Path Tracing

Michael Doggett
Department of Computer Science
Lund university

Outline

- Light transport notation
- Radiometry - Measuring light
- Illumination
  - Rendering Equation
  - Monte Carlo sampling
  - Russian roulette
- Path tracing
- Image based lighting

This is what we want:

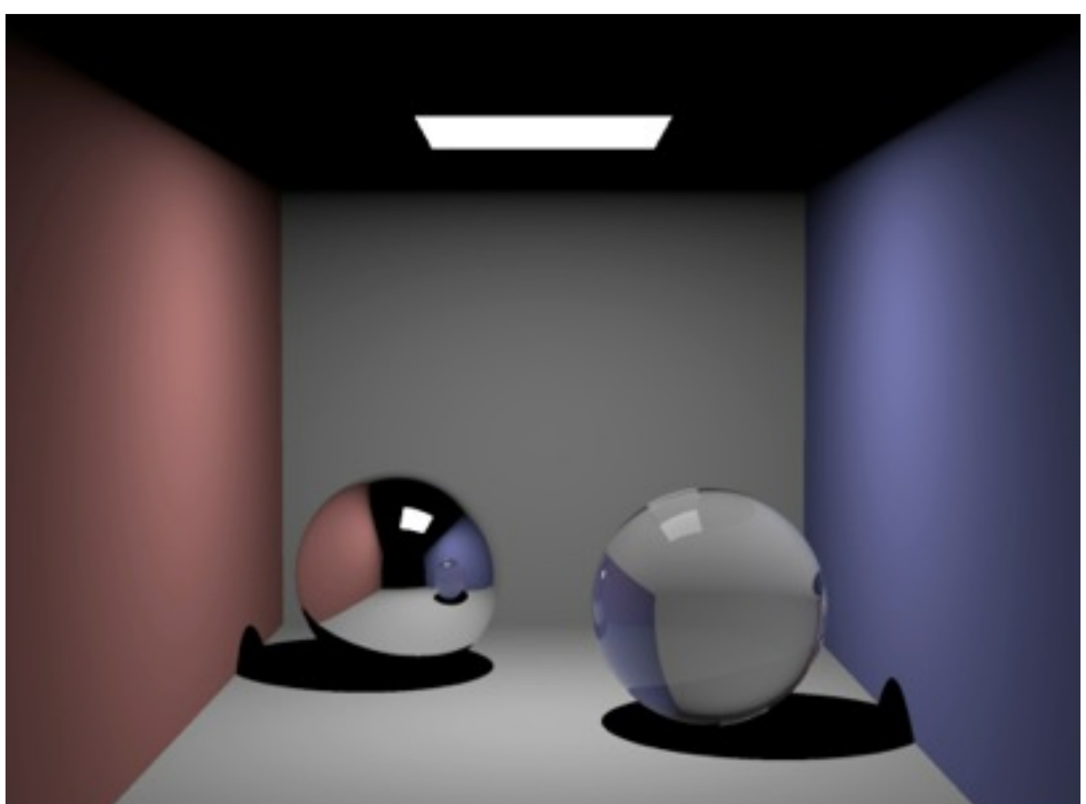
Courtesy of Henrik Wann Jensen

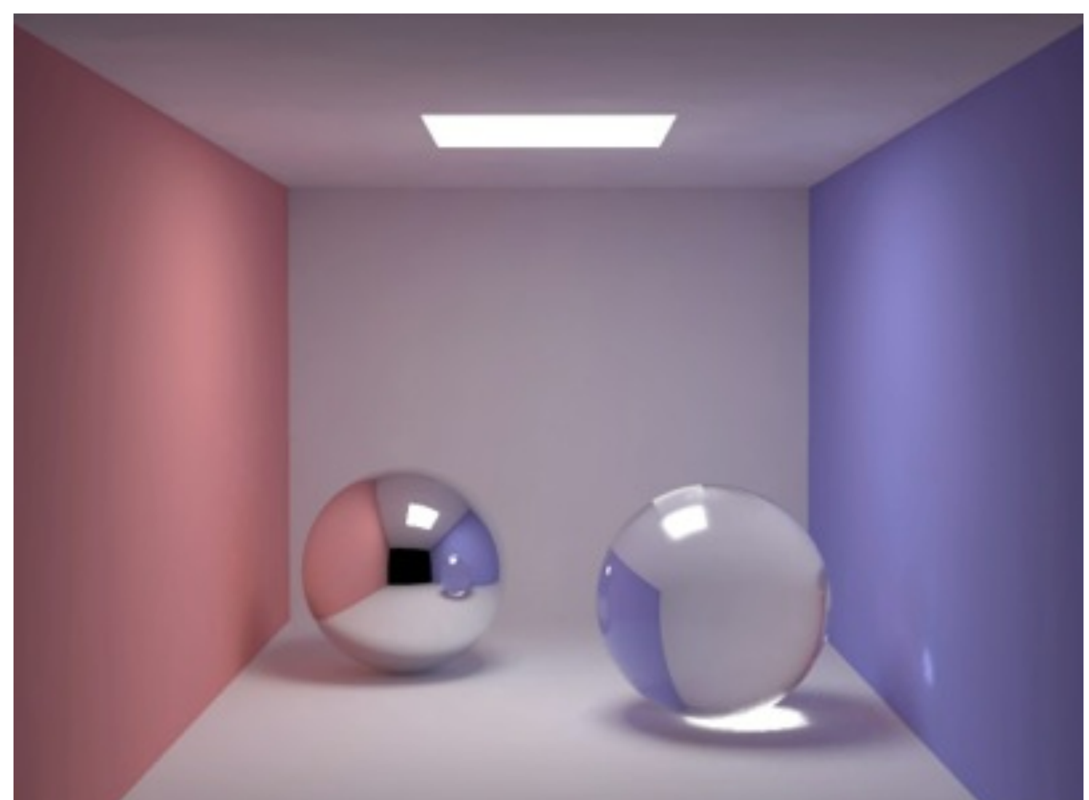
Courtesy of Paul Debevec

Images courtesy of Illuminate Labs

Isn't Whitted-style ray tracing enough?

- Does not give truly realistic images
- Why?
  - –Does not solve the entire rendering equation
  - –Miss several phenomenon:
    - Caustics
    - Indirect illumination (color bleeding, ambient lighting)
    - Soft shadows
- This lecture will present Global Illumination techniques: path tracing and image-based lighting.





Images courtesy of
Henrik Wann Jensen
University of California
San Diego

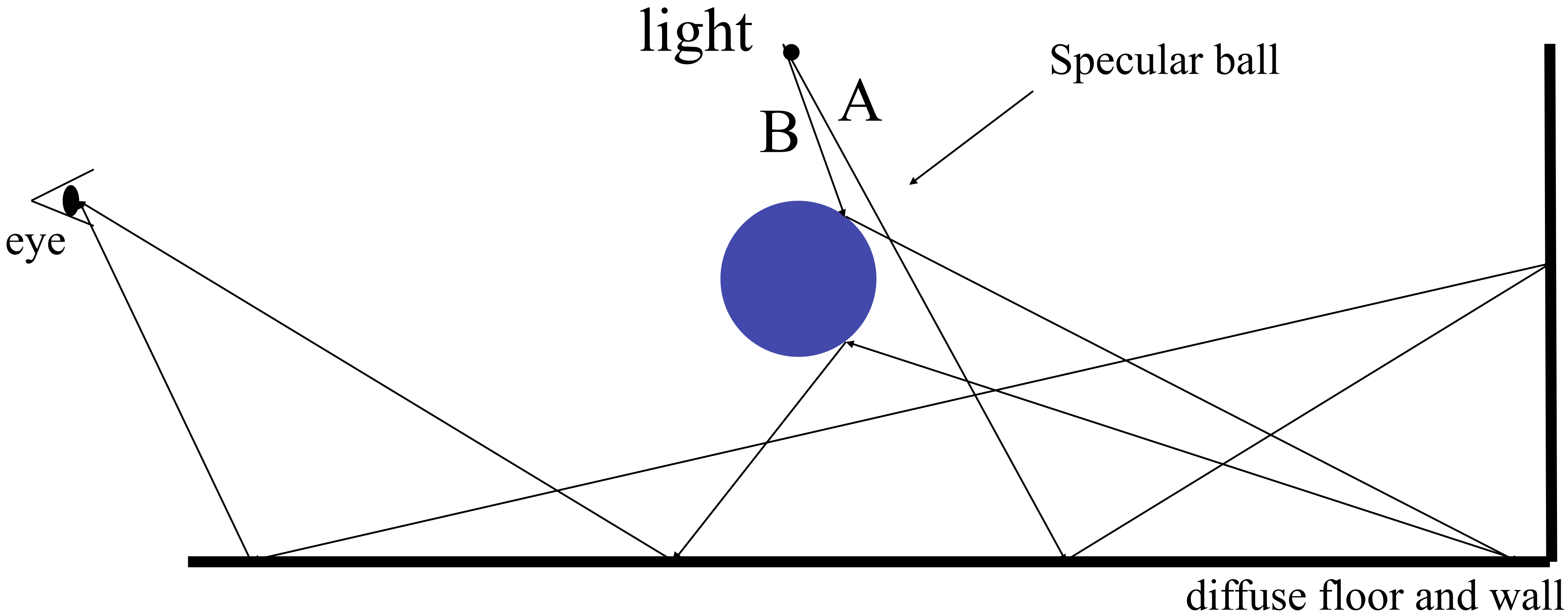
Light transport notation

Useful tool for thinking about global illumination (GI)

Notation after Paul Heckbert, SIGGRAPH 90

- Follow light paths
- The endpoints of straight paths can be:
  - L : light source
  - E : the eye
  - S : a specular reflection
  - D: a diffuse reflection
  - G: semi-diffuse (glossy) reflection
- Regular expressions can be used:
  - (K)+ : one or more of K
  - (K)*   : zero or more of K
  - (K)? : zero or one of K
  - (K | M) : a K or an M event

Examples of light transport notation



- Path A: LDDDE
- Path B: LSDSDE

Light transport notation: why?

- The ultimate goal is to simulate all light paths: L(S|G|D)*E
- Using this notation, we can find what Whitted ray tracing can handle:
  - LDS*E | LS*E = LD?S*E
    - Or if we include glossy surfaces: LD?(G|S)*E
  - This is clearly not L(S|G|D)*E !

The Rendering Equation

$$\boxed{L(x \rightarrow \Theta)} = L_e(x \rightarrow \Theta) + \int_{\Omega_x} f_r(x, \Psi \leftrightarrow \Theta) L(x \leftarrow \Psi) \cos(N_x, \Psi) d\omega_\Psi$$

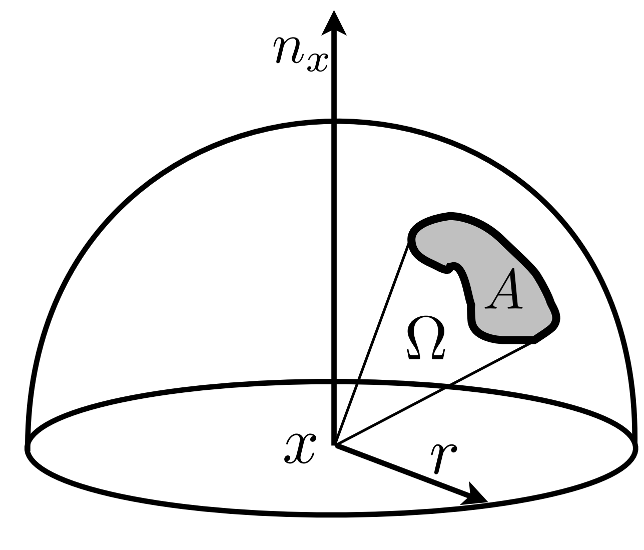
It "extends the range of optical phenomena which can be effectively simulated."
From paper of the same name, 'The Rendering Equation', by James T. Kajiya, SIGGRAPH '86

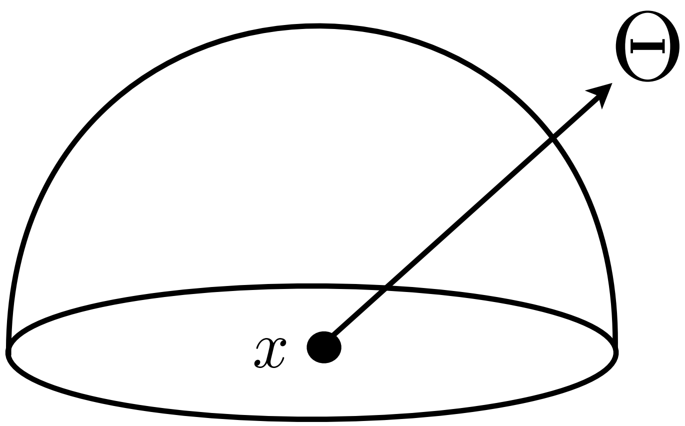
Radiometry

- Flux (radiant power): $\Phi[W]$ Big Phi
  - The total energy that flows from/to/through a surface
- Irradiance: $E = \dfrac{d\Phi}{dA}\left[\dfrac{W}{m^2}\right]$
  - Incoming radiant power per unit surface area
- Radiosity (radiant exitance)
  - Same as irradiance but outgoing

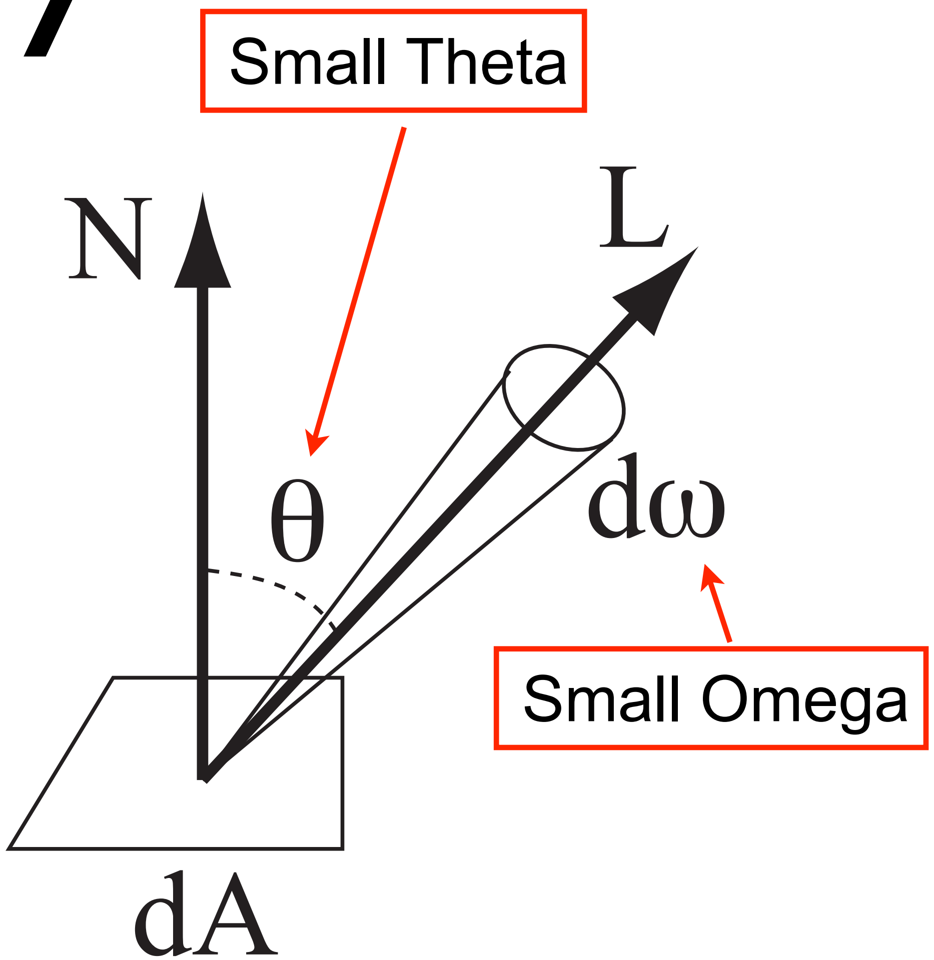
Radiometry

- **Solid Angle,** $\Omega = \frac{A}{r^2}$
  - Like the 2D version of an angle
- Shape of grey area doesn't matter
- Dimensionless
  - expressed in steradians
- Solid angle of sphere is 4π



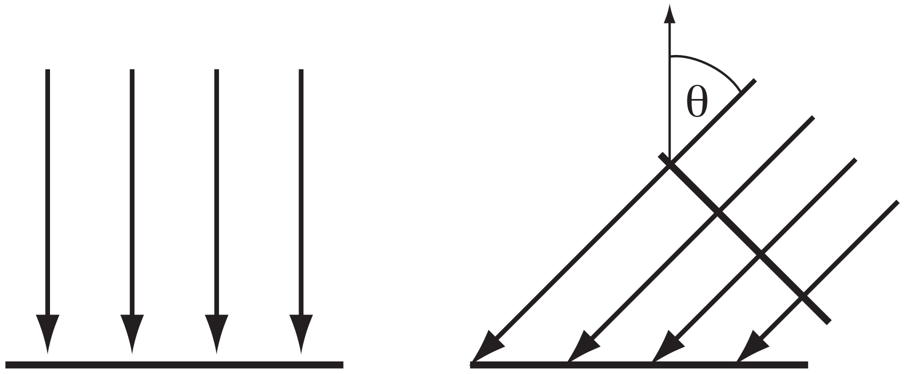


Radiometry



- Radiance: $L(x \to \Theta)$
  - L(x ->Θ) – 5 dimensions: position (x) and direction vector (Θ)
  - Important: captures ”appearance” of objects
  - Flux per unit projected area (E) per unit solid angle
    - Solid angle - subtends area on hemisphere

$$L = \frac{d^2\Phi}{d\omega dA^{\perp}} = \frac{d^2\Phi}{d\omega dA\cos\theta}$$



- Incoming flux is spread out over a larger area with bigger angles, the cosine term compensates for this

Rendering Equation

$$L(x \to \Theta) = L_e(x \to \Theta) + L_r(x \to \Theta)$$

L is radiance = most important quantity
- captures ”appearance” of objects in a scene.
- what we store in a pixel.

Energy conservation - total outgoing radiance at a point is the sum of the **emitted** and **reflected** radiance.

Incoming - Big Psi

$$L(x \to \Theta) = L_e(x \to \Theta) + \int_{\Omega_x} f_r(x, \Psi \leftrightarrow \Theta) L(x \leftarrow \Psi) \cos(N_x, \Psi) d\omega_\Psi$$

BRDF

The equation above is the **Rendering Equation** : *most important equation in graphics*

It is a Fredholm equation of the second kind: L is both to the left, and inside the integral

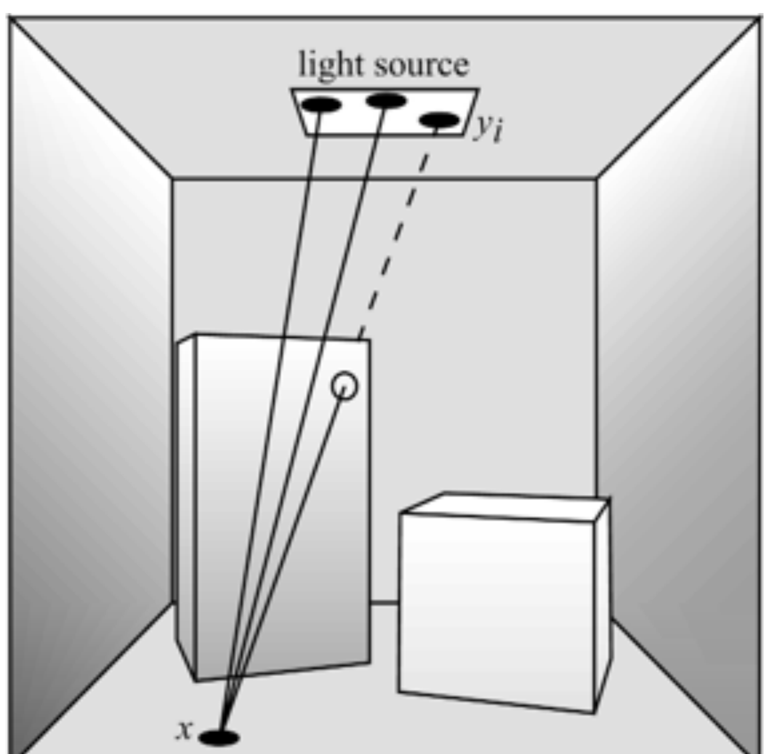
Section 11.6 in book, but it uses a different notation

Rendering Equation - L_r

$$L(x \to \Theta) = L_e(x \to \Theta) + \boxed{\int_{\Omega_x} f_r(x, \Psi \leftrightarrow \Theta) L(x \leftarrow \Psi) \cos(N_x, \Psi) d\omega_\Psi}$$

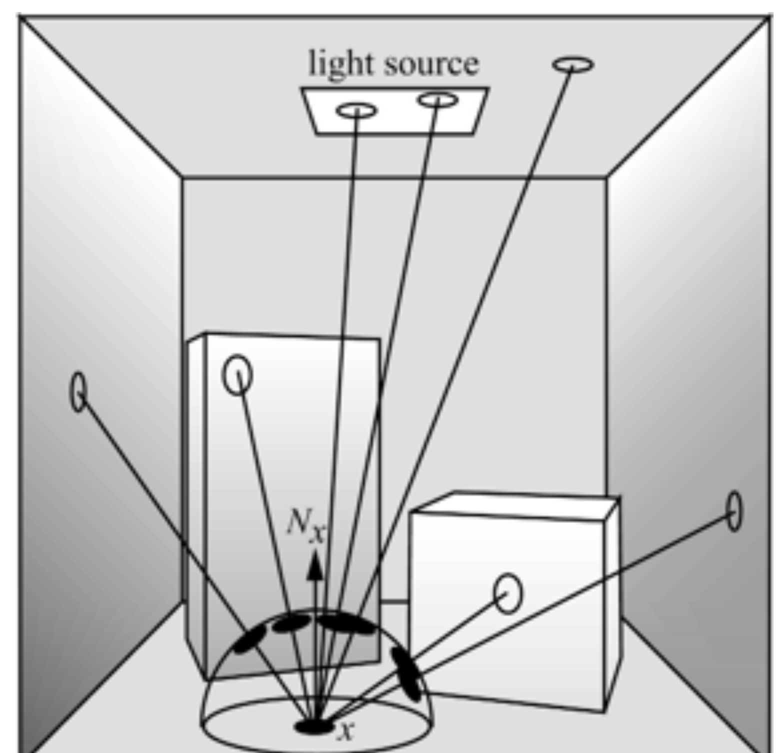
Why not sample light sources separately? They give most of the appearance (usually)!

$$L_r(x \to \Theta) = L_{direct} + L_{indirect}$$



Direct illumination

+



Indirect illumination

Direct Illumination: L_{direct}

Some definitions:

$$y = r(x, \Psi)$$

Is the closest positive intersection along ray that starts at x and has direction Ψ

$V(x,y)$ is the **visibility** function: is 0 if x occluded from y, otherwise 1.

Approach: integrate over the light surface area instead of over the hemisphere

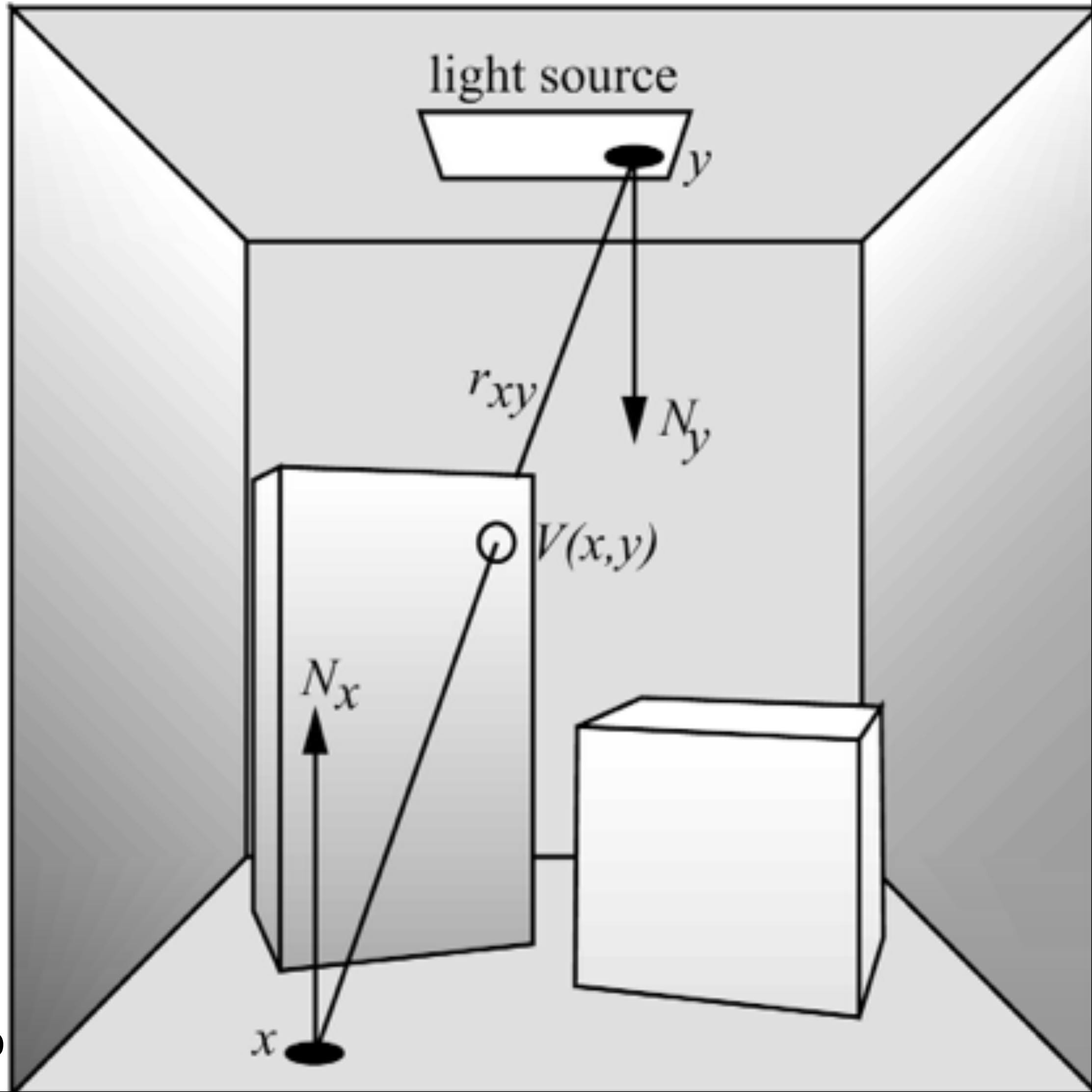
$$L_{direct} = \int_A f_r(x, \Theta \leftrightarrow \vec{xy}) L_e(y \rightarrow \vec{yx}) V(x,y) G(x,y) dA_y$$

The geometry term:

$$G(x,y) = \frac{\cos(N_x, \overrightarrow{xy}) \cos(N_y, \overrightarrow{yx})}{r_{xy}^2}$$

$$r_{xy} = \| x - y \|$$

$G(x,y)V(x,y)$ is often called ”radiance transfer”



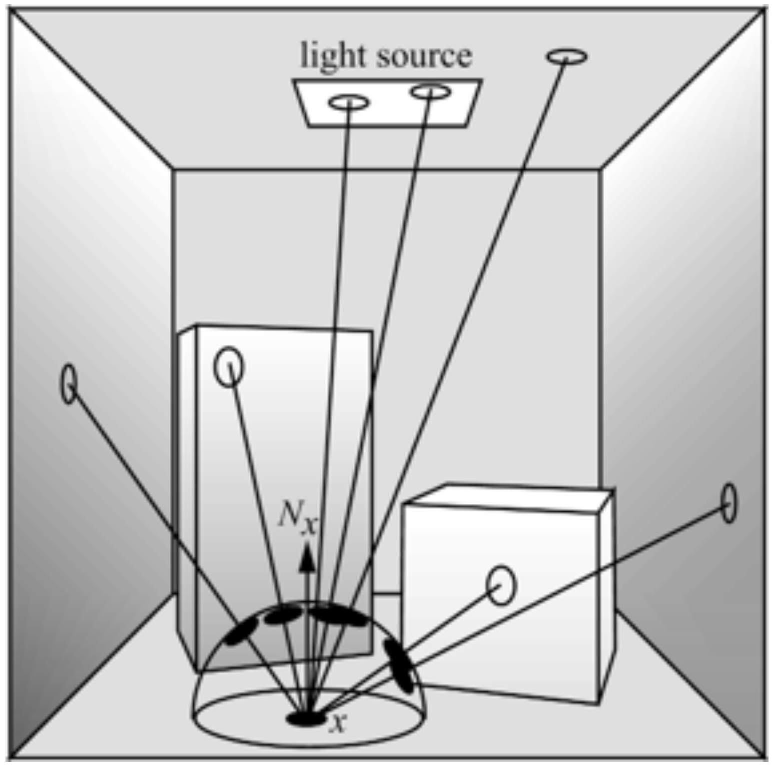
Indirect Illumination

$$L(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + \boxed{\int_{\Omega_x} f_r(x, \Psi \leftrightarrow \Theta) L(x \leftarrow \Psi) \cos(N_x, \Psi) d\omega_\Psi}$$

$$L_r(x \rightarrow \Theta) = L_{direct} + L_{indirect}$$

$$L_{indirect} = \int_{\Omega_x} f_r(x, \Psi \leftrightarrow \Theta) L_i(x \leftarrow \Psi) \cos(N_x, \Psi) d\omega_\Psi$$

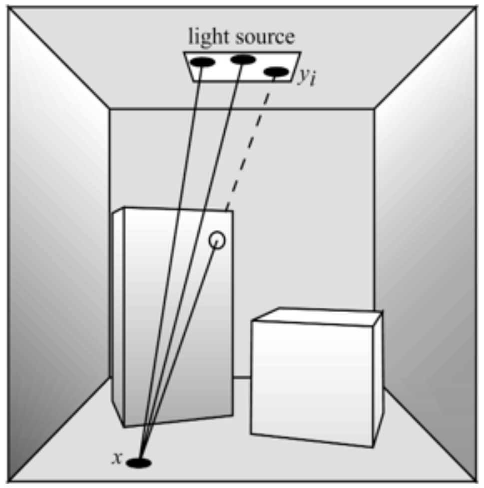
The incoming radiance at x in direction Ψ is the same as the outgoing radiance from y in direction $-\Psi$



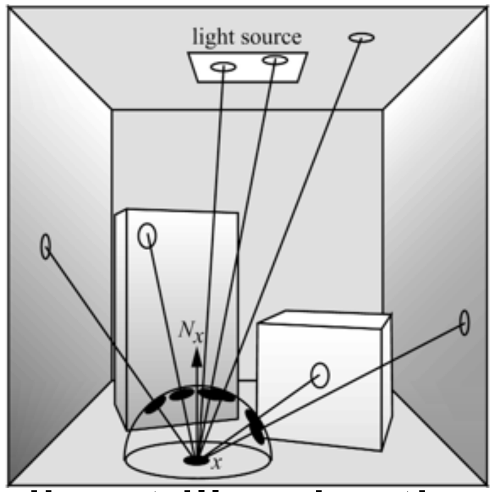
$$L_i(x \leftarrow \Psi) = L_r(r(x, \Psi) \rightarrow -\Psi)$$

$y = r(x, \Psi)$ the ray-casting operation, finds the closest positive intersection along ray that starts at x and has direction Ψ

GI illustrated



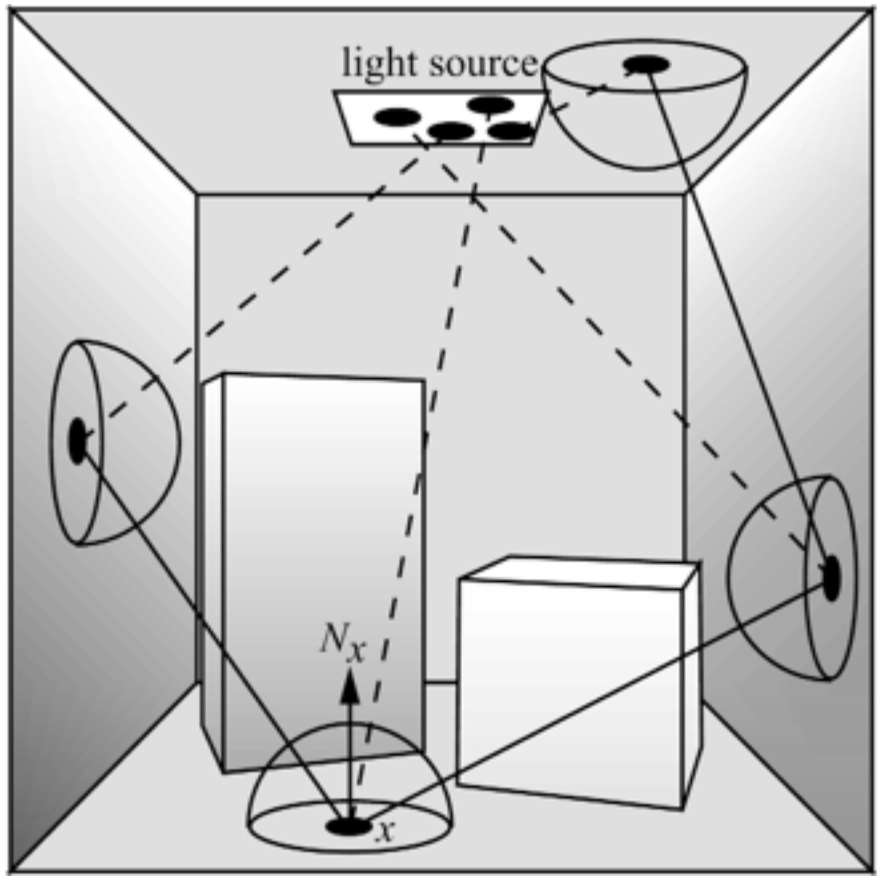
+



Direct illumination

Indirect illumination

Recursion Illustrated: (dashed lines are shadow rays)

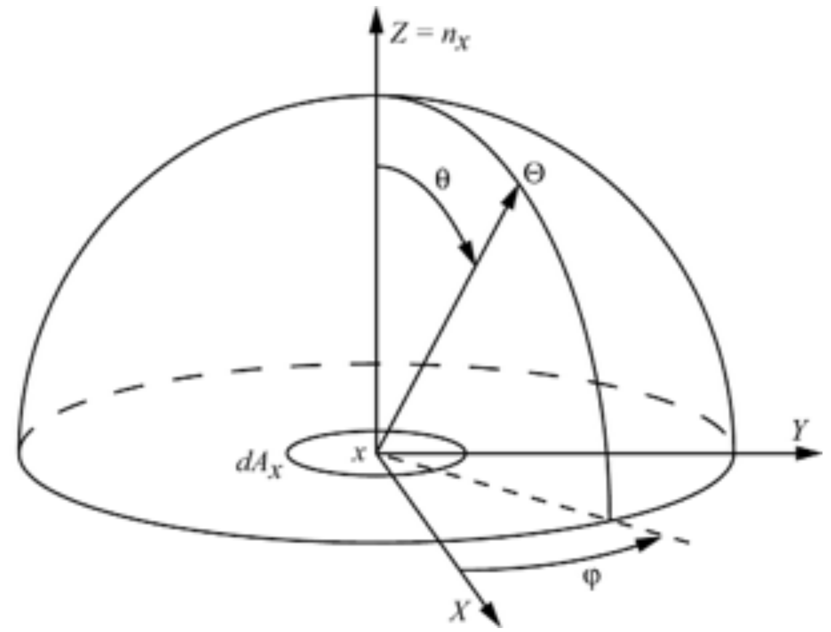


$L_{indirect}$

- Change from integrating Cartesian coordinate (solid angle) to (hemi-)spherical coordinates:

$$x = r\sin\theta\cos\varphi$$
$$y = r\sin\theta\sin\varphi$$
$$z = r\cos\theta$$

$$\int_{\Omega} f(\Theta)d\omega_{\Theta} = \int_{0}^{2\pi}\int_{0}^{\pi/2} f(\varphi,\theta)Jd\theta d\varphi$$



- J is the Jacobian of the coordinate transform

$$J = \begin{vmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \theta} & \frac{\partial x}{\partial \varphi} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \theta} & \frac{\partial y}{\partial \varphi} \\ \frac{\partial z}{\partial r} & \frac{\partial z}{\partial \theta} & \frac{\partial z}{\partial \varphi} \end{vmatrix} = \sin\theta$$

$$\Rightarrow \int_{\Omega} f(\Theta)d\omega_{\Theta} = \int_{0}^{2\pi}\int_{0}^{\pi/2} f(\varphi,\theta)\sin\theta d\theta d\varphi$$

Monte Carlo Integration

- Need to integrate incoming light over hemisphere
- Estimate definite integral using random samples

$$I_{MC} = (b-a)\frac{1}{n}\sum_{i=1}^{n} f(x_i)$$ Monte Carlo estimate

Monte Carlo Recap

X is stochastic random variable, drawn from PDF $p(x)$

$$E[X] = \int x p(x) dx$$

$$E[f(X)] = \int f(x) p(x) dx$$

How to evaluate the integral of $f(x)$ when x is drawn from PDF $p(x)$?

$$\langle I \rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)}.$$

x_i drawn from PDF

$$\begin{aligned} E[\langle I \rangle] &= E[\frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)}] = \frac{1}{N} \sum_{i=1}^{N} E[\frac{f(x_i)}{p(x_i)}] \\ &= \frac{1}{N} N \int \frac{f(x)}{p(x)} p(x) = \int f(x) dx = I \end{aligned}$$

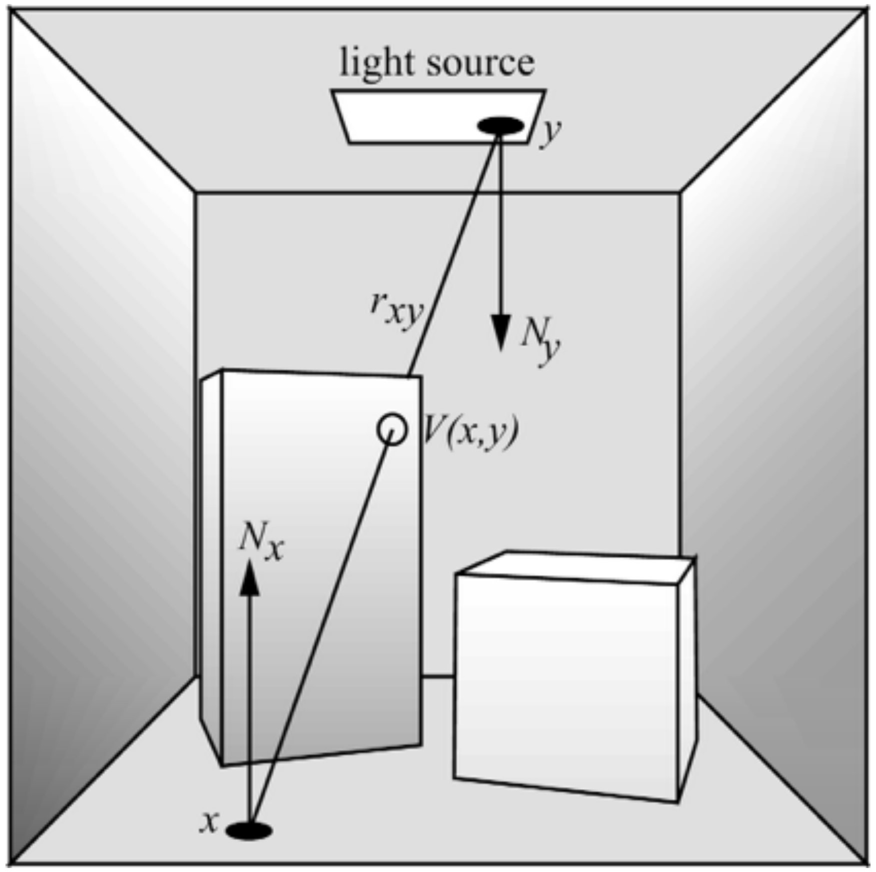
See chapter 10 for more explanation

Practical direct illumination using MC

$$L_{direct} = \int_A f_r(x, \Theta \leftrightarrow \vec{xy}) L_e(y \rightarrow \vec{yx}) V(x,y) G(x,y) dA_y$$

$$\approx \frac{1}{N_s} \sum_{i=1}^{N_s} \frac{f_r(x, \Theta \leftrightarrow \vec{xy_i}) L_e(y_i \rightarrow \vec{y_i x}) V(x, y_i) G(x, y_i)}{p(y_i)}$$

The PDF, *p(y)* may be uniform, i.e., 1/A



Russian roulette

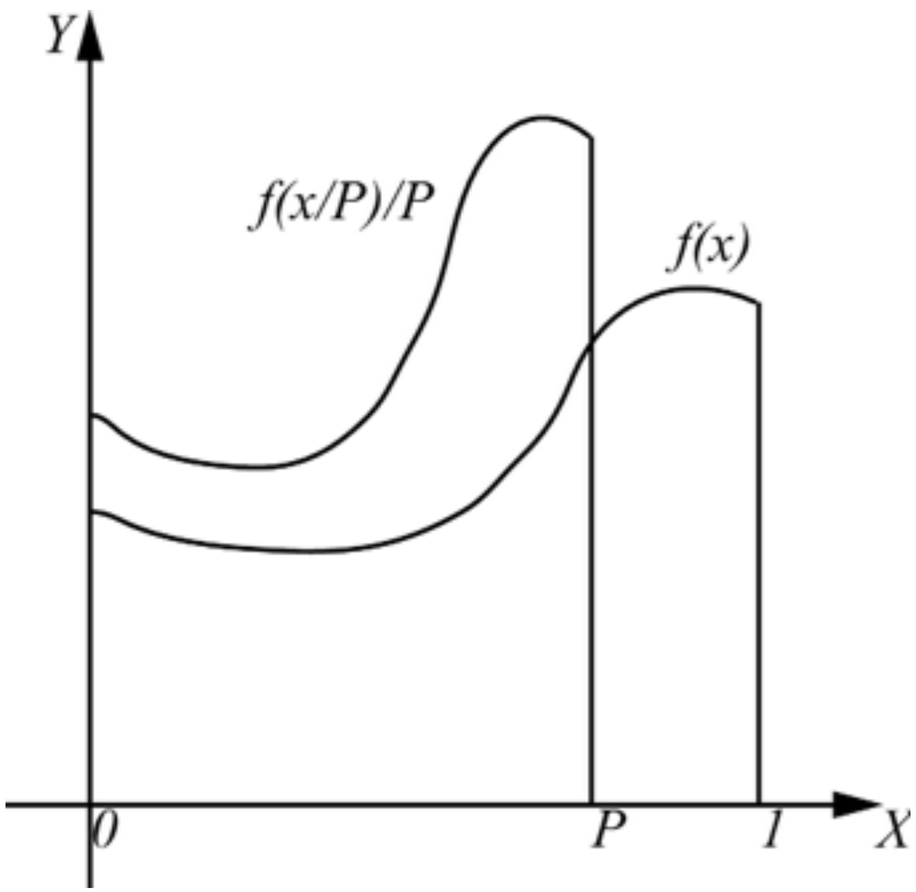
- We need to be able to terminate recursion
- Bias is the error (possibly) introduced with some technique: $E[I_{MC}]-I$ (mean of MC minus real solution)
- Avoid incorrect lighting contributions (bias)
  - –Light bounces around infinitely...
  - –How to avoid infinite path lengths without bias?

$$I = \int_0^1 f(x)dx$$

$$I_{RR} = \int_0^P \frac{1}{P} f(x/P)dx, \ \ P \le 1$$

$$< I_{RR} > = \begin{cases} \frac{1}{P} f(x/P), \text{if } x \le P \\ 0, \qquad\qquad\quad \text{if } x > P \end{cases}$$

$$E[< I_{RR} >] = (1-P)\cdot 0 + P \cdot \frac{E[I]}{P} = E[I]$$



Russian roulette for indirect illumination

- That is, we can stop recursion with a probability of $\alpha=1-P$, where α is the *absorption probability*
- If photon not absorbed, diffuse or specular?
  - Random number r from [0,1]
  - If($r < \rho_d$) then shoot diffuse ray
    - randomly on hemisphere ...
  - Else if($r < \rho_d+\rho_s$) then shoot specular ray
  - Else absorb

- Use average reflectance
  - E.g., $\rho_d=(\rho_{d,red} + \rho_{d,green} + \rho_{d,blue})/3$
  - Or special reflectivity parameter of material
    - As done in programming assignments

Practical indirect illumination

- How to choose random direction on the hemisphere?
  – Uniformly distributed

$$\int_{\Omega} f(\Theta)\, d\omega_{\Theta} = \int_{0}^{2\pi}\int_{0}^{\pi/2} f(\varphi,\theta)\sin\theta\, d\theta\, d\varphi$$

I.e., choose random θ and φ, and then use
The $\sin\theta$ to correct for non-uniform distribution
Not efficient!

- Instead:

$$\varphi = 2\pi r_1 \quad r_1, r_2 \text{ are random numbers in } [0,1]$$

$$\theta = \arccos(r_2)$$

$$\Psi_i = (\varphi, \theta)$$

$$L_{indirect} = \frac{1}{N}\sum_{i=1}^{N} \frac{f_r(x, \Psi_i \leftrightarrow \Theta) L_r(r(x,\Psi_i) \rightarrow -\Psi_i)\cos(N_x, \Psi_i))}{p(\Psi_i)}$$

PDF for $p(\Psi_i)$ is $1/2\pi$

Importance Sampling

- Choose the PDF proportional to a factor in the equation that we want to integrate
  - i.e. send more rays near the normal
- For rendering equation, this can be:
  - $\cos(\Psi_{\mathrm{i}}, N_{\mathrm{x}})$
    - Included in programming assignment 3
  - BRDF
  - Incident radiance
  - Combination...

Path tracing algorithm

- Shoot many rays through each pixel
  - Compute direct illumination
  - Compute indirect illumination (recursive)
    - Use Russian roulette to terminate recursion without bias
    - Use Hemispherical Monte Carlo sampling

- Important for path tracing:
  - Shoot **one** ray to the light sources
  - Shoot **one** ray over the hemisphere
  - Reason: do not want to spend more rays that are deeper down in the paths

Conventional Ray Tracing vs. Path Tracing

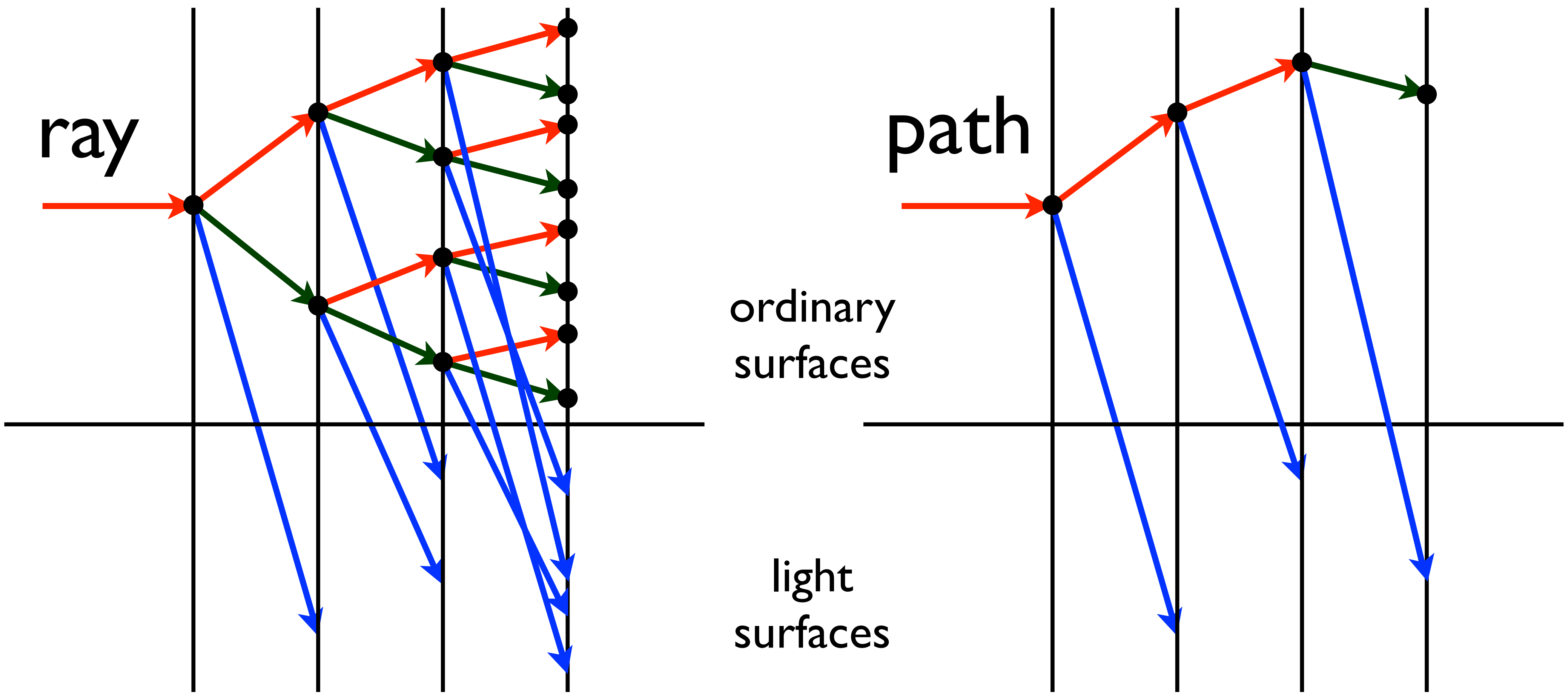
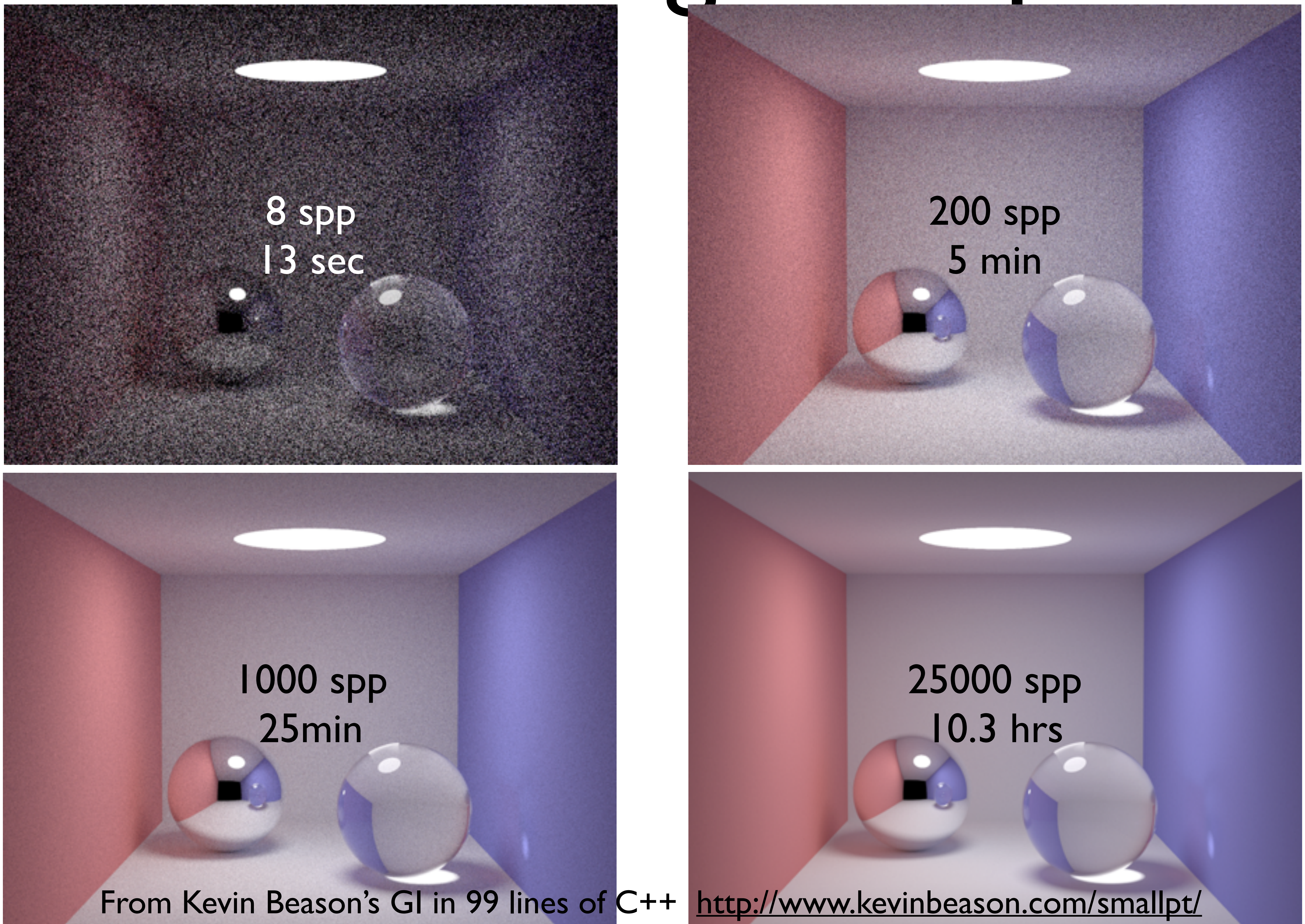


figure after “The Rendering Equation”, Jim T. Kajiya, ACM SIGGRAPH 1986

Path tracing example



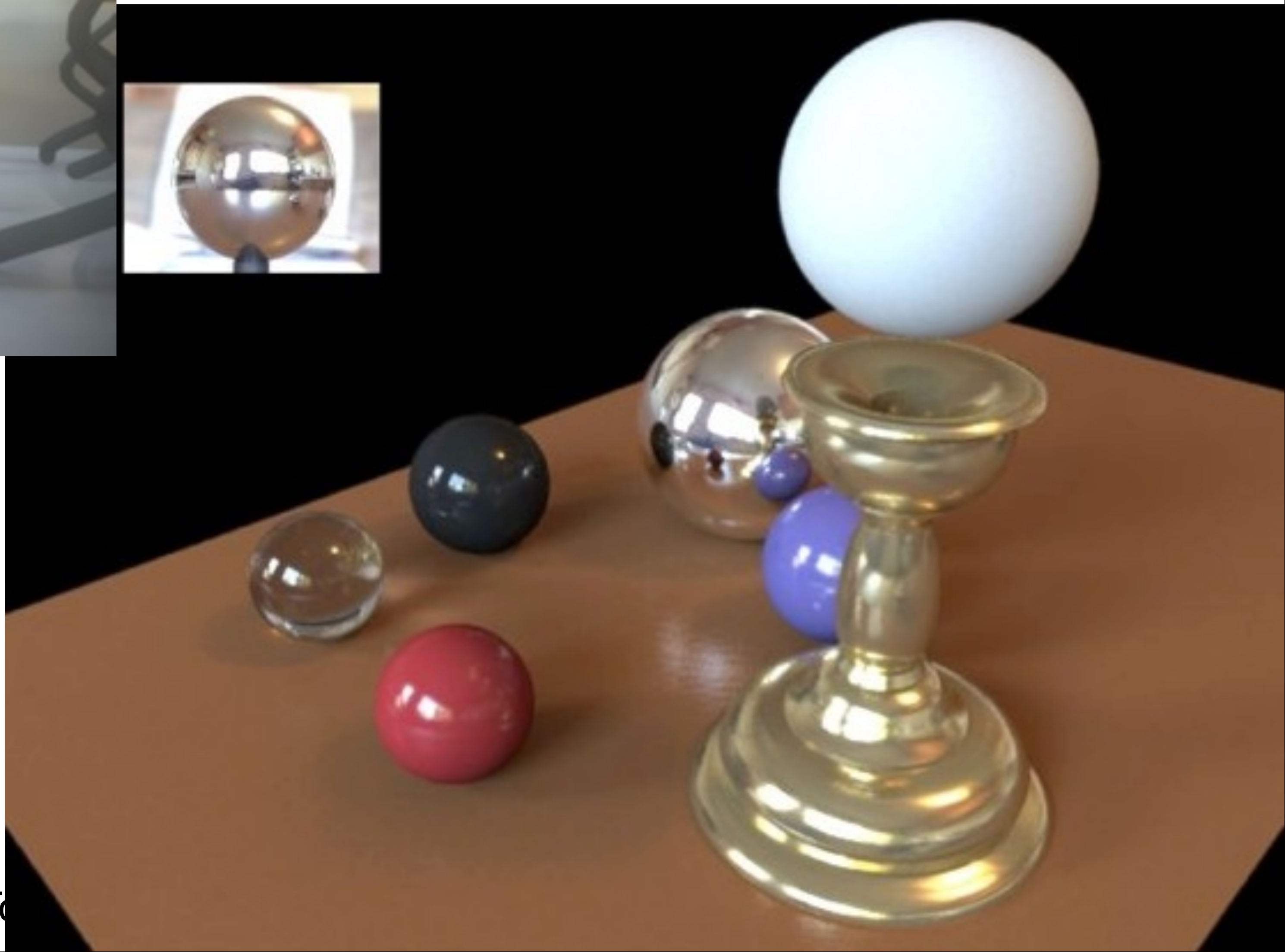
From Kevin Beason's GI in 99 lines of C++ http://www.kevinbeason.com/smallpt/

Image Based Lighting

Image-Based Lighting

- Enables very realistic lighting
  - In a simple way
- Not only primary light sources illuminate a point
- Basic idea:
  - Capture an image of the environment
  - Each pixel in this image is treated as a light source
  - During rendering, use these lights!

Example images



Images courtesy of
Paul Debevec

More example images

Images courtesy of Henrik Wann Jensen

Capturing environment maps (light probes)

- Many ways:
  - Take a photograph of a highly reflective sphere
  - Fish-eye lens: 2 images needed
  - Special device: rotating image sensors
- Need high dynamic range images!
  - 3 bytes not enough for RGB → 3 floats
- Why?
  - Because the range of ”brightness” values is larger than [0,255]

Example of image with high dynamic range

Image courtesy of Paul Debevec

- Lowest intensity is about 20
- Highest intensity is about 12,000
- Dynamic range is 1:600 !

High Dynamic Range Images

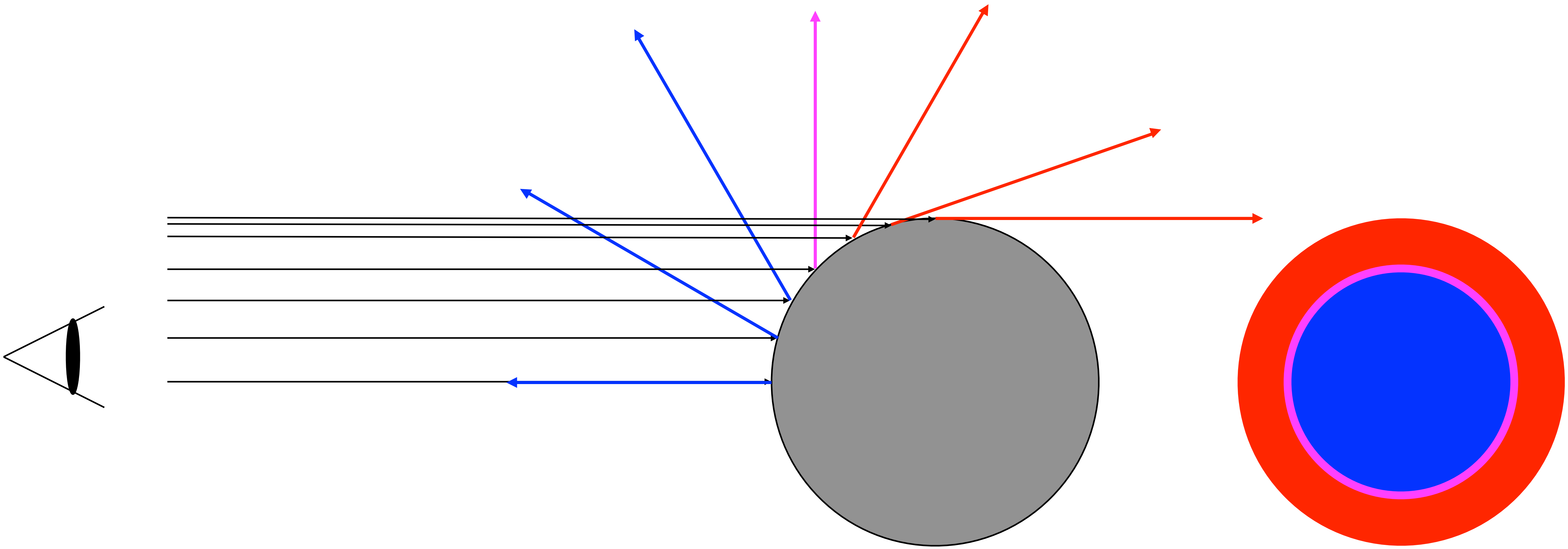
- Dynamic range of image:
  - brightest region/dimmest region
- Very briefly:
  - A digital camera uses a CCD (charge coupled device) array as a sensor. However, these values are mapped, using a non-linear function in order to display them on screen
  - To recover the entire range of the image, take several photos of the same scene with different exposure times
  - Non-linear function: from an overdetermined system using SVD
  - Use function to recover HDR image (one float per RGB)
  - There are also cameras that can take 12 bits per color component directly
- For more details: Debevec & Malik, "Recovering High Dynamic Range Radiance Maps from Photographs," *SIGGRAPH 97*, pp. 369-378.

Example of image with different exposure times

Images courtesy of Paul Debevec

How to capture an image of the environment?

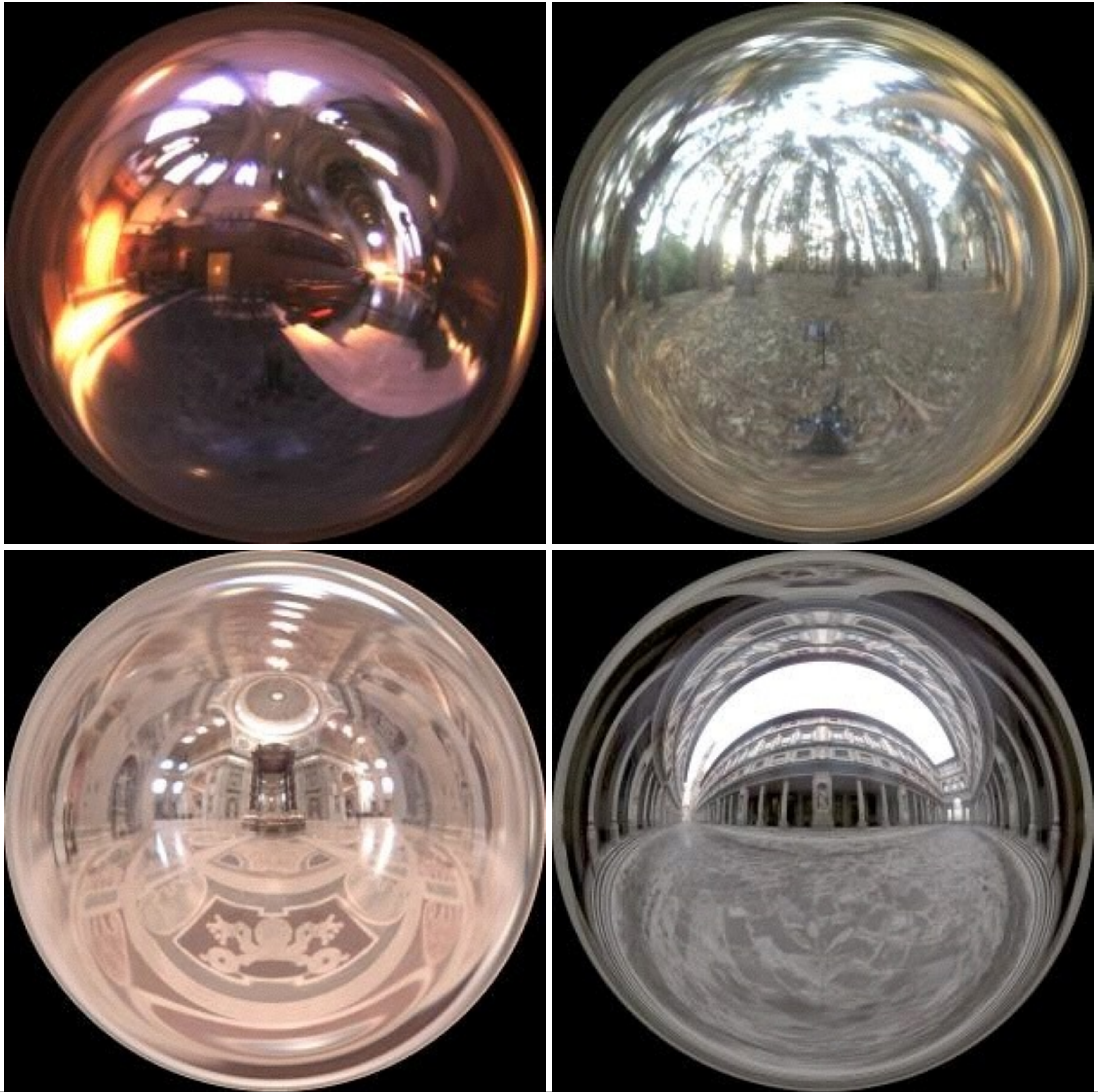
- Use simplest method: image of highly reflective sphere
  - We get an image of the entire environment, if we can assume that the camera is infinitely far away (or if the sphere is very small)



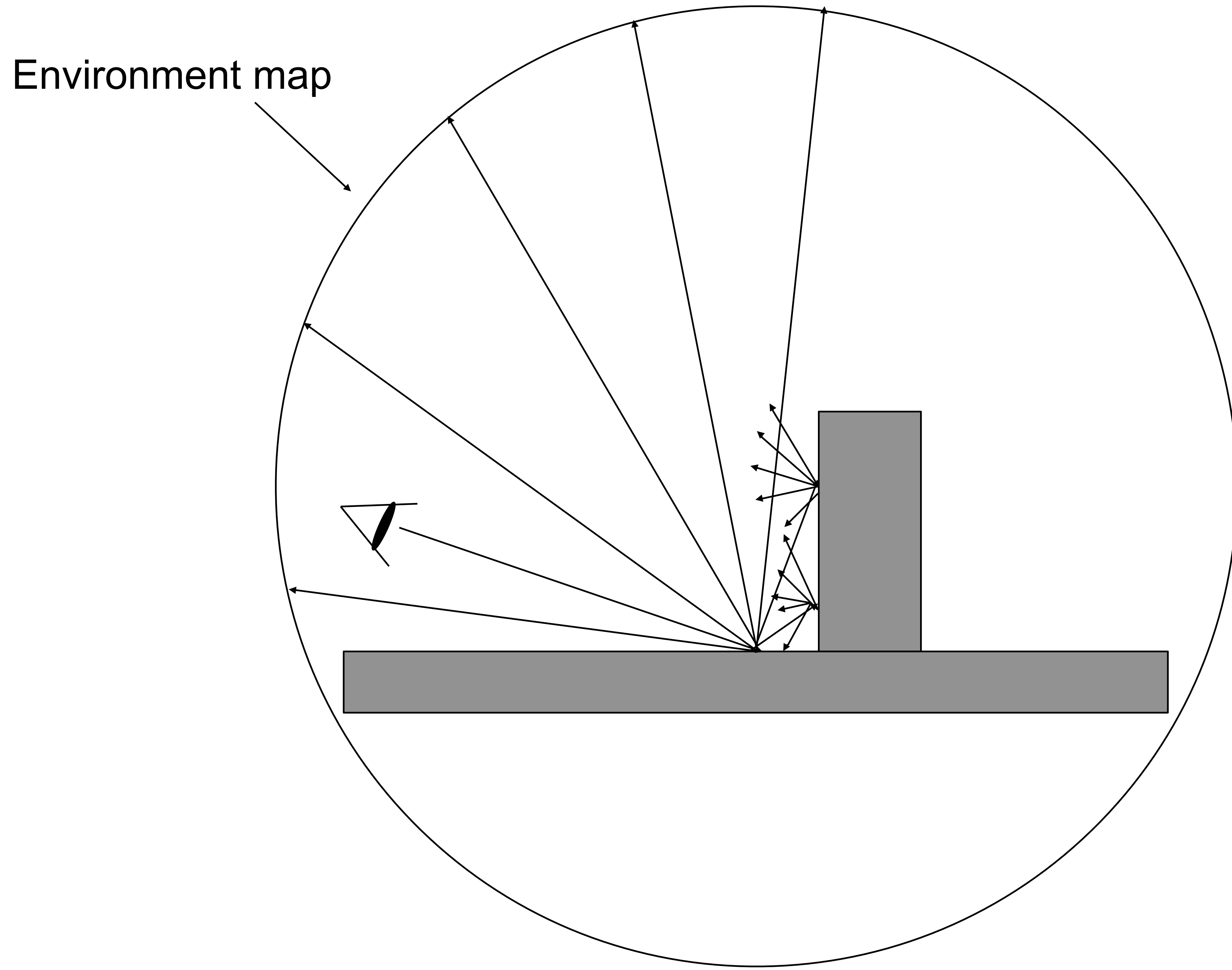
Examples of environment maps

Dynamic
Range:
1:200,000

Images courtesy of Paul Debevec



How to use environment map



Often need many rays!

May need varying number of rays depending on frequency content of env. map

Note: in path tracing, we only shoot one ray at every intersection point, but many rays through each pixel

HDR file formats

- EXR
  - ILM 16-bit float
- PFM (Portable Float Map)
  - Public open source
  - Binary, but no compression

IBL: some notes

- Even simpler to code with a cube map of the environment
  - Transform sphere map into cube map once before rendering
  - Debevec uses angular map (can use sphere map too)
- Nice tutorial by Paul Debevec, ”Image-Based Lighting,” *IEEE Computer Graphics & Applications*, March/April 2002, pp. 26-34.
  - http://www.debevec.org/
  - Check out light probe gallery + HDR shop
- How to do it fast?
  - Agarwal, Ramamoorthi, Belongie and Jensen, ”Structured Importance Sampling of Environment Maps,” *SIGGRAPH 2003*, pp. 605-612.
  - Probably takes weeks to implement though...

Next

- Friday Lab 2: BVH
- Next Week
  - Monday Seminar : Path Tracing Lab (Rasmus)
  - Tuesday Lecture : Photon Mapping
- Guest Lecture, May 12 :
  - Jonas Gustavsson, Sony
  - Production Rendering