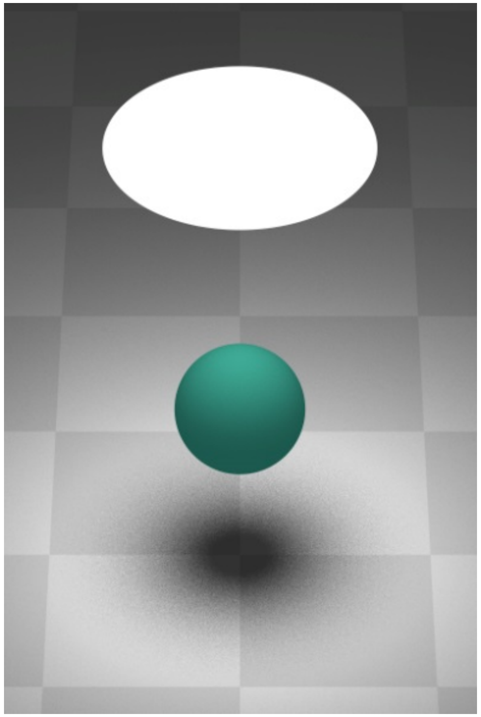
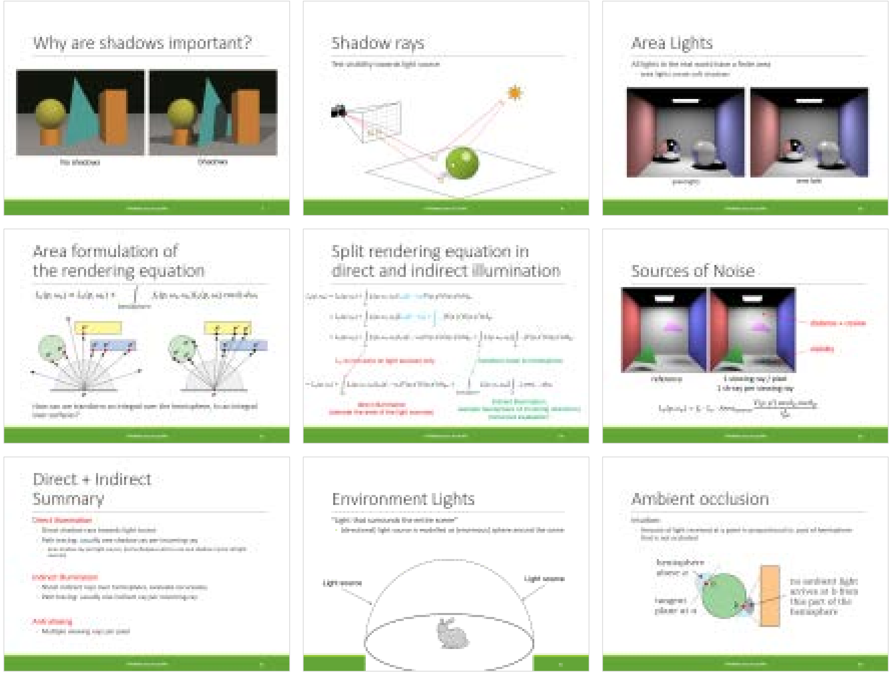
Direct Illumination



FUNDAMENTALS OF COMPUTER GRAPHICS

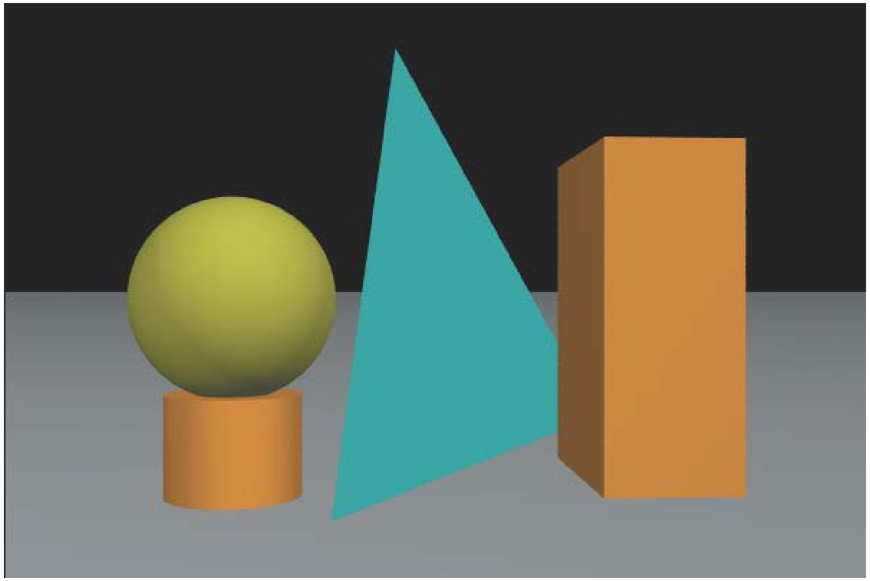
PHILIP DUTRÉ
DEPARTMENT OF COMPUTER SCIENCE

Overview Lecture

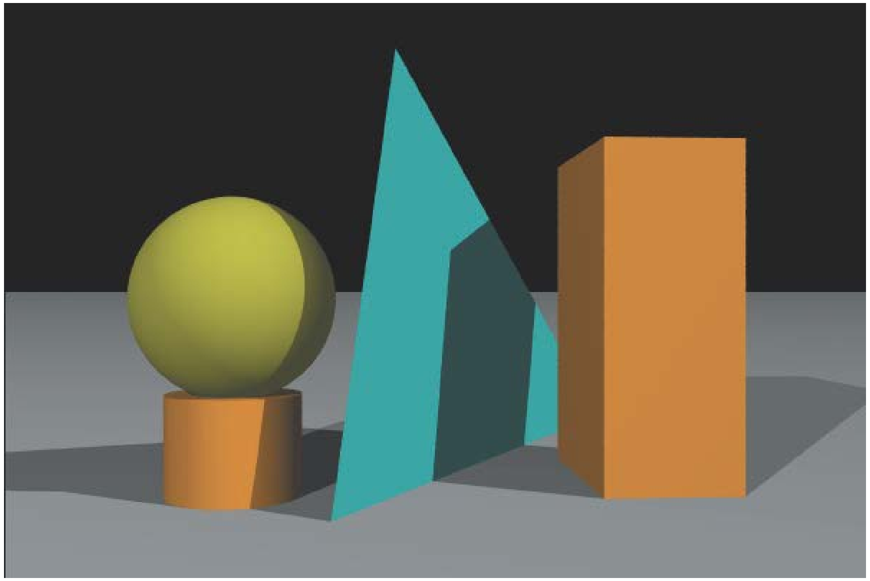


Relevant sections in book: Chapter 16, 17, 18. Useful: Chapters 5 and 7.
(Illustrations from Ray Tracing From The Ground Up, Physically-Based Rendering, Fundamentals of Computer Graphics)
(Page numbering might skip some slides due to ‘hidden’ slides in my presentation.)

Why are shadows important?

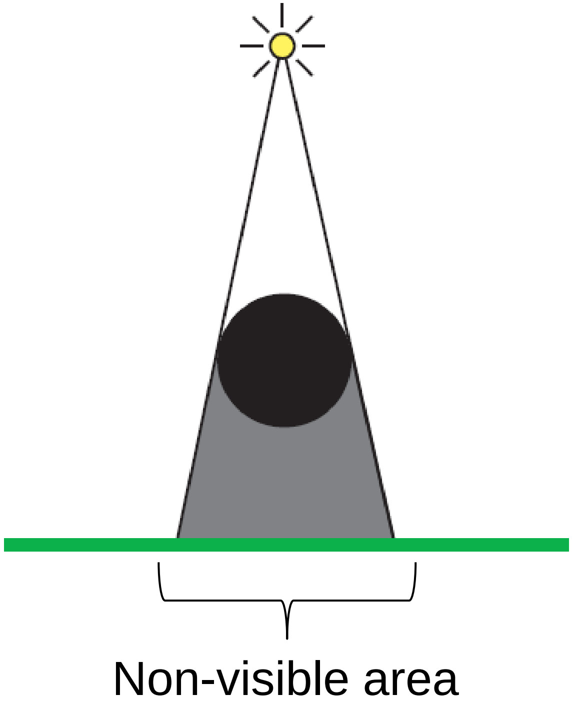


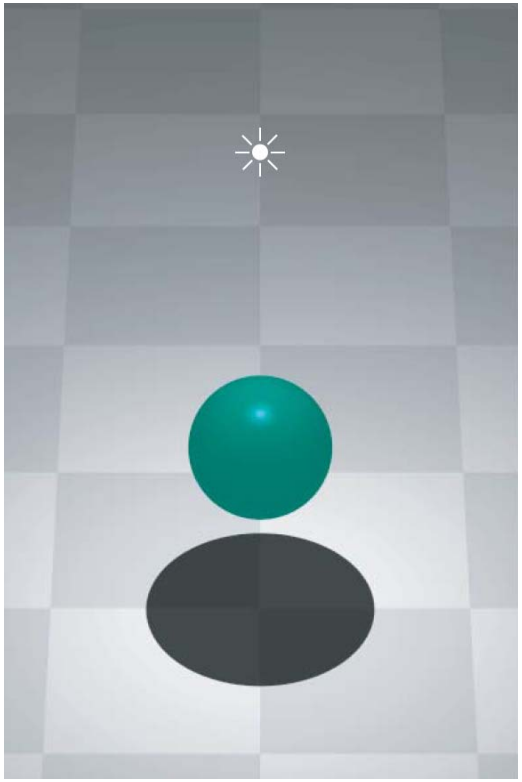
No shadows



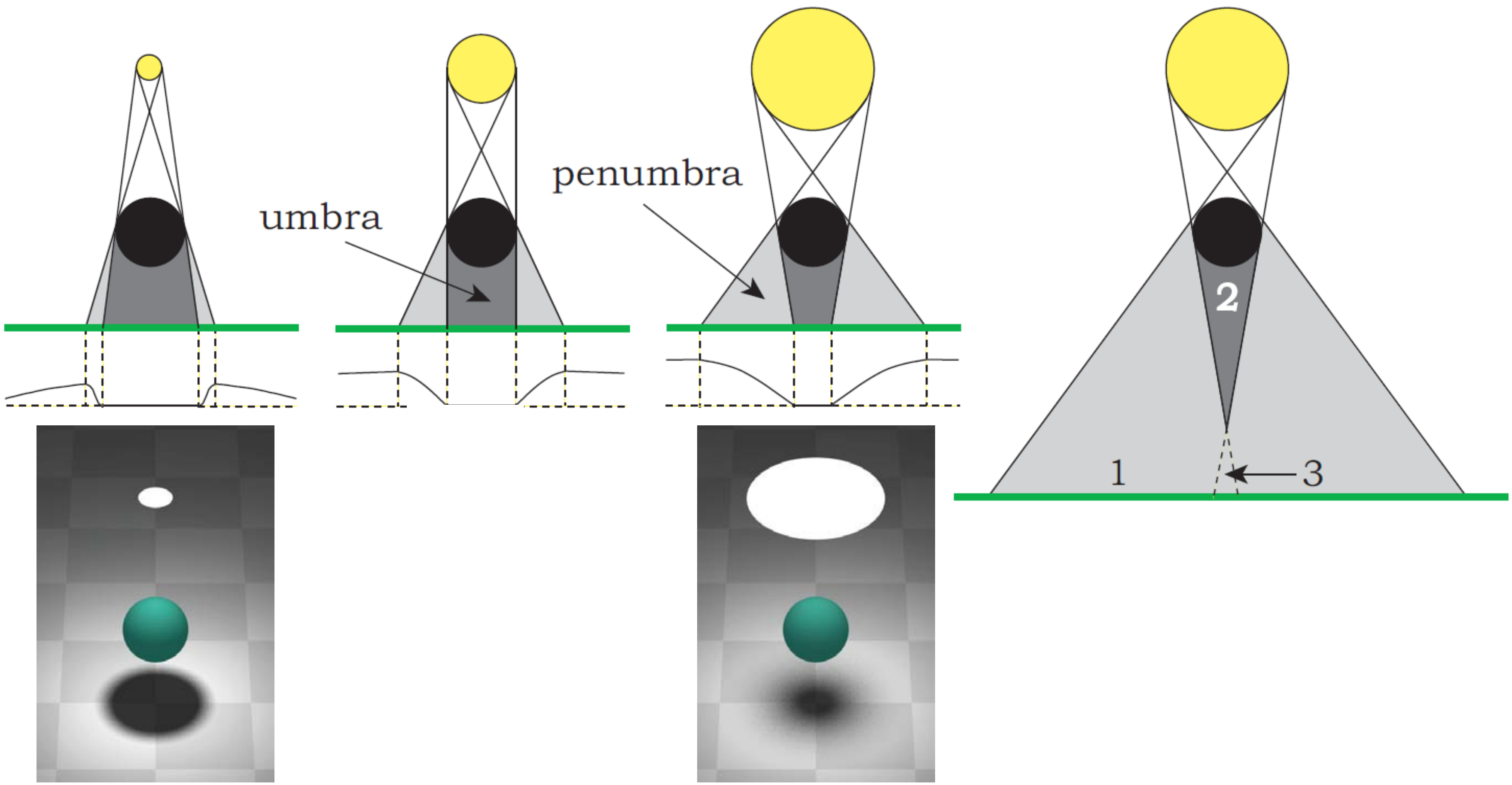
Shadows

Definitions w.r.t shadows

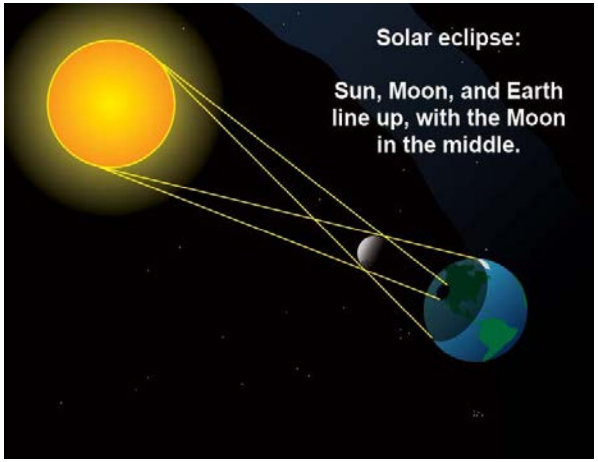


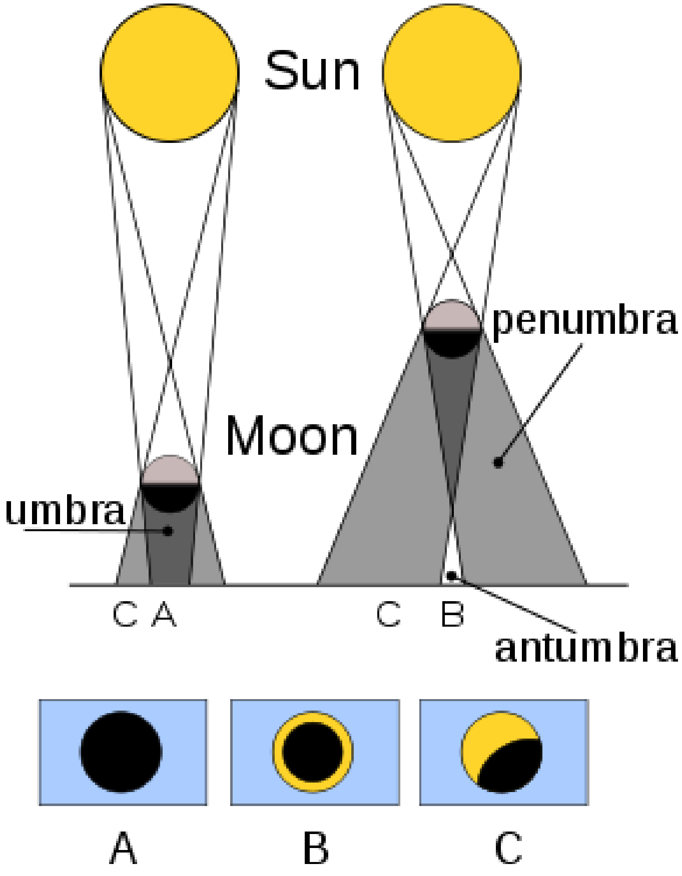


Definitions w.r.t shadows



Cfr. Solar Eclipse





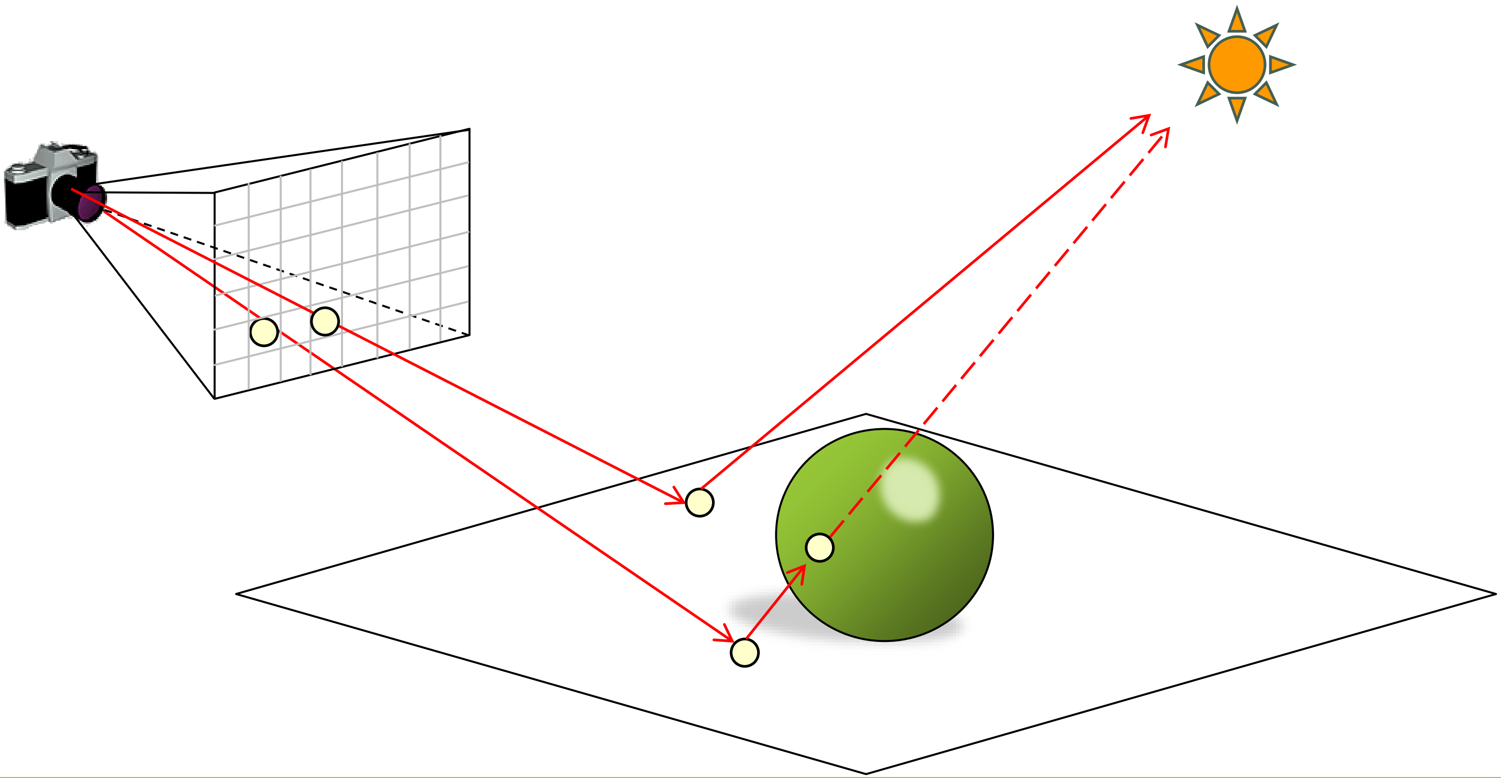
What do we need to compute shadows / direct illumination?

1. Determine what parts of the light source(s) are visible
   - Geometric operation … compute visibility
   - Tracing “shadow rays”

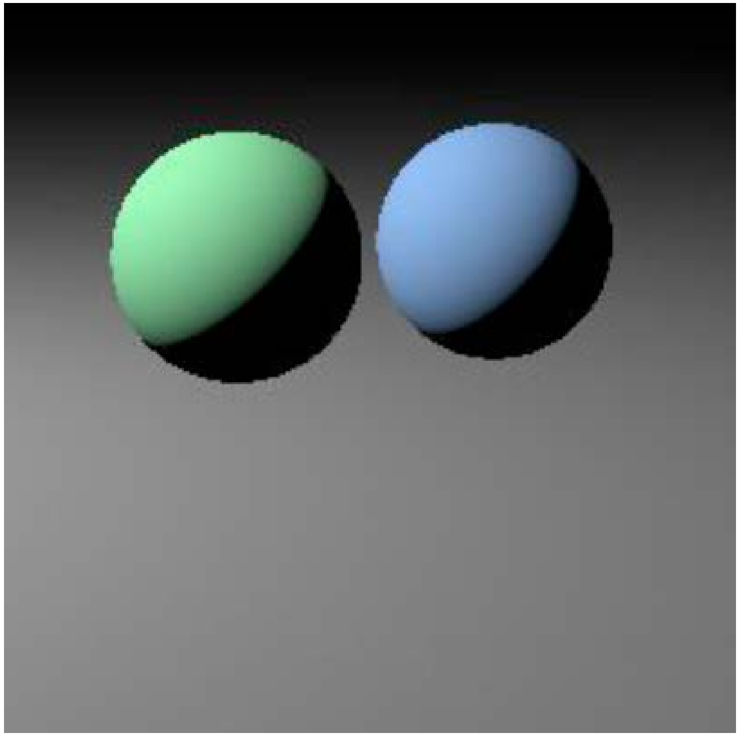
2. Compute the intensity of the illuminated areas
   - Direct illumination … compute intensity
   - Compute illumination integral using Monte Carlo

Shadow rays

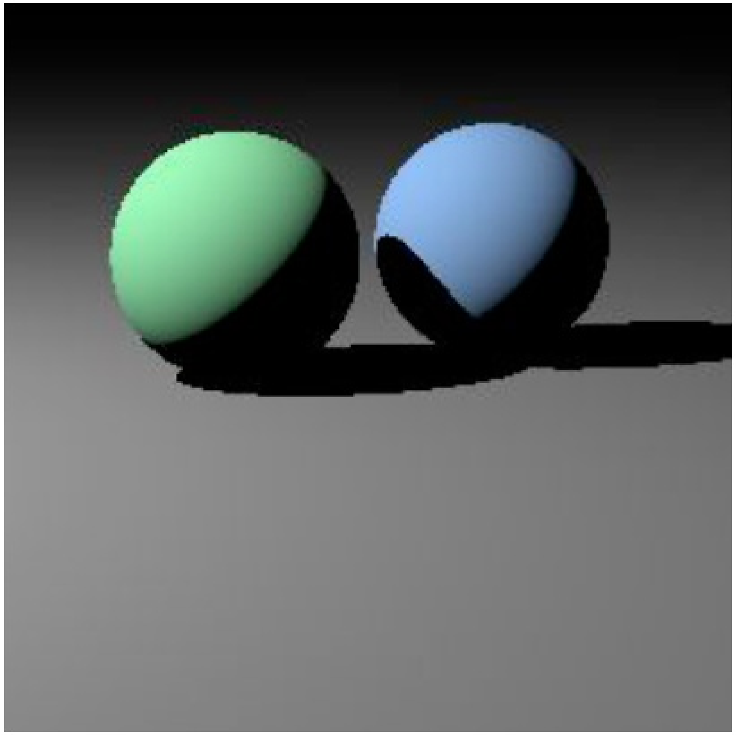
Test visibility towards light source



Shadow rays

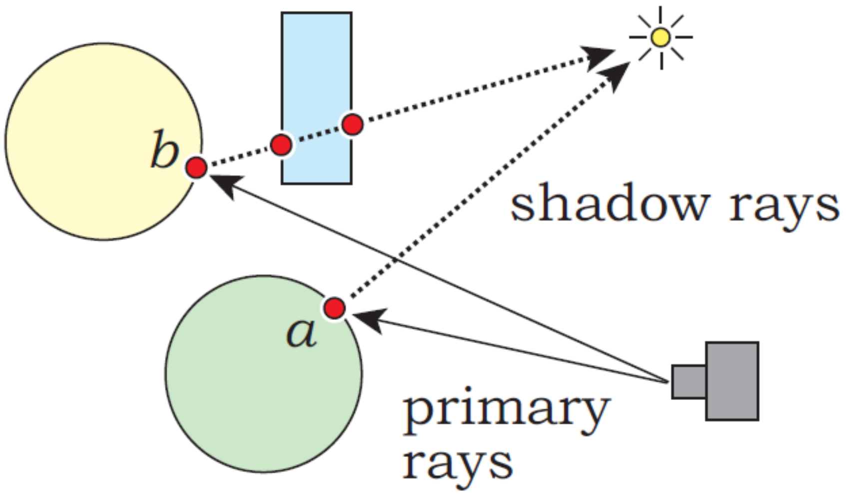


Diffuse shading model (no shadows)



Diffuse shading model (shadows)

Shadow rays



Shadow Ray:

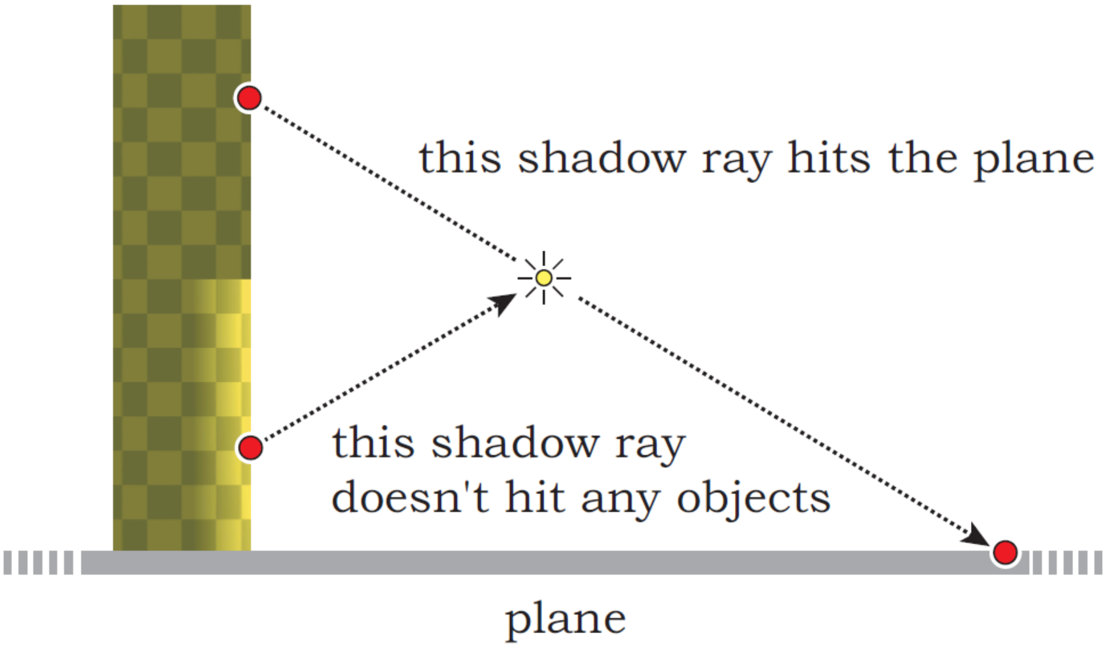
- origin: point to be shaded
- direction: towards the light source

If we find any point between surface point to be shaded and the light source → surface point is in shadow

- finding the closest intersection is not needed
- (more efficient than viewing rays; while-loop to intersect all objects can be early-aborted when we find any hitpoint)

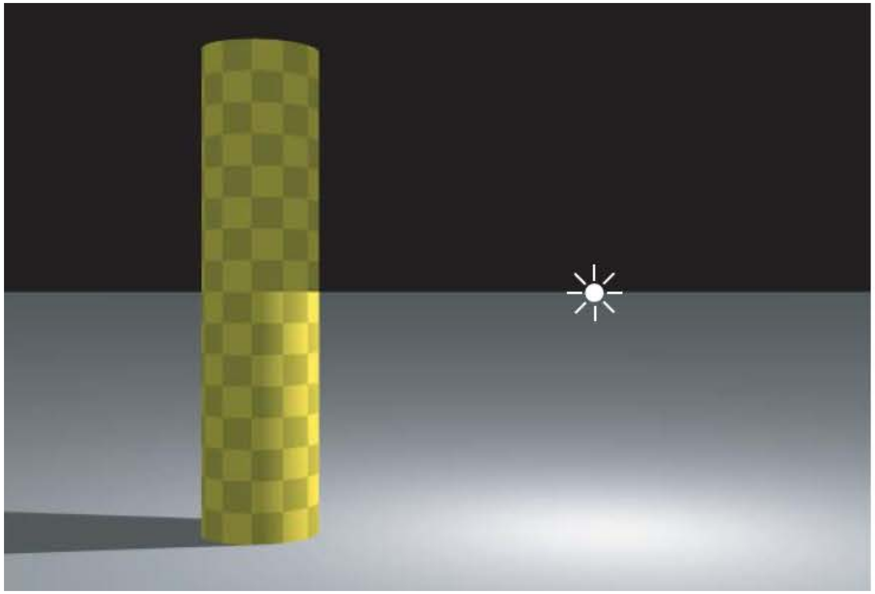
Shadow rays: problems

Geometric element casting the shadow must be located between the light source and point to be shaded

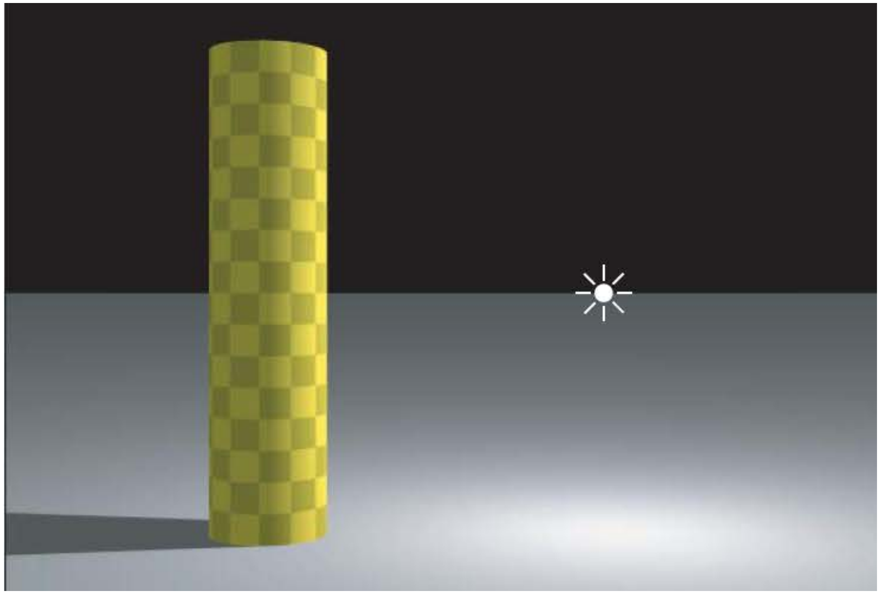


Shadow rays: problems

Geometric element casting the shadow must be located between the light source and point



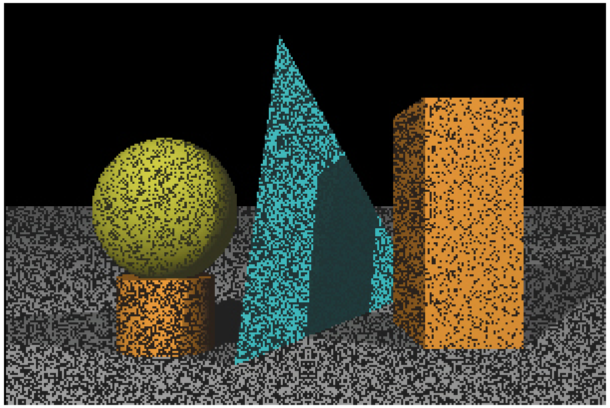
(plane casts shadow on top-half of cylinder)

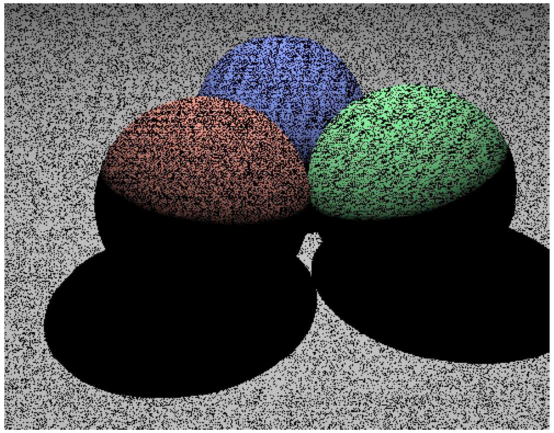


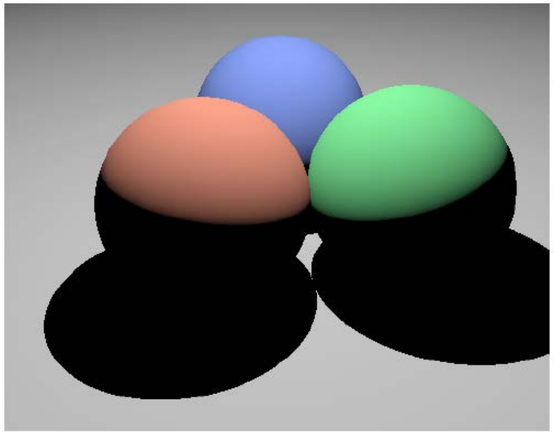
(correct shadows)

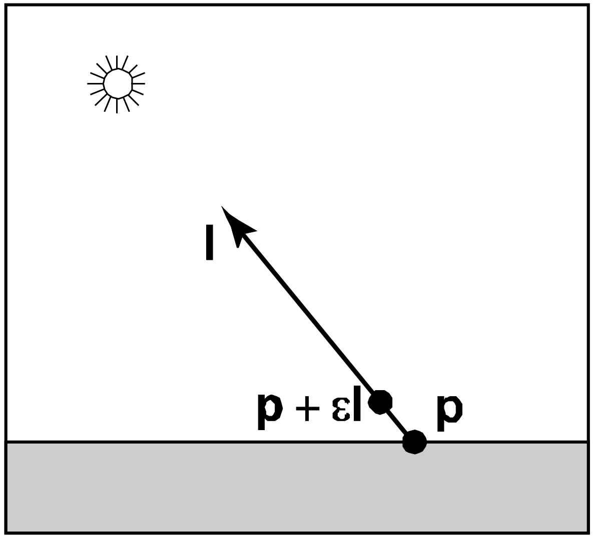
Shadow rays: problems

Self-intersection of shadow rays







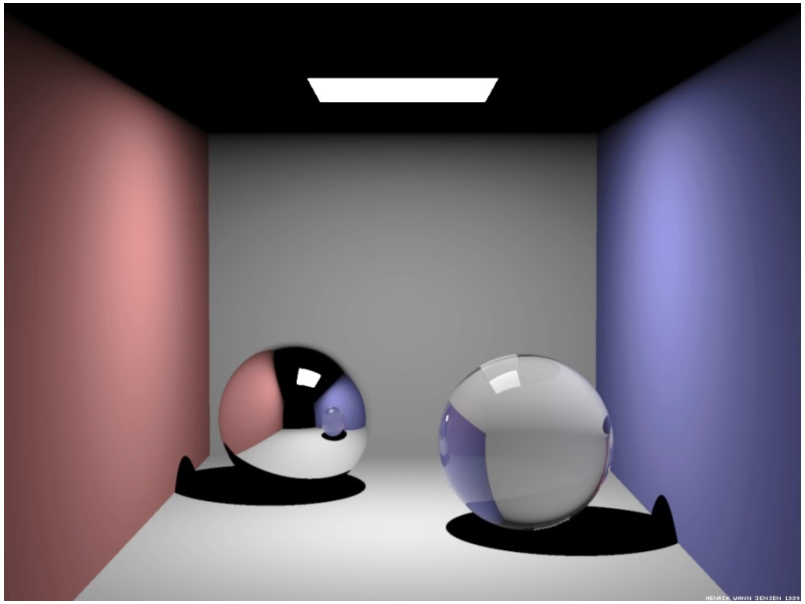


(Why is this not a problem for camera rays?)

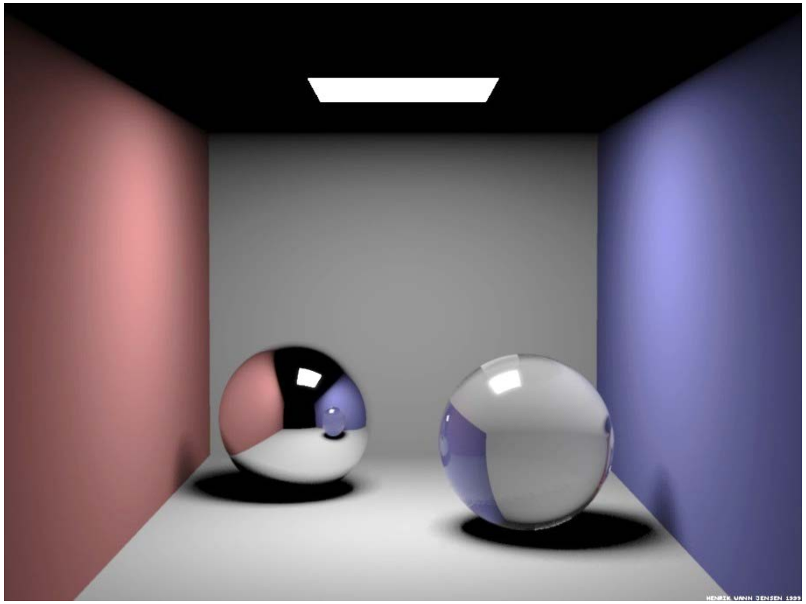
Area Lights

All lights in the real world have a finite area

- area lights create soft shadows



(point light)



(area light)

Area Lights

Rendering Equation:

$$L_o(p,\omega_o) = L_e(p,\omega_o) + \int_{hemisphere} f_r(p,\omega_i,\omega_o) L_i(p,\omega_i) \cos\theta_i \, d\omega_i$$

Path tracing algorithm (see previous lecture):

- Generate random directions over hemisphere
- Terminate paths using Russian Roulette

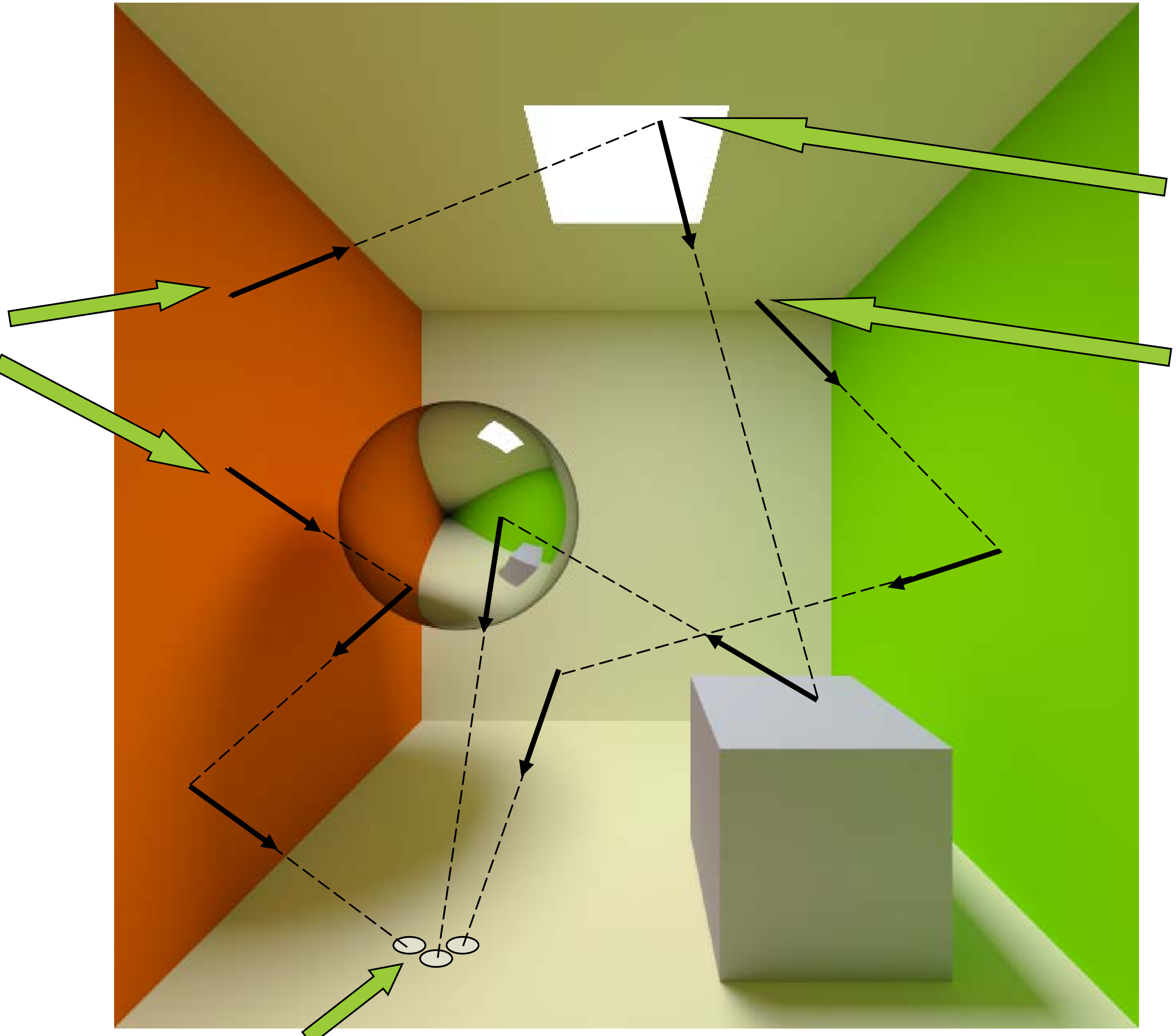
But we can do better for direct illumination:

- We know where the light is coming from …
  … light is coming from the light sources!

Light source contribution

Russian Roulette ends recursion

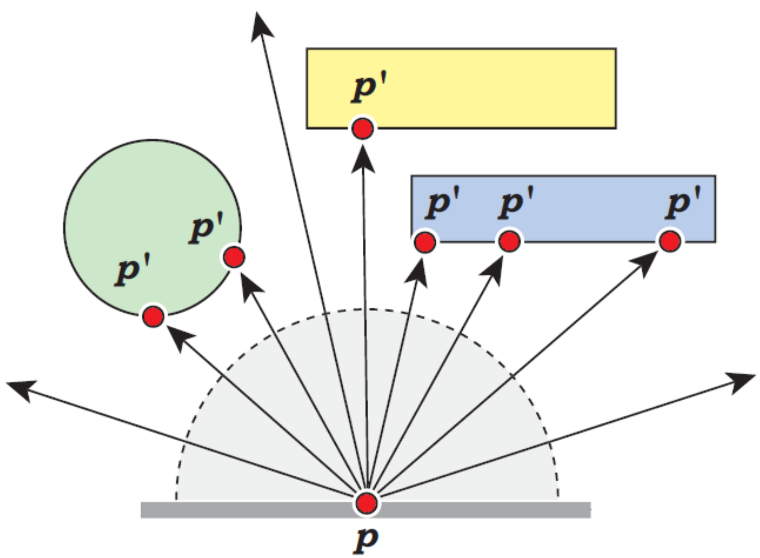
Russian Roulette ends recursion

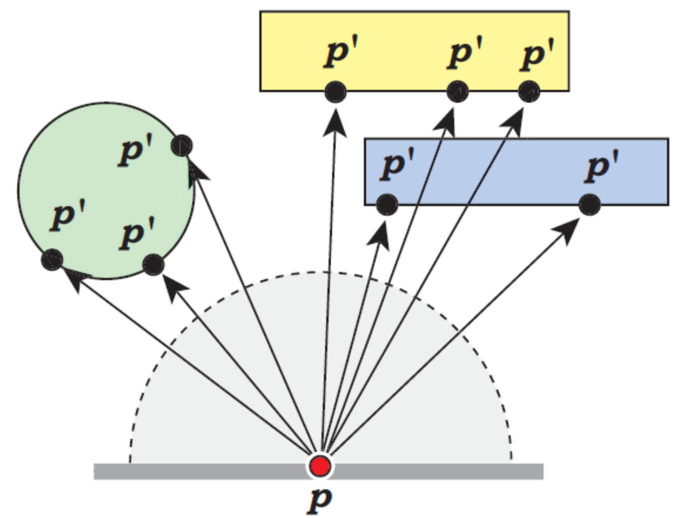


multiple viewing rays result in multiple visible points through pixel

Area formulation of the rendering equation

$$L_o(p,\omega_o) = L_e(p,\omega_o) + \int_{hemisphere} f_r(p,\omega_i,\omega_o) L_i(p,\omega_i) \cos\theta_i \, d\omega_i$$

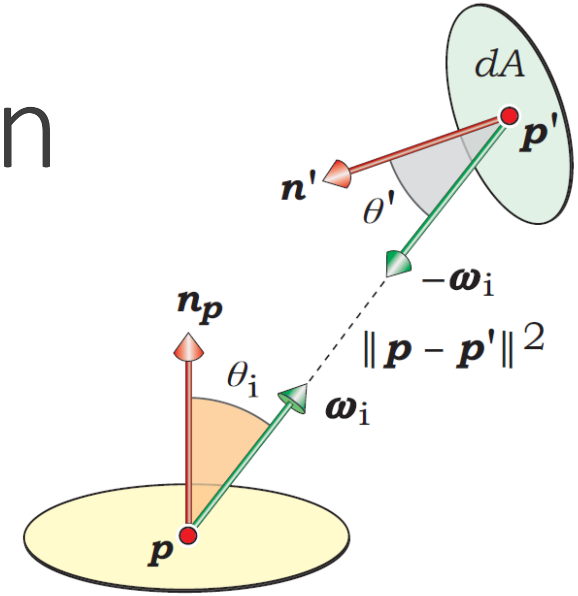




How can we transform an integral over the hemisphere, to an integral over surfaces?

Area formulation of the rendering equation

transform solid angle to area: $d\omega_i = \dfrac{\cos\theta' \, dA_{p'}}{\|p - p'\|^2}$



$$L_o(p,\omega_o) = L_e(p,\omega_o) + \int_{hemisphere} f_r(p,\omega_i,\omega_o) L_i(p,\omega_i) \cos\theta_i \, d\omega_i$$

$$= L_e(p,\omega_o) + \int_{A_{visible}} f_r(p,\omega_i,\omega_o) L_i(p,\omega_i) \cos\theta_i \frac{\cos\theta' \, dA_{p'}}{\|p - p'\|^2}$$

$$= L_e(p,\omega_o) + \int_A f_r(p,\omega_i,\omega_o) L_o(p',-\omega_i) V(p,p') G(p,p') dA_{p'}$$

visible surface points p’ to p

all surface points p’ in scene

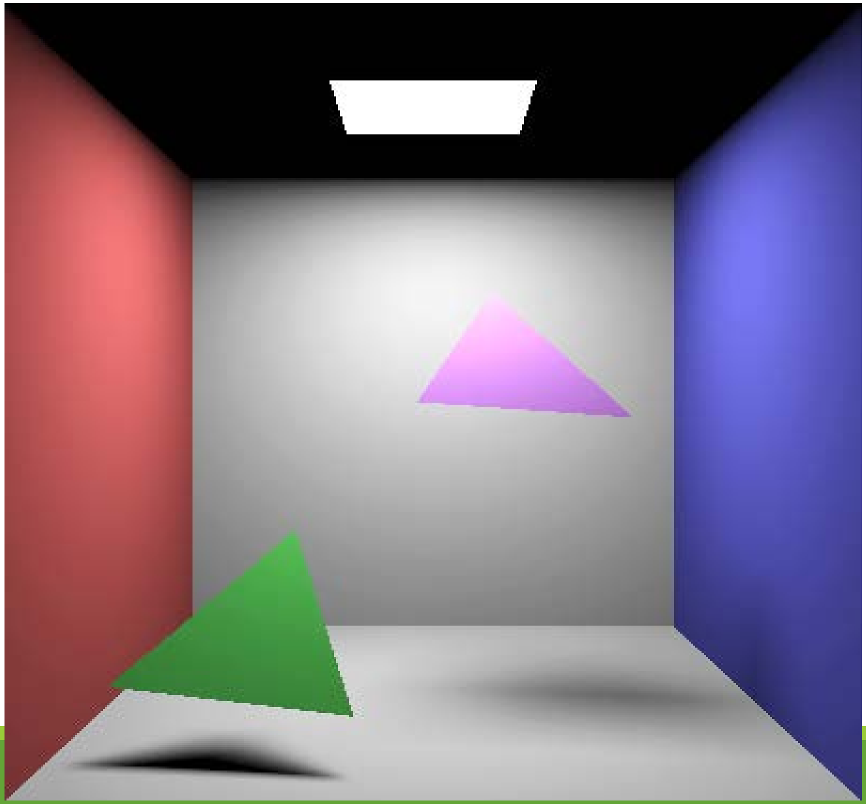
radiance emitted by p’

visibility between p and p’

Area formulation of the rendering equation

What if we apply MC to this formulation?

$$L_o(p,\omega_o) = L_e(p,\omega_o) + \int_A f_r(p,\omega_i,\omega_o) L_o(p',-\omega_i) V(p,p') G(p,p') dA_{p'}$$



Split rendering equation in direct and indirect illumination

$$L_o(p,\omega_o) = L_e(p,\omega_o) + \int_A f_r(p,\omega_i,\omega_o) L_o(p',-\omega_i) V(p,p') G(p,p') dA_{p'}$$

$$= L_e(p,\omega_o) + \int_A f_r(p,\omega_i,\omega_o)\left[L_e(p',-\omega_i) + \int_A \ldots\right] V(p,p') G(p,p') dA_{p'}$$

$$= L_e(p,\omega_o) + \int_A f_r(p,\omega_i,\omega_o) L_e(p',-\omega_i) V(p,p') G(p,p') dA_{p'} + \int_A f_r(p,\omega_i,\omega_o)\left[\int_A \ldots\right] V(p,p') G(p,p') dA_{p'}$$

L_e is non-zero on light sources only

transform back to hemisphere

$$= L_e(p,\omega_o) + \int_{A_L} f_r(p,\omega_i,\omega_o) L_e(p',-\omega_i) V(p,p') G(p,p') dA_{p'} + \int_{hemisphere} f_r(p,\omega_i,\omega_o)\left[\int_A \ldots\right] \cos(\ldots) d\omega_i$$

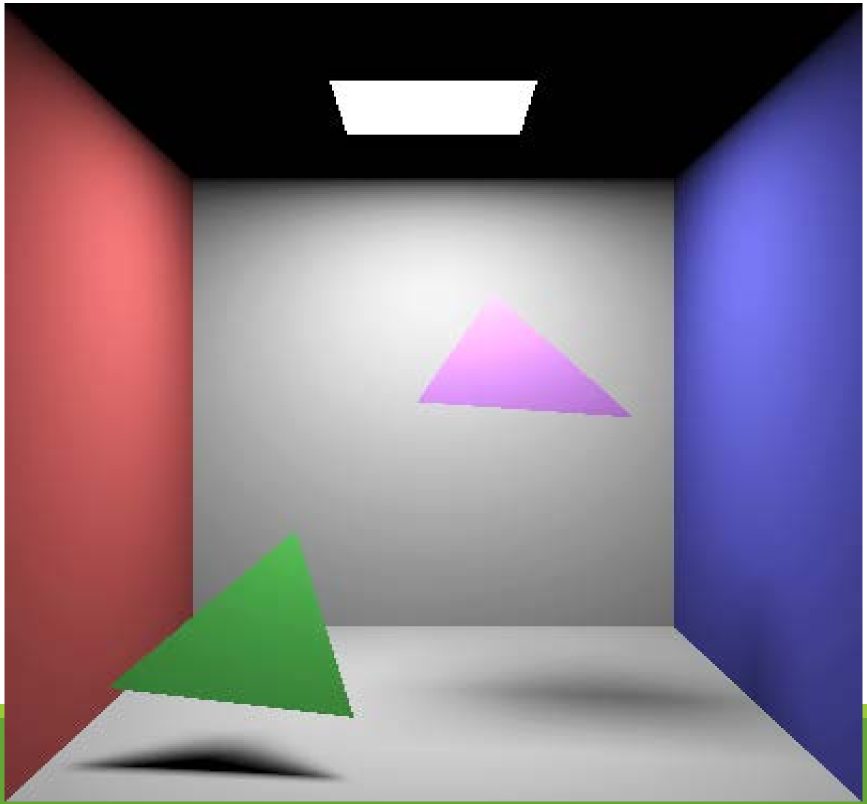
direct illumination
(sample the area of the light sources)

indirect illumination
(sample hemisphere of incoming directions)
(recursive evaluation)

Direct illumination only …

What if we apply MC to this formulation?

$$L_o(p,\omega_o) = \int_{A_L} f_r(p,\omega_i,\omega_o) L_e(p',-\omega_i) V(p,p') G(p,p') dA_{p'}$$



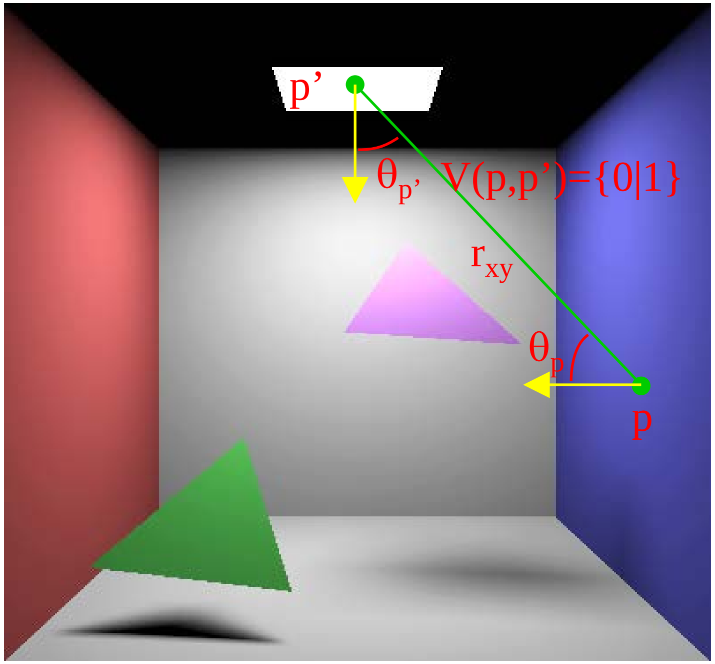
Direct diffuse illumination

Generate random points on surface of the light source, using a probability density function $pdf(p')$

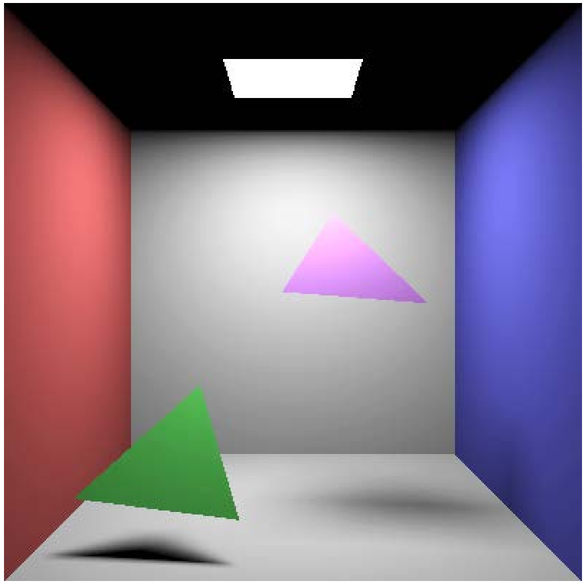
$$L_o(p,\omega_o) = \int_{A_L} f_r(p,\omega_i,\omega_o) L_e(p',-\omega_i) V(p,p') G(p,p') dA_{p'}$$

$$= f_r \cdot L_e \int_{A_L} V(p,p') \frac{\cos\theta_p \cos\theta_{p'}}{r_{pp'}^2} dA_{p'}$$

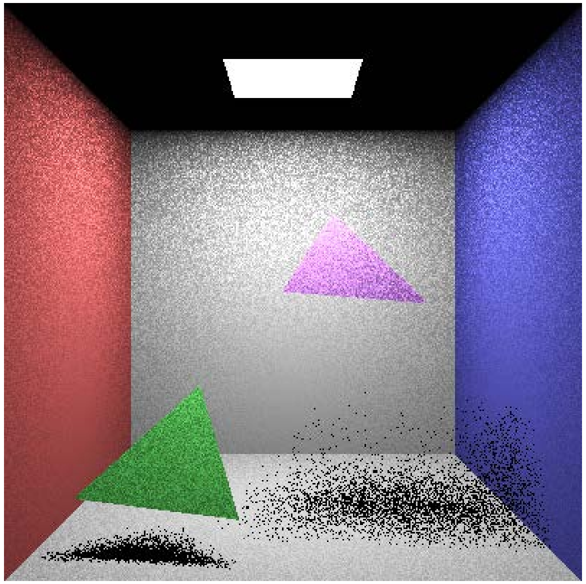
$$\approx \frac{f_r \cdot L_e}{N} \sum_{j=1}^{N} \frac{V(p,p') \cos\theta_p \cos\theta_{p'}}{pdf(p') r_{pp'}^2}$$



Direct diffuse illumination



reference

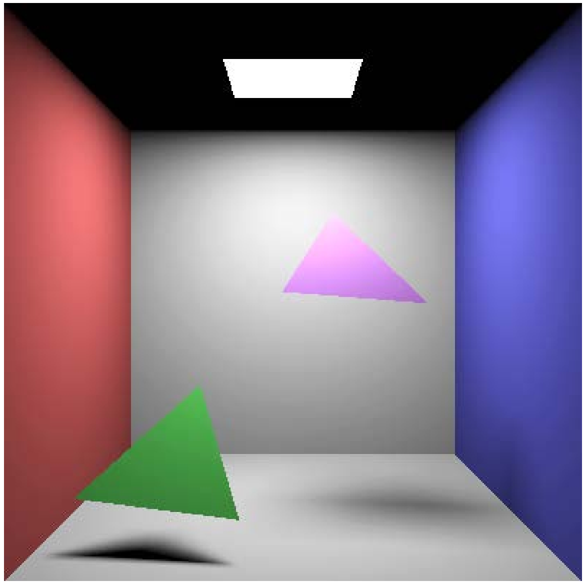


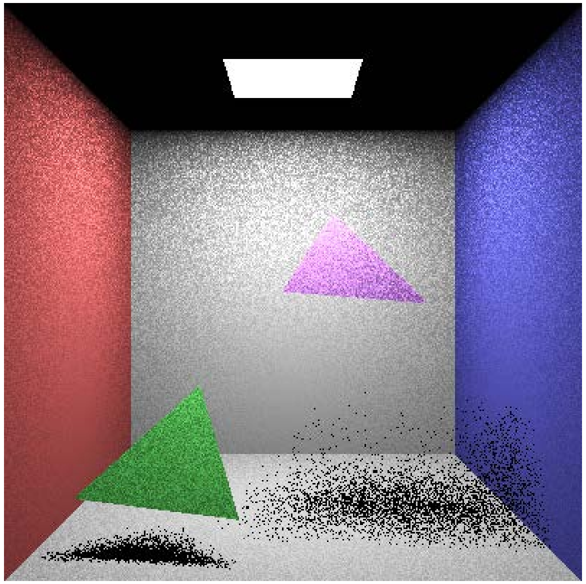
1 point on light source (*1 shadow ray*)

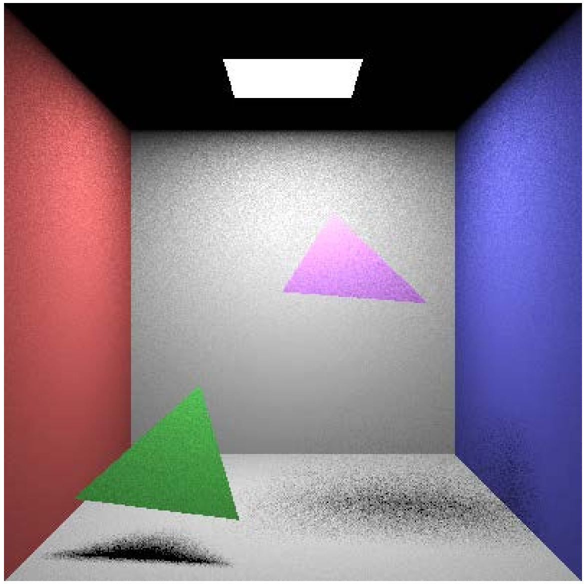
$$pdf(p') = \frac{1}{Area_{source}}$$

$$L_o(p, \omega_o) \approx f_r \cdot L_e \cdot Area_{source} \frac{V(p, p') \cos\theta_p \cos\theta_{p'}}{r_{pp'}^2}$$

Direct diffuse illumination


reference

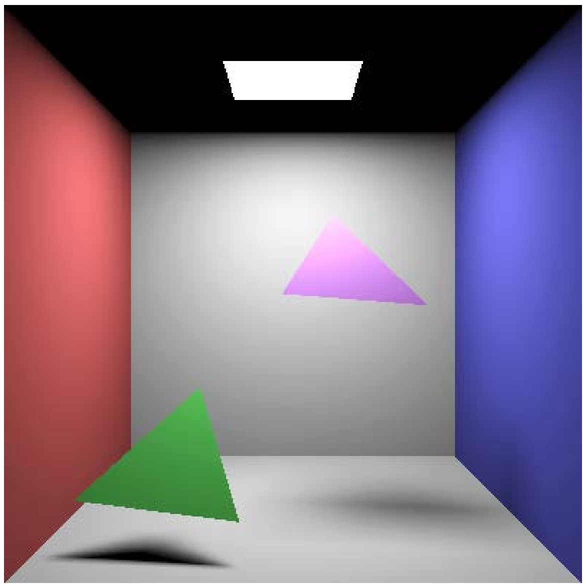

1 shadow ray

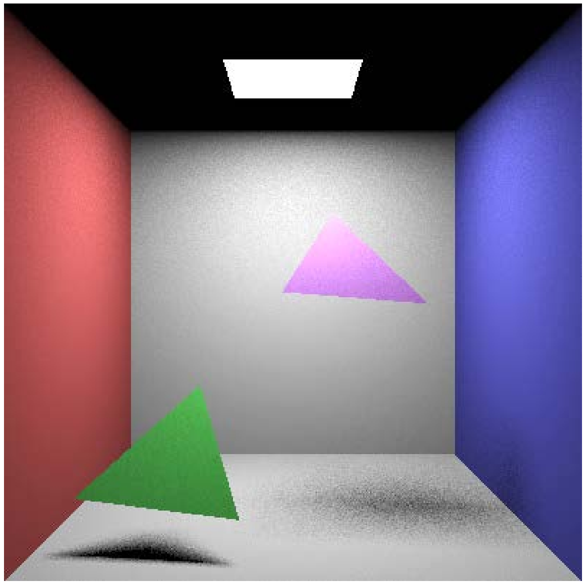

9 shadow rays

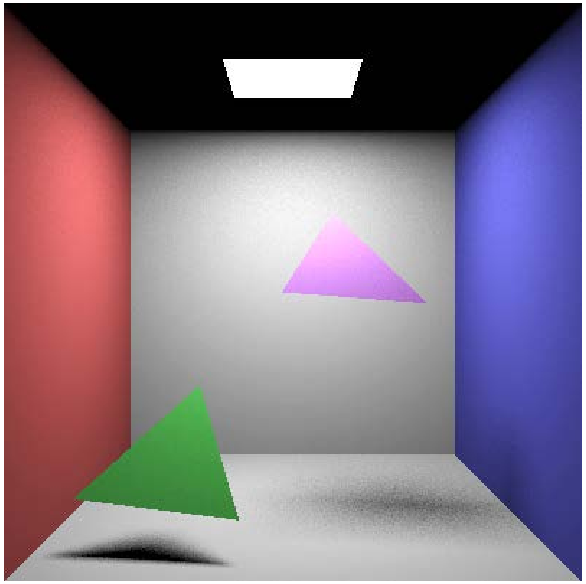
$$pdf(p') = \frac{1}{Area_{source}}$$

$$L_o(p, \omega_o) \approx f_r \cdot L_e \cdot Area_{source} \frac{V(p, p') \cos\theta_p \cos\theta_{p'}}{r_{pp'}^2}$$

Direct diffuse illumination


reference


36 shadow rays


100 shadow rays

$$pdf(p') = \frac{1}{Area_{source}}$$

$$L_o(p, \omega_o) \approx f_r \cdot L_e \cdot Area_{source} \frac{V(p, p') \cos\theta_p \cos\theta_{p'}}{r_{pp'}^2}$$

Choices for Direct illumination

Sampling scheme for viewing rays

- "How to select viewing rays within pixel area?"

Sampling scheme for shadow rays

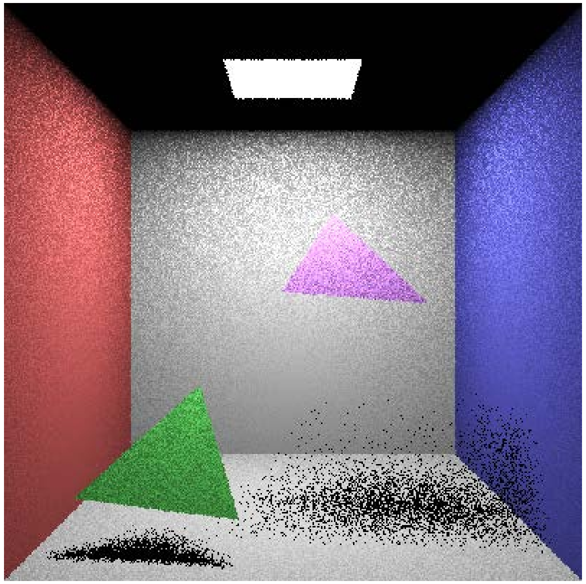
- "How to select shadow rays over the light source?"

'path tracing':

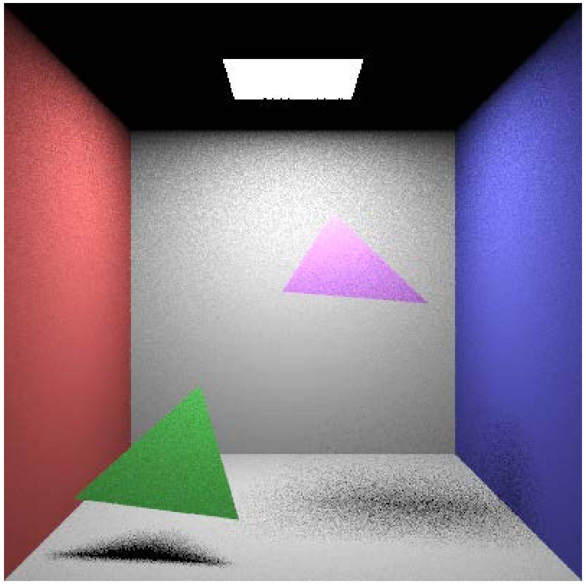
- Exactly 1 shadow ray per viewing ray
- Power of Monte Carlo: 1 sample in multi-dimensional space
  - 2 random numbers for position in pixel
  - 2 random numbers for position of shadow ray on light source

  ➔ 1 single quadruplet of random numbers

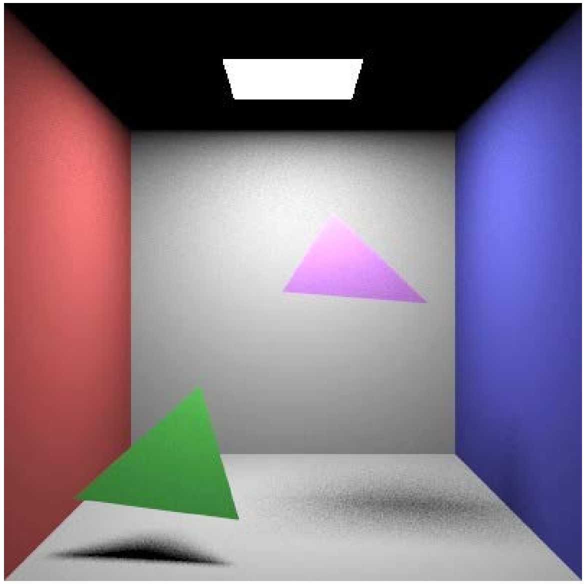
Path tracing
Direct illumination



1 viewing ray / pixel
1 sh-ray per viewing ray

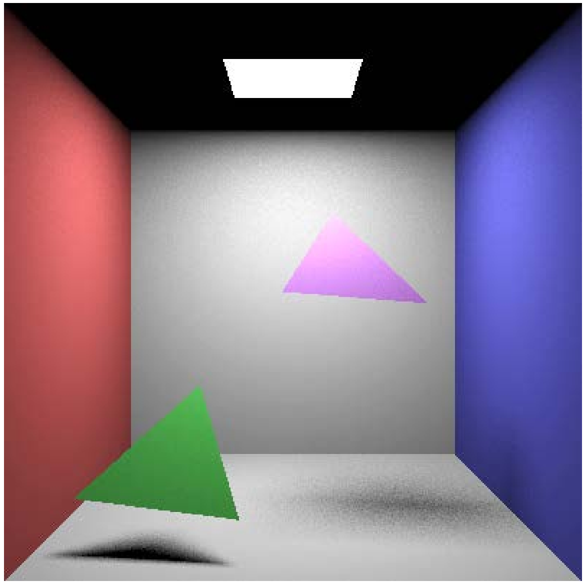


10 viewing rays / pixel
1 sh-ray per viewing ray



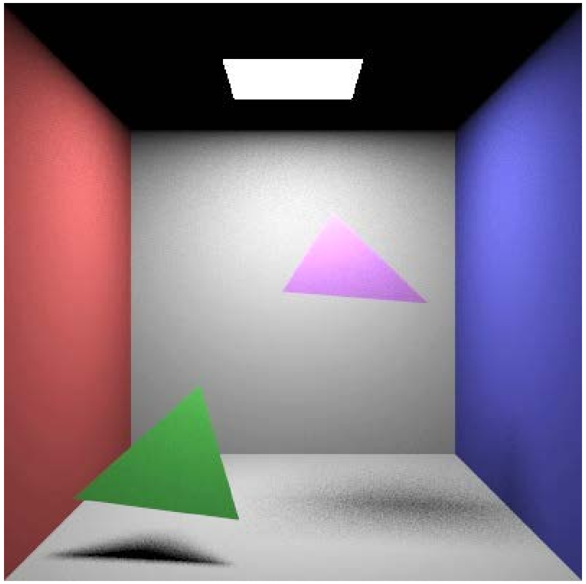
100 viewing rays / pixel
1 sh-ray per viewing ray

Path tracing
Direct illumination



1 centered viewing ray
100 random shadow rays per viewing ray

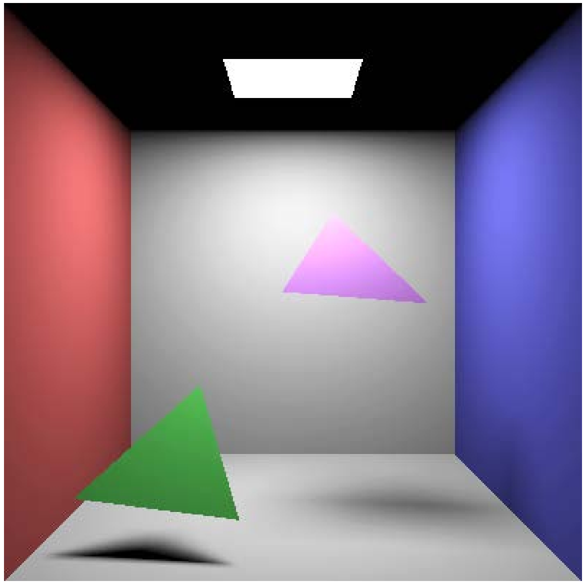
101 rays total per pixel



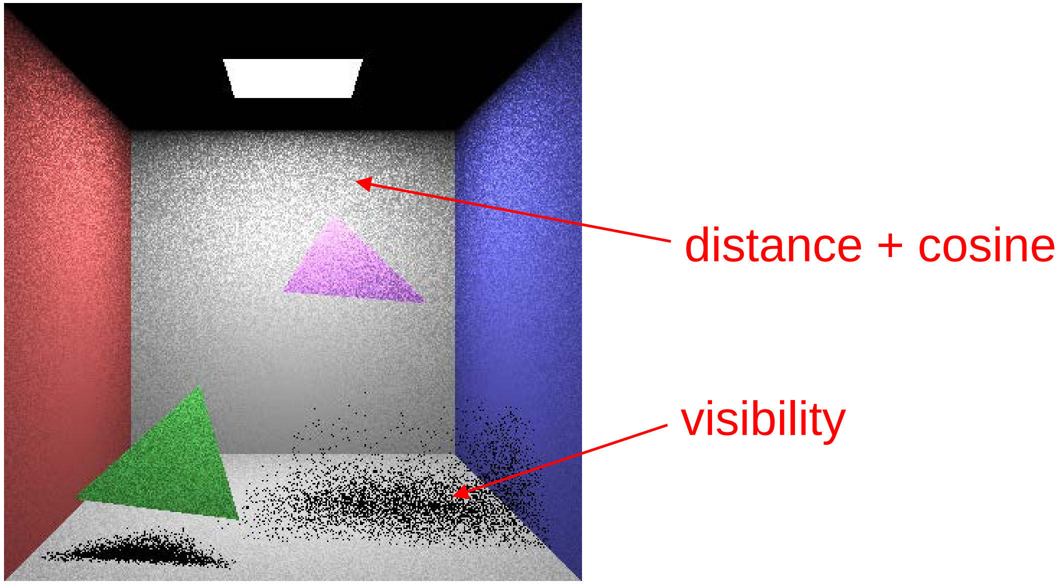
100 random viewing rays
1 random shadow ray per viewing ray

200 rays total per pixel

Sources of Noise



reference

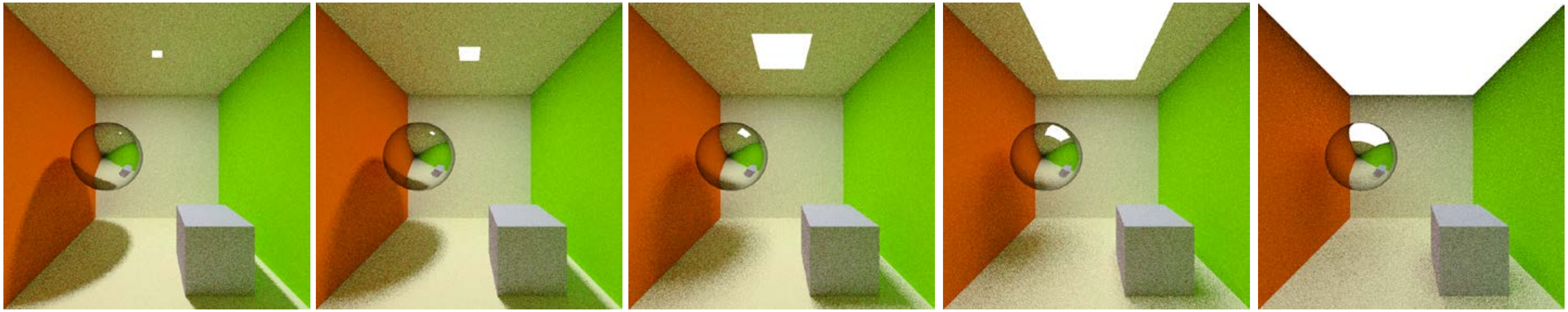


1 viewing ray / pixel
1 sh-ray per viewing ray

$$L_o(p,\omega_o) \approx f_r \cdot L_e \cdot Area_{source} \frac{V(p,p')\cos\theta_p \cos\theta_{p'}}{r_{pp'}^2}$$

Sources of Noise

Size of light source



$$L_o(p,\omega_o) \approx f_r \cdot L_e \cdot Area_{source} \frac{V(p,p')\cos\theta_p \cos\theta_{p'}}{r_{pp'}^2}$$

Direct + Indirect Summary

Direct illumination

- Shoot shadow rays towards light source
- Path tracing: usually one shadow ray per incoming ray
  - (one shadow ray per light source, but techniques exist to use one shadow ray for all light sources)

Indirect illumination

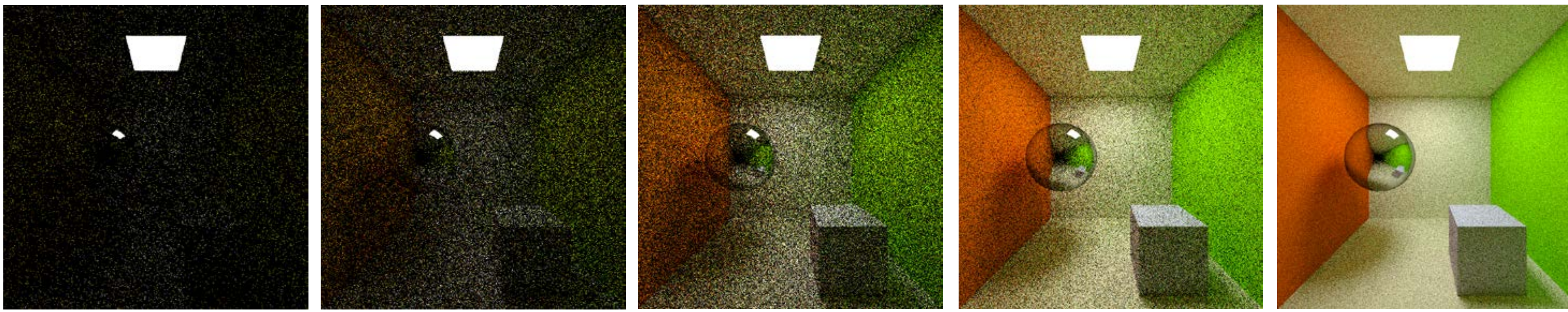
- Shoot indirect rays over hemisphere, evaluate recursively
- Path tracing: usually one indirect ray per incoming ray

Anti-aliasing

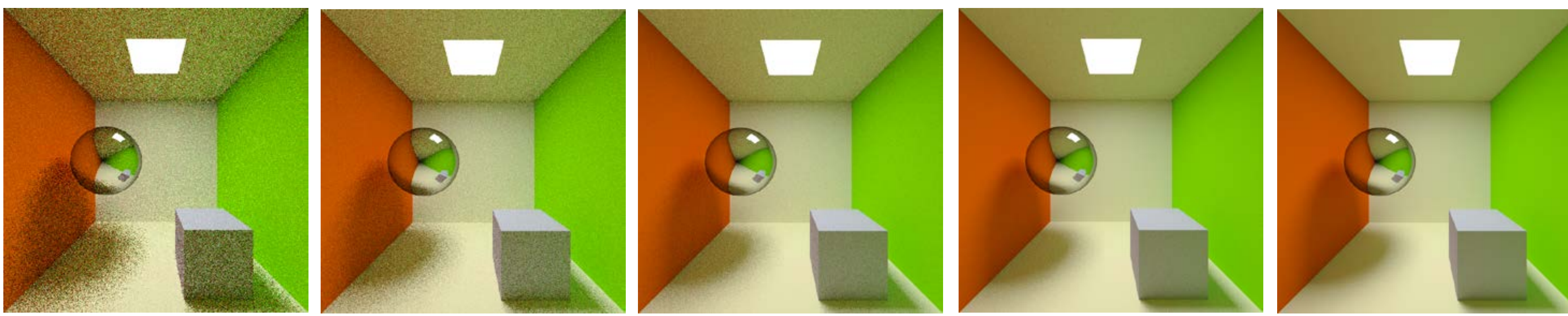
- Multiple viewing rays per pixel

Direct + Indirect Summary

Path tracing, no explicit sampling of light sources



1 path per pixel 4 16 64 256

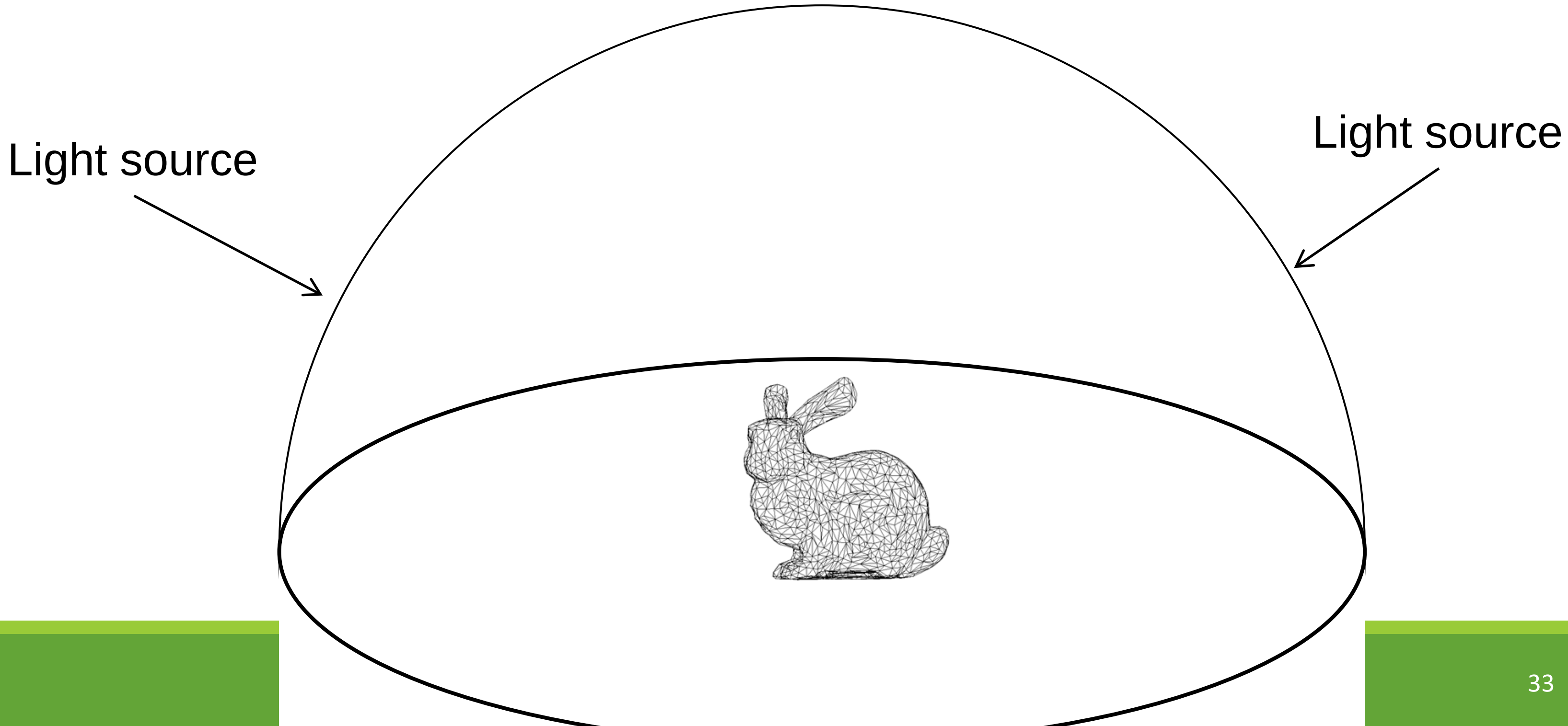


Path tracing, direct + indirect computed separately

Environment Lights

"Light that surrounds the entire scene"

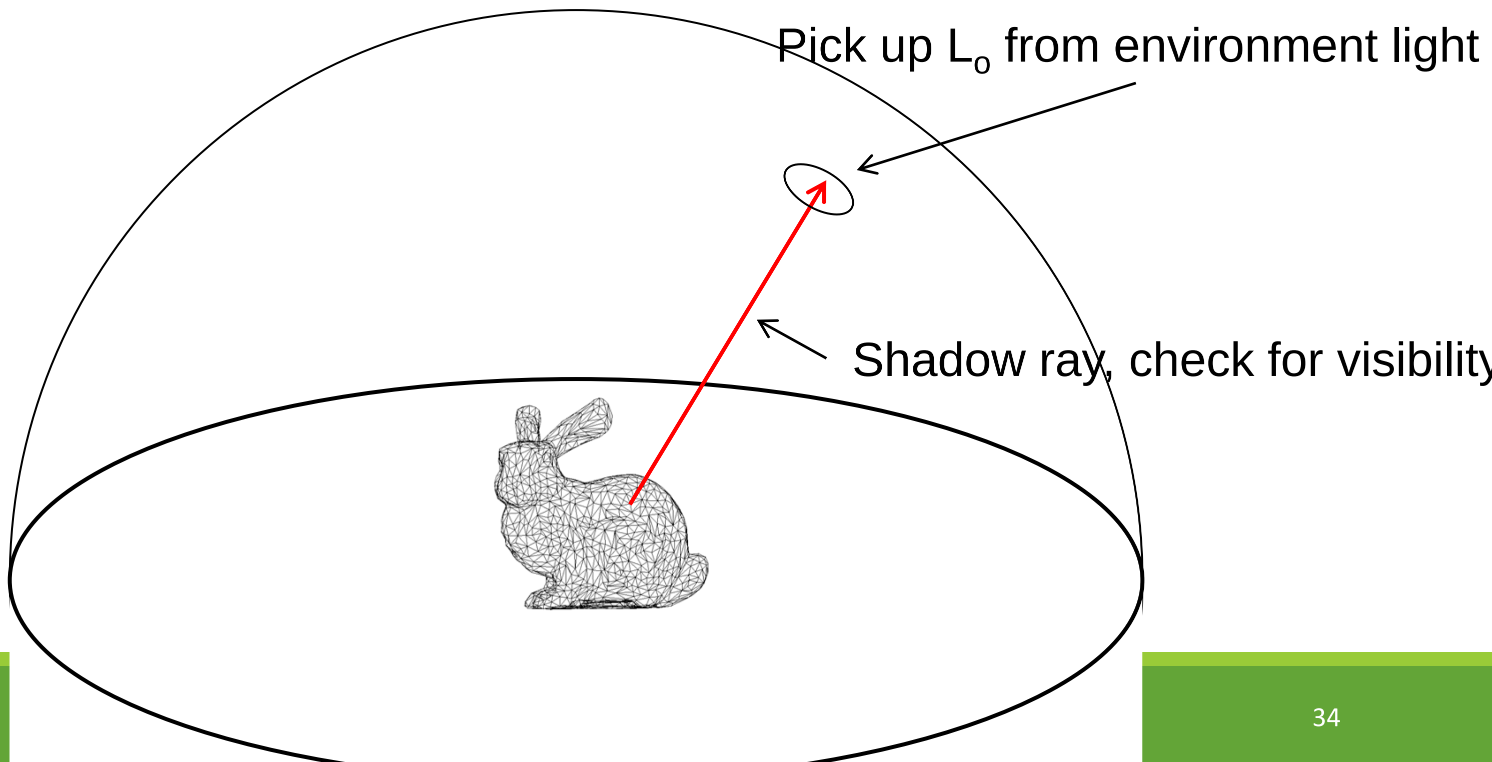
- (directional) light source is modelled as (enormous) sphere around the scene



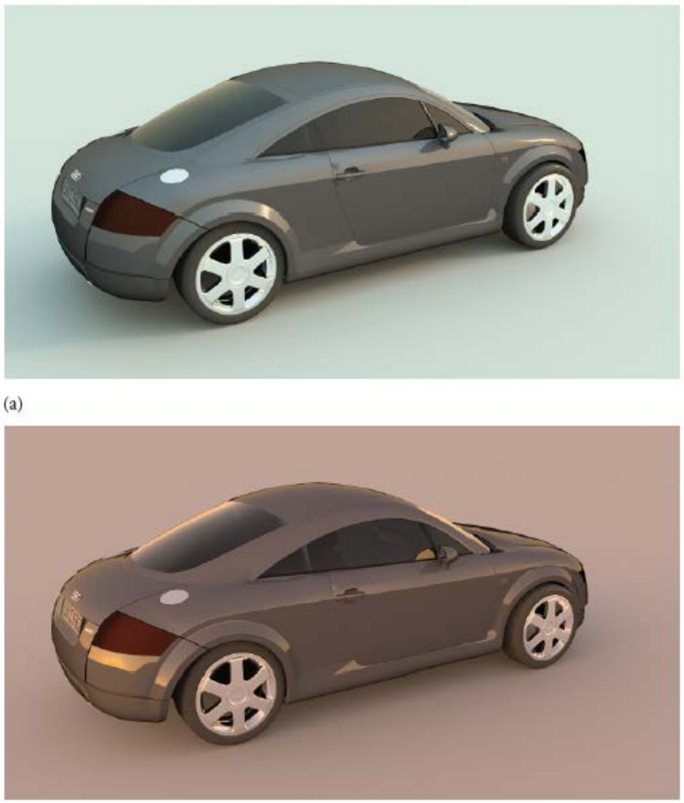
Environment Lights

Write direct illumination as integral over hemisphere directions:

$$L_o(p,\omega_o) = L_e(p,\omega_o) + \int_{hemisphere} f_r(p,\omega_i,\omega_o) L_o(-\omega_i) V(\omega_i) \cos d\,\omega_i$$



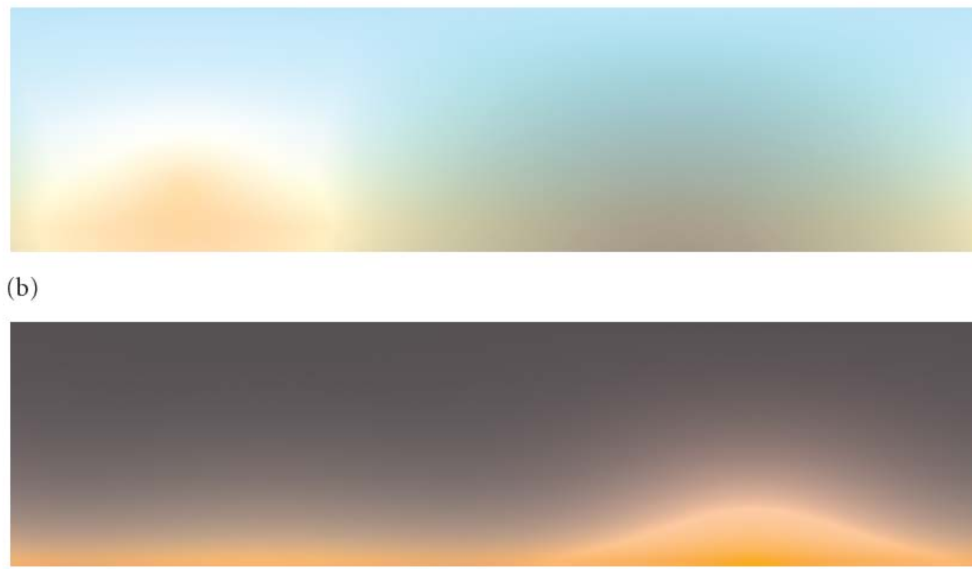
Environment Lights



(a)

(b)

Figure 12.15: Changing just the environment map used for illumination gives quite different results in the final image: (a) using a midday skylight distribution and (b) using a sunset environment map.



(b)

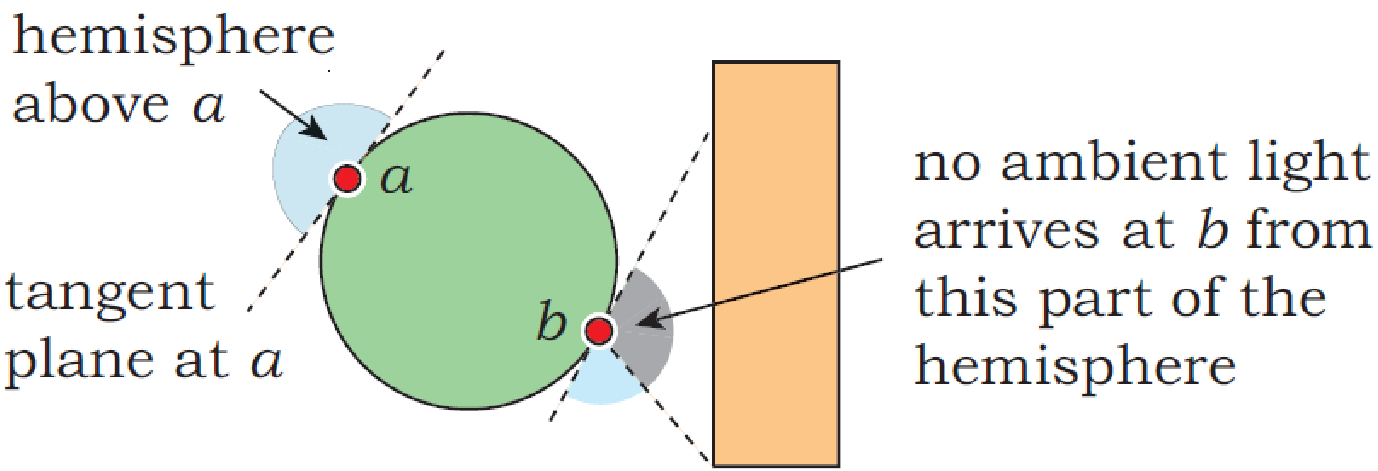
(c)

Figure 12.16: Environment Maps Used for Illumination in Figures 12.14 and 12.15. (a) Morning, (b) midday, and (c) sunset sky. (The bottom halves of these maps aren't shown here, since they are just black pixels.)

Ambient occlusion

Intuition:

- Amount of light received at a point is proportional to part of hemisphere that is not occluded



Ambient occlusion: formulation

$$L_o(p,\omega_o) = [L_e(p,\omega_o)] + \int_{hemisphere} f_r(p,\omega_i,\omega_o) L_i(p,\omega_i)\cos\theta_i \, d\omega_i$$

Incoming light equal intensity from all directio
but some light might be blocked

$$L_o(p,\omega_o) = L_i \int_{hemisphere} f_r(p,\omega_i,\omega_o) V(p,\omega_i)\cos\theta_i \, d\omega_i$$

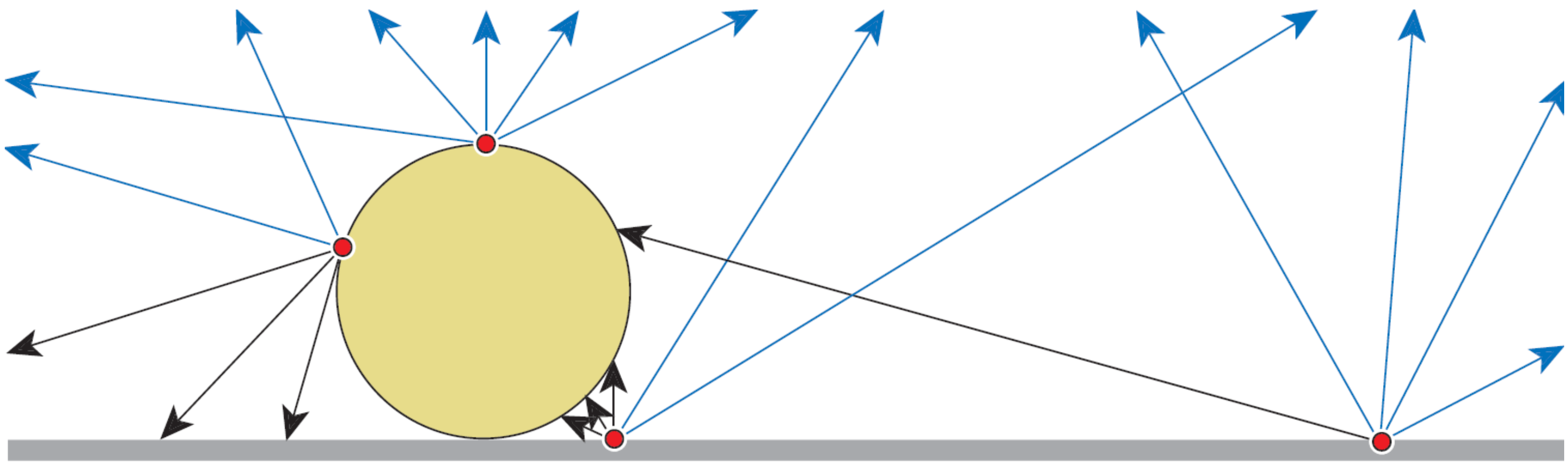
Diffuse brdf (brdf = constant)

$$L_o(p,\omega_o) = f_r \cdot L_i \int_{hemisphere} V(p,\omega_i)\cos\theta_i \, d\omega_i$$

Ambient occlusion: formulation

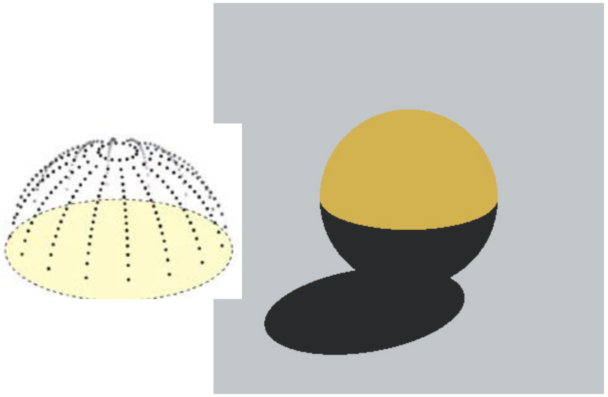
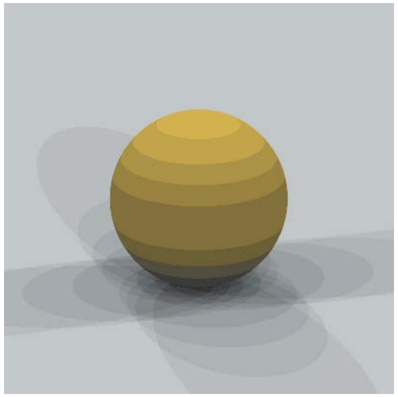
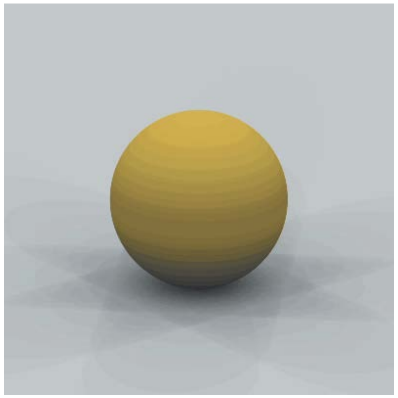
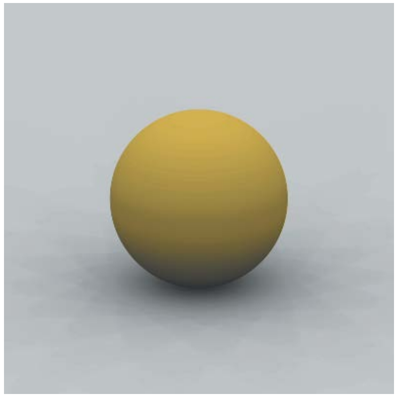
$$L_o(p,\omega_o) = f_r \cdot L_i \underbrace{\int_{hemisphere} V(p,\omega_i)\cos\theta_i \, d\omega_i}$$

- Can be pre-computed (e.g. at vertices)
- Only dependent on geometry

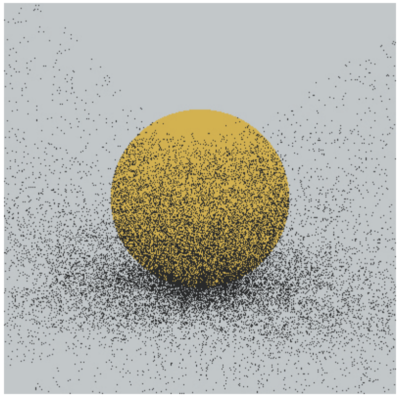
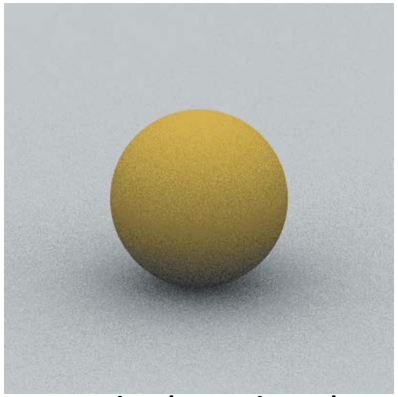
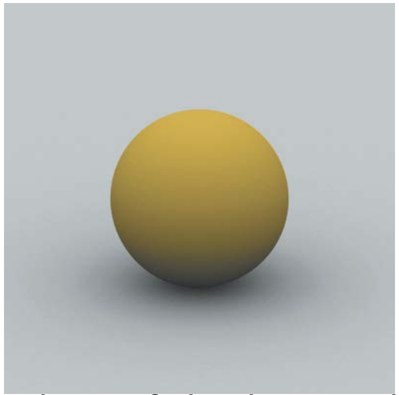
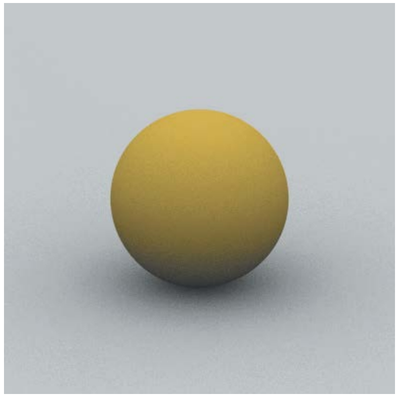


Ambient occlusion: sampling

Regular sampling of the hemisphere

1 sample | 16 samples | 64 samples | 256 samples

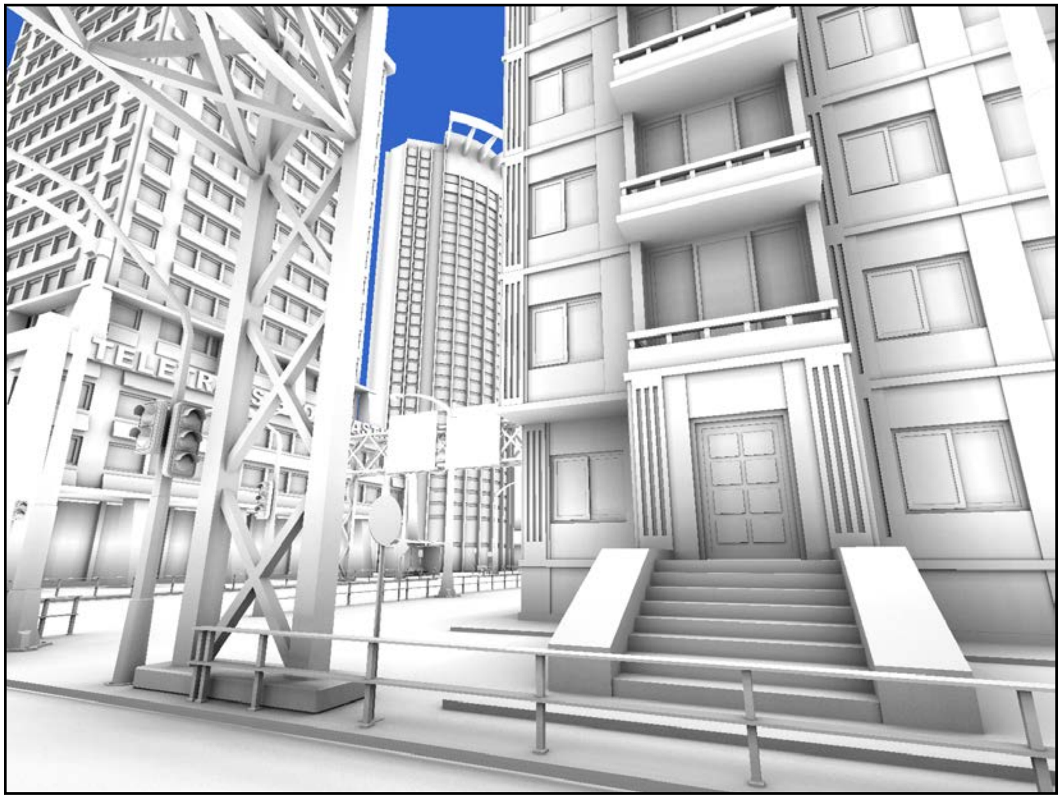





Monte Carlo (random) sampling of the hemisphere

Ambient occlusion: result

http://forums.3dtotal.com/showthread.php?t=75339

Ambient occlusion: result



Two Methods for Fast Ray-Cast Ambient Occlusion
Samuli Laine and Tero Karras (EGSR 2010)