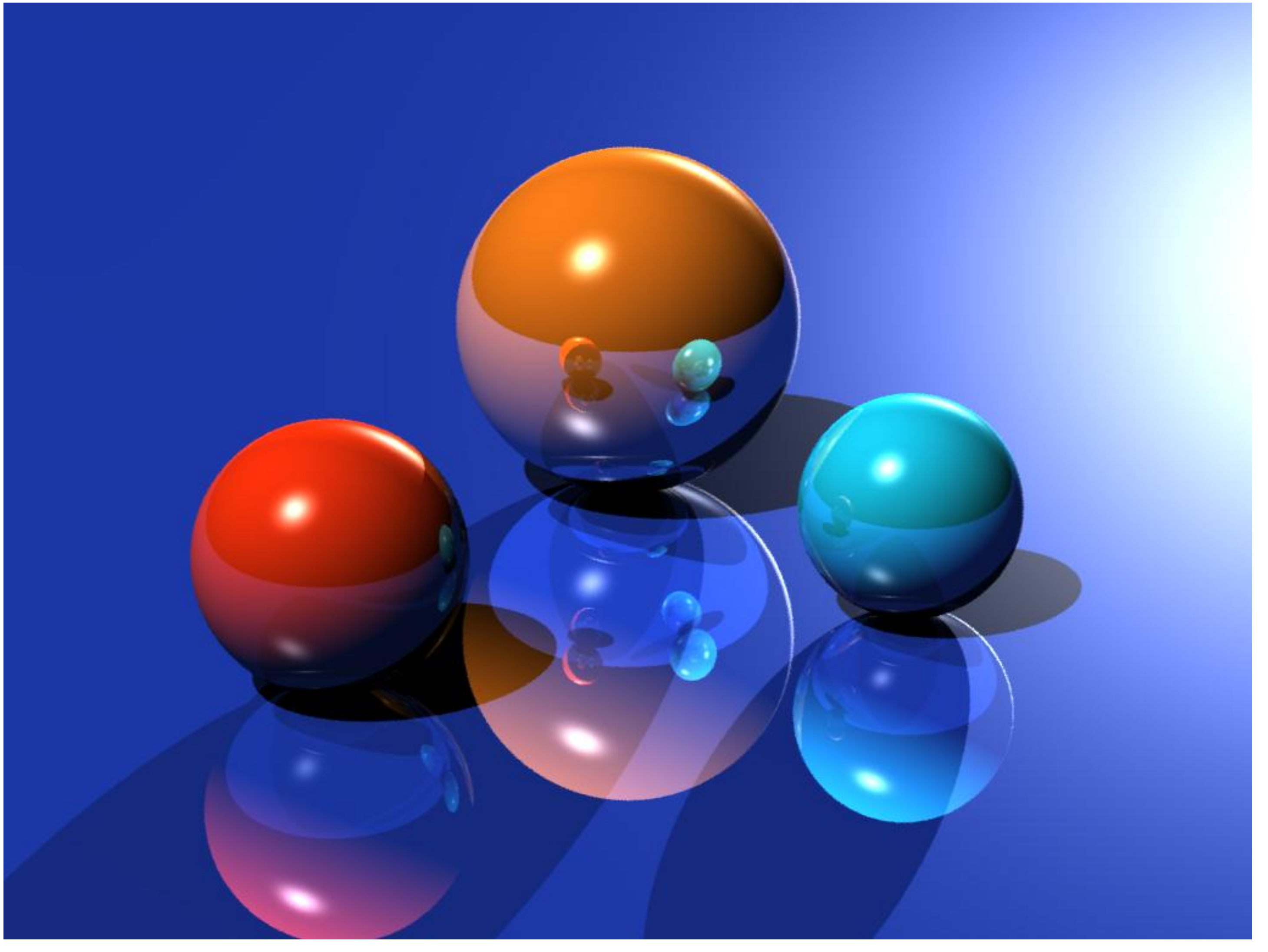
SOME THEORY BEHIND REAL-TIME RENDERING

Jaroslav Křivánek
Charles University in Prague

Off-line realistic rendering (not yet in rea-time)

Ray tracing




corona
FERRERO ROCHER
Chakib Rabia©
2012


corona
Image created by *Bertrand Benoit*
Rendered in *Corona Renderer*


corona
Image created by *Jeff Patton*
Rendered in *Corona Renderer*


2SA 8801

A room with a view

Put a rocking chair in front of your favourite window and experience how relaxing it is to get away from it all by just coming home. Life is in full swing outside, but you feel totally calm.

IKEA® Seat cushions filled with high resilience foam ...ide comfortable support for your body when ...pe when you rise. Cover: 53% linen, 47% viscose/rayon. [illegible] Risane natural.

02 New **FABRIKÖR glass-door cabinet $399** The shelves in the cabinet are adjustable – makes it easy to adjust the height to suit what you want to store. May be completed with DIODER LED lighting strip. Powder coated steel and tempered glass. Designer: Nike Karlsson. W57×D47, H150cm. Light green 702.422.94

03 VÄRMDÖ rocking-chair $169 Wooden furniture that is suitable for both indoor and outdoor use. Solid pine. Designer: Nike Karlsson. W65×D74, H106cm. Black 002.059.59

04 BJÖRNLOKA rug, flatwoven $199 The durable, soil-resistant wool surface makes this rug perfect in your living room or under your dining table. The rug is machine-woven. User surface: 100% pure new wool. W170×L240cm. Beige/black 402.290.05

05 HEMNES coffee table $229 Stained, clear lacquered solid pine. Designer: Carina Bengs. L90×W90, H46cm. Grey-brown 402.579.51

Show products (3)

Image created by *Weta Digital*

Image created by *Weta Digital*
© 20th Century Fox

vimeo >> “The Great Gatsby VFX”

vimeo >> “The Great Gatsby VFX”

Global illumination

Global illumination – Color bleeding

Image credit: Michael Bunnell

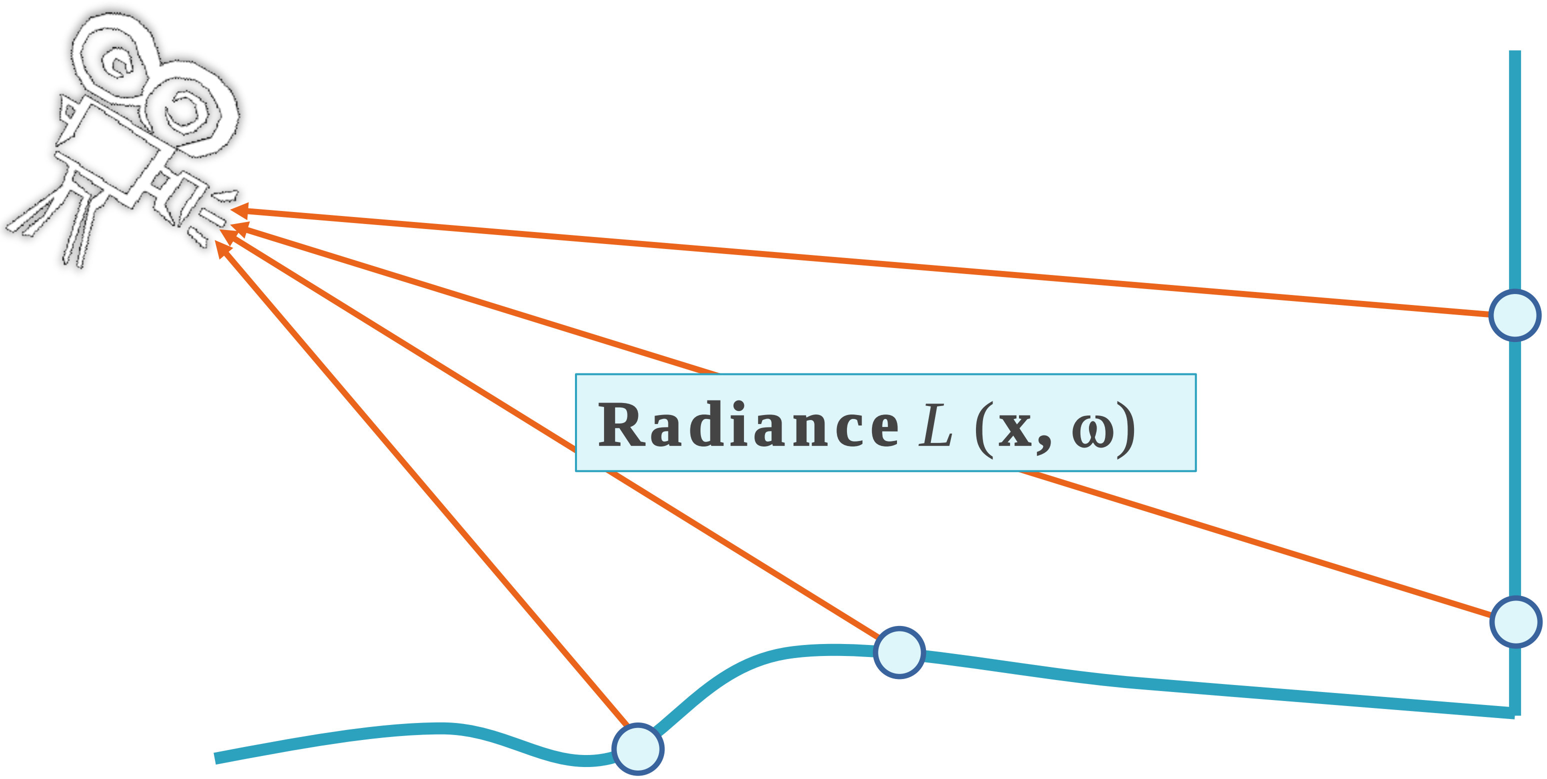
Global illumination – Caustics

Global illumination in games

Basic light transport theory

Image generation

- **Pixel value =** average **radiance** reflected from surfaces visible through the pixel
- Generating an image involves (some sort of) **light transport simulation**



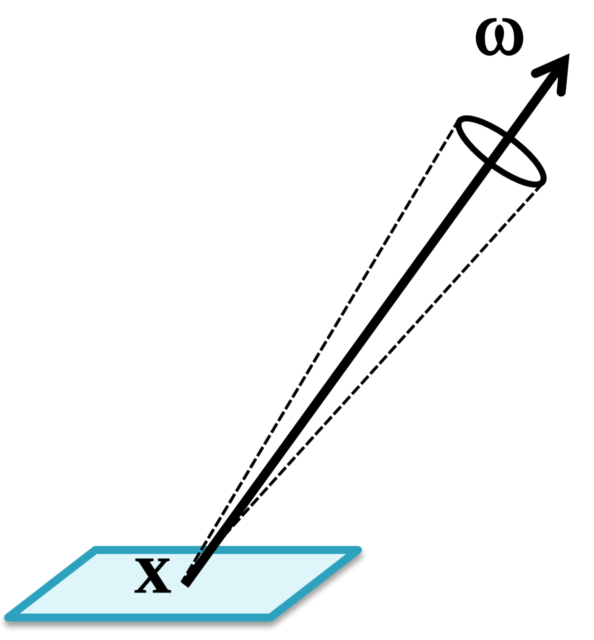
Radiance

- Radiometric quantity measuring "amount of light energy" along a ray

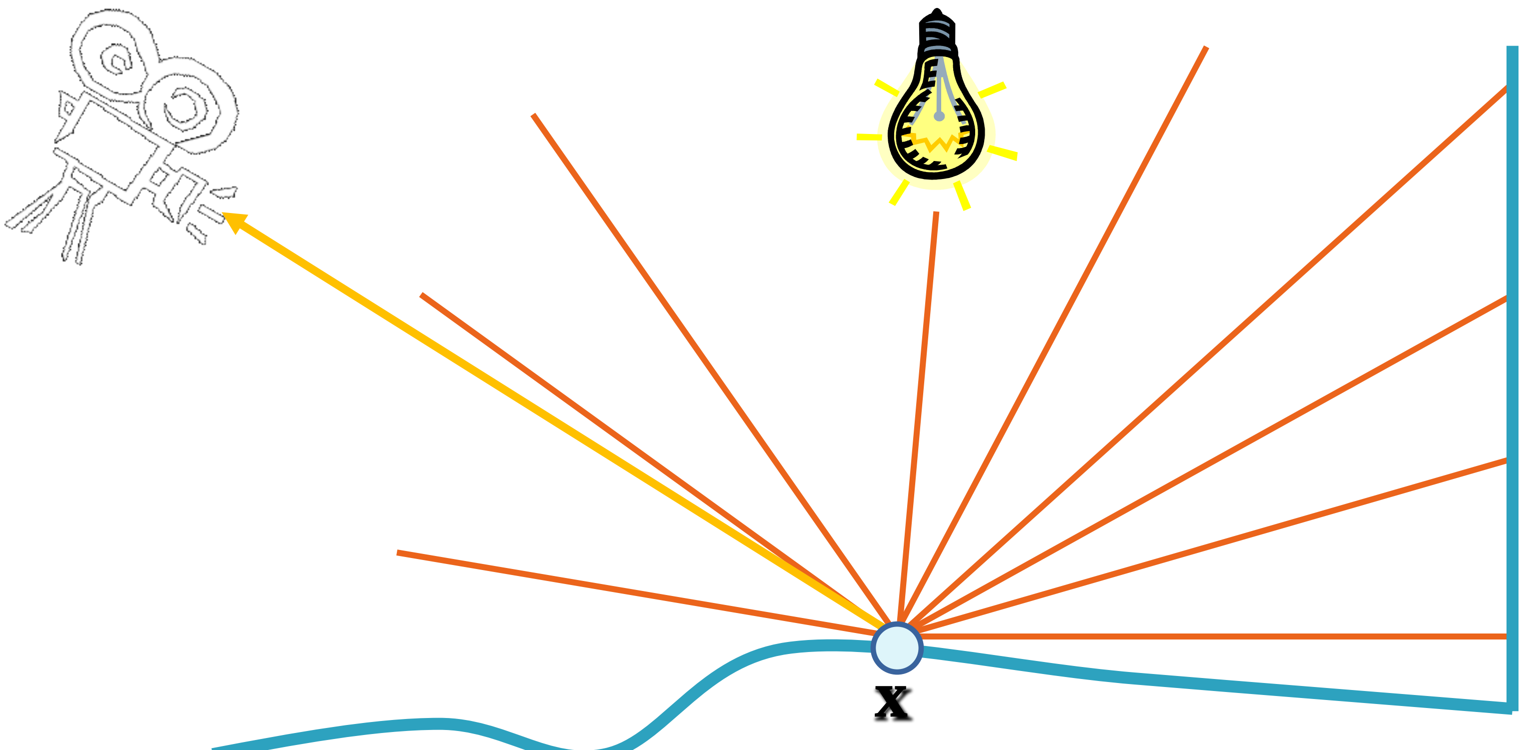
- Denoted L (**x**, ω)
  - **x** ... position
  - **ω** ... direction

- Units [W / m^2 sr]

- Essential properties of radiance
  - Proportional to perceived brightness
    - xxxxx
  - Constant along a ray
    - xxxxxx

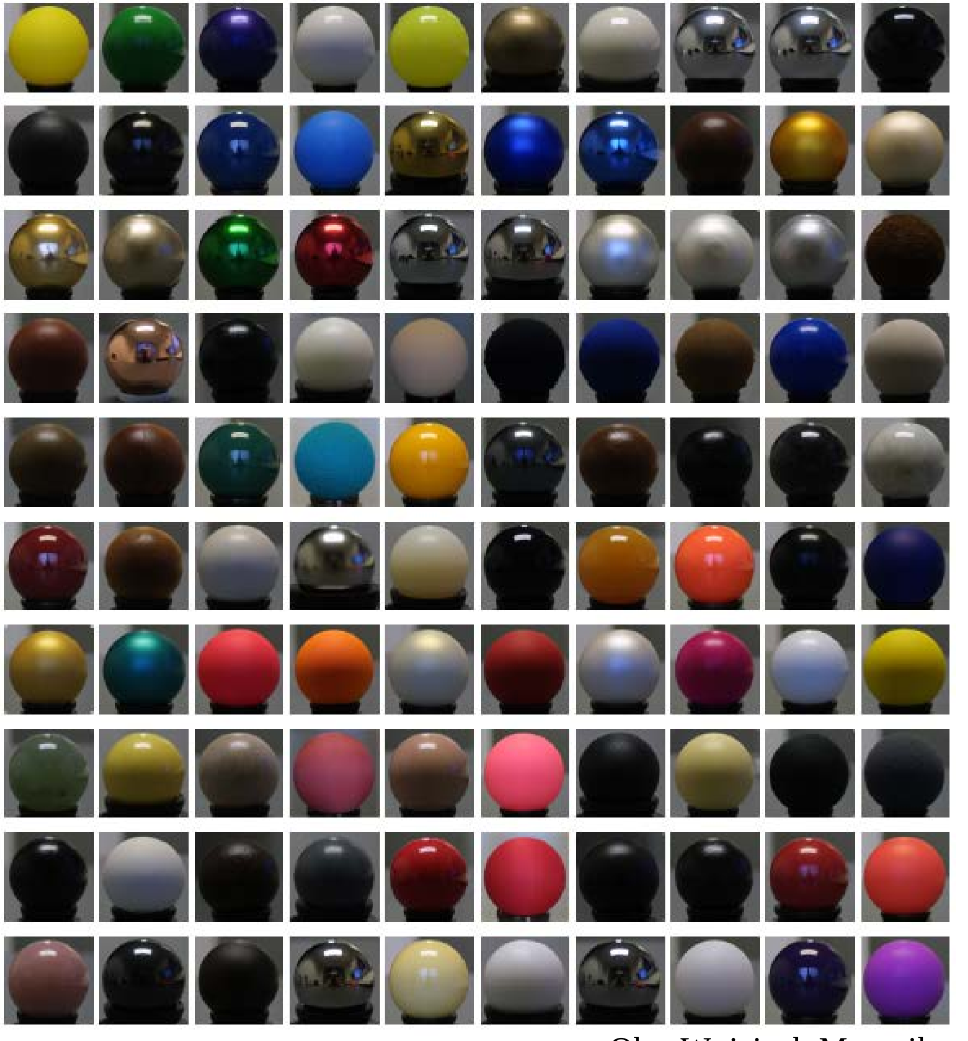


Light reflection



Light reflection

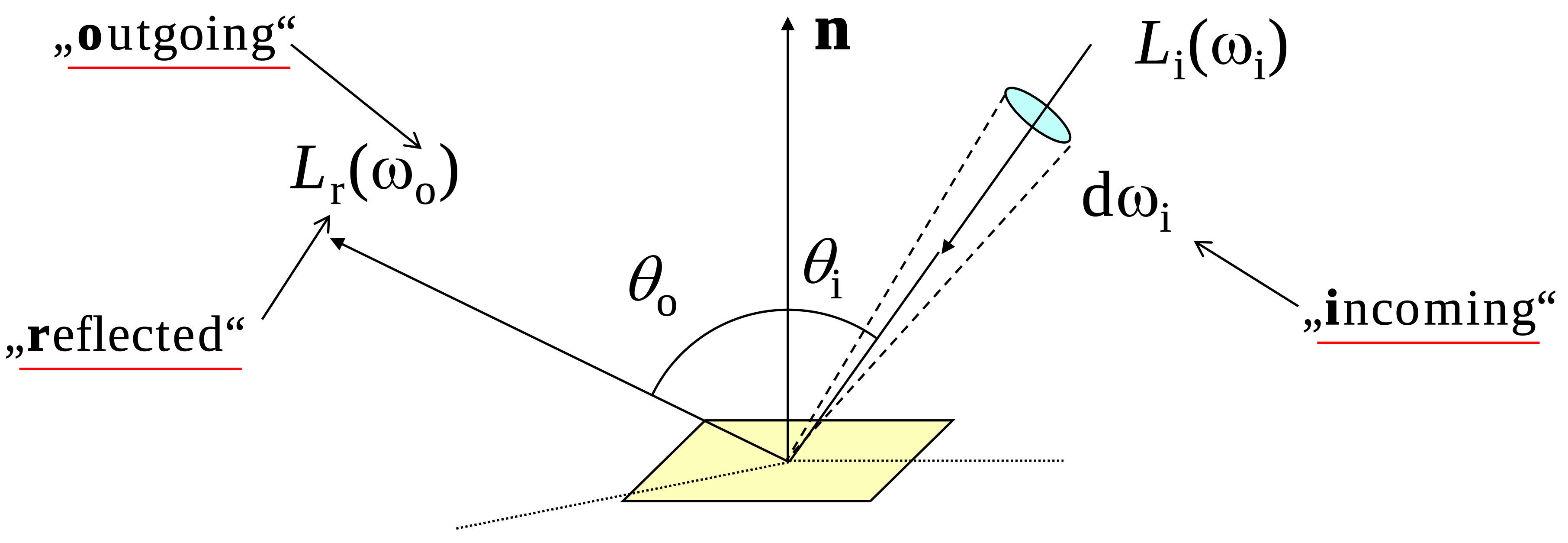
- **BRDF**
  - **B**i-directional **R**eflectance **D**istribution **F**unction



Obr. Wojciech Matusik

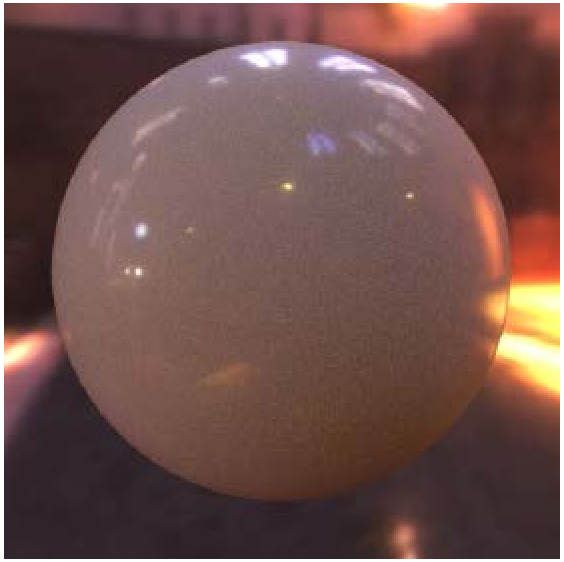
BRDF – Formal definition

- **B**idirectional **R**eflectance **D**istribution **F**unction

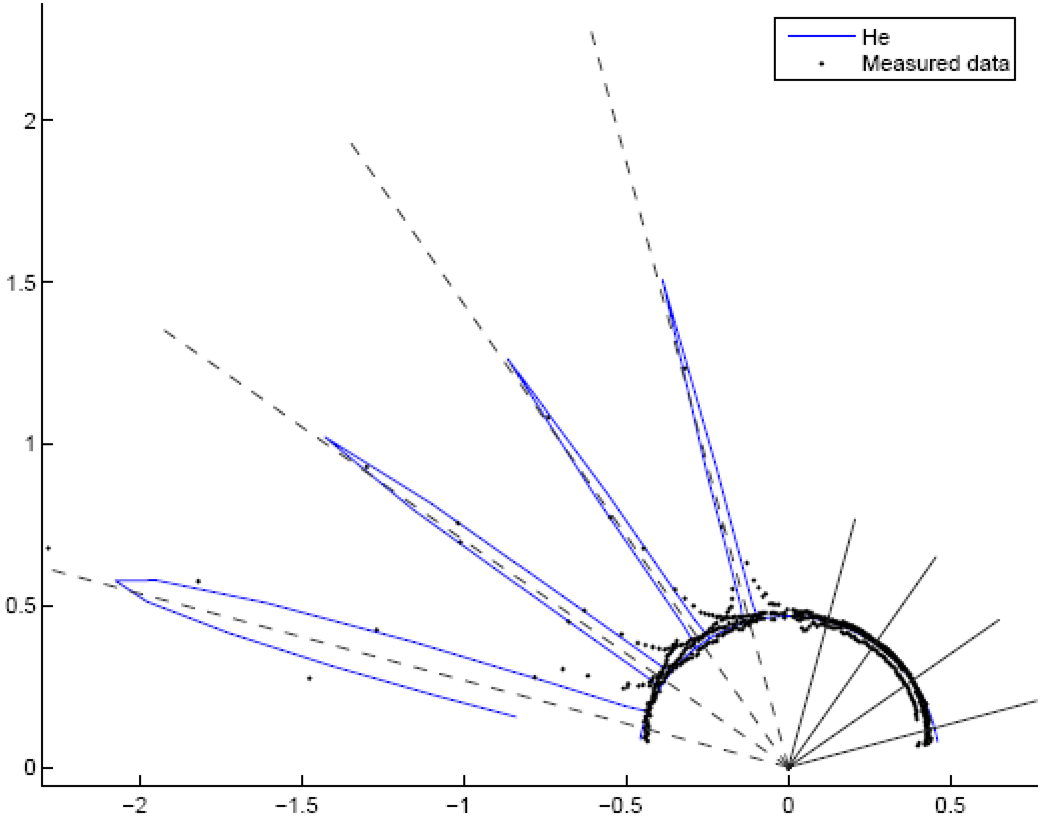


$$f_r(\omega_i \rightarrow \omega_o) = \frac{dL_r(\omega_o)}{L_i(\omega_i) \cdot \cos\theta_i \cdot d\omega_i} \quad [\mathrm{sr}^{-1}]$$

Surface appearance and the BRDF



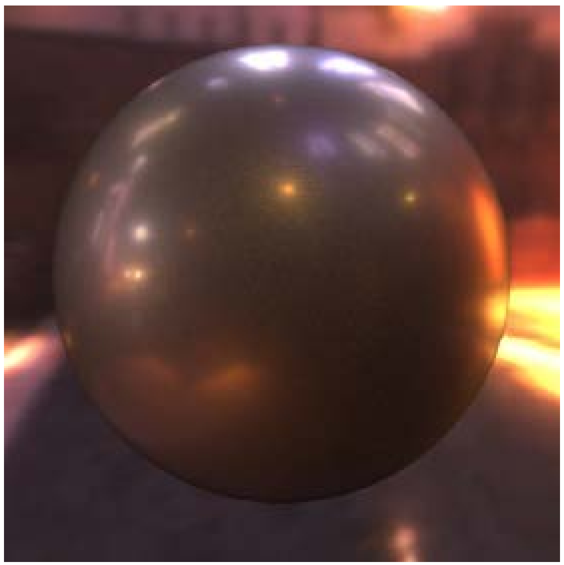
Appearance



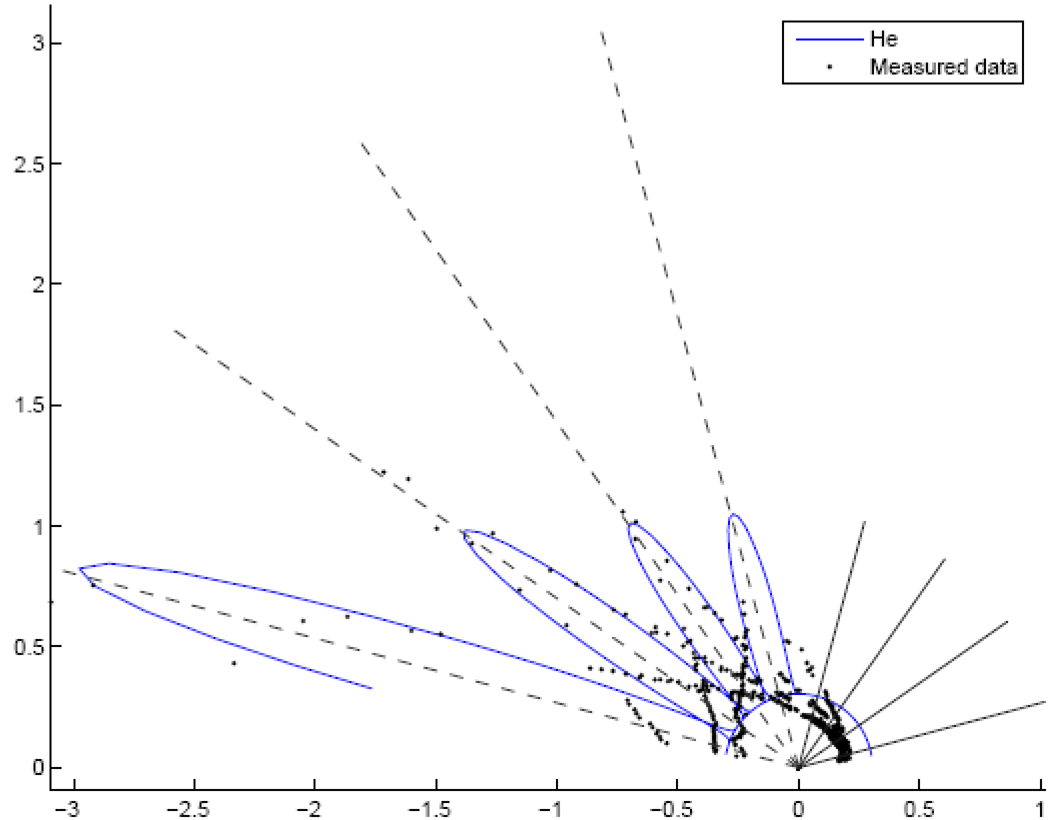
BRDF lobe
(for four different viewing directions)

Souce: Ngan et al. Experimental analysis of BRDF models, http://people.csail.mit.edu/addy/research/brdf/

Surface appearance and the BRDF



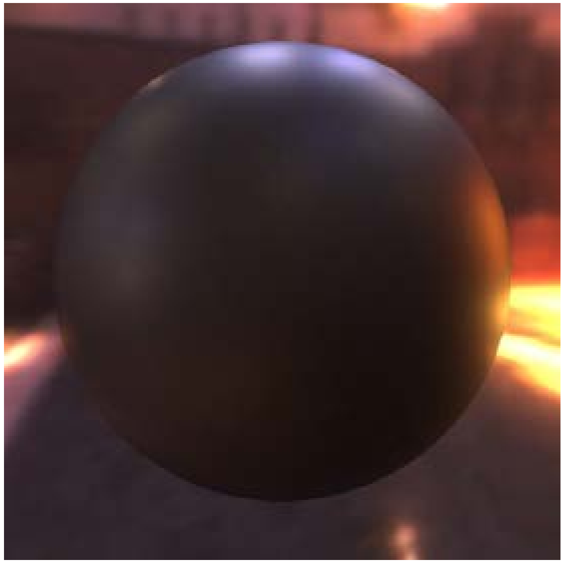
Appearance



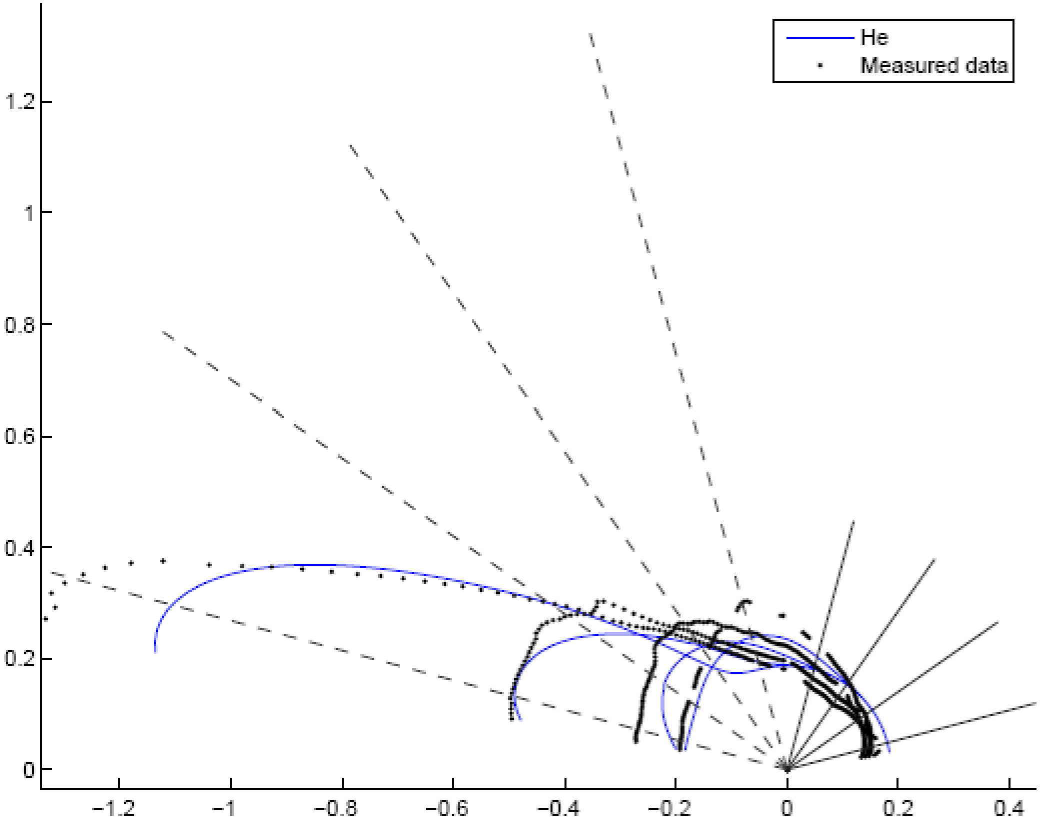
BRDF lobe
(for four different viewing directions)

Souce: Ngan et al. Experimental analysis of BRDF models, http://people.csail.mit.edu/addy/research/brdf/

Surface appearance and the BRDF



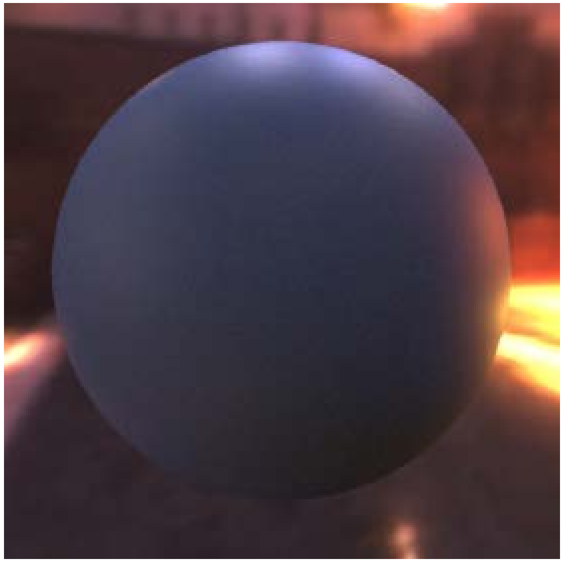
Appearance



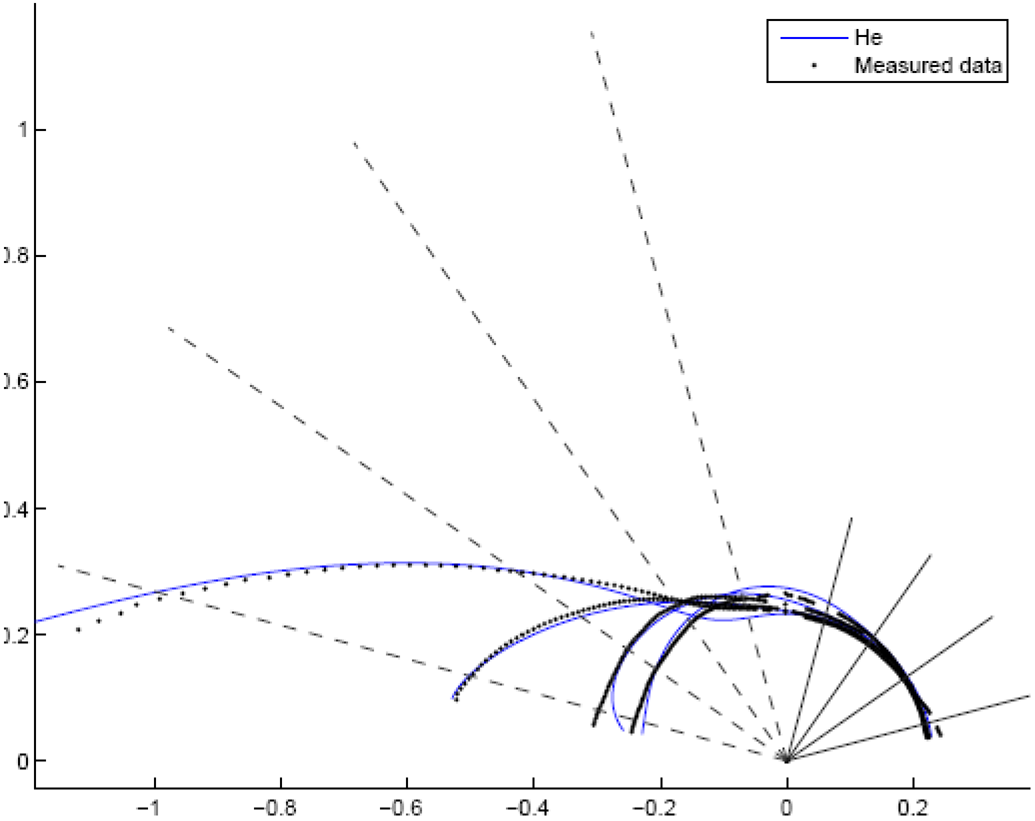
BRDF lobe
(for four different viewing directions)

Souce: Ngan et al. Experimental analysis of BRDF models, http://people.csail.mit.edu/addy/research/brdf/

Surface appearance and the BRDF



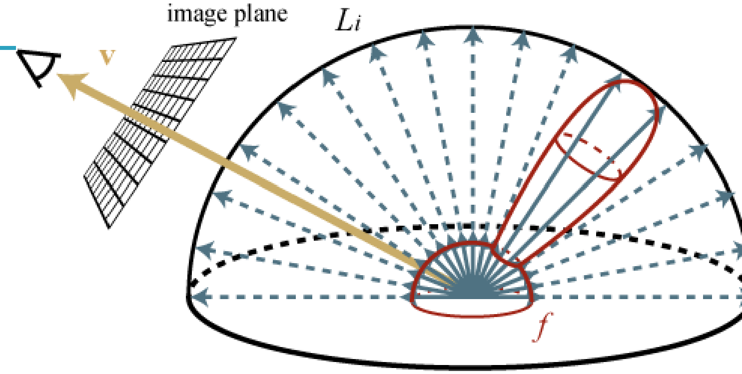
Appearance



BRDF lobe
(for four different viewing directions)

Souce: Ngan et al. Experimental analysis of BRDF models, http://people.csail.mit.edu/addy/research/brdf/

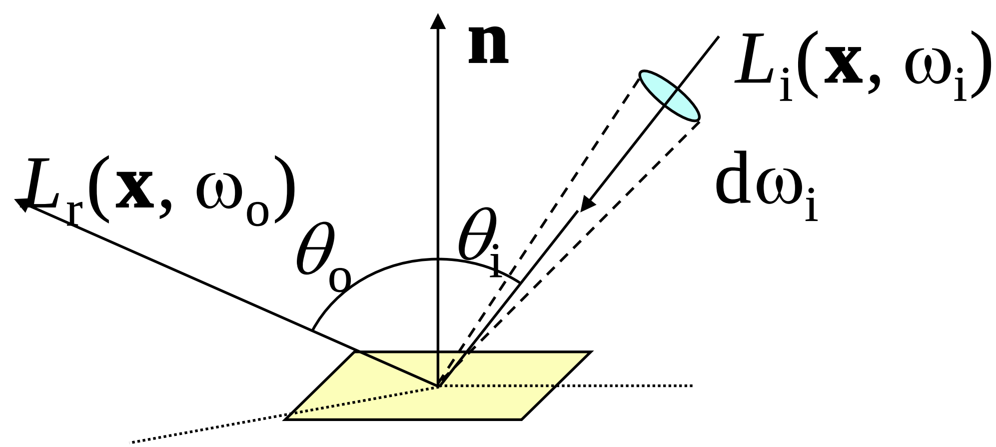
Reflection equation



- Total reflected radiance: integrate contributions of incident radiance, weighted by the BRDF, over the hemisphere

$$L_r(\mathbf{x},\omega_o) = \int_{H(\mathbf{x})} L_i(\mathbf{x},\omega_i)\cdot f_r(\mathbf{x},\omega_i \to \omega_o)\cdot \cos\theta_i \, d\omega_i$$

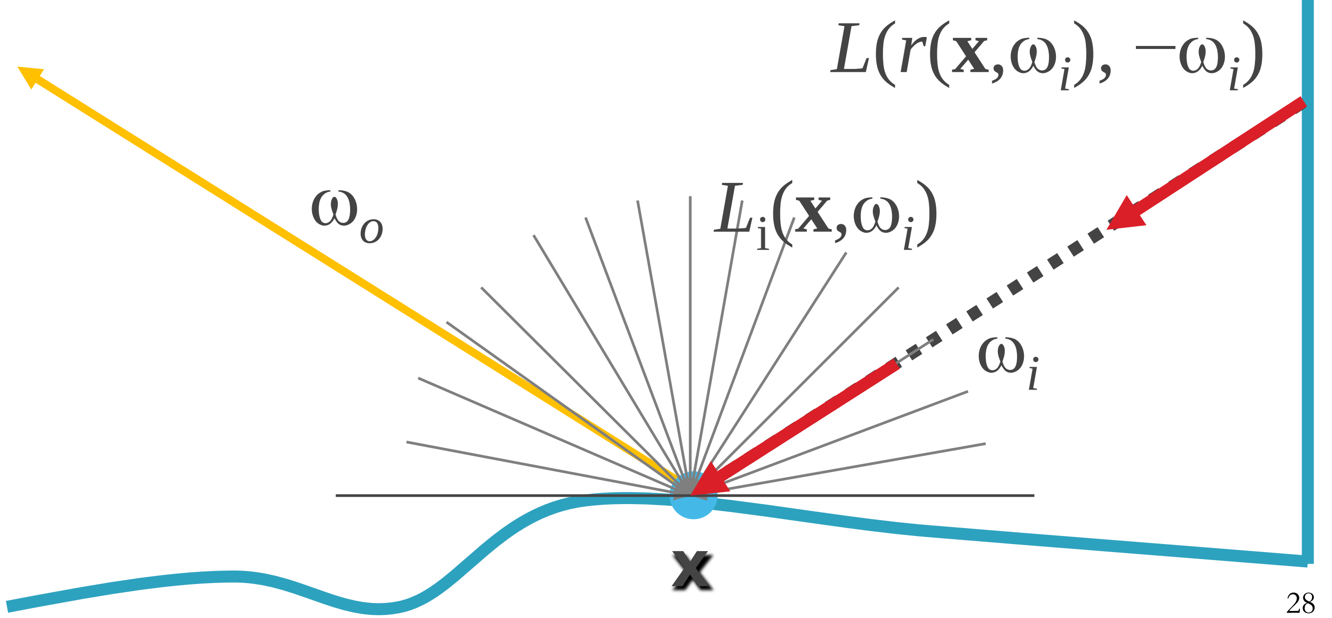
upper hemisphere over x



Rendering Equation

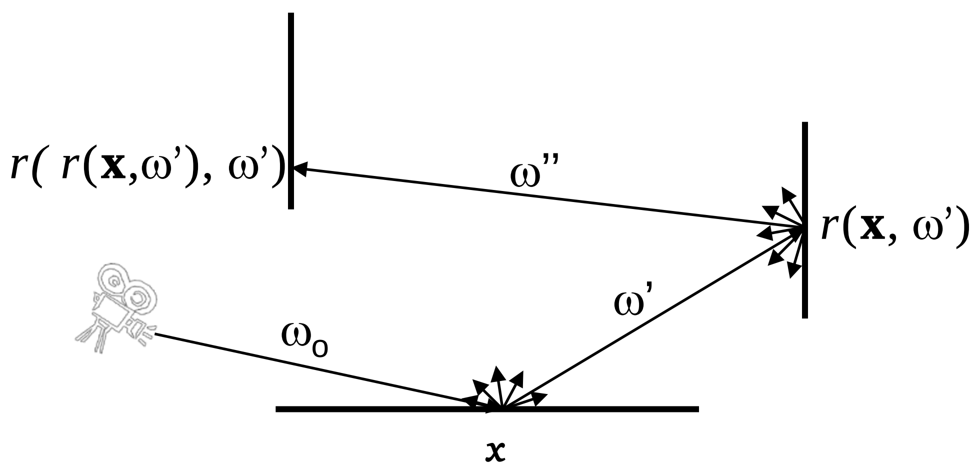
- But where does the incident radiance come from? Other surfaces in the scene!

$$L_o(\mathbf{x},\omega_o) = L_e(\mathbf{x},\omega_o) + \int_{H(\mathbf{x})} L_o(r(\mathbf{x},\omega_i),-\omega_i)\cdot f_r(\mathbf{x},\omega_i\rightarrow\omega_o)\cdot\cos\theta_i \, d\omega_i$$



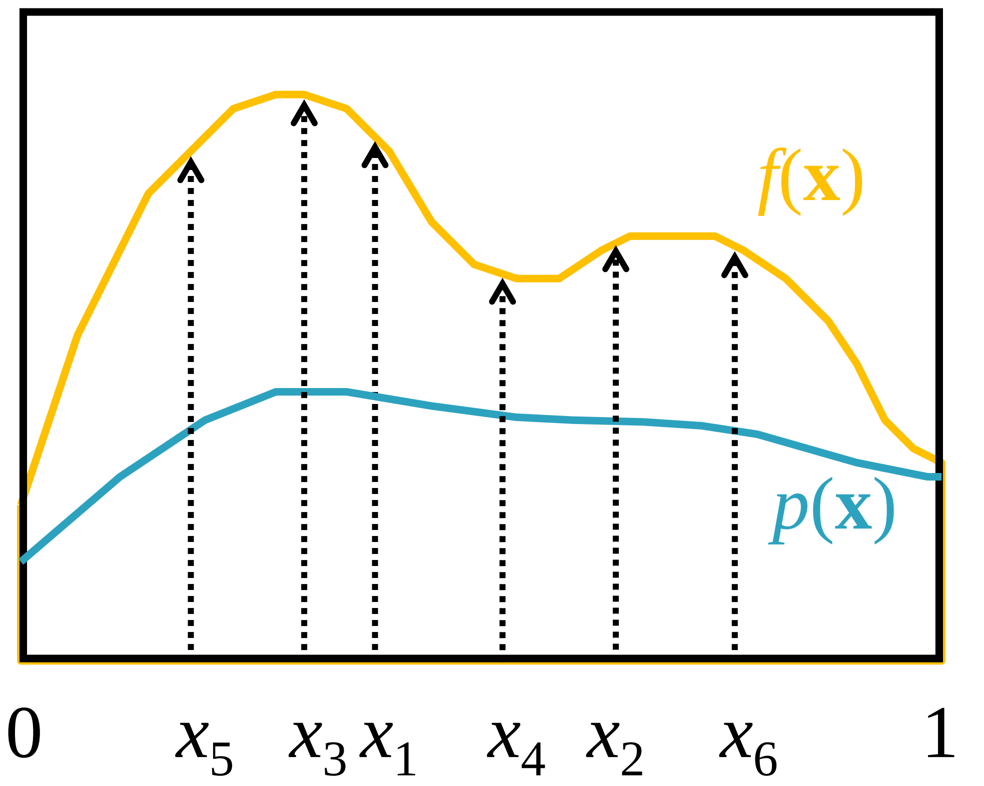
Recursive calculation of rendering equation

- To calculate $L(\mathbf{x}, \omega_o)$ I need to calculate $L(r(\mathbf{x}, \omega'), -\omega')$ for all directions ω' around the point **x**.
- For the calculation of each $L(r(\mathbf{x}, \omega'), -\omega')$ I need to do the same thing recursively
- At each step, apply Monte Carlo integration to sample the hemisphere
- etc.



Monte Carlo integration

- General approach to numerical evaluation of integrals



Integral:

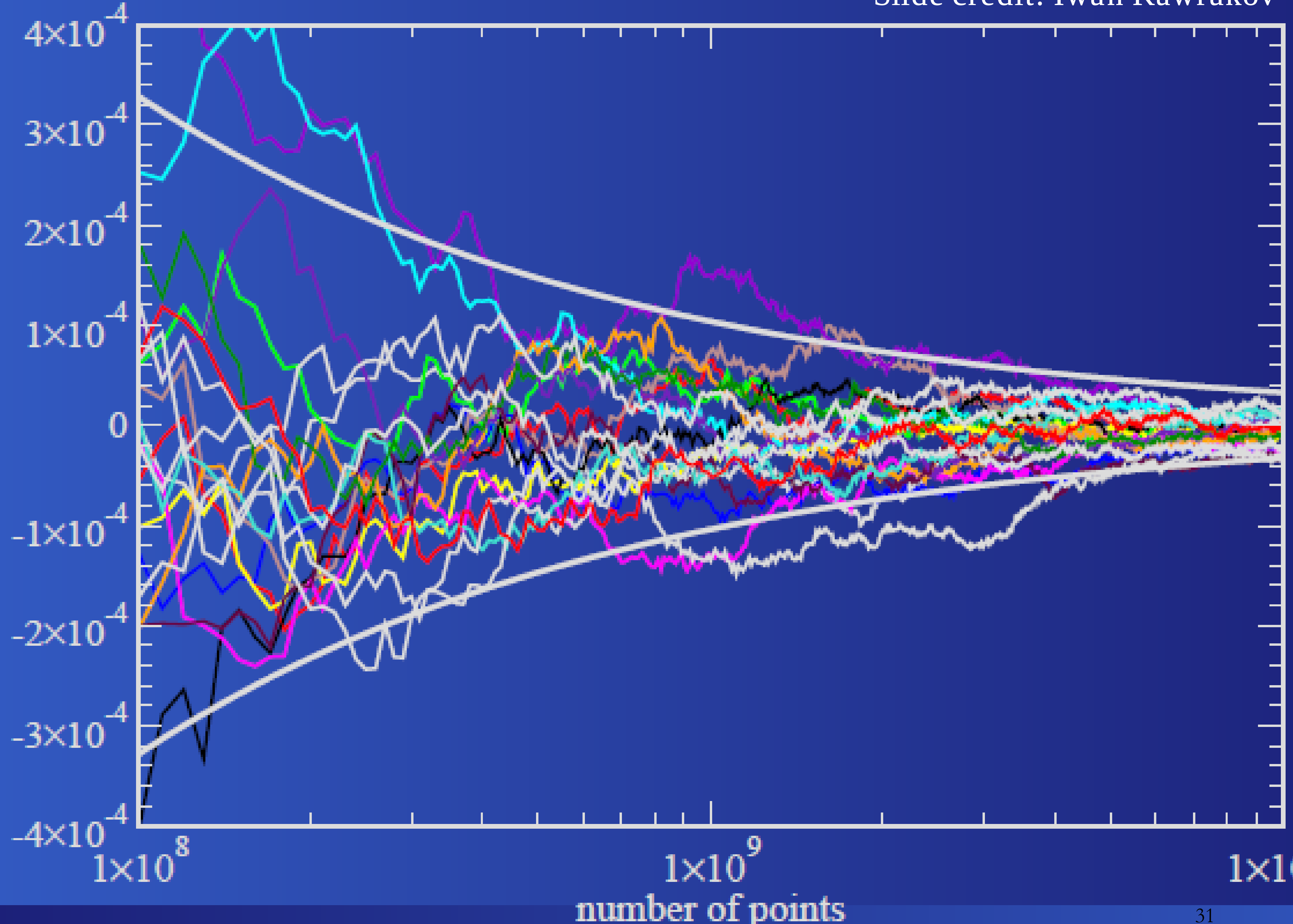
$$I = \int f(x)\mathrm{d}x$$

Monte Carlo estimate of I:

$$\langle I \rangle = \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)}; \quad x_i \propto p(x)$$

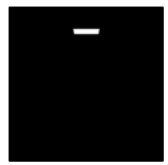
Correct „on average“:

$$E[\langle I \rangle] = I$$


4×10-4
3×10-4
2×10-4
1×10-4
0
-1×10-4
-2×10-4
-3×10-4
-4×10-4
1×10
8
1×10
9
number of points

Rendering equation – Operator form

$$L = L_e + T \circ L$$

 $=$  $+ \; T \circ$ 

- L: Equilibrium radiance
- L_e: Emitted radiance
- T: Transport & scattering operator

Progressive approximation

- Each application of T corresponds to one step of reflection & light propagation

$$L = L_e + TL_e + T^2 L_e + T^3 L_e + \dots$$

emission

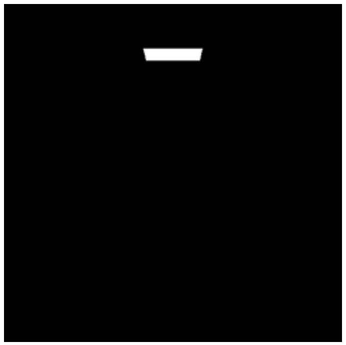
direct illumination

one-bounce indirect illumination

two-bounce indirect illumination

OpenGL shading

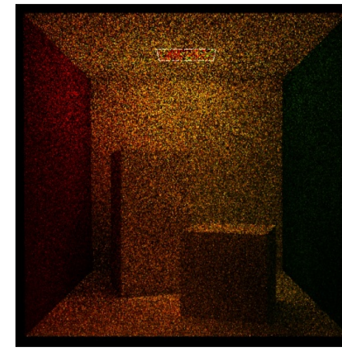
Progressive approximation



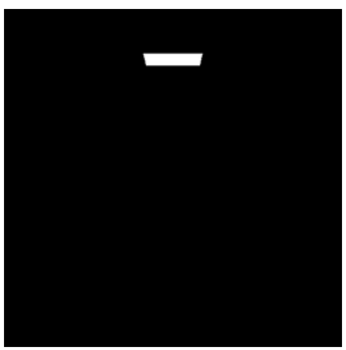
L_e

$T \circ L_e$

$T \circ T \circ L_e$

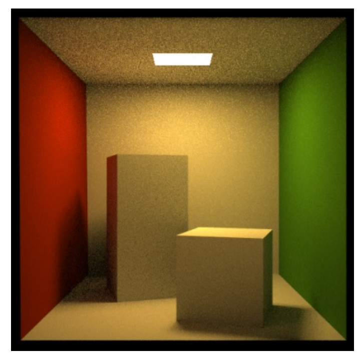


$T \circ T \circ T \circ L_e$

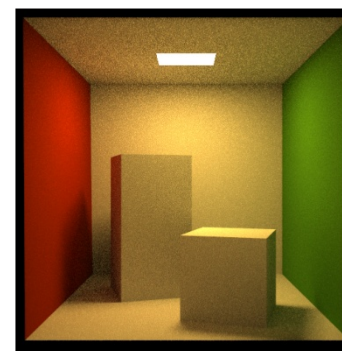


L_e

$L_e + T \circ L_e$



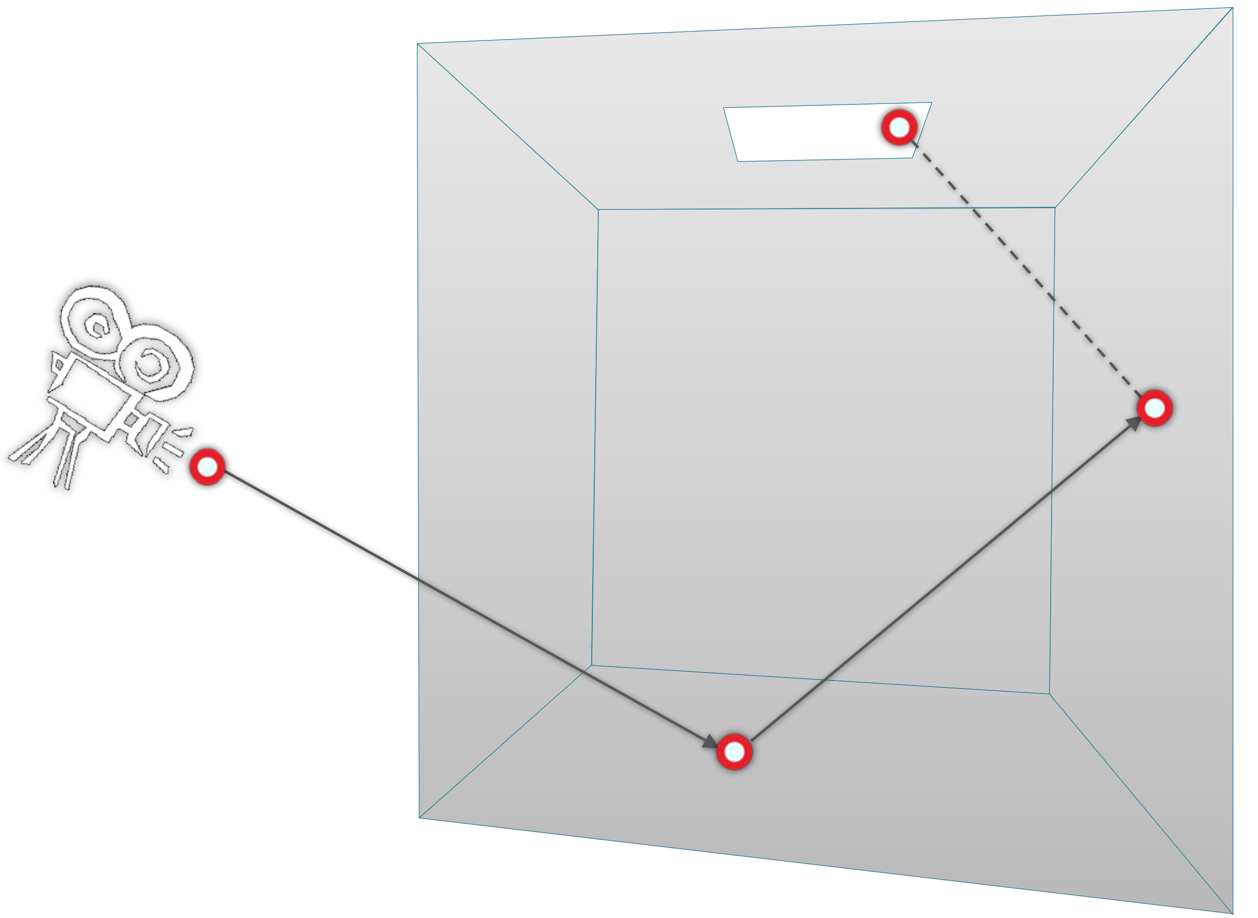
$L_e + TL_e + T^2 L_e$



$L_e + \ldots + T^3 L_e$

Idea: Distributed ray tracing (Cook '84)

Path tracing



Path tracing, Naive

```
getLi (x, ω):
    y = traceRay(x, ω)
    return
        Le(y, –ω) +                          // emitted radiance
        Lr (y, –ω)                           // reflected radiance

Lr(y, ω):
    ω′ = genUniformRandomDir( n(y) )
    return getLi (y, ω′) * brdf(y, ω, ω′) * dot(ω′, n(y)) * 2π
```

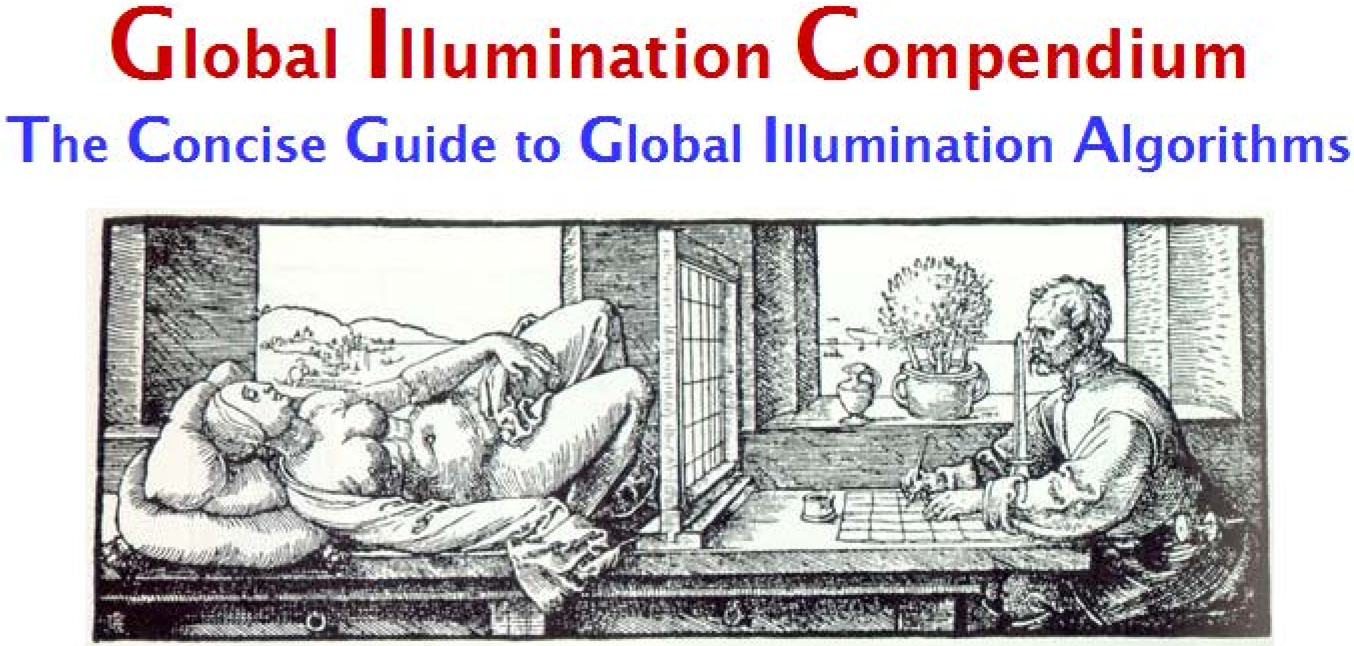
Path tracing, Naive

```
getLi (x, ω):
    y = traceRay(x, ω)
    return
        Le(y, –ω) +                    // emitted radiance
        Lr (y, –ω)                     // reflected radiance

Lr(y, ω):
    ω′ = genUniformRandomDir( n(y) )
    return getLi (y, ω′) * brdf(y, ω, ω′) * dot(ω′, n(y)) * 2π
```

Practical formulas for light transport

- P. Dutré: **Global Illumination Compendium**, http://people.cs.kuleuven.be/~philip.dutre/GI/

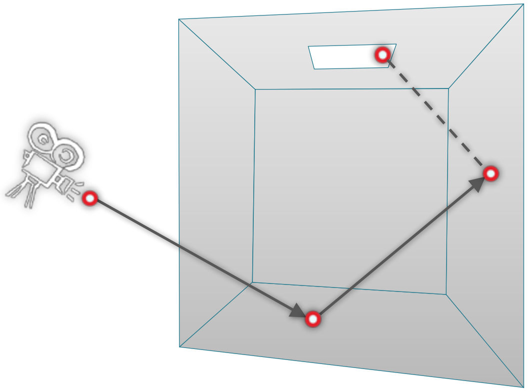


Albrecht Duerer, *Underweysung der Messung mit dem Zirkel und Richtscheyt* (Nurenberg, 1525), Book 3, figure 67.

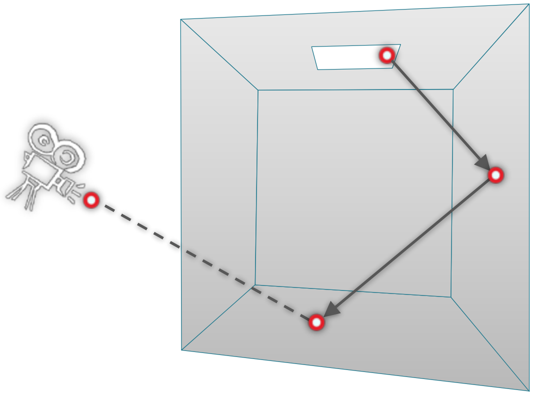
ROBUST GI CALCULATION

Path sampling techniques

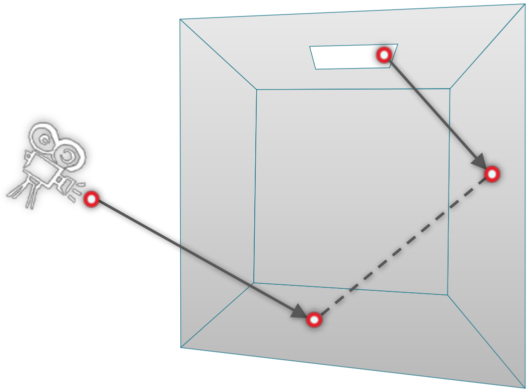
Path tracing



Light tracing

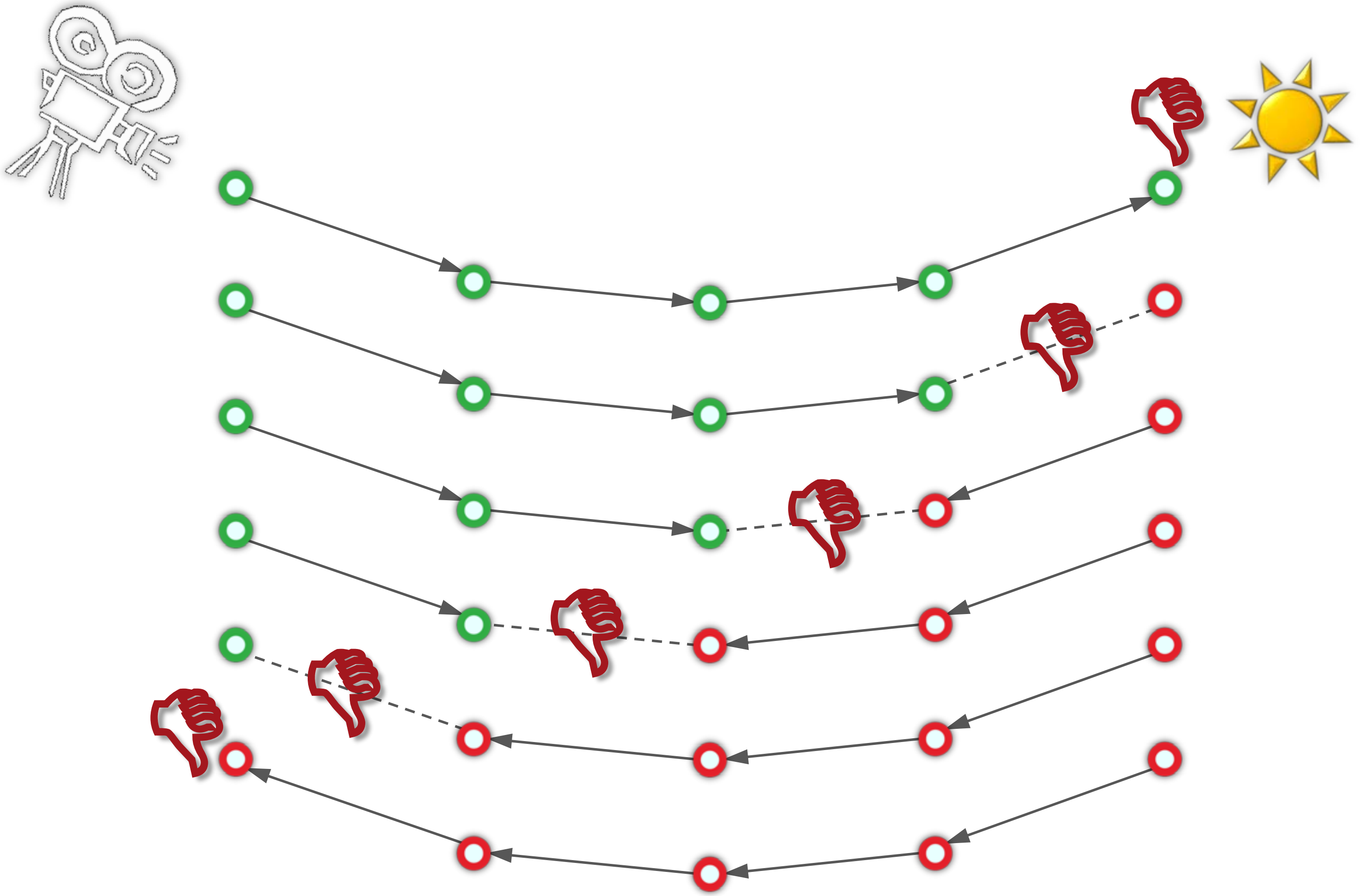


Bidirectional path tracing



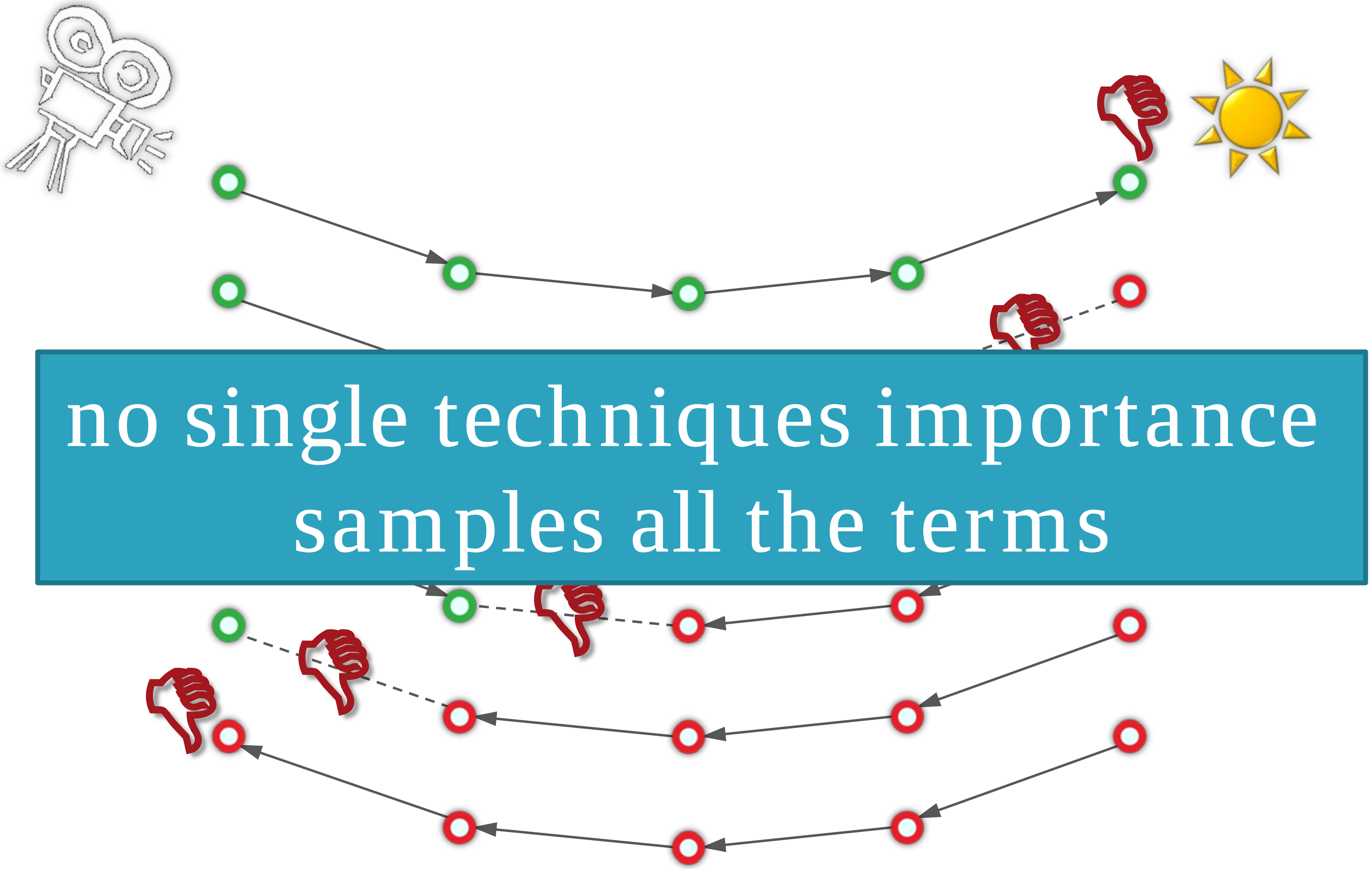
All possible bidirectional techniques

- vertex on a **light sub-path**
- vertex on en **eye subpath**



All possible bidirectional techniques

- vertex on a **light sub-path**
- vertex on en **eye subpath**

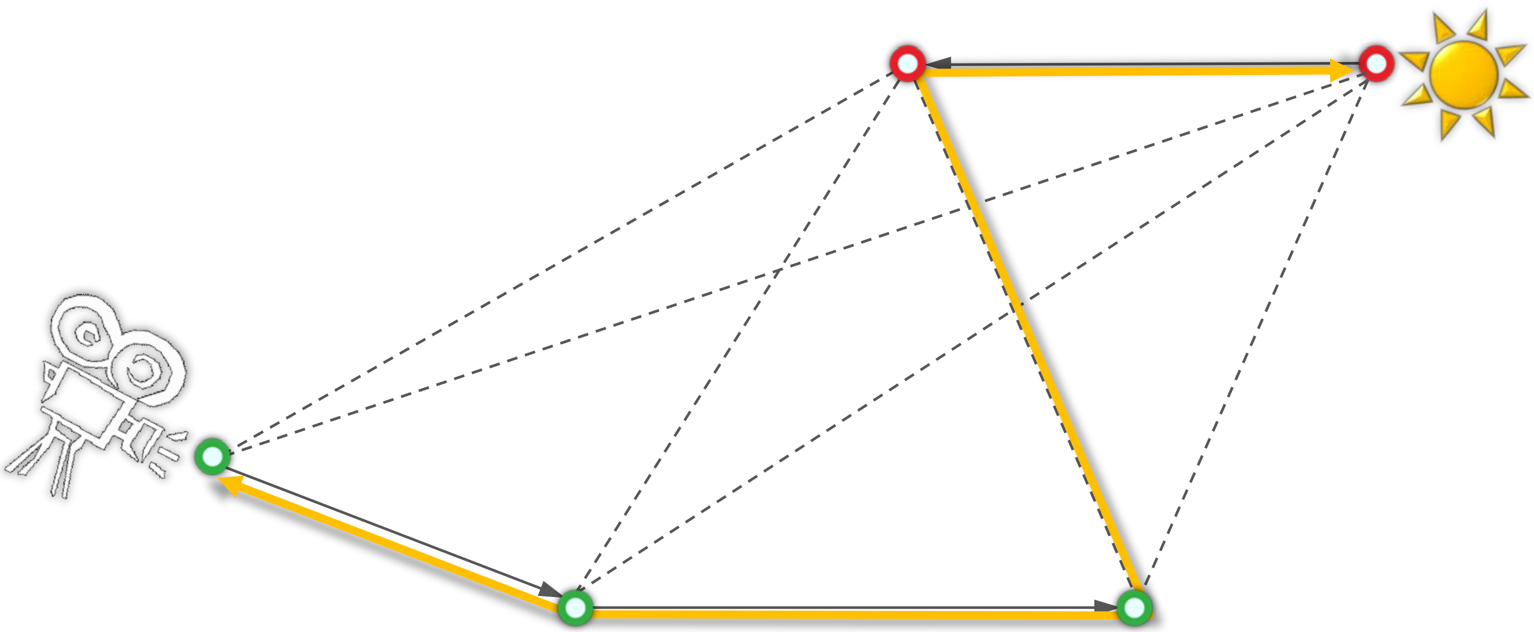


no single techniques importance samples all the terms

Bidirectional path tracing

- Use **all** of the above sampling techniques
- Combine using **Multiple Importance Sampling**

BPT Implementation

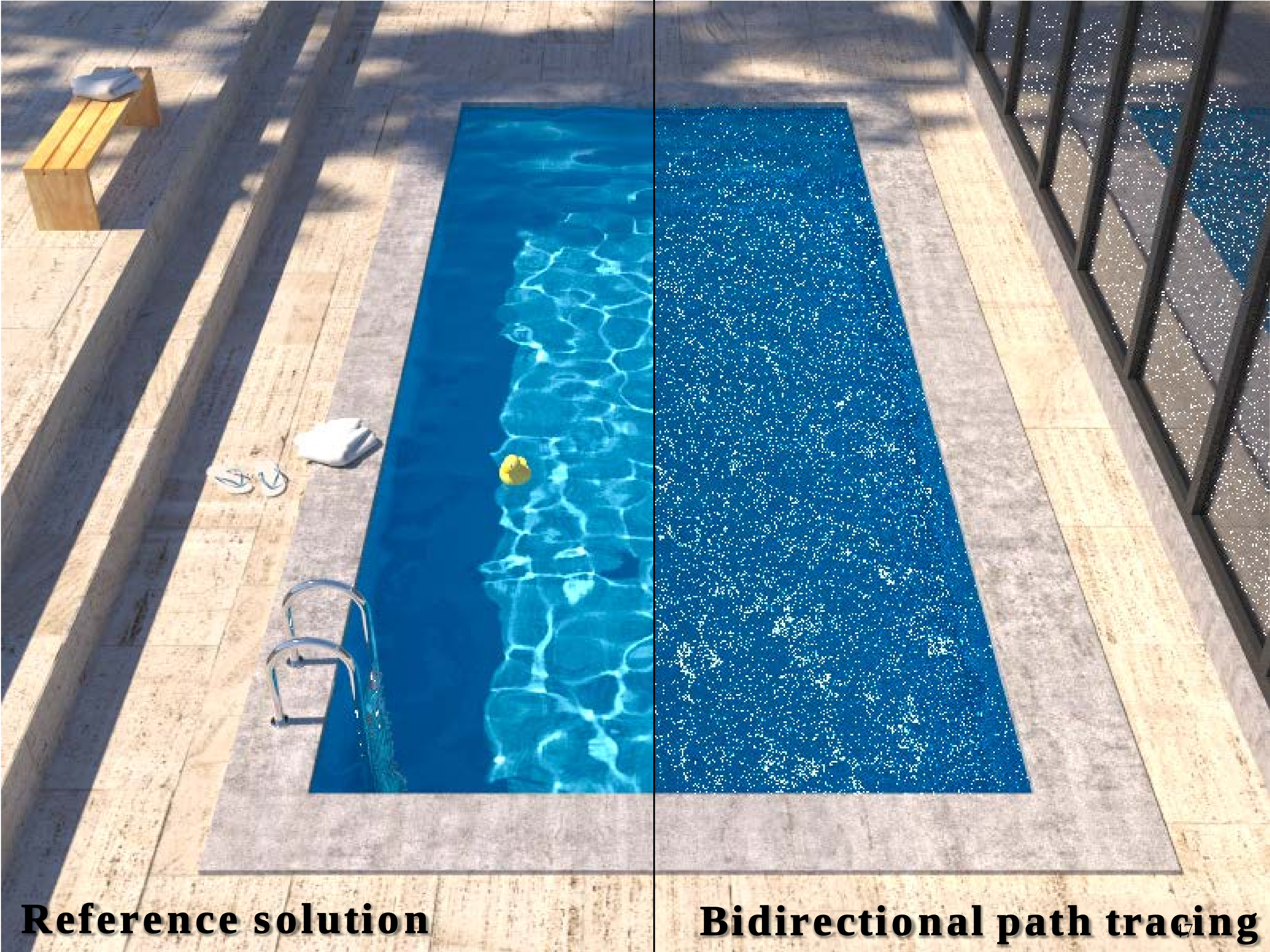


Results

BPT, 25 samples per pixel

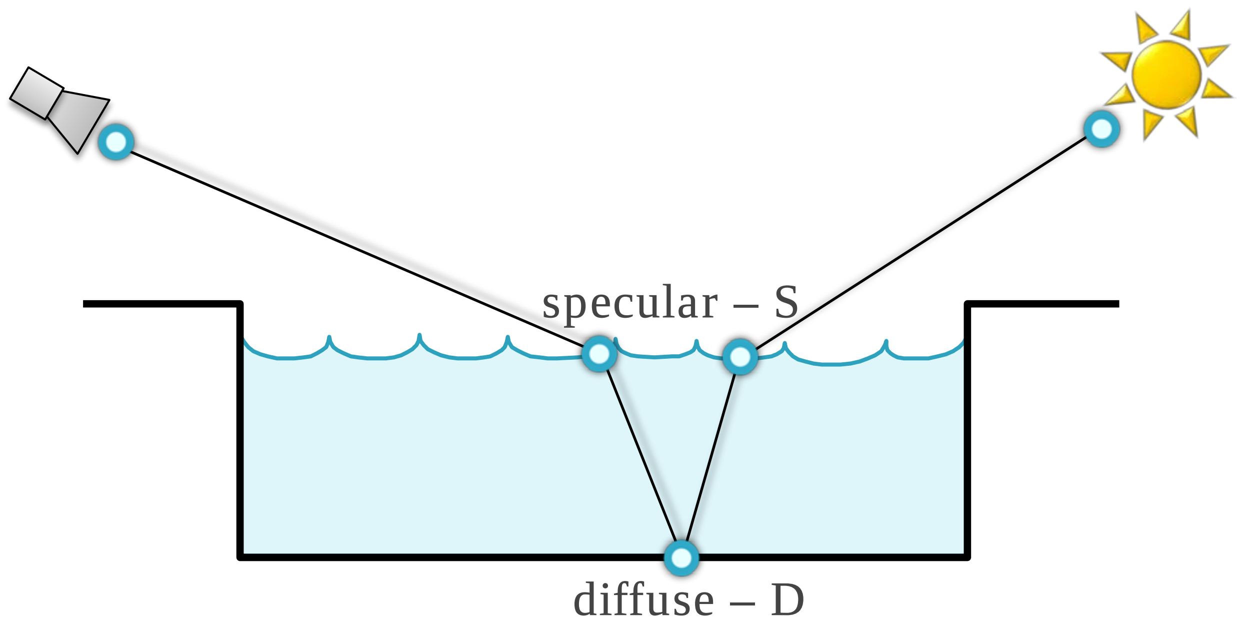
PT, 56 samples per pixel

Images: Eric Veach


Reference solution
Bidirectional path tracing

Insufficient path sampling techniques

- In BPT, **Specular-Diffuse-Specular paths** sampled with zero (or very small) probability

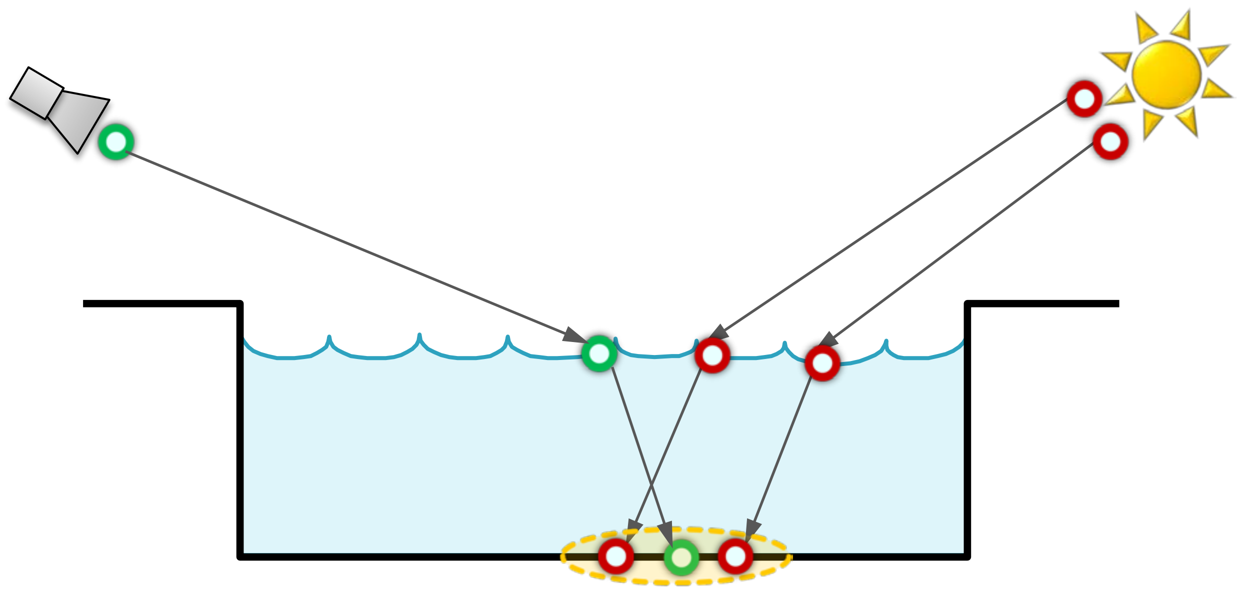


PHOTON MAPPING

(DENSITY ESTIMATION)

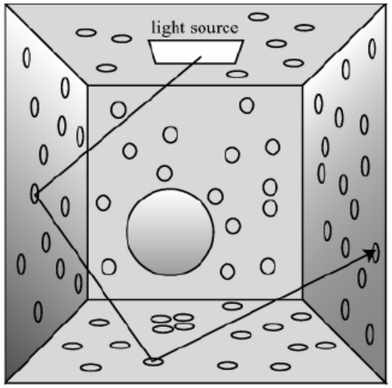
Photon mapping (Density estimation)

1. Many fwd walks + store collisions (“photon map”)
2. Radiance estimate: (Kernel) **density estimation**

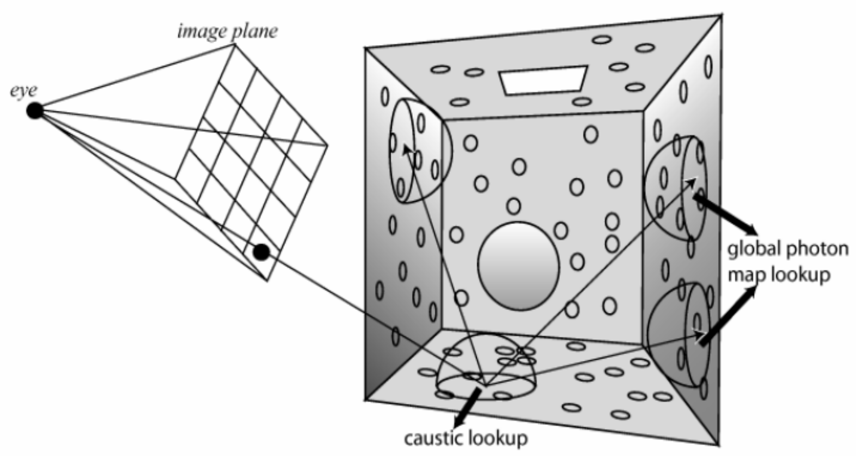


Photon mapping

1. **Photon tracing**
   - „Photon“ emission
   - Particle tracing
   - Storage in the „photon map“



2. **Rendering with photon map**
   - Like path tracing
   - Photon map query instead of recursion



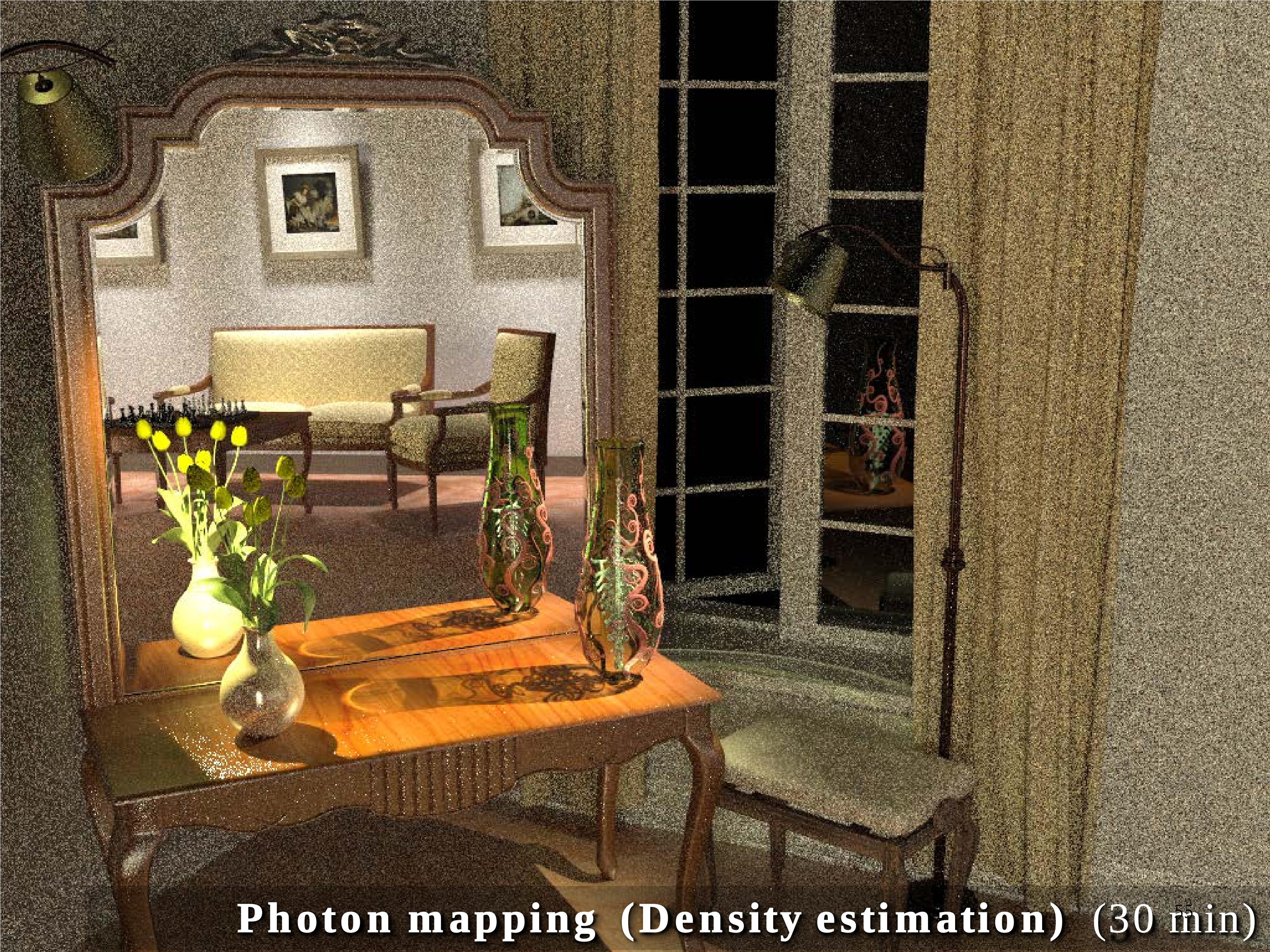
Photon mapping – SDS paths

© H.W.Jensen

© Wojciech Jarosz

Our work:
Vertex Connection and Merging
BPT + PM combination

Bidirectional path tracing (30 min)



Photon mapping (Density estimation) (30 min)

Vertex connection and merging (30 min)

THANK YOU!

Questions?

Computer
Graphics
Charles
University

[cgg.mff.cuni.cz/~jaroslav]