Selected Topics in Global Illumination Computation

Jaroslav Křivánek
Charles University, Prague
Jaroslav.Krivanek@mff.cuni.cz

Global illumination

- Light bouncing around in a scene

www.photos-of-the-year.com

Diffuse inter-reflection

- May go unnoticed, but looks odd if missing

Slide credit: Michael Bunnell

Why is GI important?

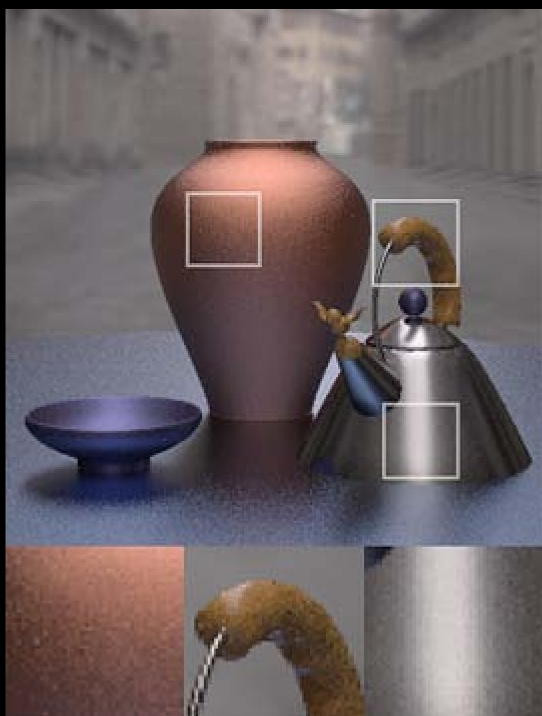
- Architectural visualization
- Interior design
- Product design
- Animated movies, special effects
- Games

- Quality criteria depend on the application

Syllabus discussion

Data-driven importance sampling

Measured BRDF sampling



[Lawrence et al. 2004]

Direct illumination sampling

[Wang & Åkerlund 2009]

Sampling various illumination effects

[Cline et al. 2008]

Many-light rendering methods

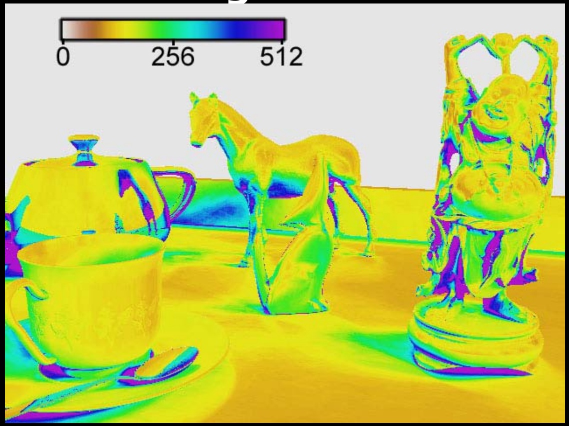
Basic formulation (VPLs)

[Keller 1997]

Speeding it up: Real-time methods

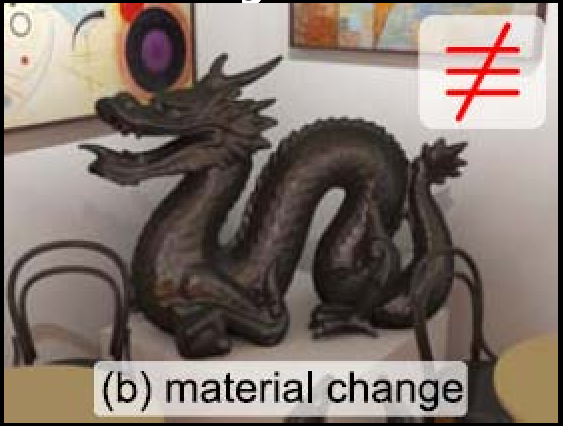
[Ritschel et al. 2008]

Making it scalable



[Walter et al. 2005]

Making it robust



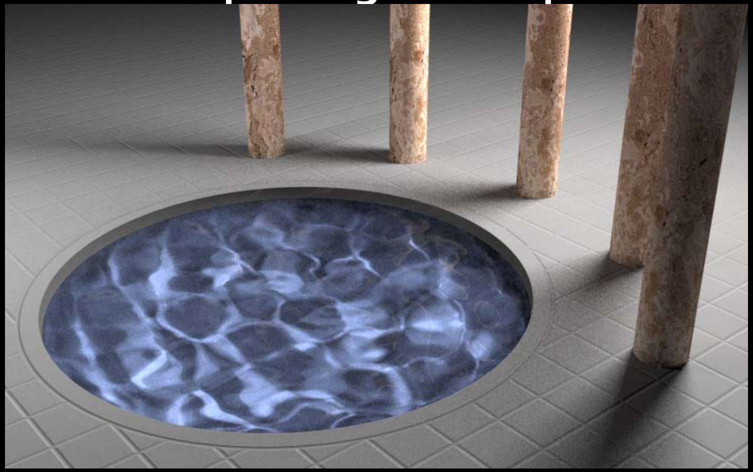
[Křivánek et al. 2010]

Real-time GI for Games & Other Apps

- No-precomputation
  - Real-time many-light methods
  - Other stuff
- Precomputed radiance transfer (PRT)
  - Light transport as a linear operator
  - Spherical harmonics PRT
  - Wavelet PRT
  - Separable BRDF approximation
  - Direct-to-Indirect Transfer
  - Non-linear operator approximations

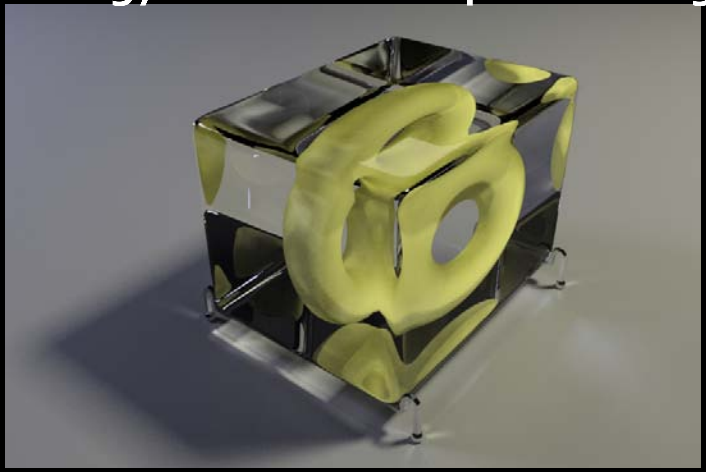
Metropolis sampling & apps

Metropolis light transport



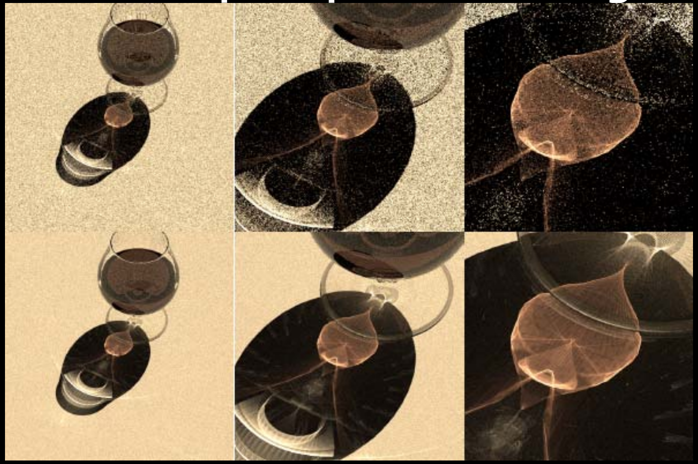
[Veach & Guibas, 1997]

Energy redistribution path tracing



[Cline et al. 2005]

Metropolis photon tracing



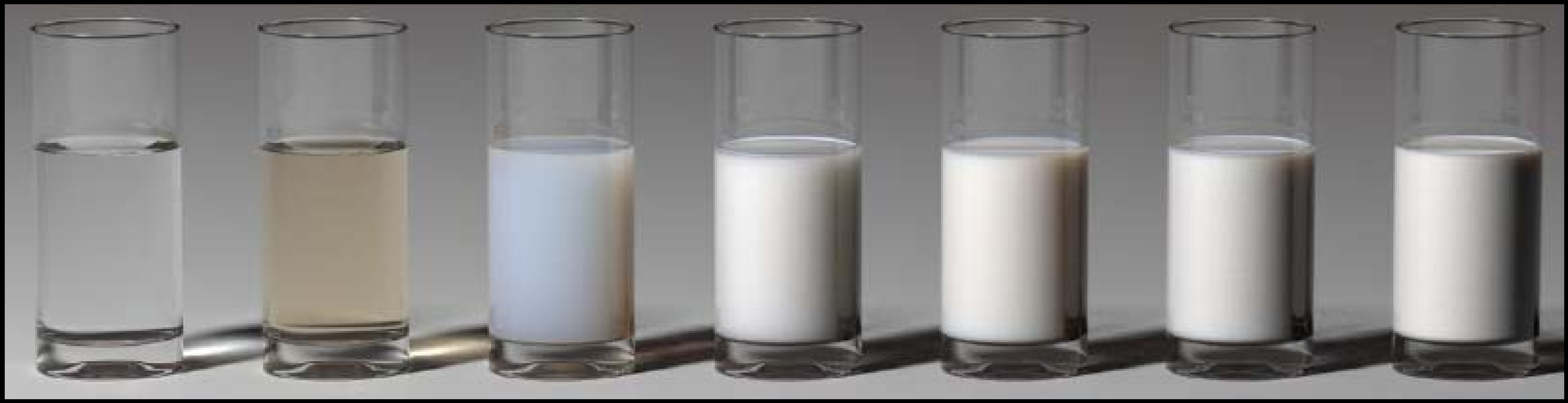
[Hachisuka & Jensen 2011]

Metropolis env. map sampling

[Ghosh & Heindrich 2006]

Participating media

- Radiative transport equation
- Single scattering solutions
- Multiple scattering solutions
  - Volumetric (bidirectional) path tracing
  - Volumetric photon mapping
  - Volumetric radiance caching



Subsurface scattering

- Diffusion approximation
- BSSRDF
- Fast hierarchical computation
- Multi-layered materials
- Real-time solutions

Appearance modeling

- Surface BRDF models
- Hair
  - Kajiya-Kay, Marschner reflection model
  - Multiple scattering in hair
- Measurement & data-driven models

[d'Eon et al. 2011]

A little more exotic stuff

- Light transport measurement
  - Nystrom kernel method
  - Compressed sensing
  - Separation of direct and indirect illumination
- Fabrication
  - Reflectance fabrication
  - Volumetric scattering fabrication

(a) Scene (b) Direct Component (c) Global Component

[Nayar et al. 2006]

Rules & Organization

Student participation

- Either
  - **Lecture notes**
    - Prepare notes based on lectures given in the class
  - **Give a 45-minute lecture**
    - Topic chosen by the student
    - Must be closely related to the class
- Or
  - **Research / Implementation**
    - Topic chosen by the student or assigned by the instructor
    - Preferably (but not necessarily) original, unpublished work

Student evaluation

- Student participation
- Oral exam
  - 2 questions from the lecture material
  - 1 out of 3 papers (must be different from the lecture material)

Books & other sources

- M. Pharr, G. Humphreys: *Physically-based rendering*. Morgan-Kaufmann 2004 (2nd ed. 2010)
- Dutre, Bala, Bekaert: *Advanced Global Illumination*. AK Peters, 2006.
- Szirmay-Kalos: *Monte-Carlo Methods in Global Illumination*, Vienna University of Technology, 2000. http://sirkan.iit.bme.hu/~szirmay/script.pdf
- Paper references on the class page: http://cgg.mff.cuni.cz/~jaroslav/teaching/2012-npgr031/index.html

A brief review of physically-based rendering

Required knowledge

- Radiometry / light reflection (BRDF)
- Rendering equation
- Monte Carlo quadrature
  - Primary estimator, Importance sampling, Stratified sampling, Multiple-importance sampling
- Path / light tracing
- Bidirectional path tracing
- (Progressive) photon mapping
- Irradiance caching

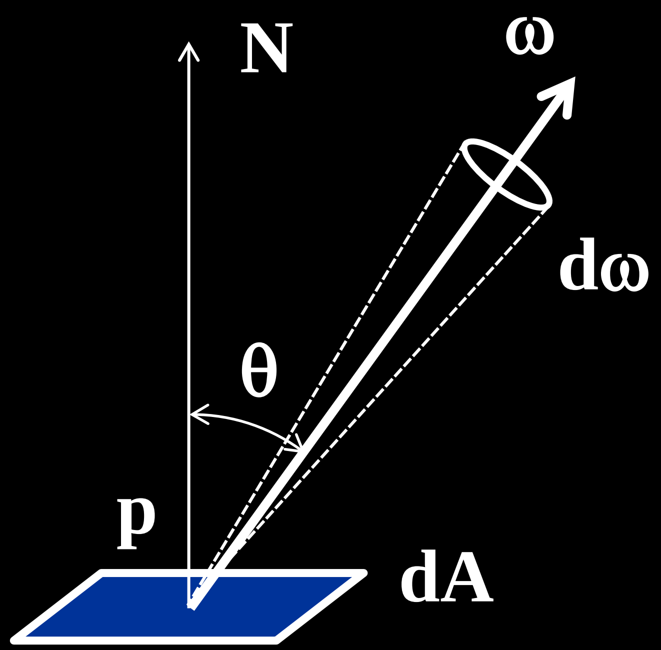
Rendering

- For each visible point **p** in the scene
  - How much light is reflected towards the camera

Radiance

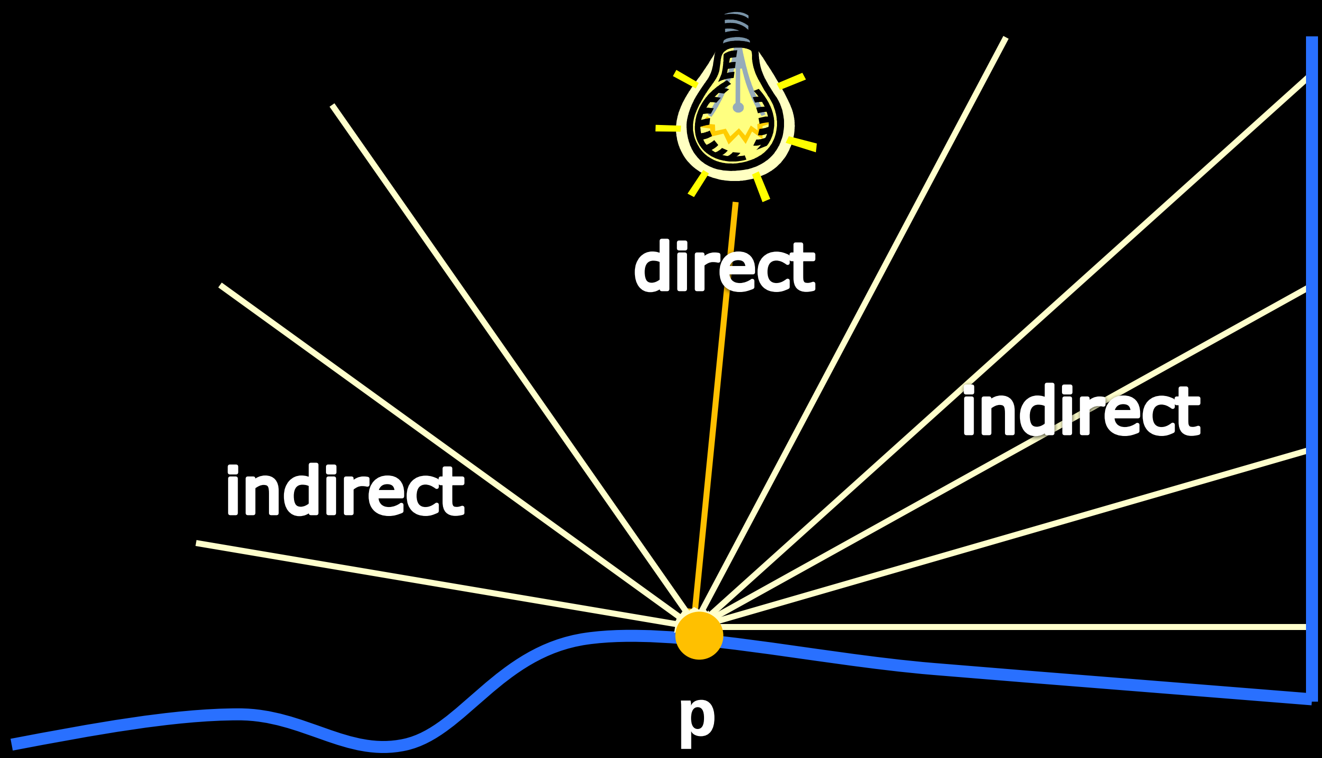
- "Amount of light" transported by a ray
  - To / from point **p**
  - To / from direction **ω**
- L (**p**, **ω**) [W / m^2 sr]

- Constant along a ray
- Proportional to perceived <u>brightness</u>



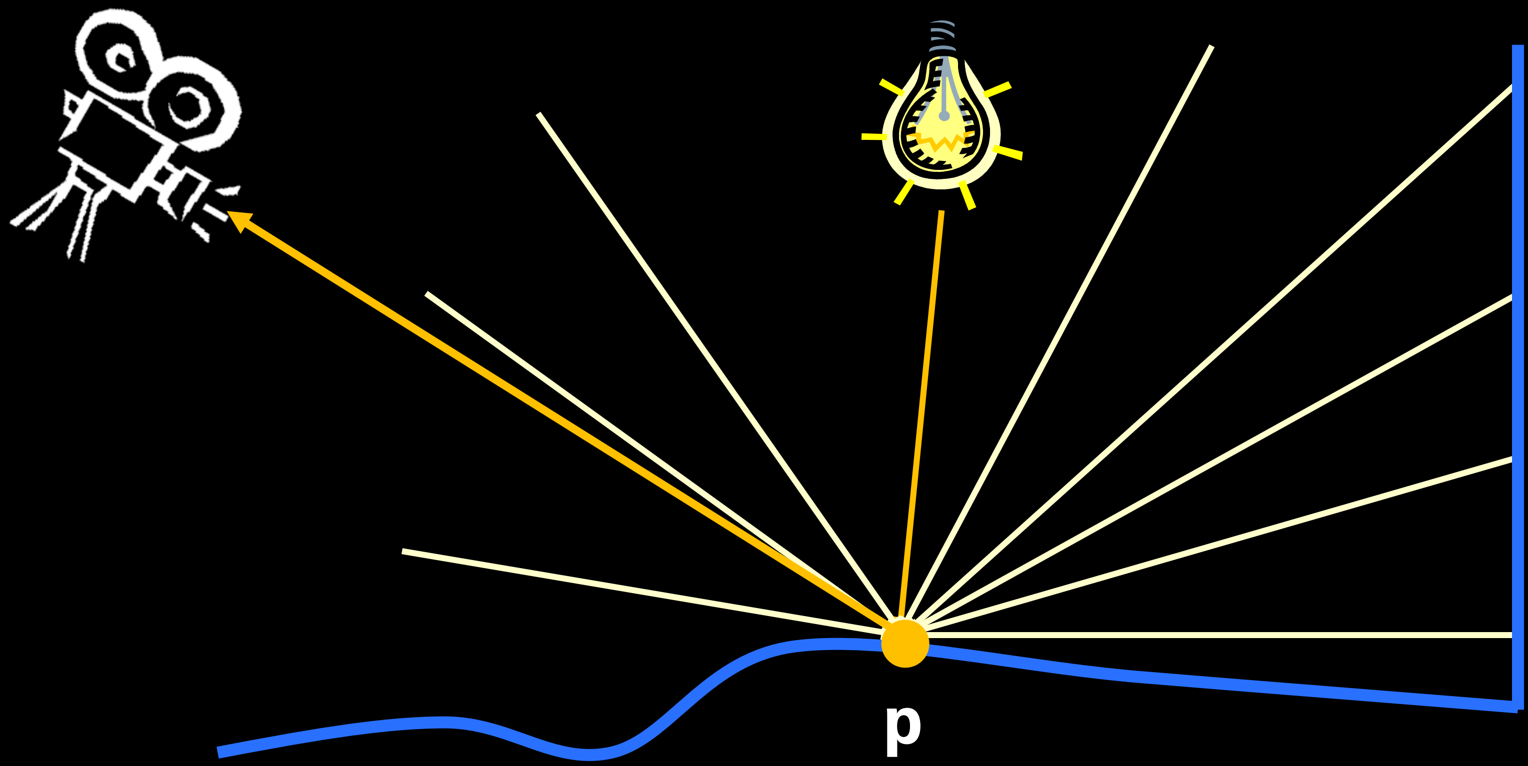
Direct vs. indirect illumination

- Where does the light come from?
  - Light sources (*direct illumination*)
  - Scene surfaces (*indirect illumination*)



Where does the light go then?

- Light reflection – material reflectance



Light reflection

- **BRDF**
  - Bi-directional Reflectance Distribution Function
- Implementation:
  - Shader

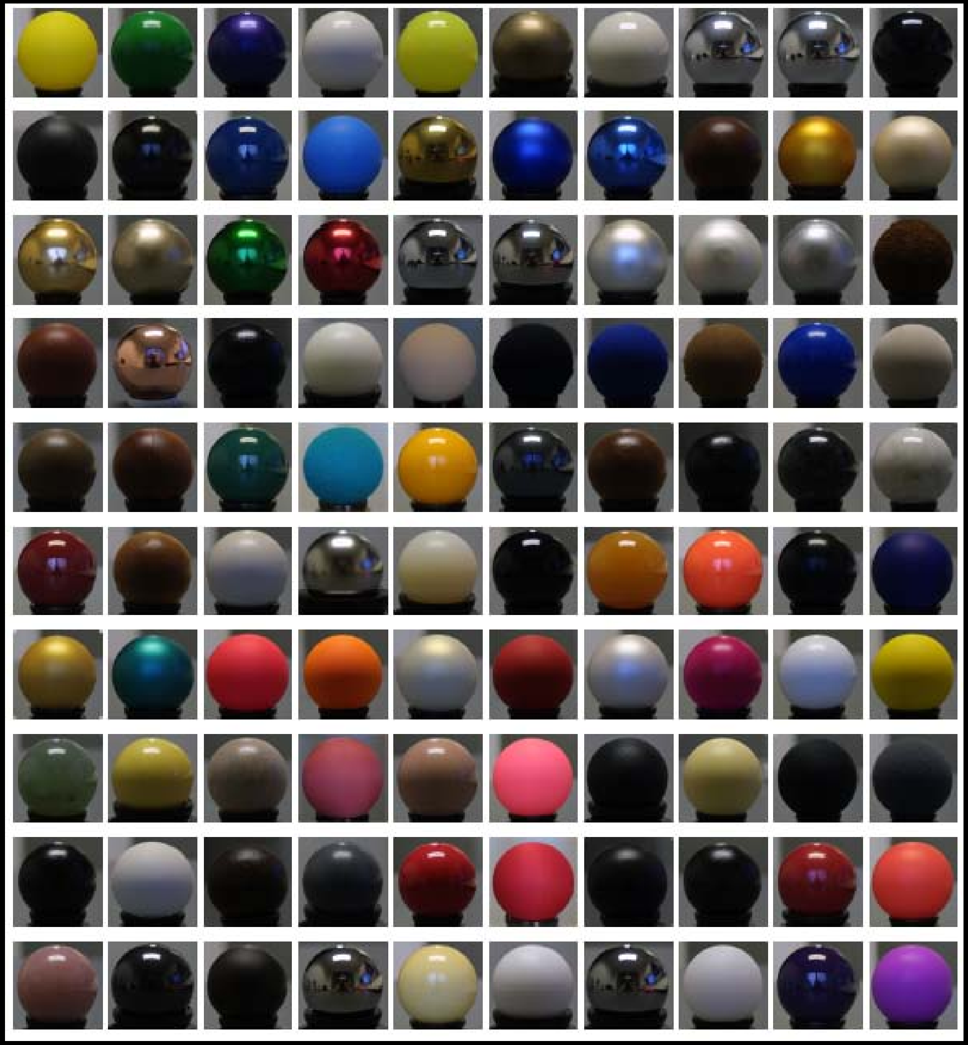
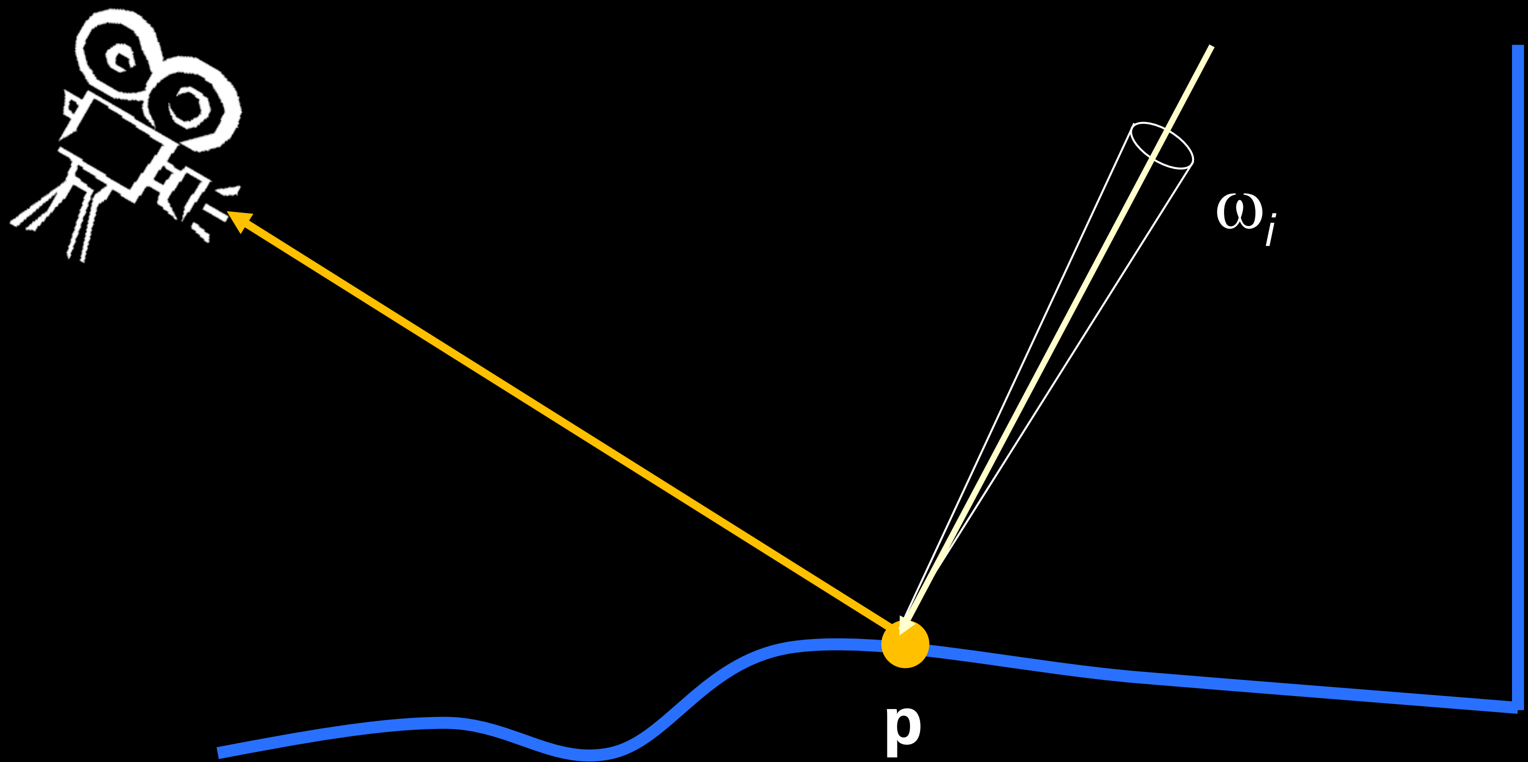


Image courtesy Wojciech Matusik

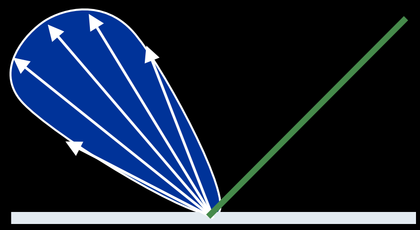
Light reflection – BRDF

- **<u>B</u>**i-directional **<u>R</u>**eflectance **<u>D</u>**istribution **<u>F</u>**unction

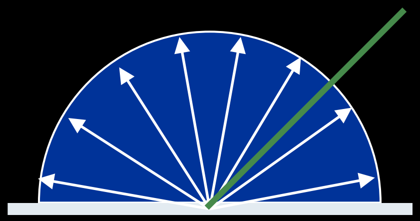
$$f_r(\omega_i, \omega_o) = \frac{dL_o(\omega_o)}{L_i(\omega_i)\cos\theta_i d\omega_i} = \frac{\text{light reflected to } \omega_o}{\text{light coming from } \omega_i}$$



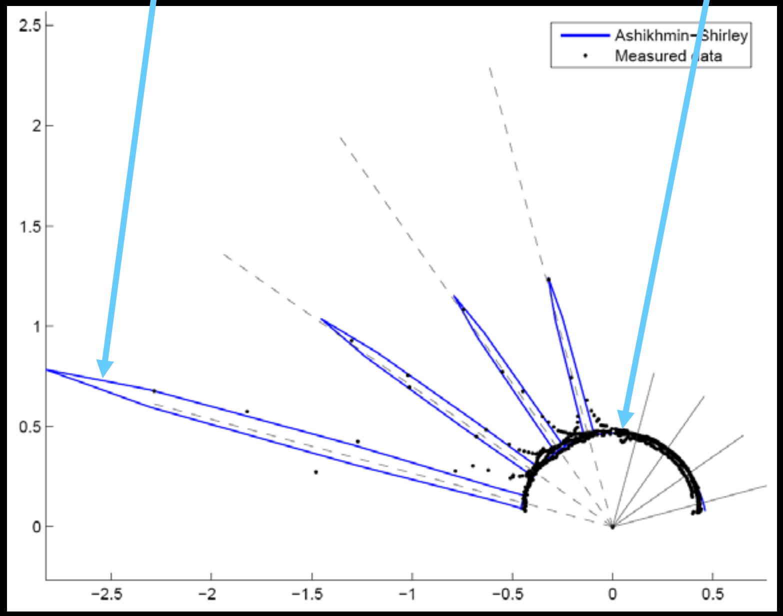
BRDF components

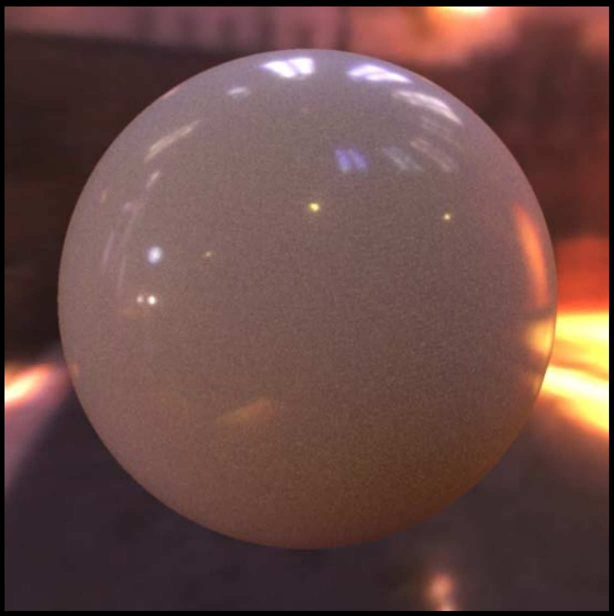


Glossy / specular



Diffuse



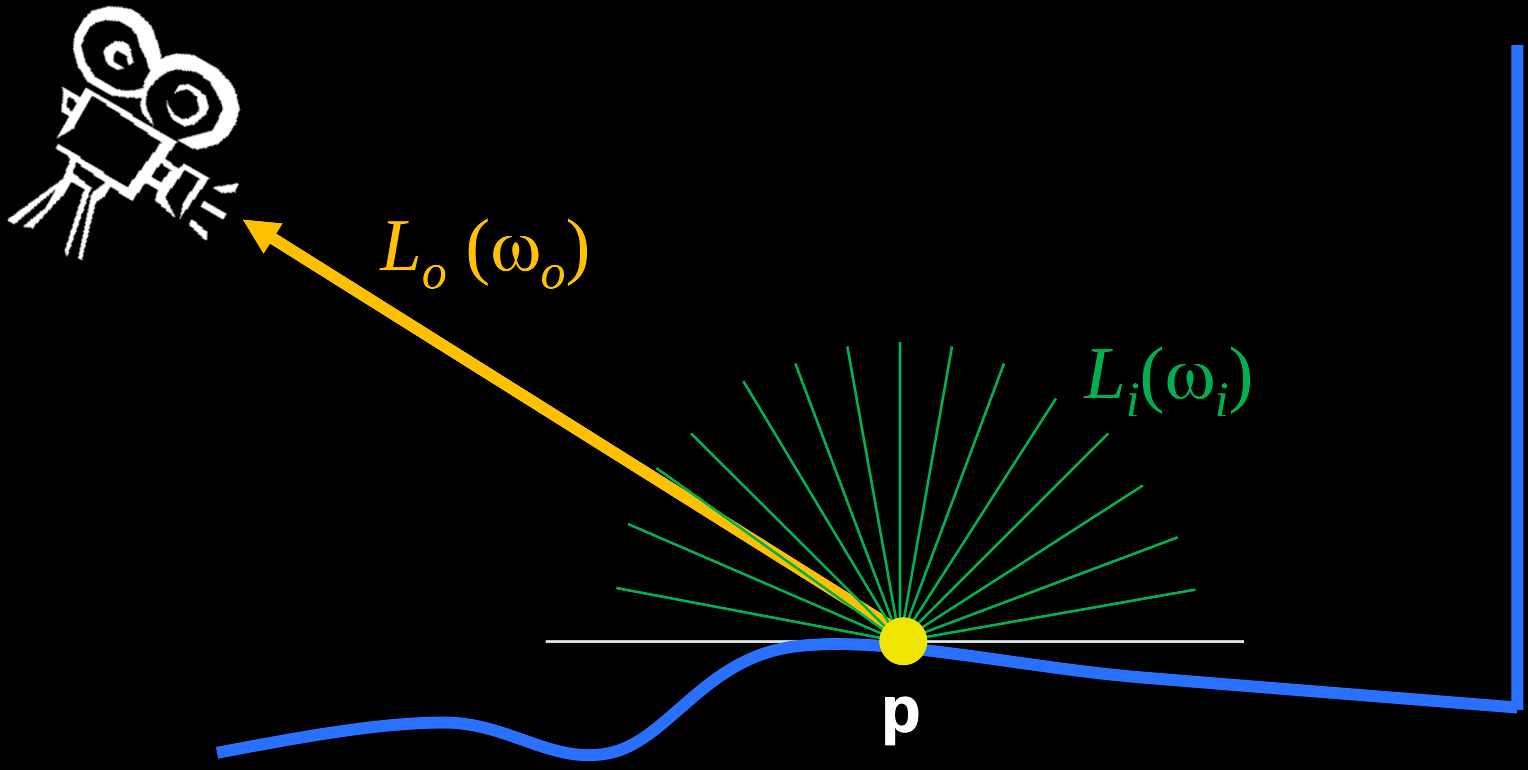


Images by Addy Ngan

Local reflection integral

- Total amount of light reflected to ω_o:

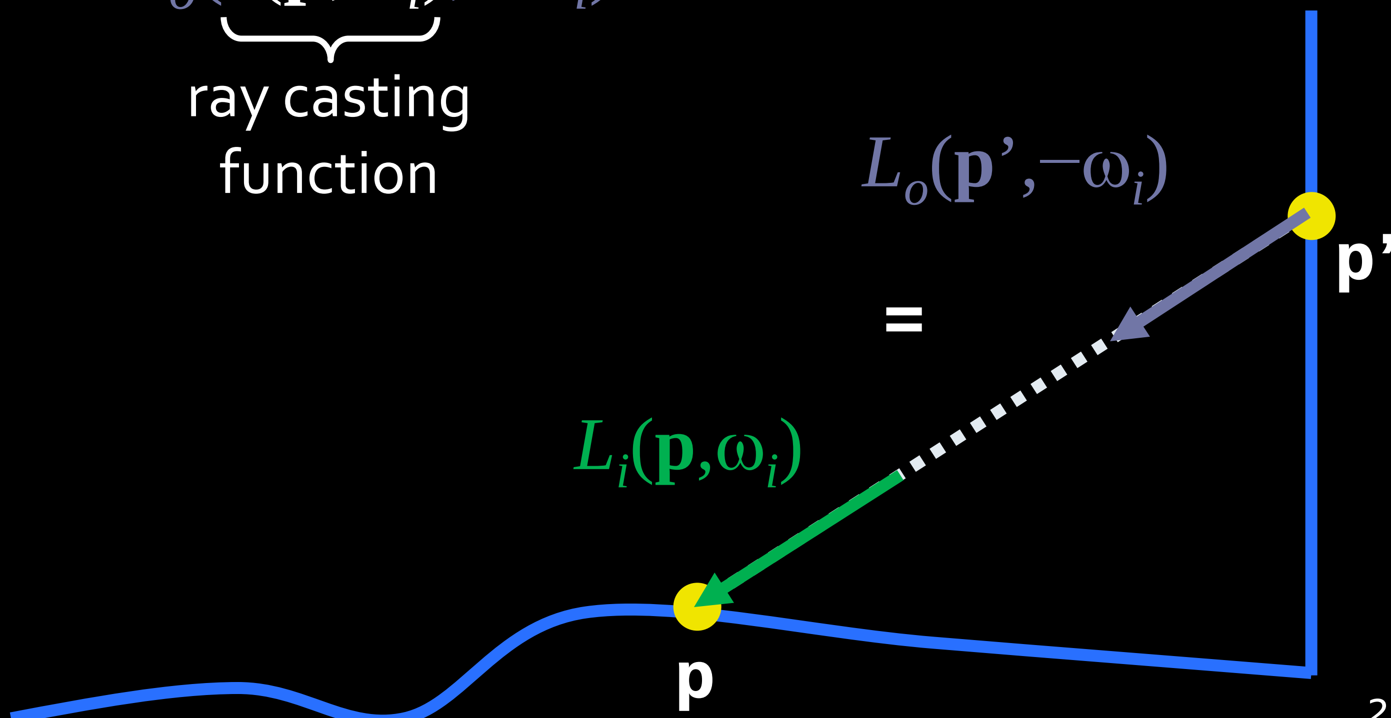
$$L_o(\omega_o) = L_e(\omega_o) + \int L_i(\omega_i)\,\mathrm{BRDF}(\omega_i, \omega_o)\cos\theta_i \, d\omega_i$$



Light transport

- **Q**: How much light is coming from ω_i ?
- **A**: Radiance constant along rays, so:

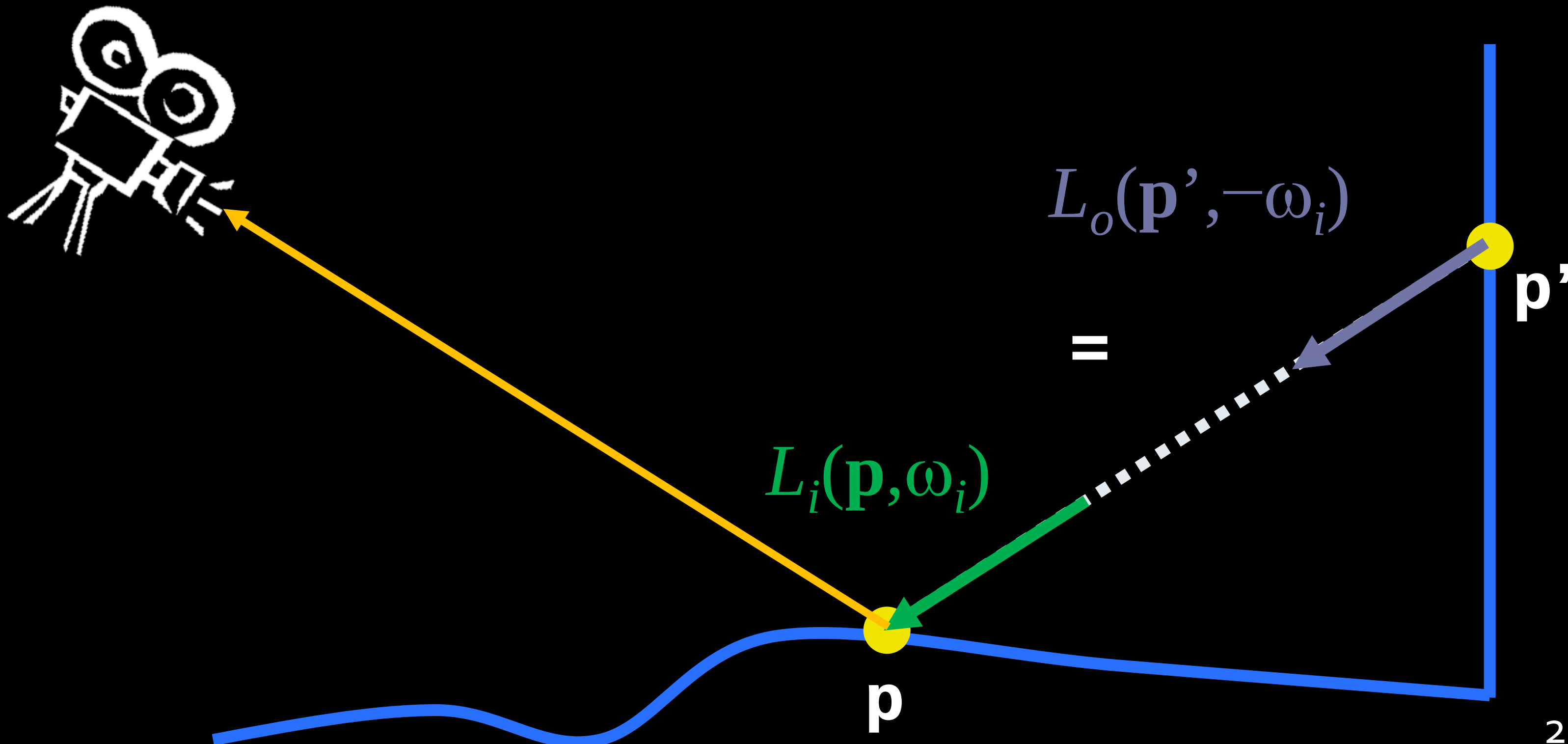
$$
\begin{aligned}
L_i(\mathbf{p},\omega_i) &= L_o(\mathbf{p}',-\omega_i) \\
&= L_o(\underbrace{r(\mathbf{p},\omega_i)}_{\text{ray casting function}},-\omega_i)
\end{aligned}
$$



Rendering equation

$$L_o(\mathbf{p}, \omega_o) = L_e(\omega_o) + \int L_i(\mathbf{p}, \omega_i)\ \mathrm{BRDF}(\mathbf{p}, \omega_i, \omega_o) \cos\theta_i \, \mathrm{d}\omega_i$$

$$L_o(\mathbf{p}, \omega_o) = L_e(\omega_o) + \int L_o(r(\mathbf{p},\omega_i),-\omega_i)\ \mathrm{BRDF}(\mathbf{p}, \omega_i, \omega_o) \cos\theta_i \, \mathrm{d}\omega_i$$



Rendering eqn vs. reflection integral

- Reflection Integral
  - Local light reflection
  - Integral to compute L_o given that we know L_i

- Rendering Equation
  - Condition on light distribution in the scene
  - Integral equation – the unknown, L, on both sides

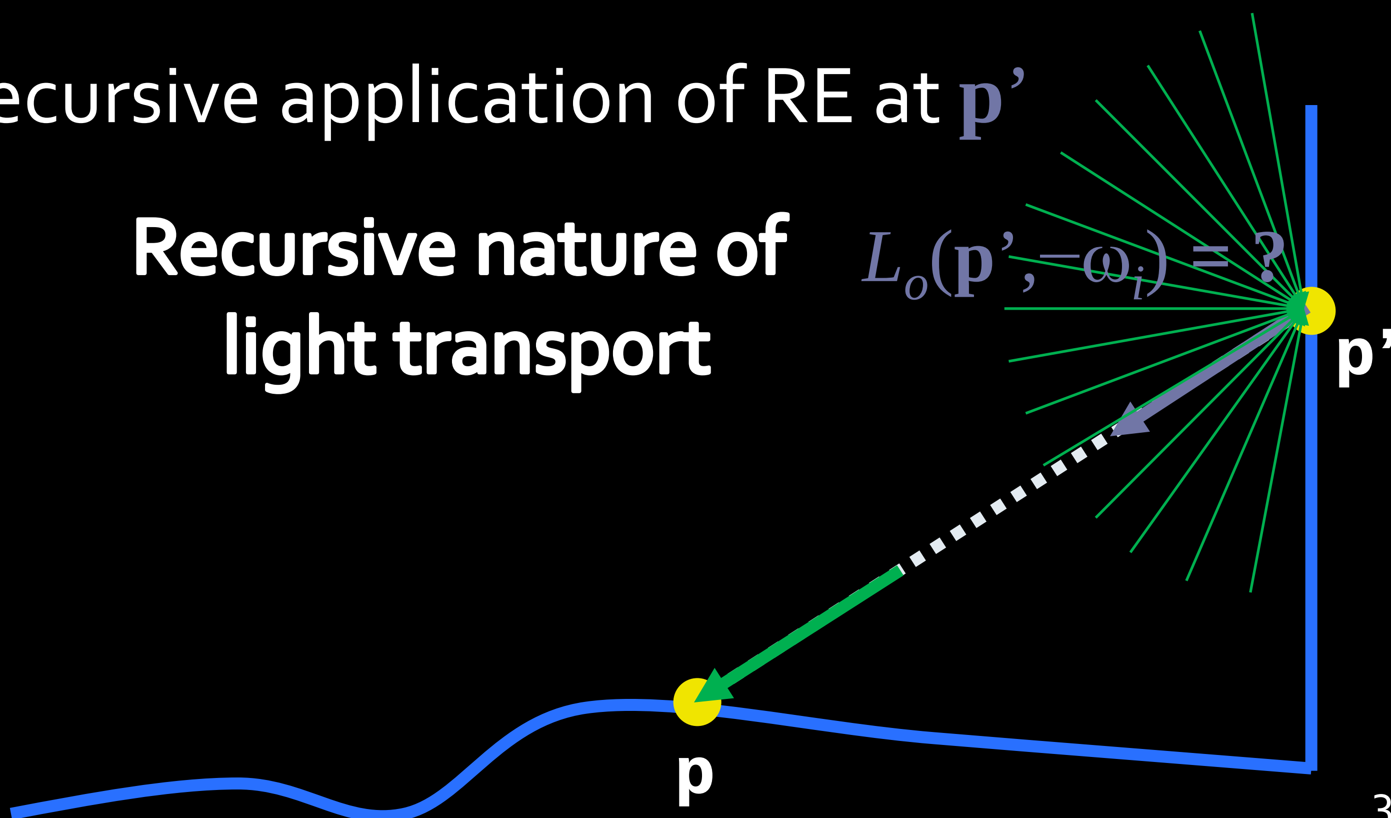
Solving RE – Recursion

$$L_o(\mathbf{p}, \omega_o) = L_e(\omega_o) + \int L_o(r(\mathbf{p},\omega_i),-\omega_i)\ \mathrm{BRDF}(\mathbf{p}, \omega_i, \omega_o) \cos\theta_i \, d\omega_i$$

- **Q**: How much light is reflected from $\mathbf{p}'$?

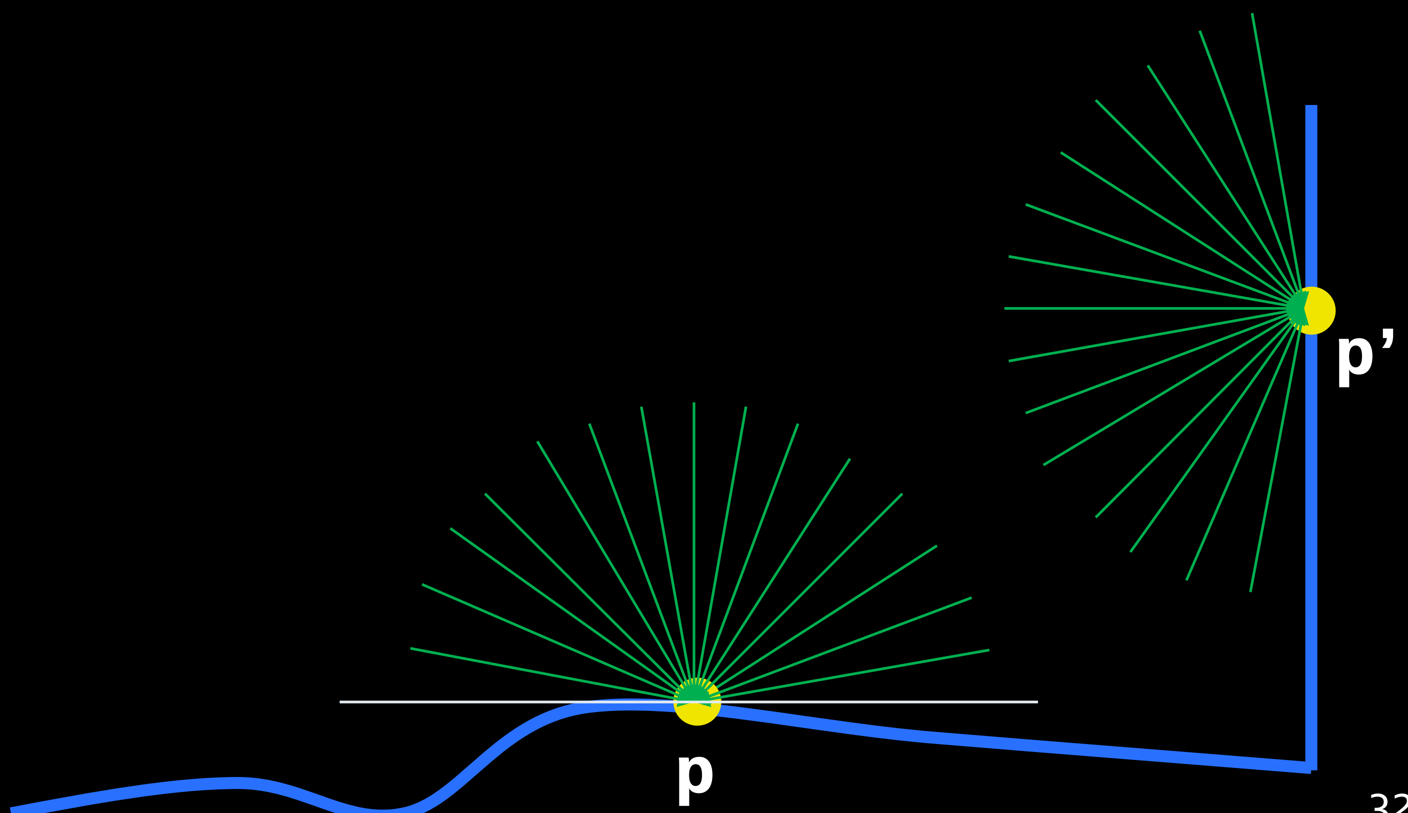
Recursive application of RE at $\mathbf{p}'$

Recursive nature of light transport



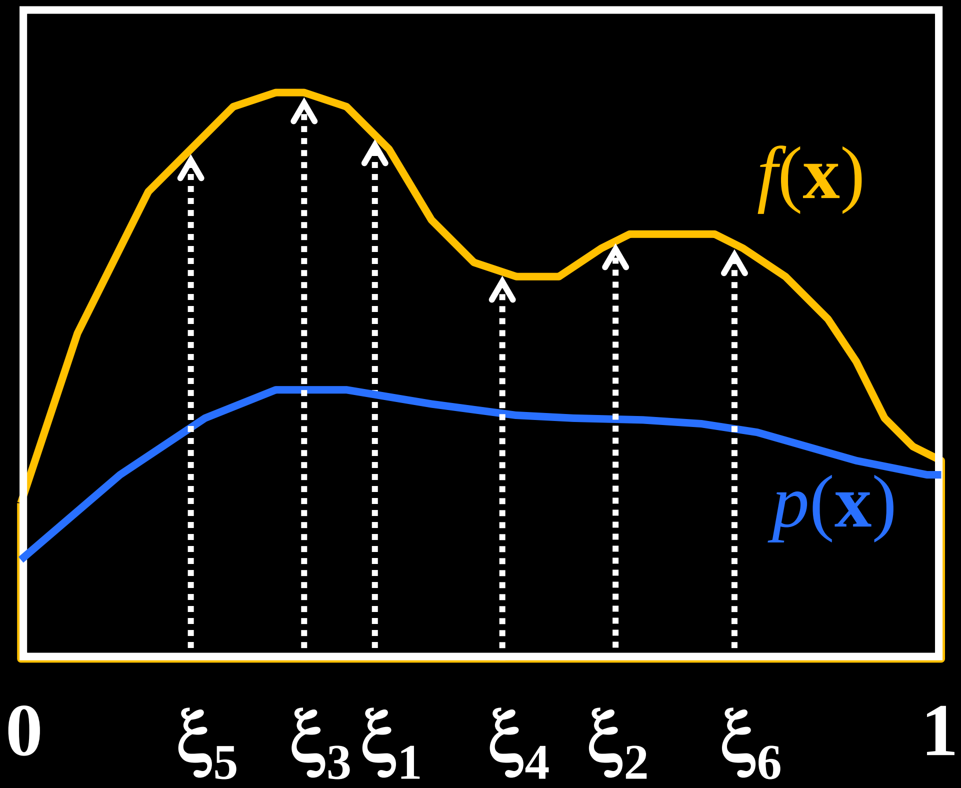
Solving RE – Recursion

- Recursive evaluation of illumination integral at different points (p, p’, …)



Monte Carlo integration

- General approach for numerical estimation of integrals



Integral to evaluate:

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

Monte Carlo estimator:

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

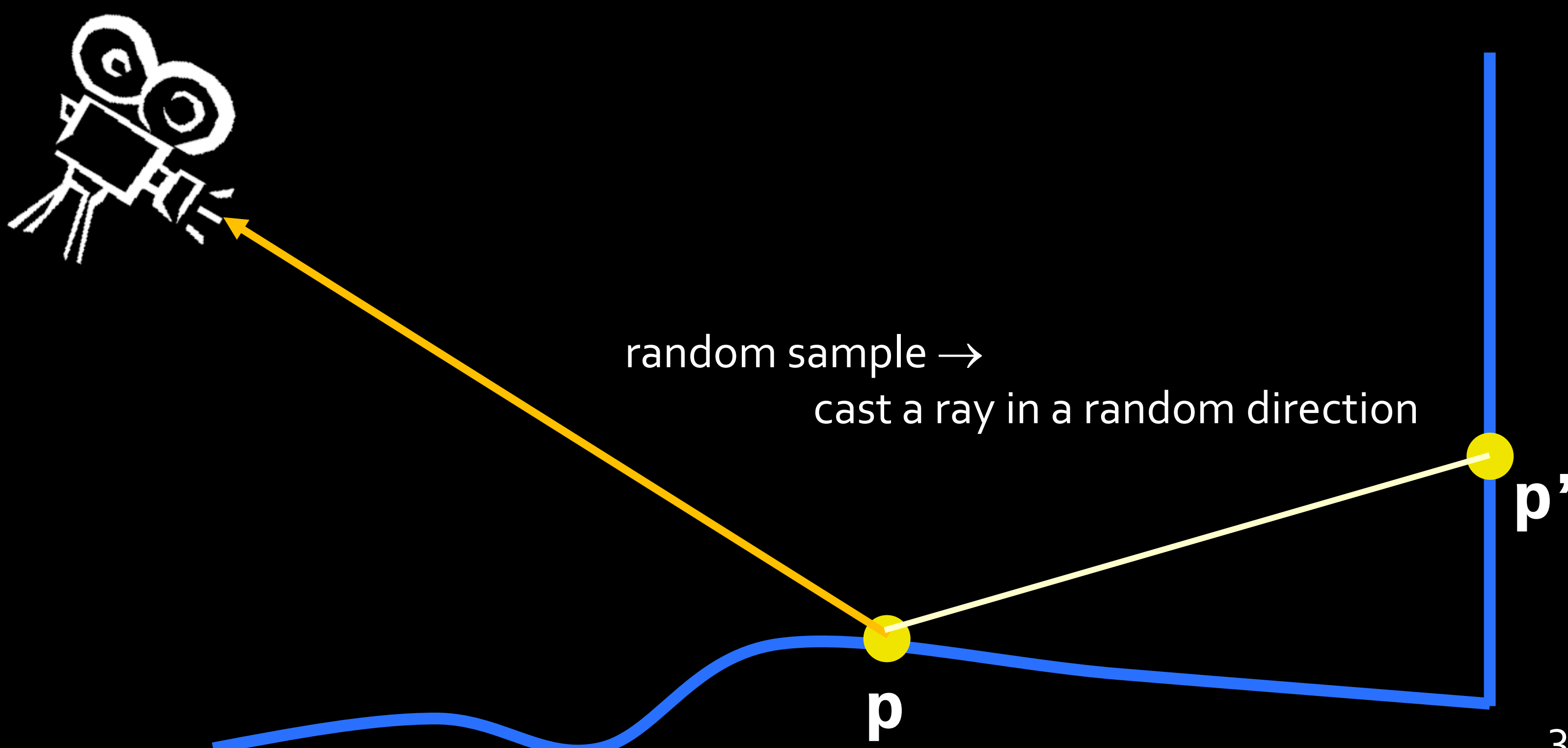
Monte Carlo integration

- Estimator $\langle I \rangle$ gives an unbiased estimate of I:

$$E\left[\langle I \rangle\right] = \frac{1}{N}\sum_{i=1}^{N} E\left[\frac{f(\xi_i)}{p(\xi_i)}\right]$$

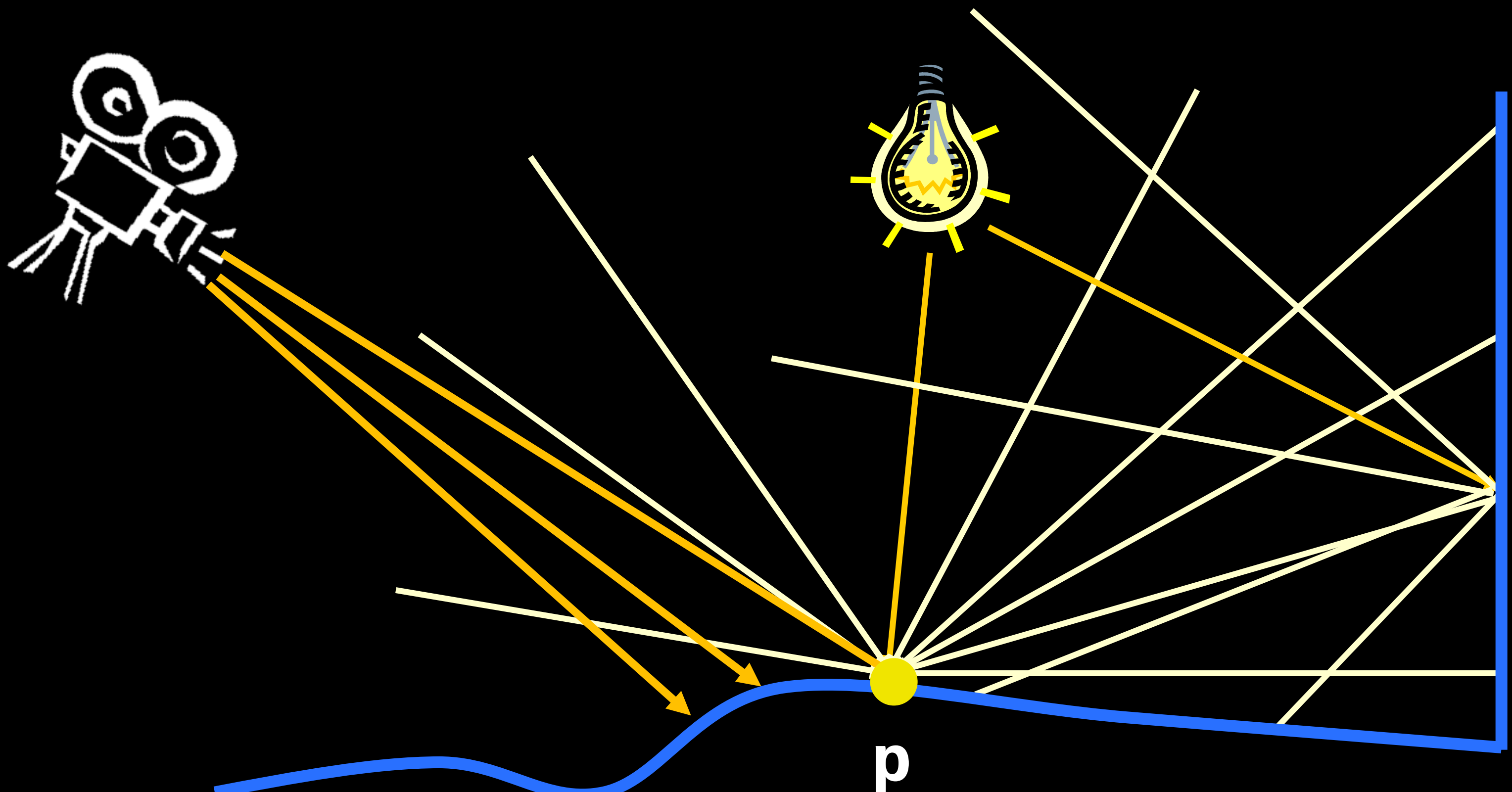
Applying MC to rendering

$$L_o(\mathbf{p}, \omega_o) = L_e(\omega_o) + \int \underbrace{L_o(r(\mathbf{p},\omega_i),-\omega_i)\ \mathrm{BRDF}(\mathbf{p}, \omega_i, \omega_o)\ \cos\theta_i\ d\omega_i}_{\text{integrand } f(\omega_i)}$$



Distribution Ray Tracing

- Recursive nature – ray tracing
- Direct illumination separated from indirect



Distribution Ray Tracing

- Cook '84
- Breakthrough at the time

- Problem: Terribly slow!
  - Number of rays exponential with recursion depth
- Solution: Irradiance caching / path tracing / …

Unbiased vs. consistent estimator

- Unbiased estimator
  - No systematic error, only variance
- Consistent estimator
  - May has systematic error
  - Converges to the correct result

Unbiased / consistent GI algorithms

- Path Tracing – *unbiased*
- Light Tracing – *unbiased*
- Bi-directional Path Tracing – *unbiased*
- Metropolis Light Transport – *unbiased*
- Photon Mapping – *biased, not consistent*
- Progressive photon mapping – *biased, consistent*
- Irradiance caching – *biased, not consistent*
- Radiance caching – *biased, not consistent*

GI Algorithms: Pros and Cons

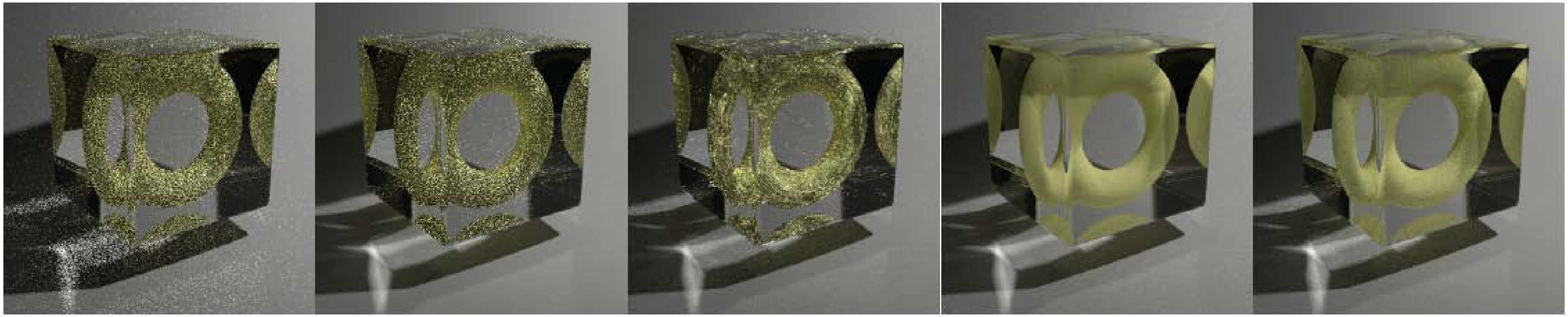


Figure 7: *Torus embedded in a glass cube. The reference image on the far right have been rendered using path tracing with 51500 samples per pixel. The Monte Carlo ray tracing methods fail to capture the lighting within the glass cube, while progressive photon mapping provides a smooth result using the same rendering time.*

Image credit: Toshiya Hachisuka

GI Algorithms: Pros and Cons

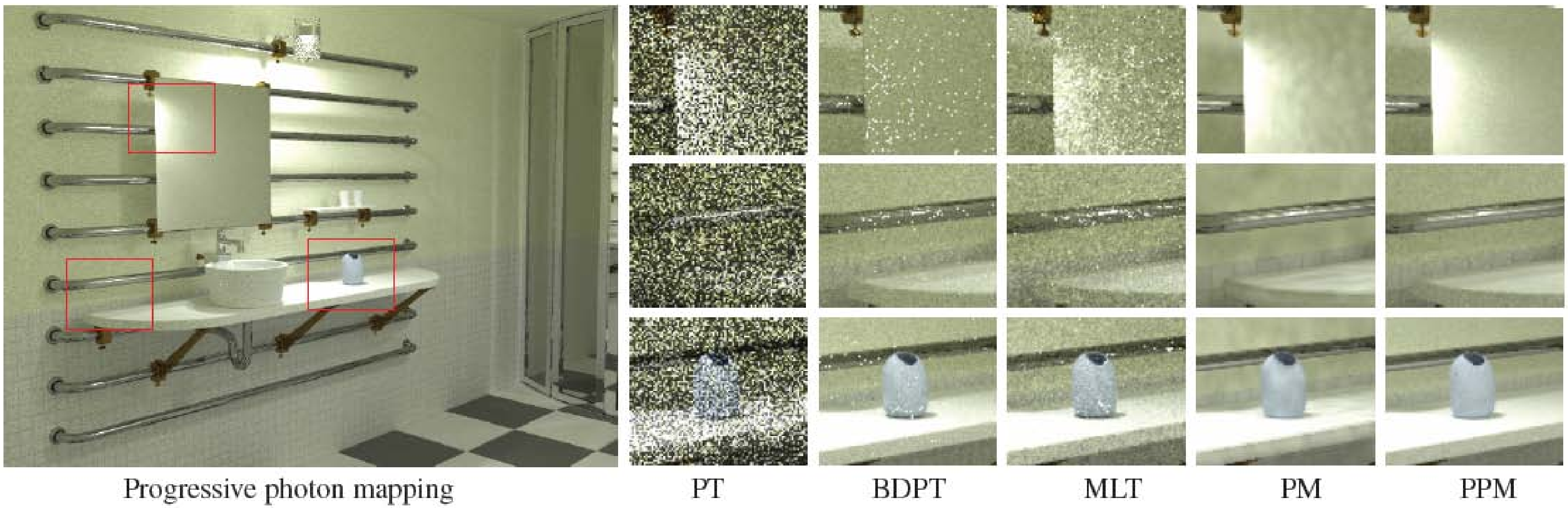


Figure 8: *Lighting simulation in a bathroom. The scene is illuminated by a small lighting fixture consisting of a light source embedded in glass. The illumination in the mirror cannot be resolved using Monte Carlo ray tracing. Photon mapping with 20 million photons results in a noisy and blurry image, while progressive photon mapping is able to resolve the details in the mirror and in the illumination without noise.*

Image credit: Toshiya Hachisuka

Unbiased vs. consistent GI algorithm

- Practice
  - Prefer less noise at the cost of bias
  - Systematic error is more acceptable than noise if “looks good” is our only measure of image quality

There’s more to realistic rendering

- We’ve seen
  - GI, i.e. Light transport simulation
- There’s also
  - Emission modeling
    - How do various objects emit light?
  - Appearance modeling
    - What does light do after it hits a specific surface?
  - Tone mapping
    - Radiance remapping for display