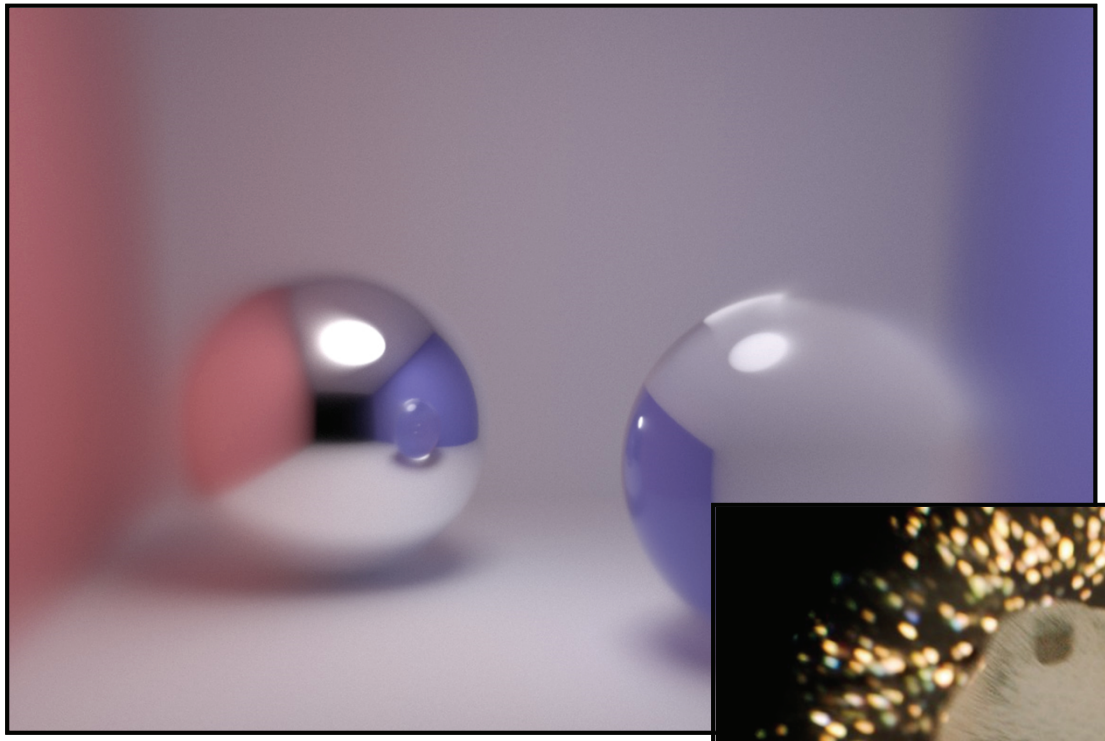


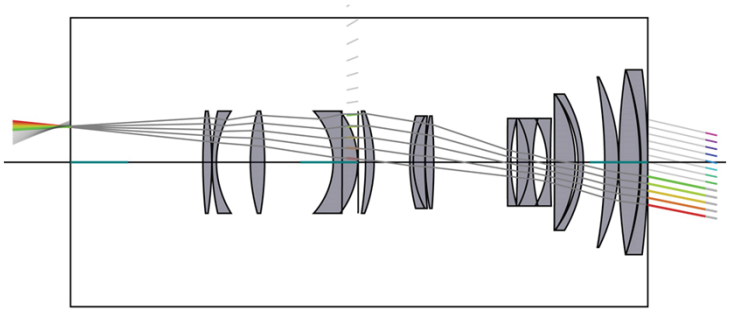
Forschung: Bispektrale BRDFs

- Acquisition and Analysis of Bispectral Bidirectional Reflectance and Reradiation Distribution Functions, M. Hullin, J. Hanika, B. Ajdin, J. Kautz, H.-P. Seidel, H. Lensch



Forschung: Rendering mit Linsensystemen





Forschung: Partizipierende Medien und Streuung

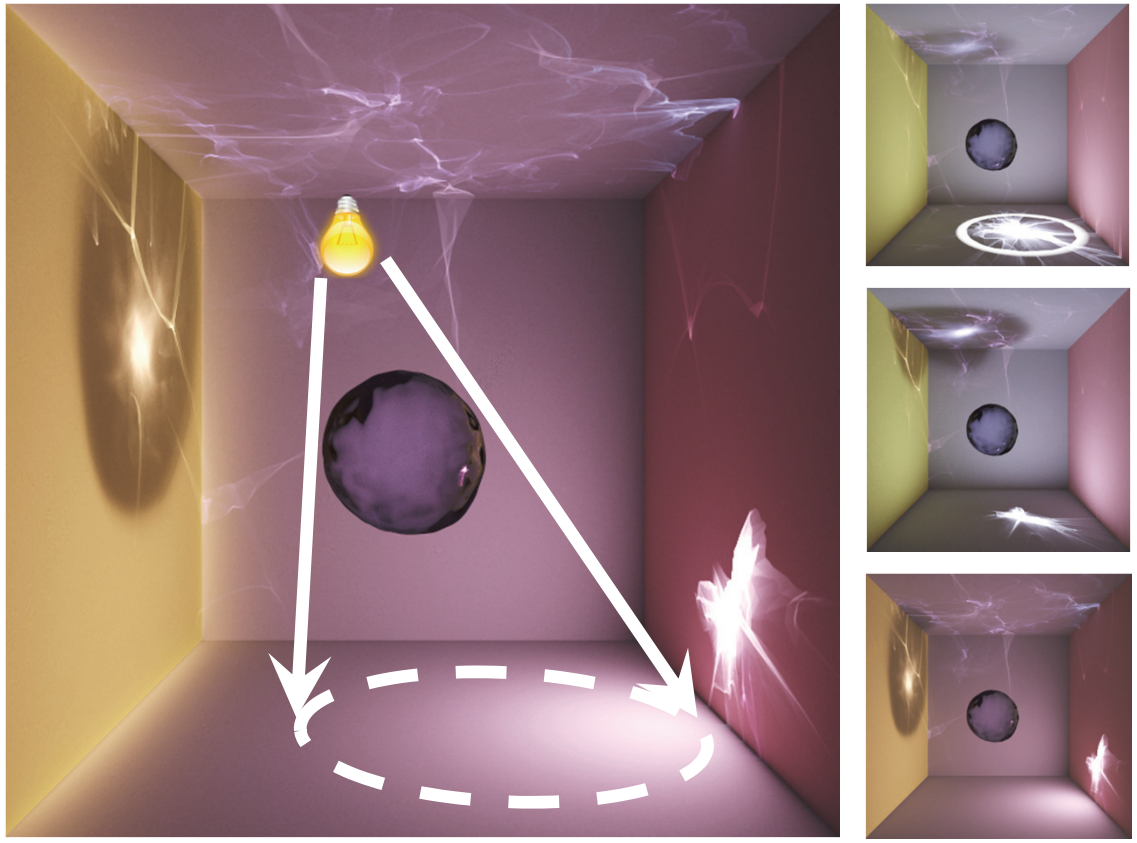
surface reflection

single scattering

multiple scattering

Spekulare Interaktionen

Light Transport Manipulation



Light Transport Manipulation