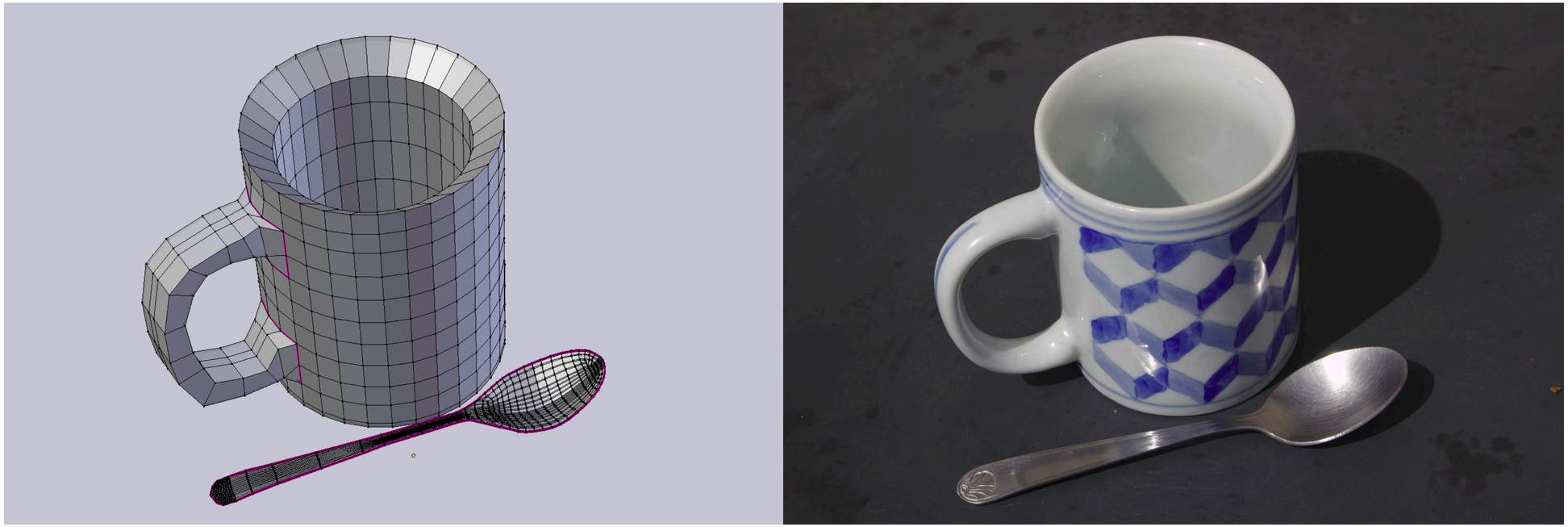
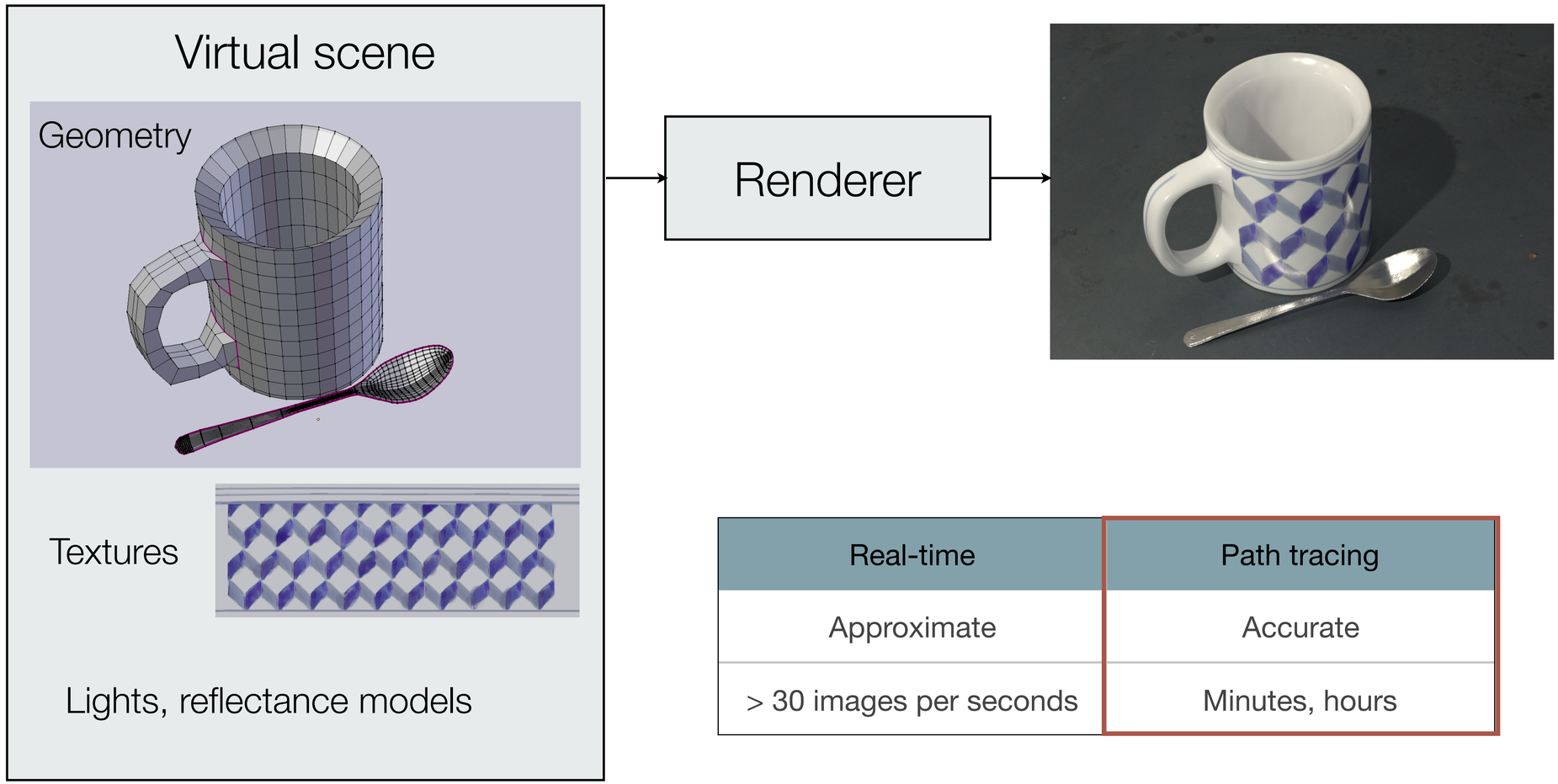


Differentiable path tracing for optimising virtual scenes

Guillaume Loubet Inria/EPFL PostDoc Fellowship
Wenzel Jakob EPFL
Nicolas Holzschuch Inria Grenoble

Photo-realistic rendering



Real-time	Path tracing
Approximate	Accurate
> 30 images per seconds	Minutes, hours

Photo-realistic rendering with Path Tracing

Dawn of the Planet of the Apes, *Weta*, 2014

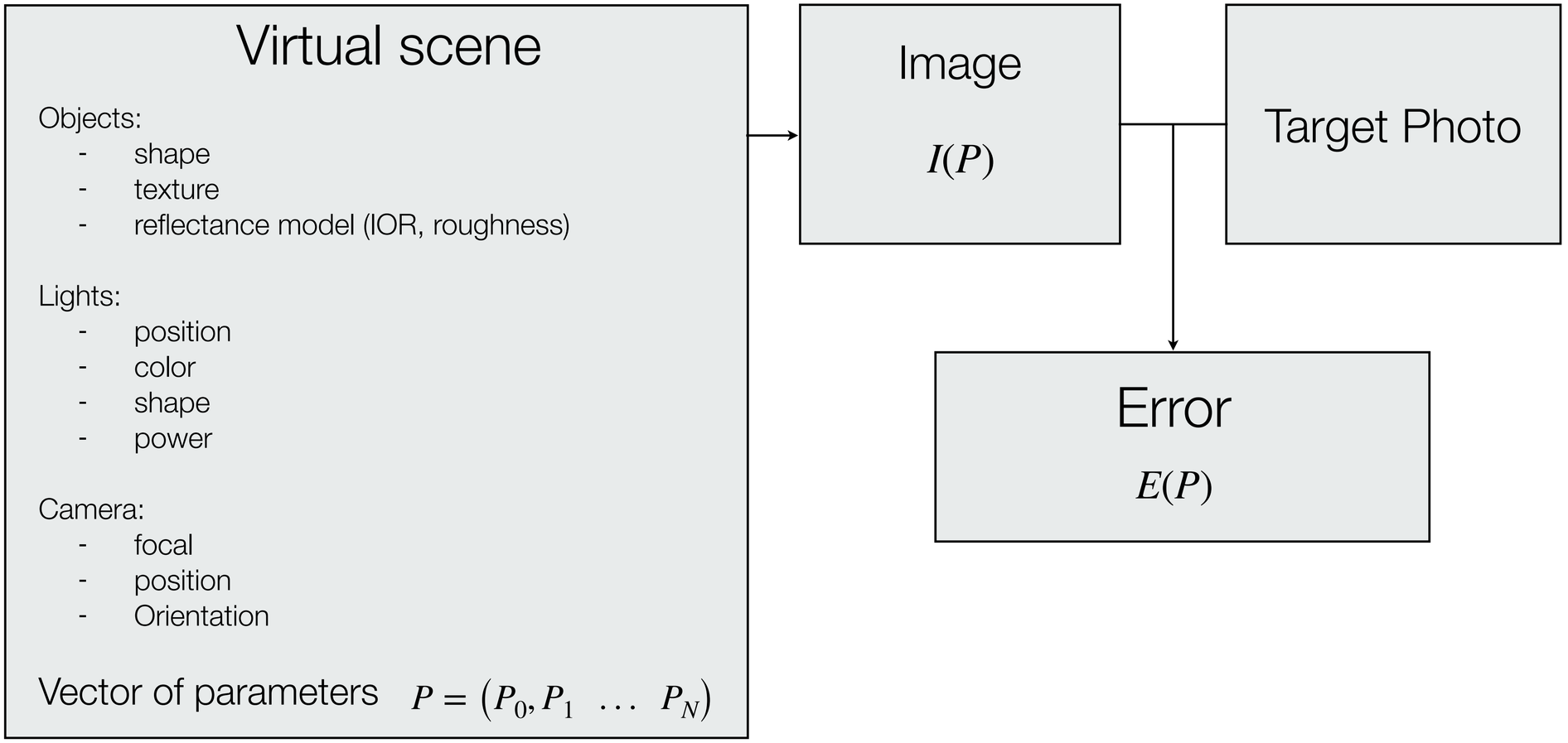
Creating realistic virtual scenes

Photo

Rendered with path tracing

- Problem: hard to create realistic virtual scenes, too many parameters
- Can we optimise a virtual scene with photos?

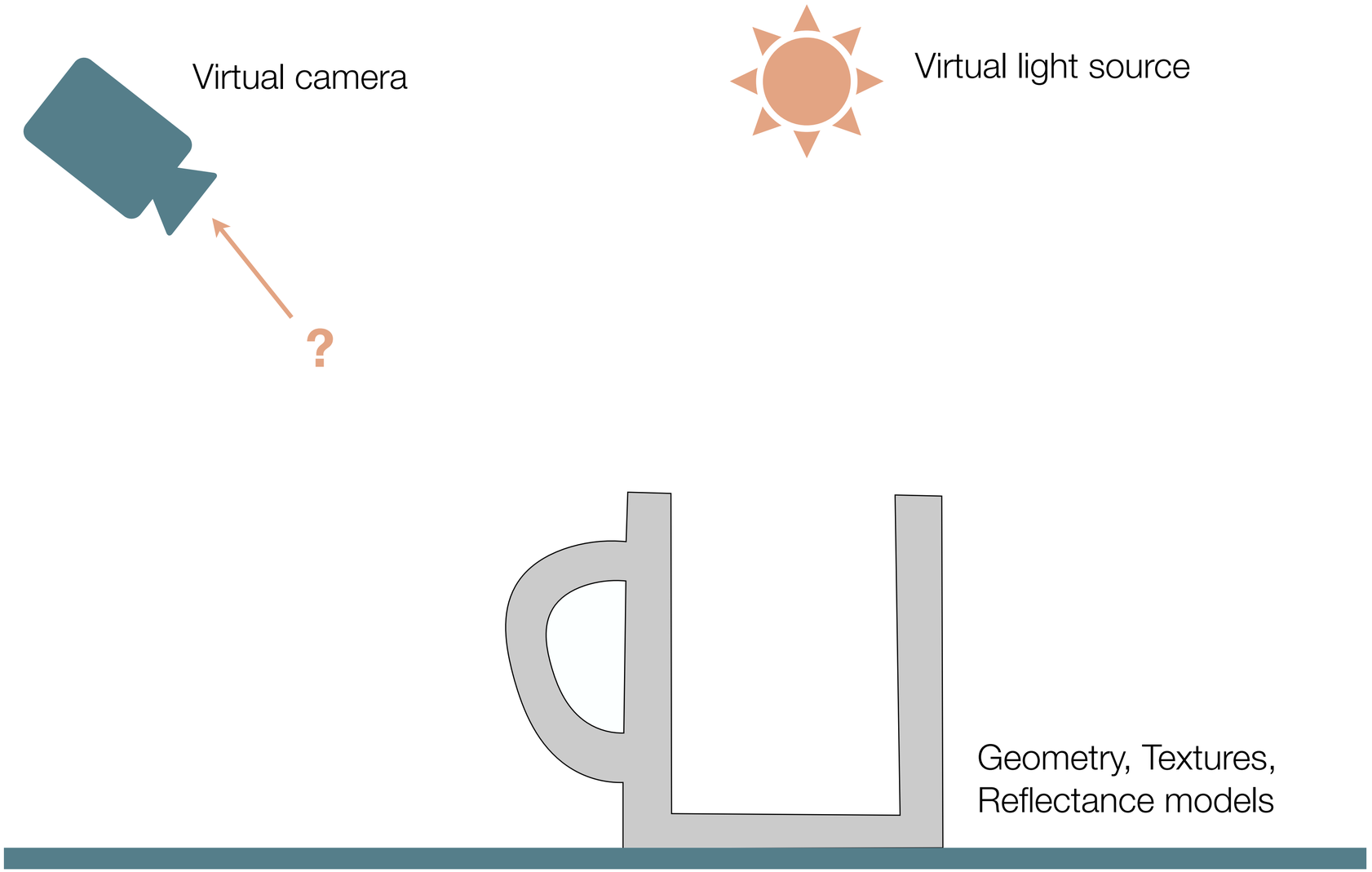
Optimising virtual scenes



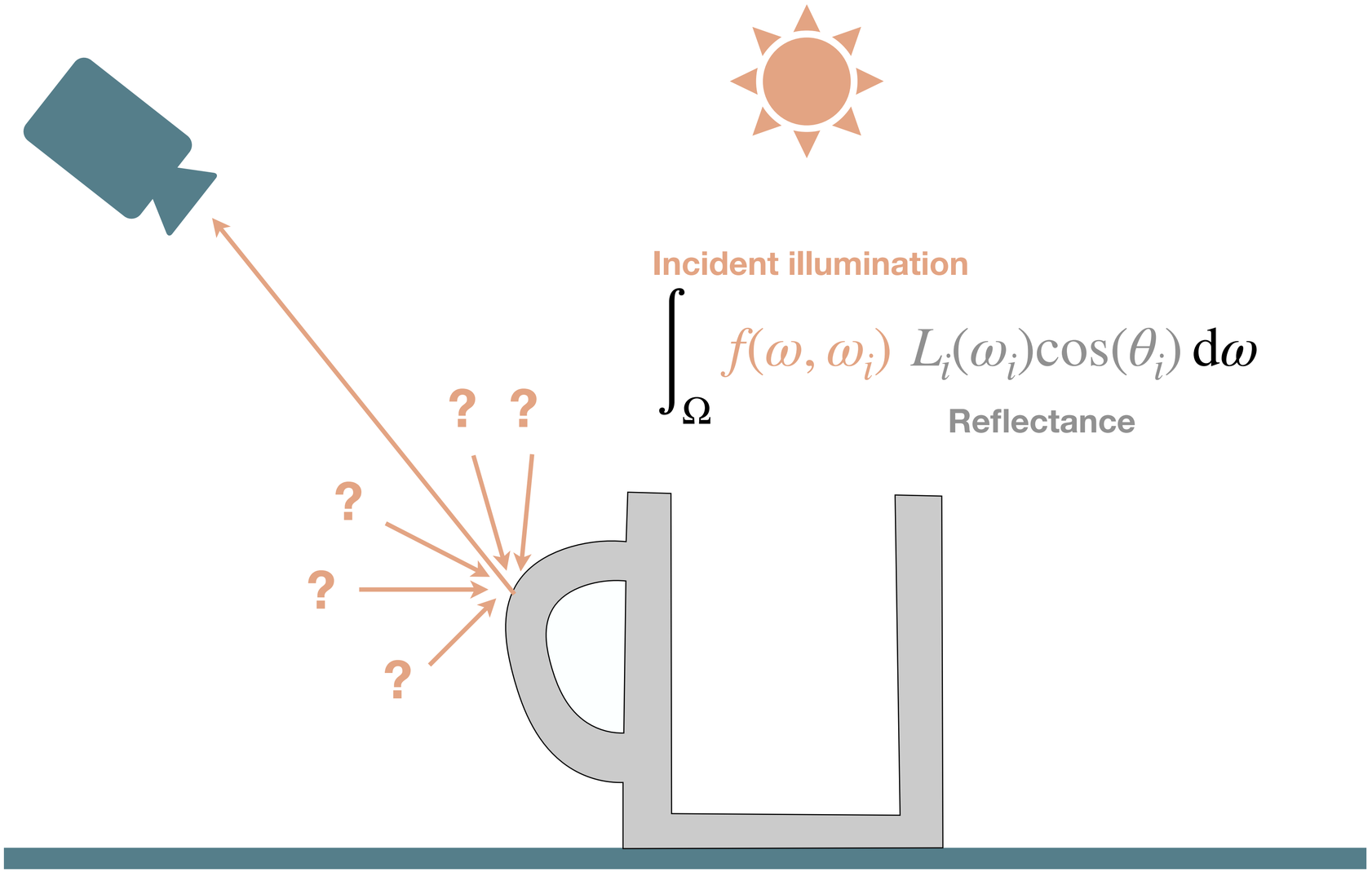
Gradient descent: we need all the $\frac{\partial I}{\partial P_i}$

Difficult to compute
Our job

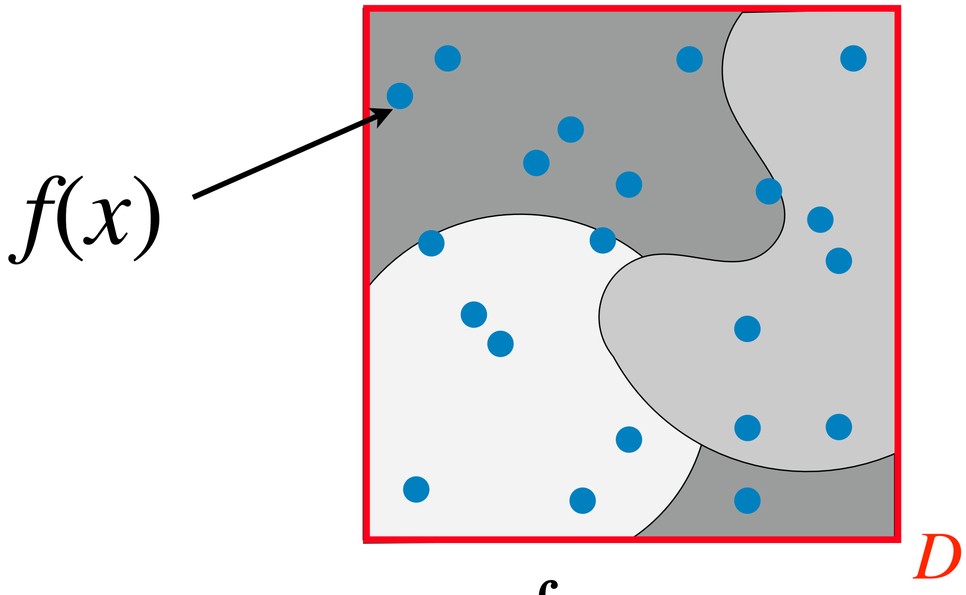
Rendering with Path Tracing



Rendering with Path Tracing



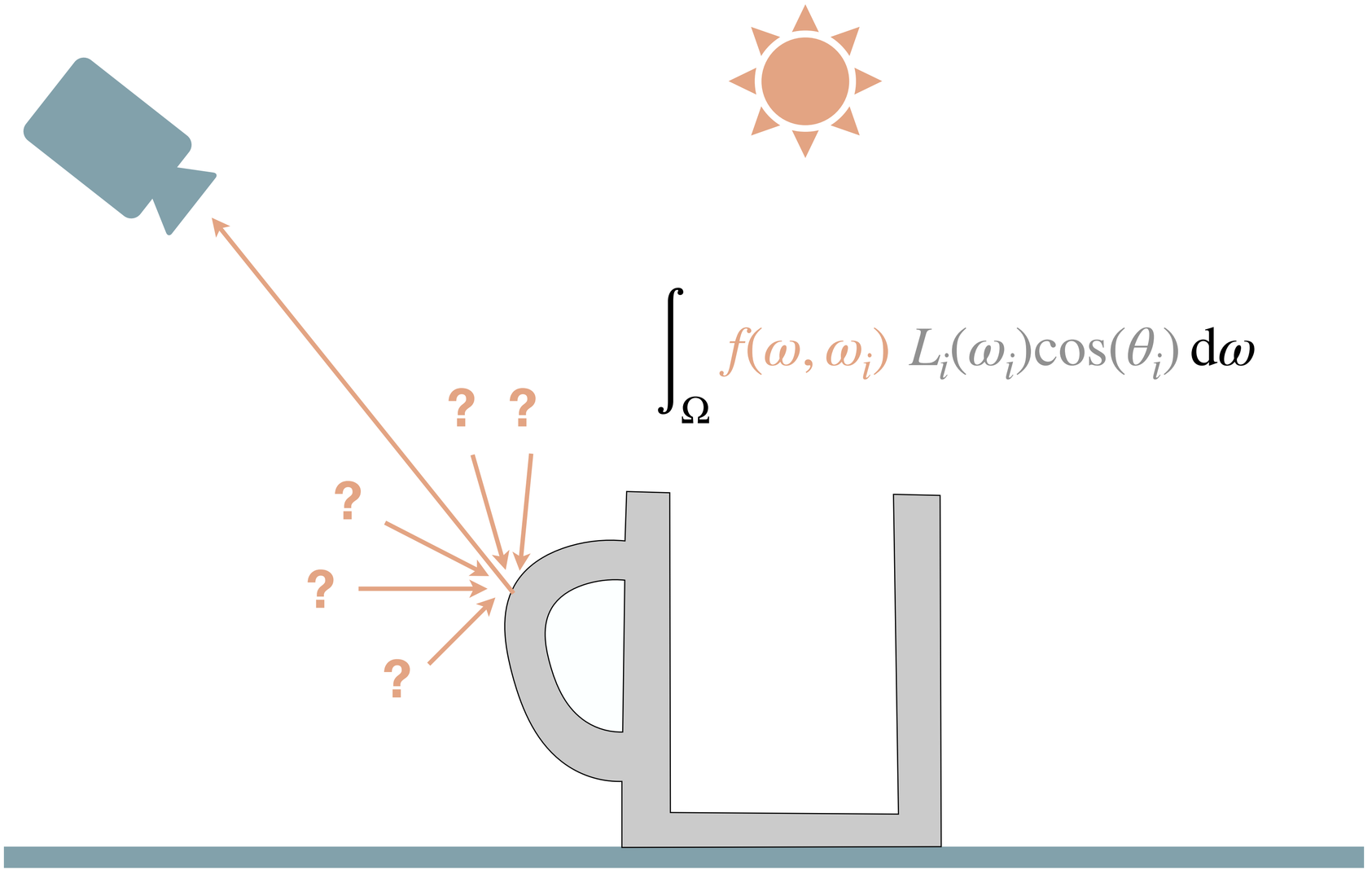
Monte Carlo Integration



$$I = \int_D f(x)\,\mathrm{d}x = ?$$

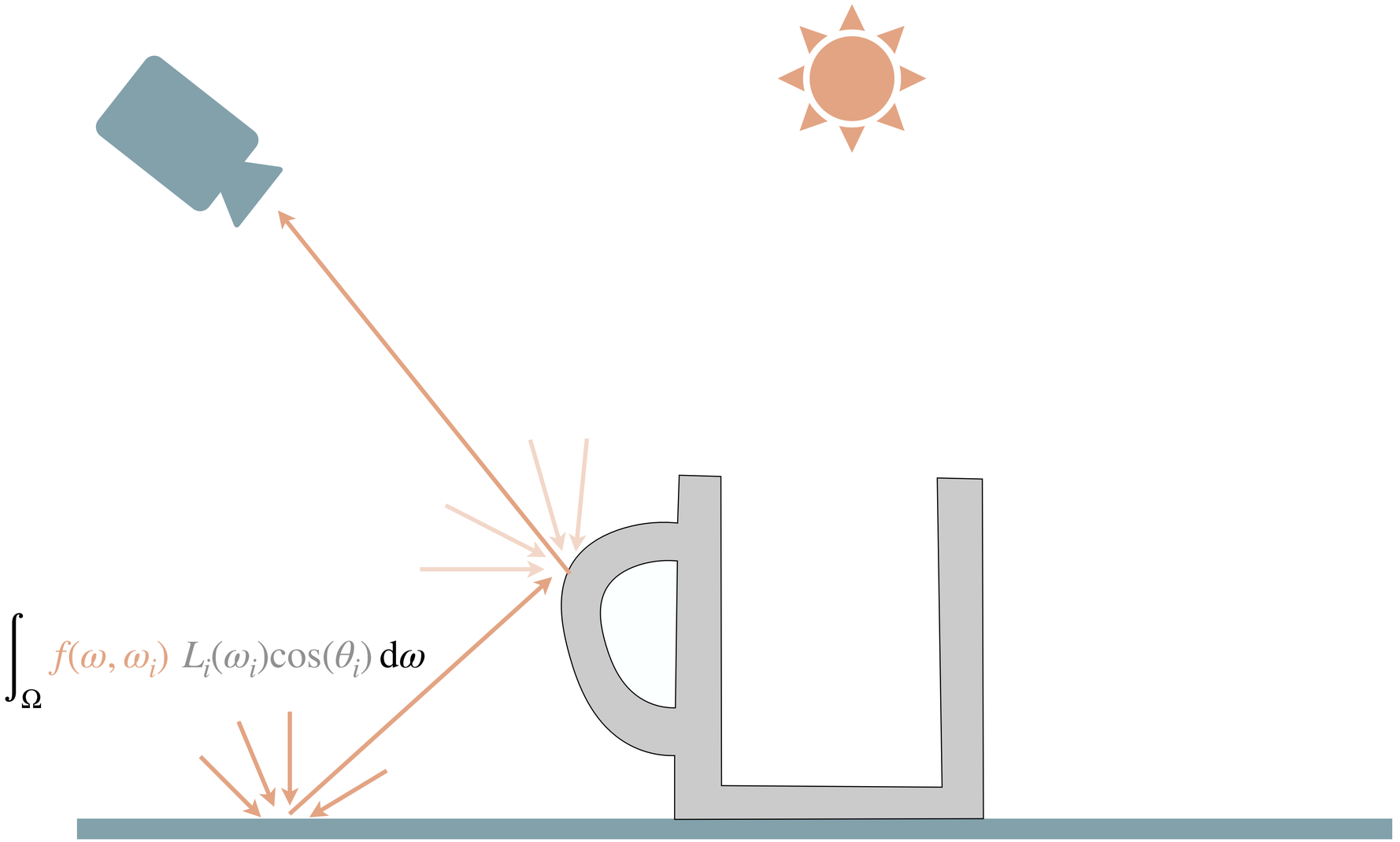
Monte Carlo Integration: $$I = \int_D f(x)\,\mathrm{d}x \approx \frac{C}{N}\sum_{i=1}^{N} f(x_i)$$

Rendering with Path Tracing

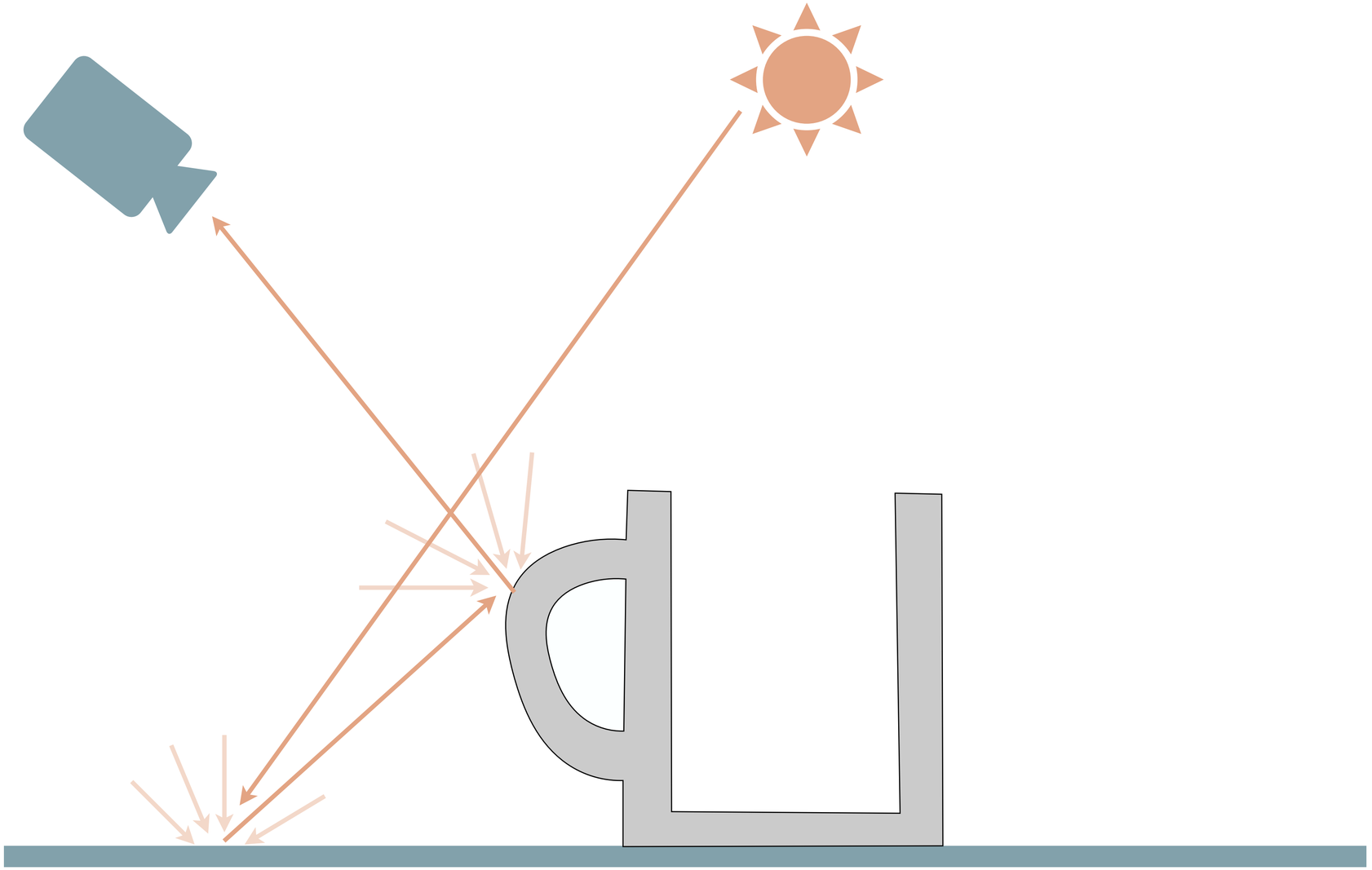


$$\int_{\Omega} f(\omega, \omega_i)\ L_i(\omega_i)\cos(\theta_i)\,\mathrm{d}\omega$$

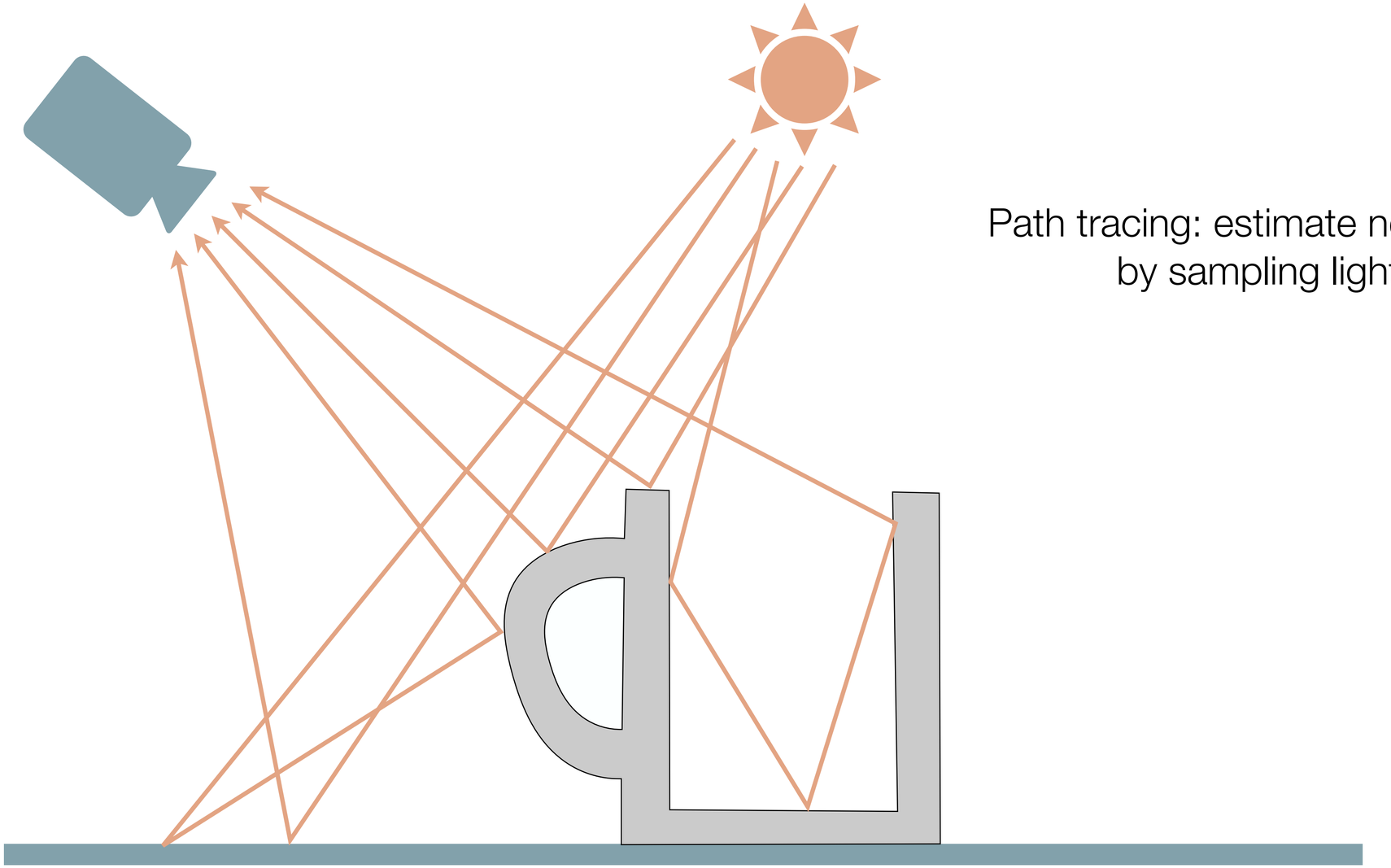
Rendering with Path Tracing



Rendering with Path Tracing



Rendering with Path Tracing



Path tracing: estimate nested integrals
by sampling light paths

Differentiable rendering

Monte Carlo samples

$$I = \int f(x)\,\mathrm{d}x \approx \frac{C}{N}\sum f(x_i)$$

$$\frac{\partial I}{\partial P_i} = \frac{\partial}{\partial P_i}\int f(x)\,\mathrm{d}x \approx \frac{C}{N}\sum \frac{\partial f(x_i)}{\partial P_i}$$

Differentiable rendering

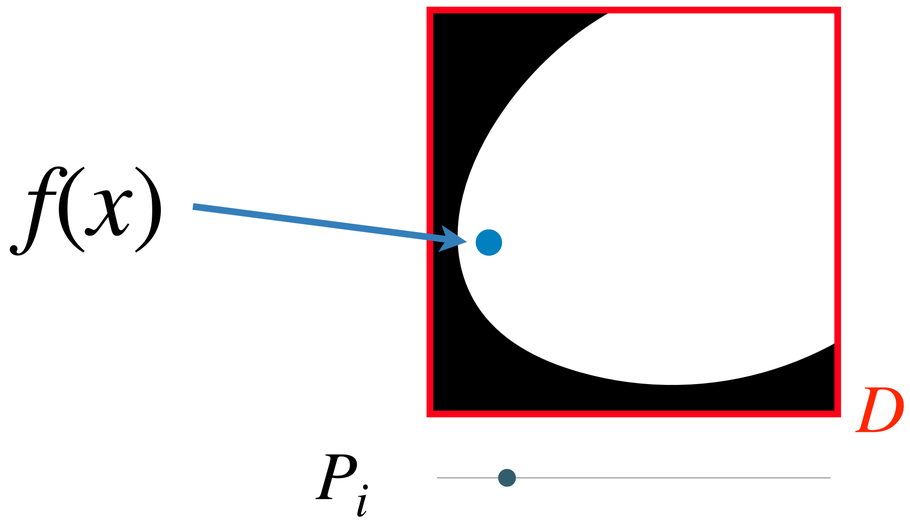
Monte Carlo samples

$$I = \int f(x)\,\mathrm{d}x \approx \frac{C}{N}\sum f(x_i)$$

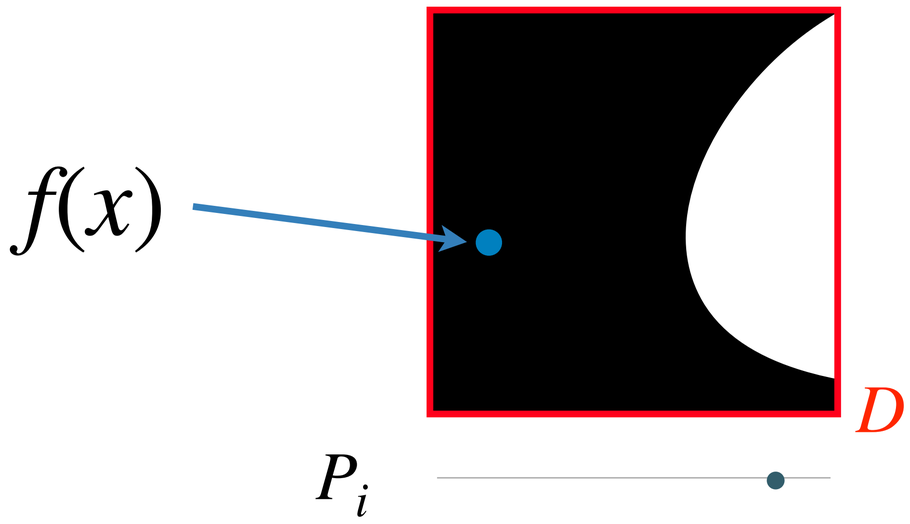
$$\frac{\partial I}{\partial P_i} = \frac{\partial}{\partial P_i}\int f(x)\,\mathrm{d}x \approx \frac{C}{N}\sum \frac{\partial f(x_i)}{\partial P_i}$$

Problem: only if f is differentiable *wrt* P_i

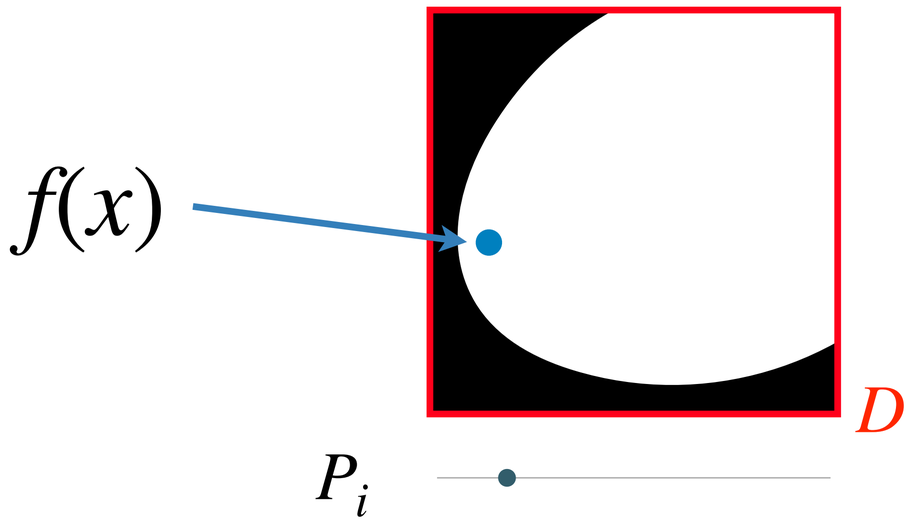
Differentiable rendering



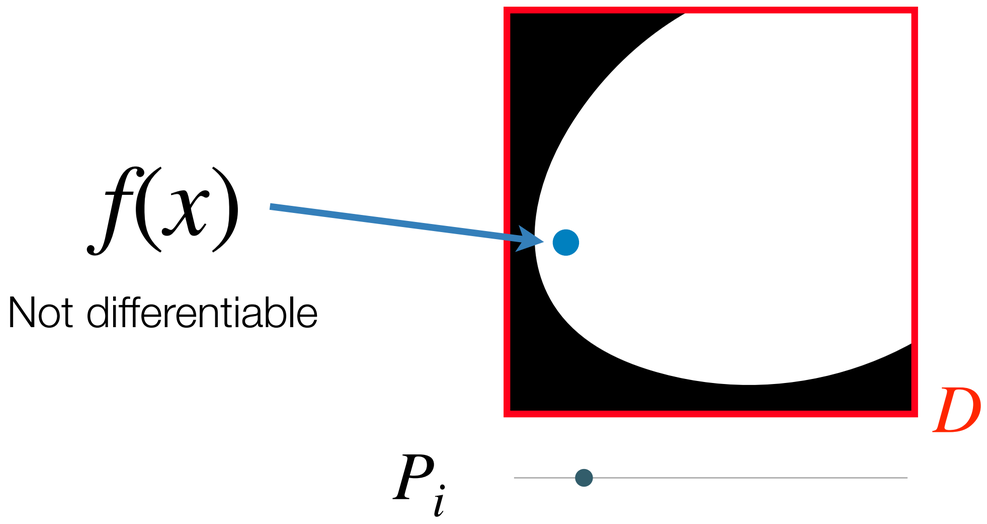
Differentiable rendering



Differentiable rendering



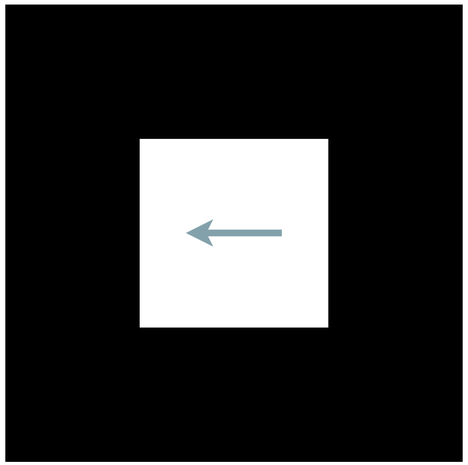
Differentiable rendering



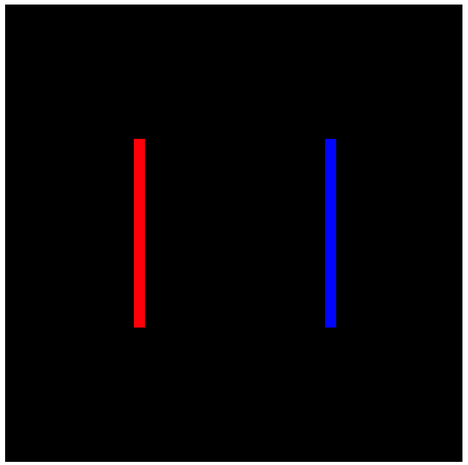
$\int_D f(x)\,\mathrm{d}x$ is differentiable, but cannot differentiate Monte Carlo estimator

$$\frac{\partial I}{\partial P_i} = \frac{\partial}{\partial P_i}\int f(x)\,\mathrm{d}x \approx \frac{C}{N}\sum \frac{\partial f(x_i)}{\partial P_i}$$

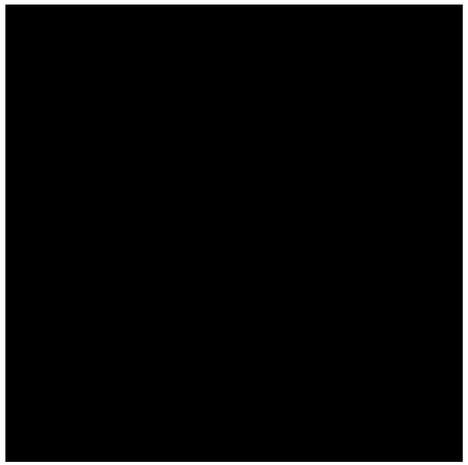
Examples



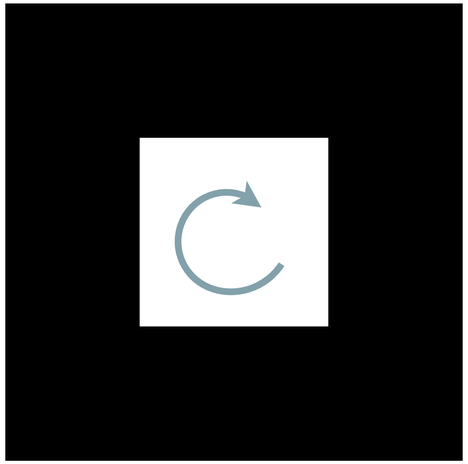
Scene



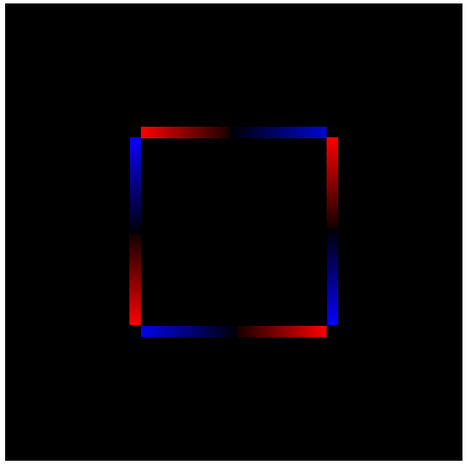
Expected



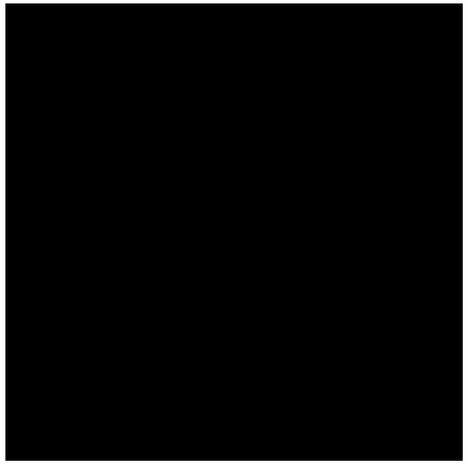
Result (if ignoring discontinuities)



Scene



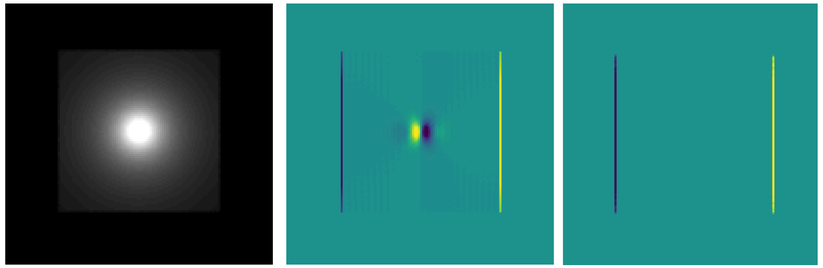
Expected



Result (if ignoring discontinuities)

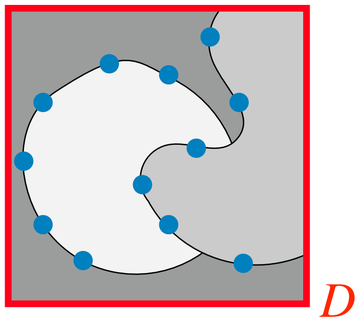
Previous work

- Approx differentiable renderers *[Loper 2014] [Rhodin 2015] [Kato 2018]*

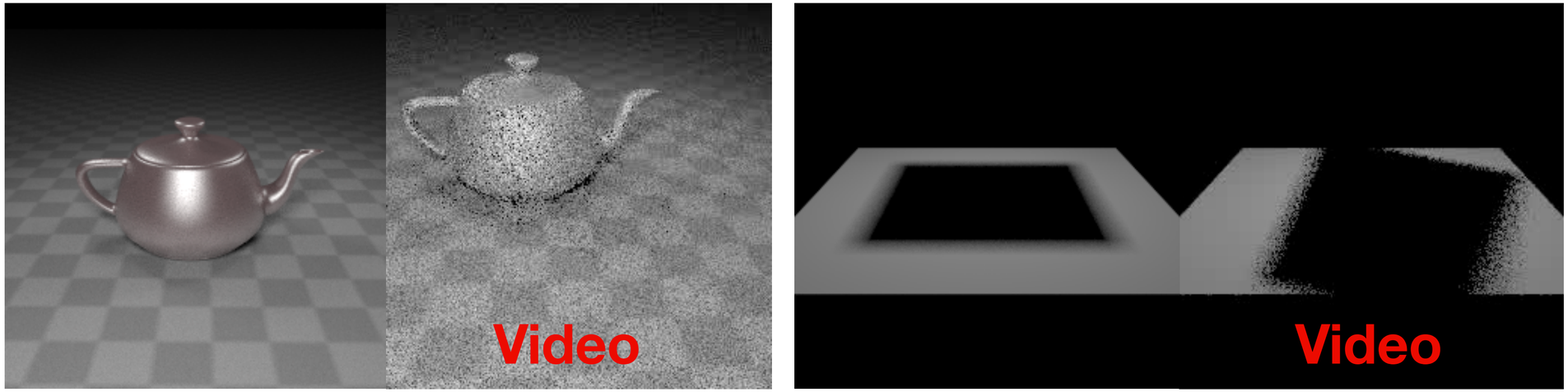


Scene OpenDR Expected

- Edge sampling *[Li 2018]*: first accurate diff. renderer !



Previous work: Edge Sampling [Li 2018]



Target Optimisation Target Optimisation

- Slow, hard to sample important discontinuities
- We need faster approx. gradients

Some slides have been removed here (ongoing research)

Takeaways

- Accurate differentiable renderers are useful:
  - Shape optimisation from photos
  - Get reflectance parameters from photos
- Difficult because of:
  - Complexity of physically-based rendering algorithms
  - Discontinuities in integrals, monte Carlo sampling
- Idea: new differentiable Monte Carlo estimator