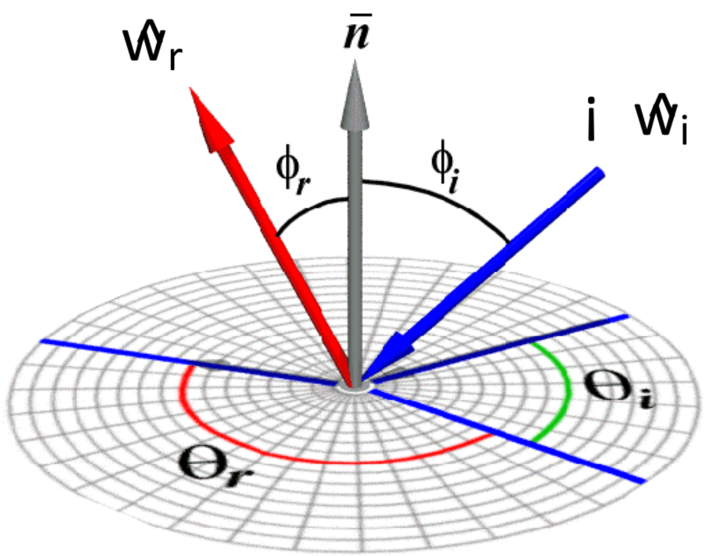
Global Illumination: Path Tracing and Radiosity

Recall: The Rendering Equation

$$L_{\text{out}}(\theta_r, \phi_r) = \int_{\theta_i} \int_{\phi_i} f_r(\theta_r, \phi_r, \theta_i, \phi_i) L_{\text{in}}(\theta_i, \phi_i) \cos\theta_i$$

$$L_{\text{out}}(\hat{w}_r) = \int_{\hat{w}_i \in \text{hemisphere}} f_r(\hat{w}_r, \hat{w}_i)\ L_{\text{in}}(\hat{w}_i)\ \hat{w}_i \cdot \hat{n}$$



BRDF

"Bidirectional Reflectance Distribution Function" (encodes material)

Recall: The Rendering Equation

Diffuse shading, reflection, and refraction all **special cases** of a simple BRDF

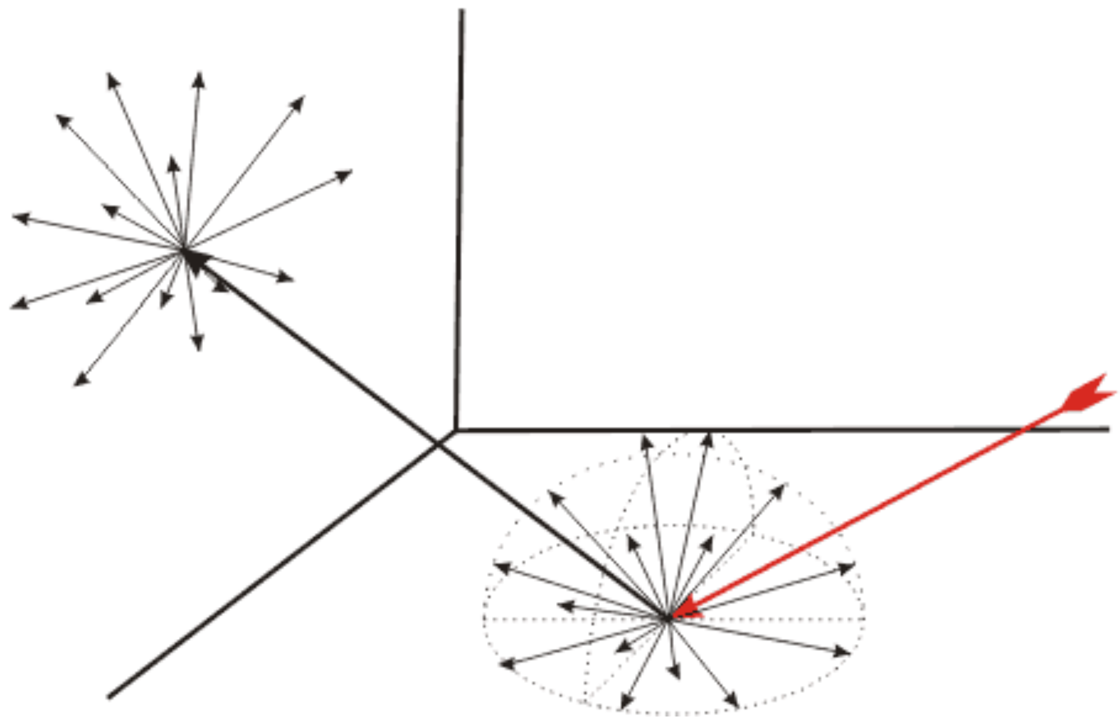
$$L_{\text{out}}(\hat{w}_r) = \int_{\hat{w}_i \in \text{hemisphere}} f_r(\hat{w}_r, \hat{w}_i)\ L_{\text{in}}(\hat{w}_i)\ \hat{w}_i \cdot \hat{n}$$

Global illumination: render using the full rendering equation

Main Idea

Light **leaving surface** depends on incoming light from all directions

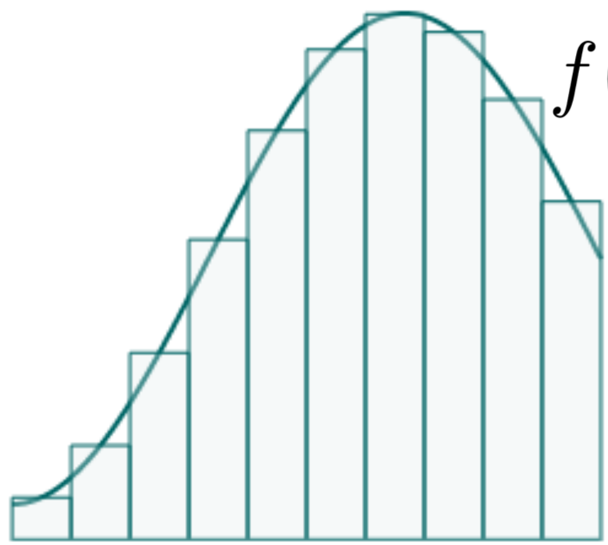
Recursive



Revisiting Integration

Problem: calculate integral of function

- area under the curve



Classic approach: Riemann sum

Monte Carlo Integration

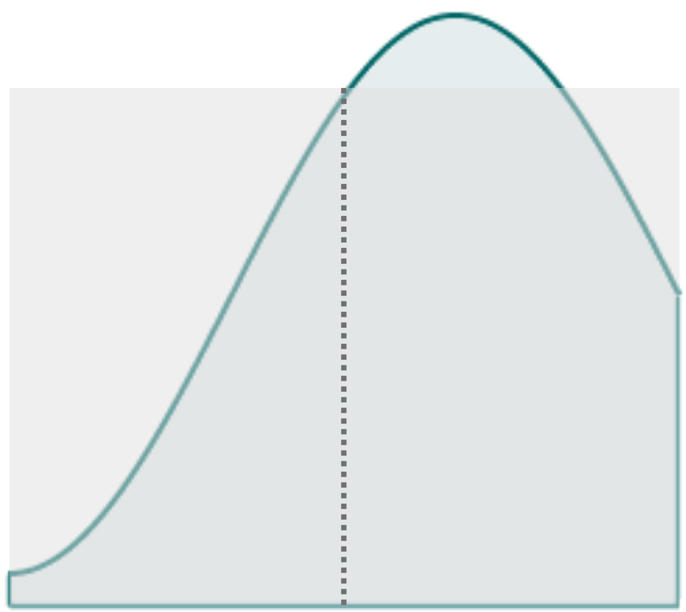
Problem: calculate integral of function

Allowed to evaluate function **once only**

Monte Carlo Integration

Problem: calculate integral of function

Allowed to evaluate function **once only**



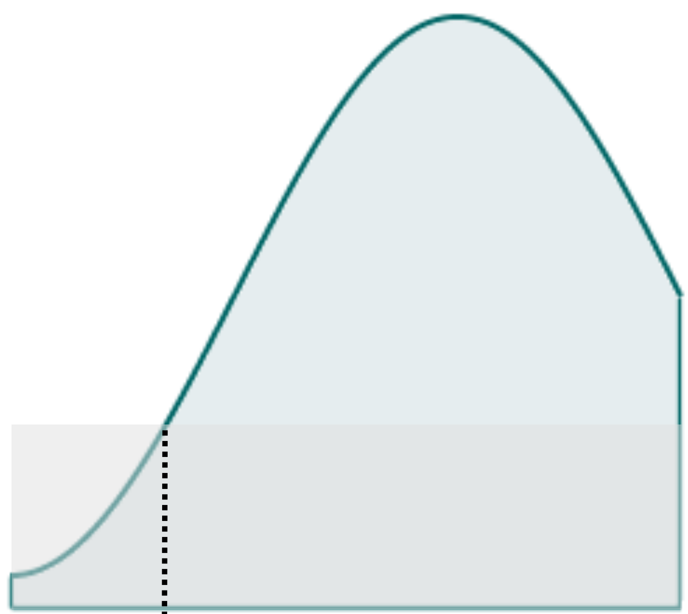
$$\int_0^1 f(x)\,dx \approx f\left(\frac{1}{2}\right)$$

Downside: value at x=½ may not be "typical"

Monte Carlo Integration

Problem: calculate integral of function

Allowed to evaluate function **once only**



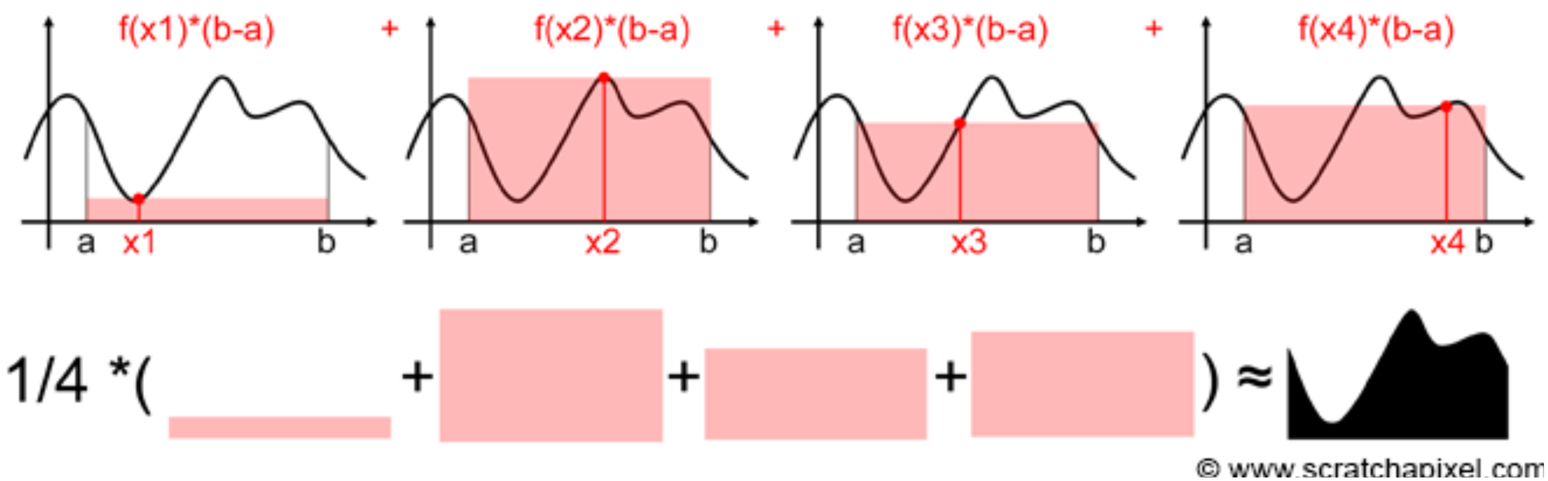
$$\int_0^1 f(x)\,dx \approx f\,(\mathrm{rand}())$$

unbiased estimate
of integral

Monte Carlo Integration

Problem: calculate integral of function

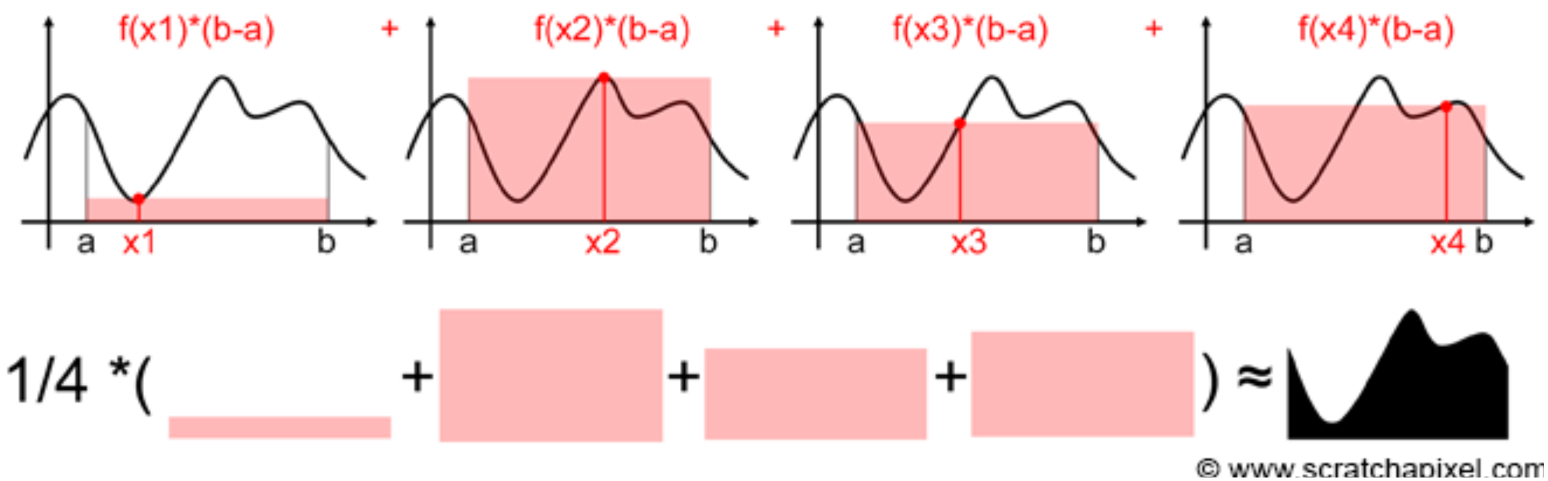
- sample function randomly at N points
- take the average



Monte Carlo Integration

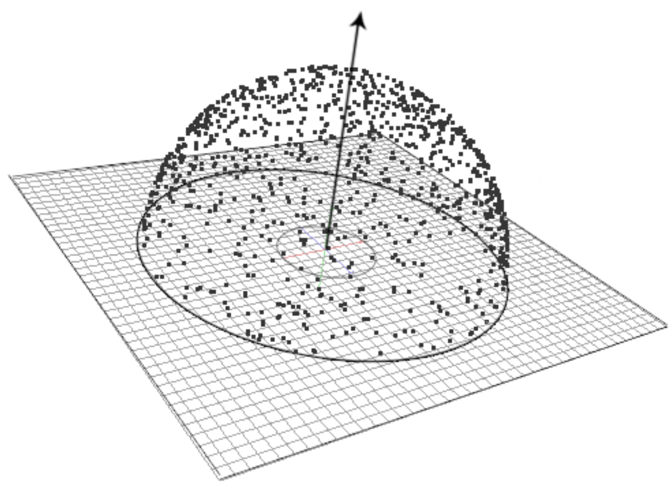
Pros:

- unbiased estimate of integral
- easy to code, easy to make adaptive
- works equally well in high dimensions



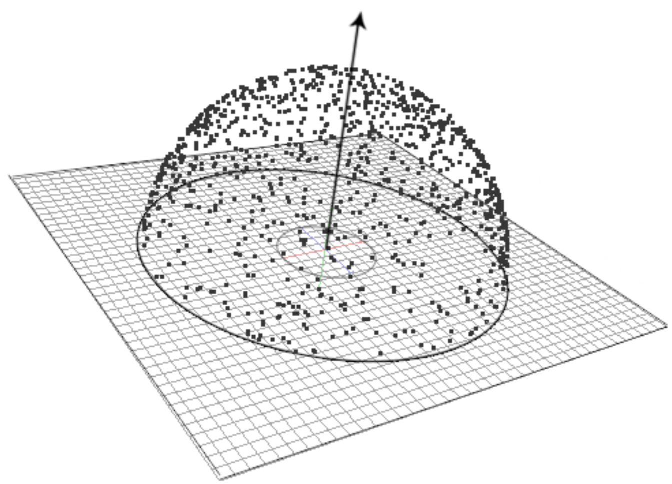
Monte Carlo Integration

$$L_{\text{out}}(\hat{w}_r) = \int_{\hat{w}_i \in \text{hemisphere}} f_r(\hat{w}_r, \hat{w}_i)\ L_{\text{in}}(\hat{w}_i)\ \hat{w}_i \cdot \hat{n}$$



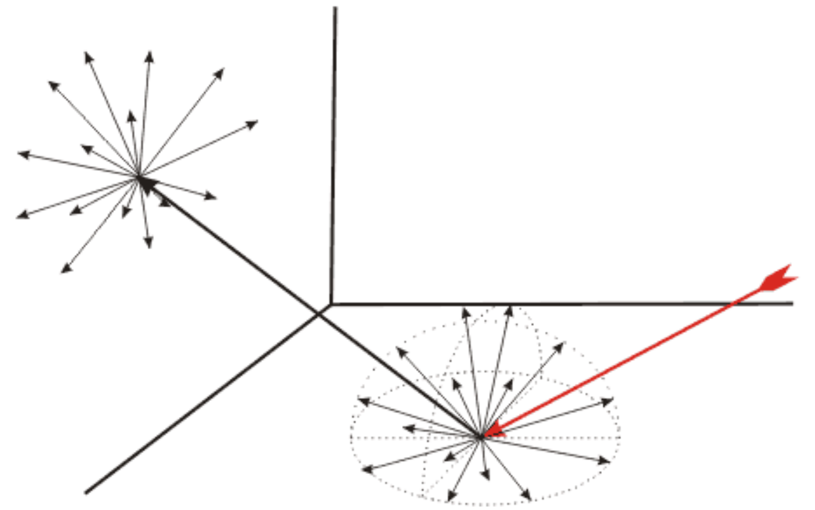
Monte Carlo Integration

$$L_{\text{out}}(\hat{w}_r) = \int_{\hat{w}_i \in \text{hemisphere}} f_r(\hat{w}_r, \hat{w}_i)\ L_{\text{in}}(\hat{w}_i)\ \hat{w}_i \cdot \hat{n}$$



$$L_{\text{out}}(\hat{w}_r) \approx \frac{2\pi}{N} \sum_{j=1}^{N} f_r(\hat{w}_r, \hat{w}_i^j)\ L_{\text{in}}(\hat{w}_i^j)\ \hat{w}_i^j \cdot \hat{n}$$

Path Tracing



Shoot primary ray

At intersection point,

- choose N random secondary ray dirs
- shoot each secondary ray
  - (recursively shoot tertiary rays, …)
- compute

$$L_{\text{out}}(\hat{w}_r) \approx \frac{2\pi}{N} \sum_{j=1}^{N} f_r(\hat{w}_r, \hat{w}_i^j)\ L_{\text{in}}(\hat{w}_i^j)\ \hat{w}_i^j \cdot \hat{n}$$

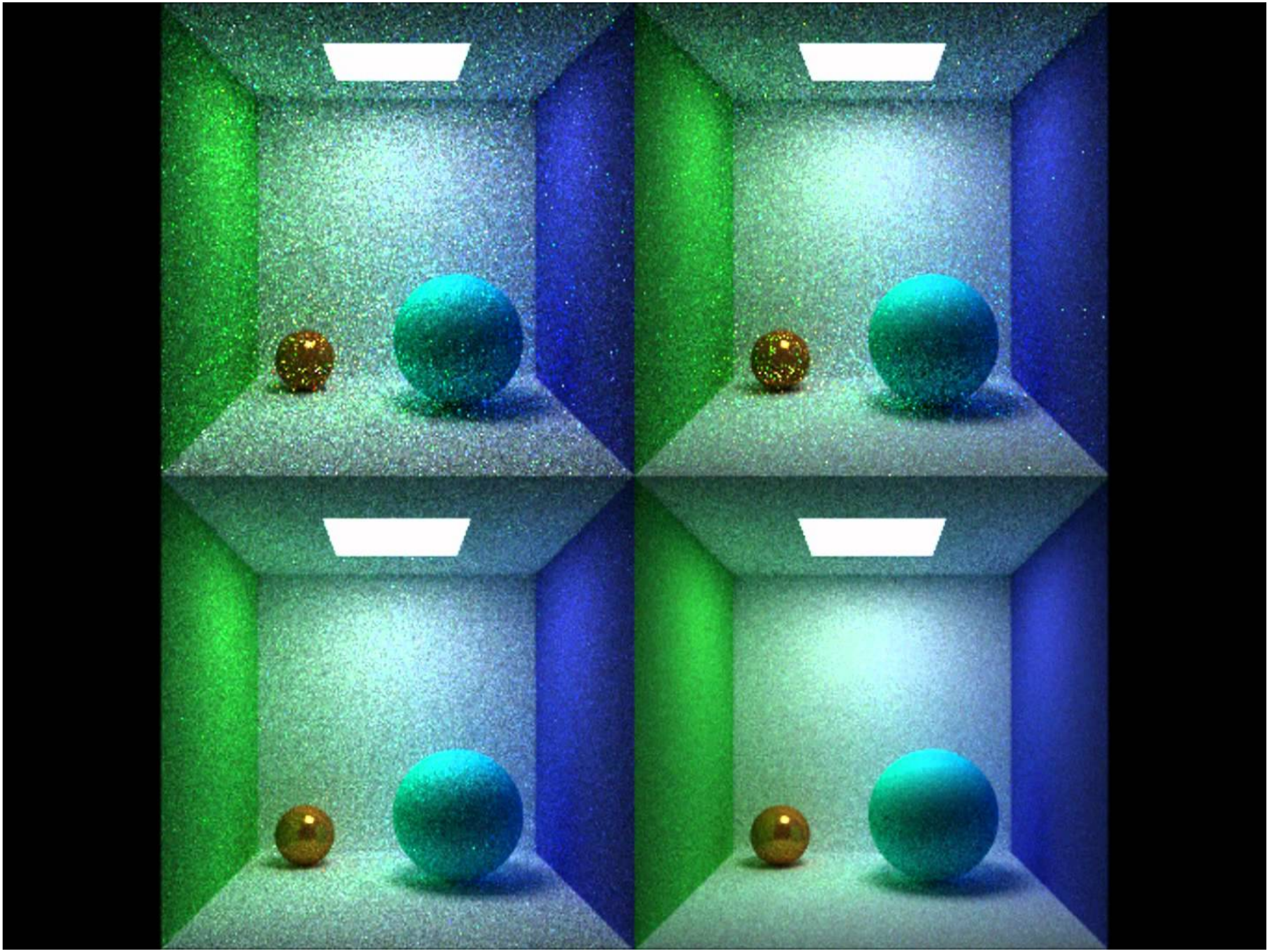
Path Tracing

Two knobs:

1. number of rays to shoot at each level
2. maximum recursion depth

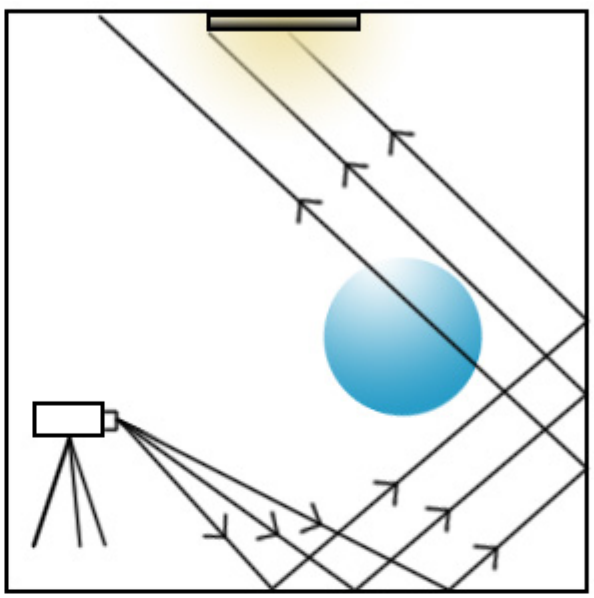
True image is limit as **both** go to infinity

- called **ground truth**
- **SLOW**: combinatorial explosion

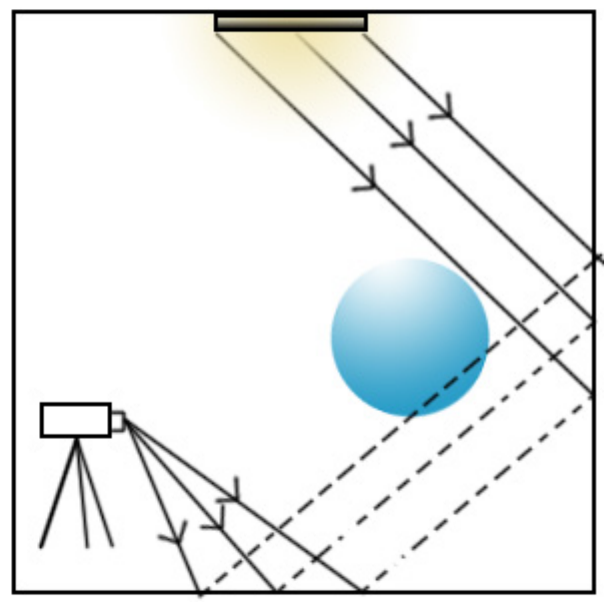


Path Tracing: Improvements

Bidirectional path tracing: shoot rays also from the light sources



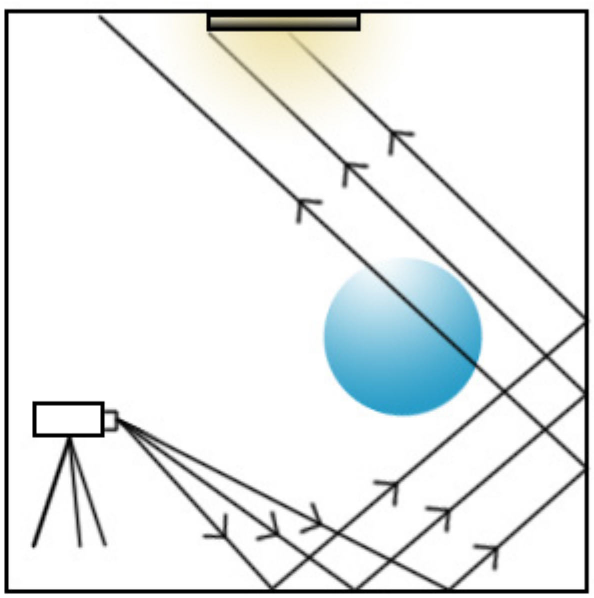
Normal Pathtracing



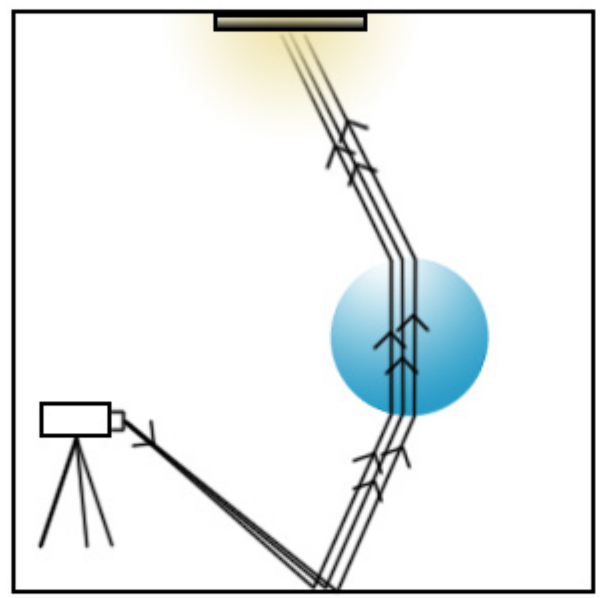
Bidirectional pathtracing

Path Tracing: Improvements

Metropolis-Hastings: instead of random rays, perturb known good rays



Normal Pathtracing

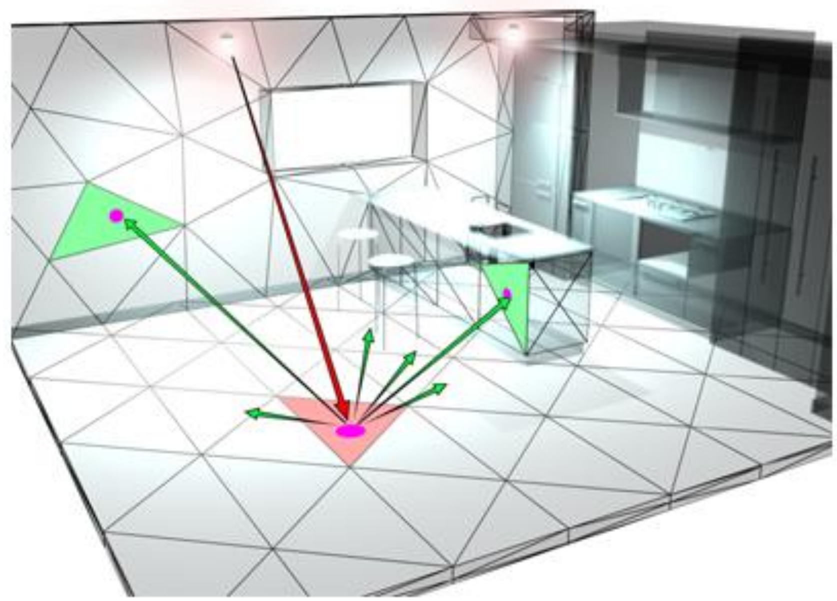


MLT pathtracing

Radiosity

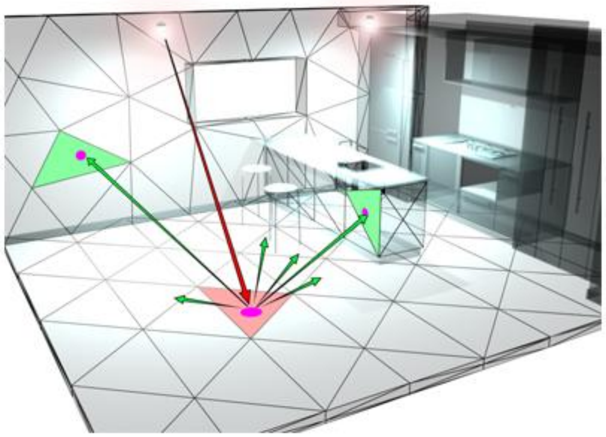
Faster method when surfaces are diffuse

Divide geometry
into patches



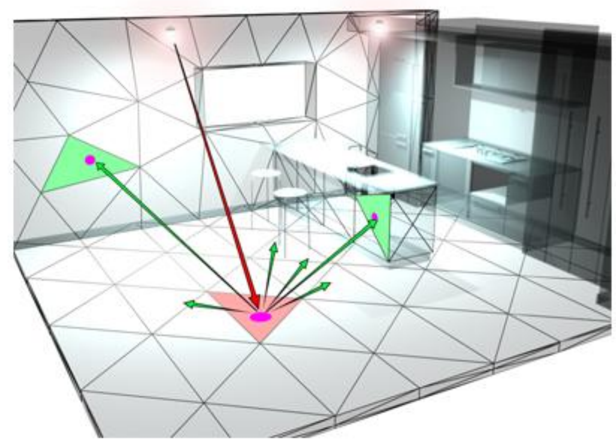
Each patch **receives**
and **transmits** light
to other patches

Radiosity



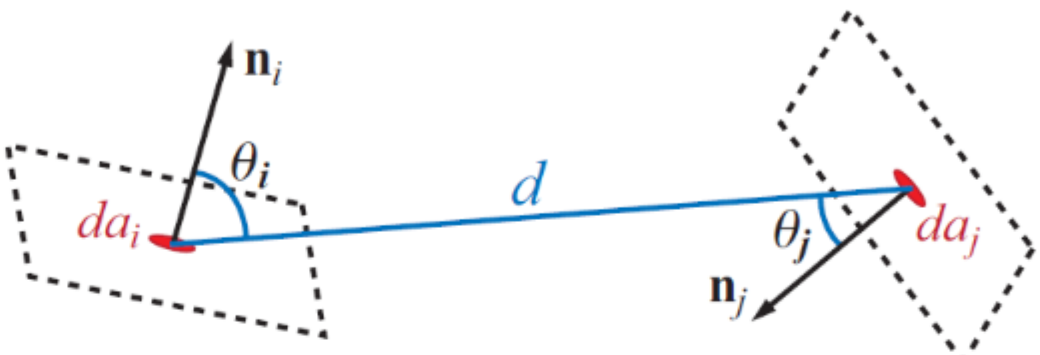
How much light transfers between two patches?

Radiosity



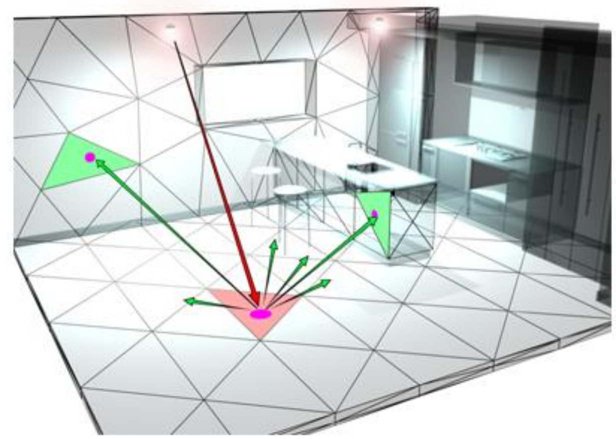
How much light transfers between two patches?

Depends on **distance** and **angle** of patches ("form factors")



$$\omega_{ij} = \frac{\cos\theta_i \cos\theta_j}{\pi d^2}$$

Radiosity

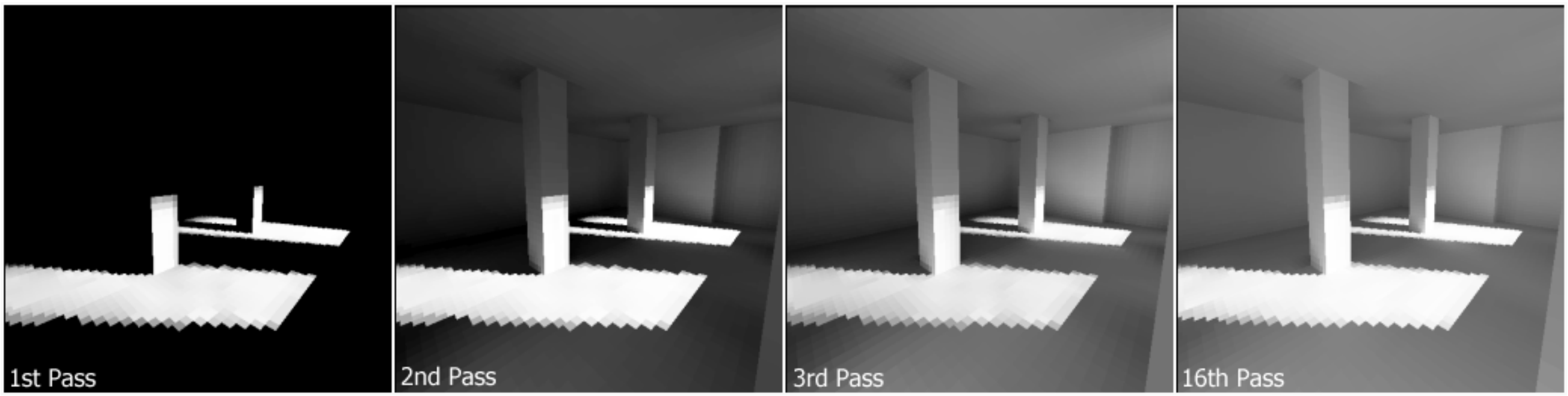


How much light transfers between two patches?

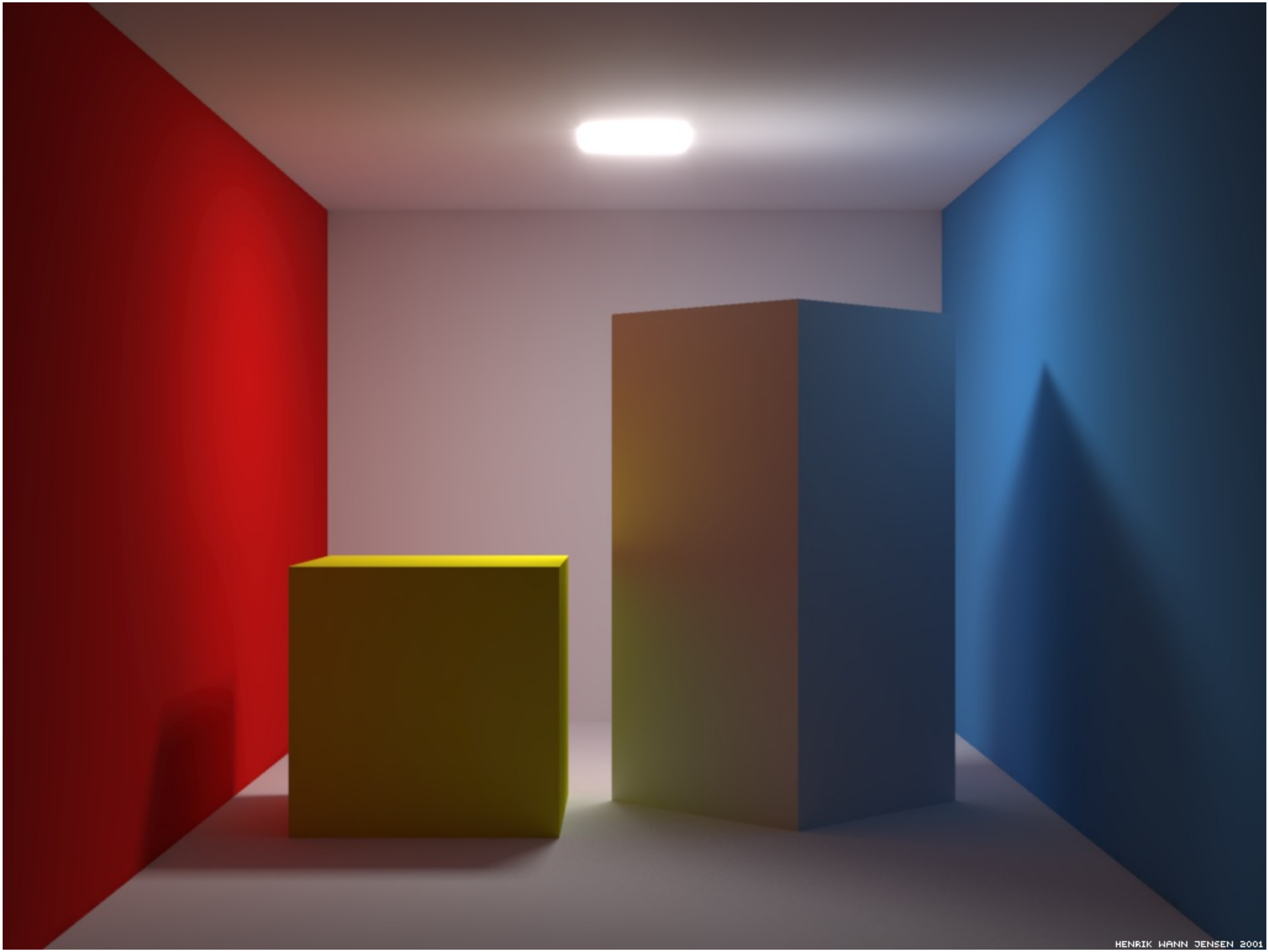
Depends on **distance** and **angle** of patches (“form factors”)

Propagate light around scene until steady state is reached

Radiosity



Propagate light around scene until steady state is reached


HENRIK WANN JENSEN 2001

Radiosity

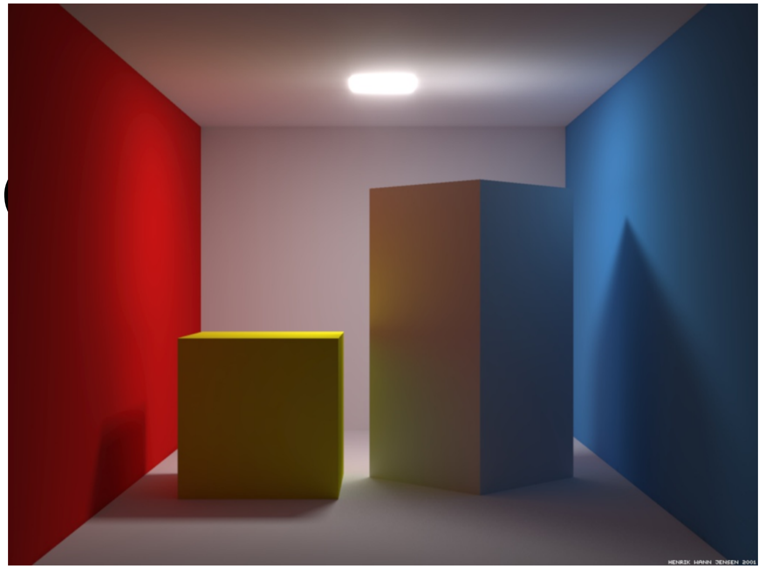
Pros:

- beautiful soft shadows
- up to 10x faster than path tracing

Cons:

- diffuse only (no reflections)
- tessellation artifacts

Ambient Occlusi

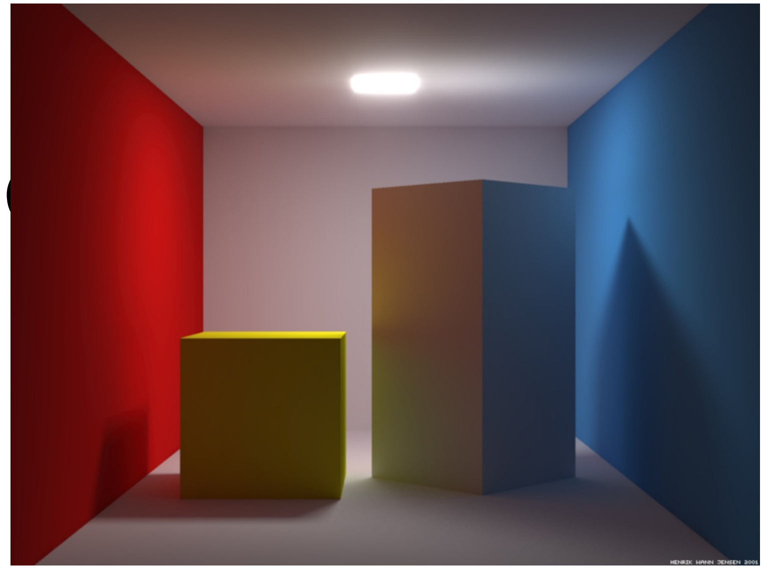


Notice:

corners of walls are darker than centers

Why?

Ambient Occlusi



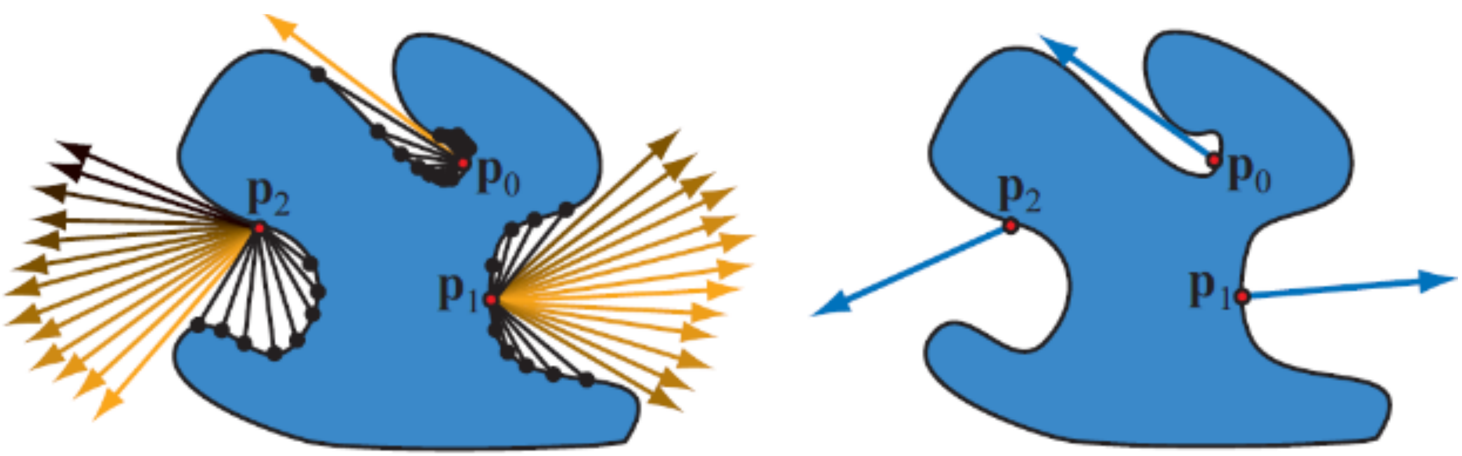
Notice:
corners of walls are darker than centers

Why?

- fewer direction light can come in from

Ambient Occlusion

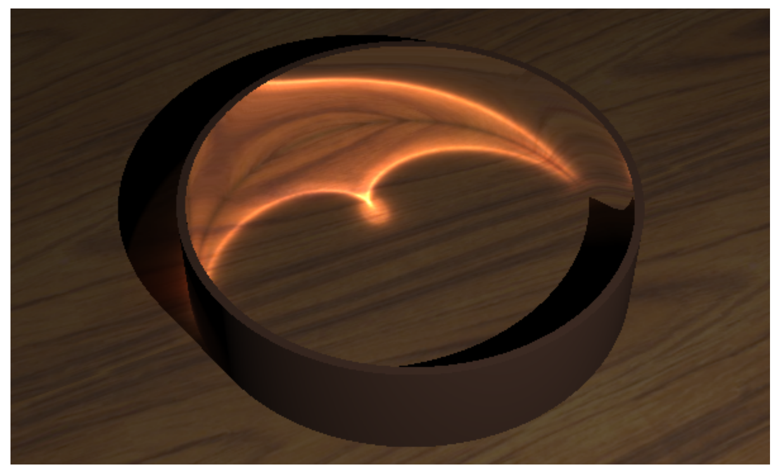
To approximate this effect:
  from each point on surface, shoot rays
  for small distance in all directions



More rays blocked → darker pixel

Caustics

What is going on?

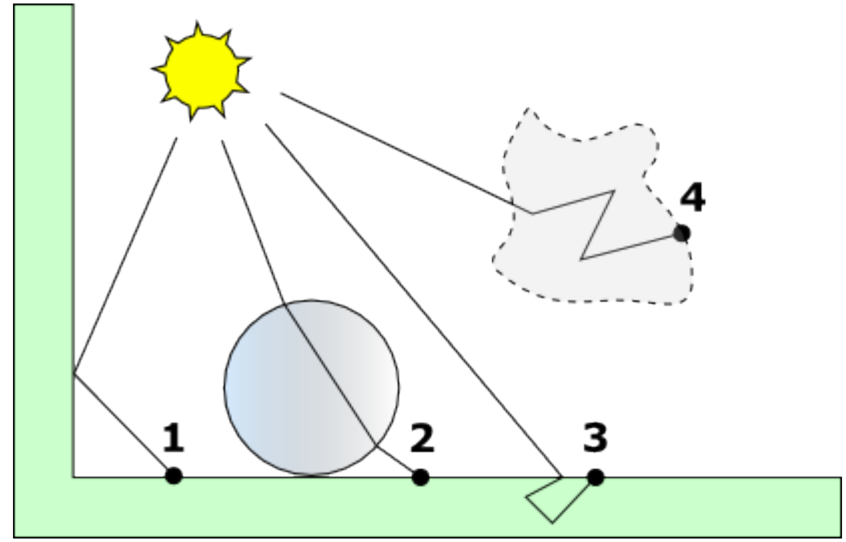


Why is it hard?

Photon Mapping

Shoot rays (photons) from light into scene

Keep going until they hit eye?

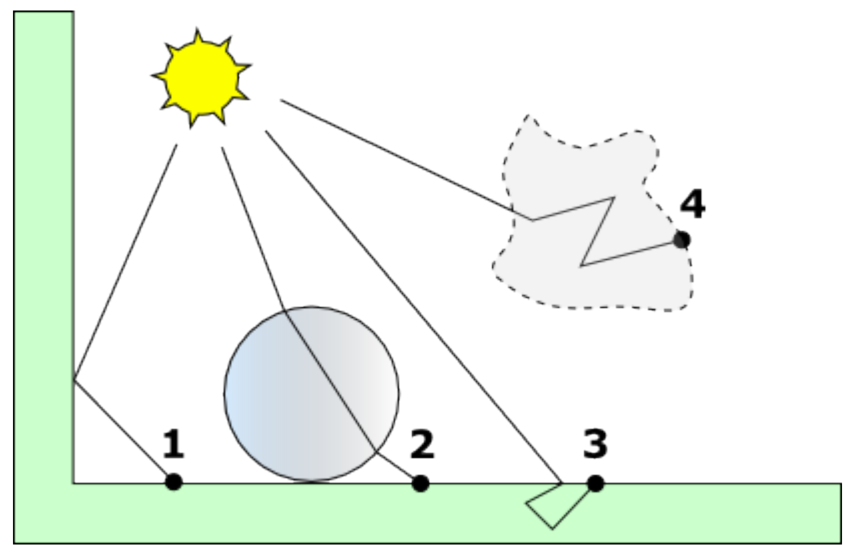


Photon Mapping

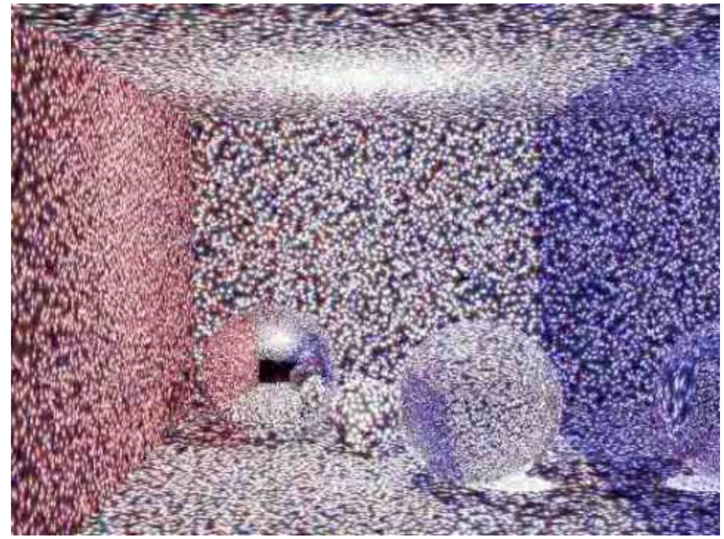
Shoot rays (photons) from light into scene

Keep going until they hit eye?

Impossible (too many photons)



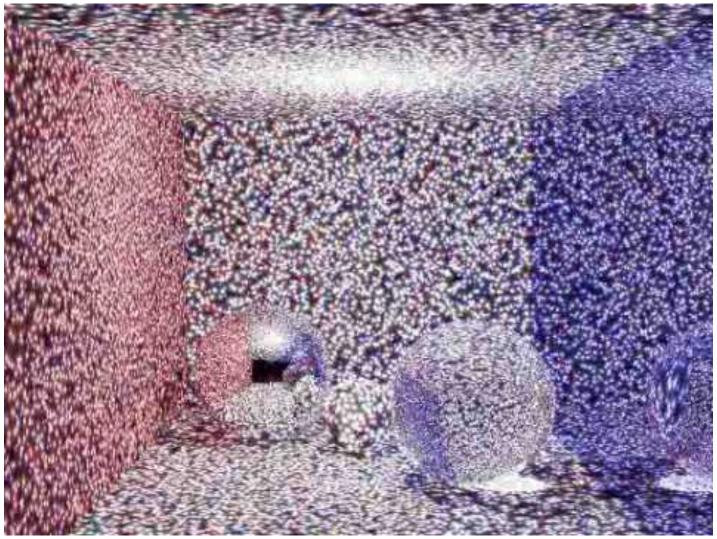
Photon Mapping



Phase 1: Shoot photons

- “small” number
- reflect, refract, scatter, etc
- stop after some depth, **store** in scene

Photon Mapping



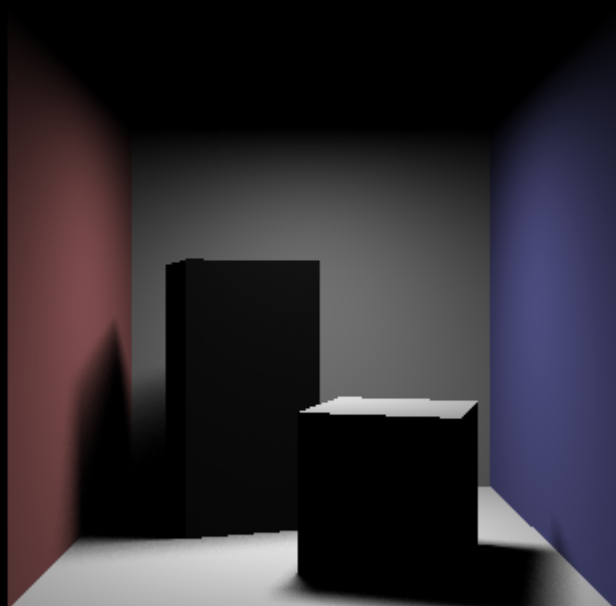
Phase 1: Shoot photons

- "small" number
- reflect, refract, scatter, etc
- stop after some depth, **store** in scene

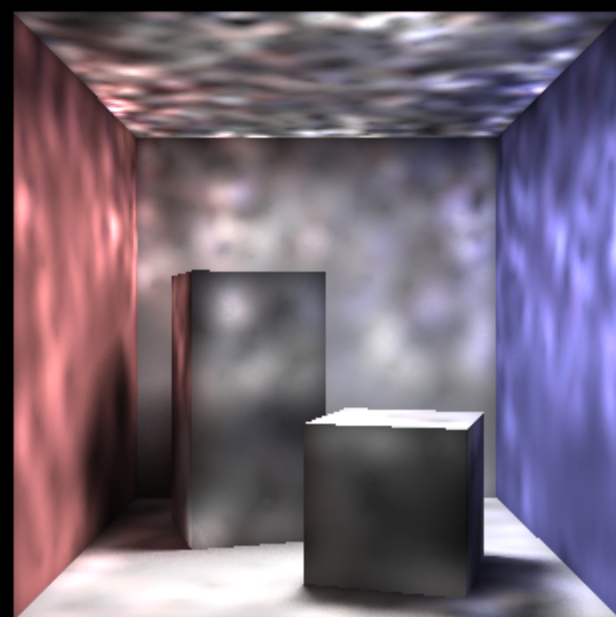
Phase 2: Ray trace the scene

- ordinary shading for direct lighting
- use photons for indirect lighting

Photon Mapping

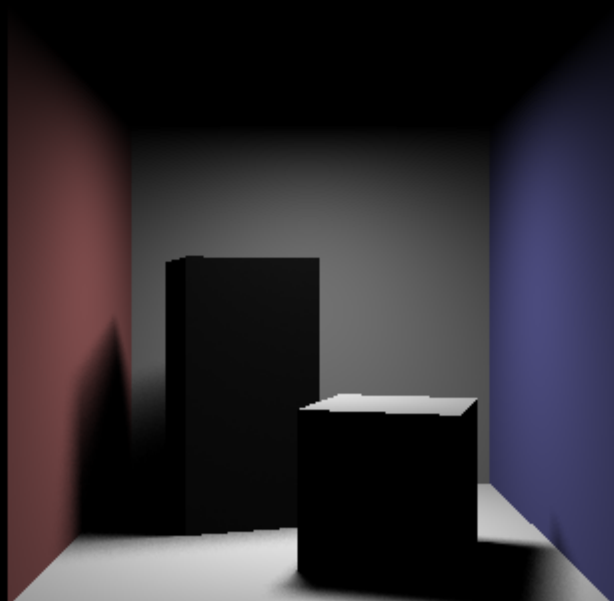


Direct Lighting

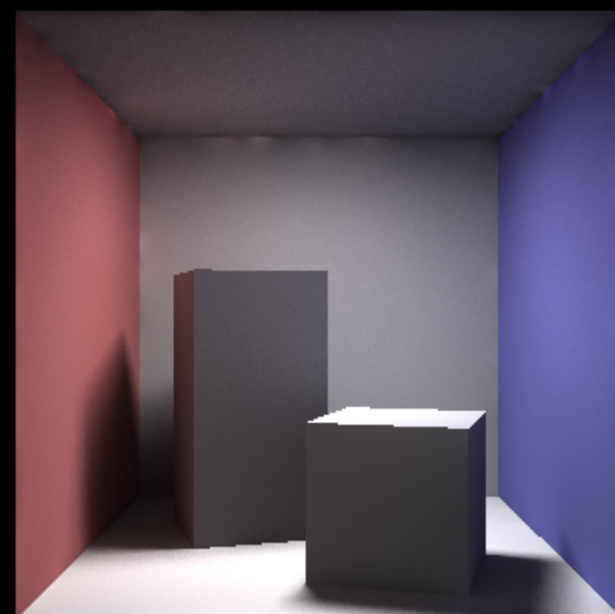


With Photon Map

Photon Mapping



Direct Lighting



With Photon Map
(depth = 2)