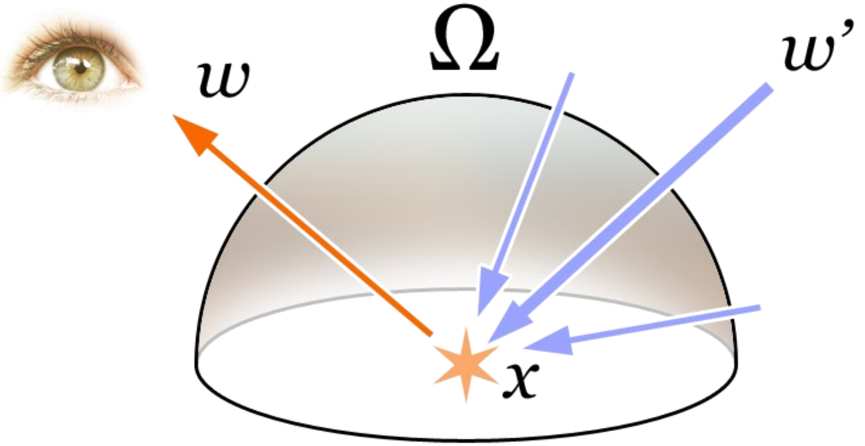


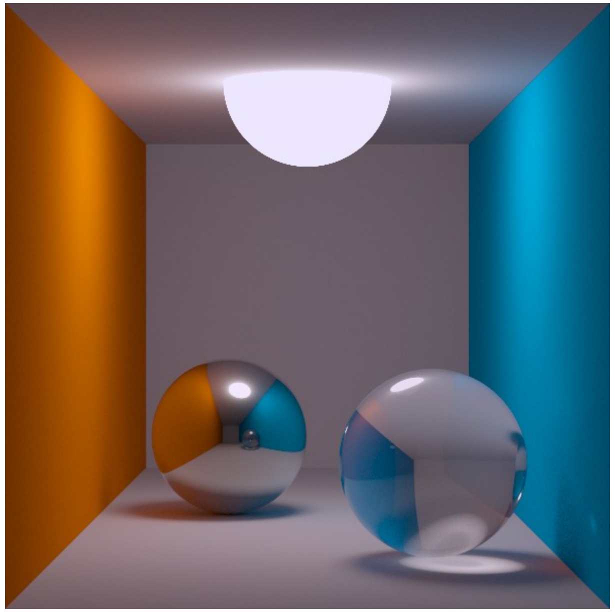
Radiometry and Monte Carlo integration

Adapted from slides by Derek Nowrouzezahrai

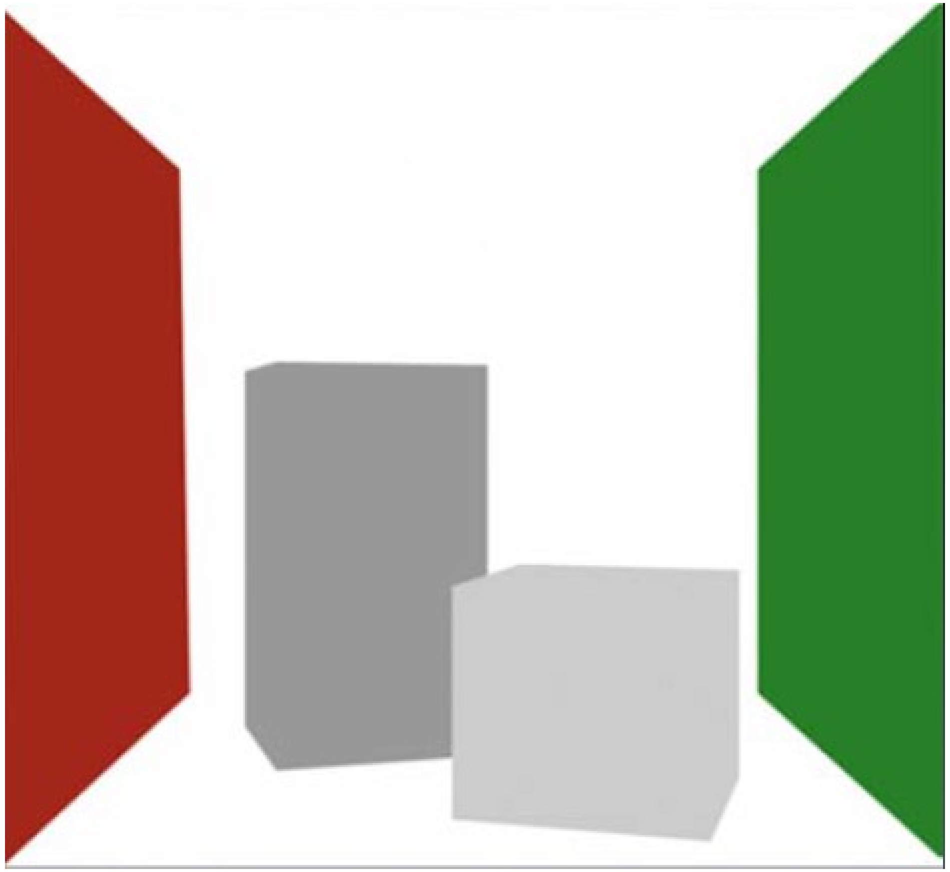
What will we cover?

- The theory of physically-based rendering (PBR)
  - Radiometry
  - Monte Carlo integration
    - Ambient occlusion
    - Direct illumination
    - Global illumination
- How to go from theory to simple, practical algorithms
  - You'll be able to generate images like these:



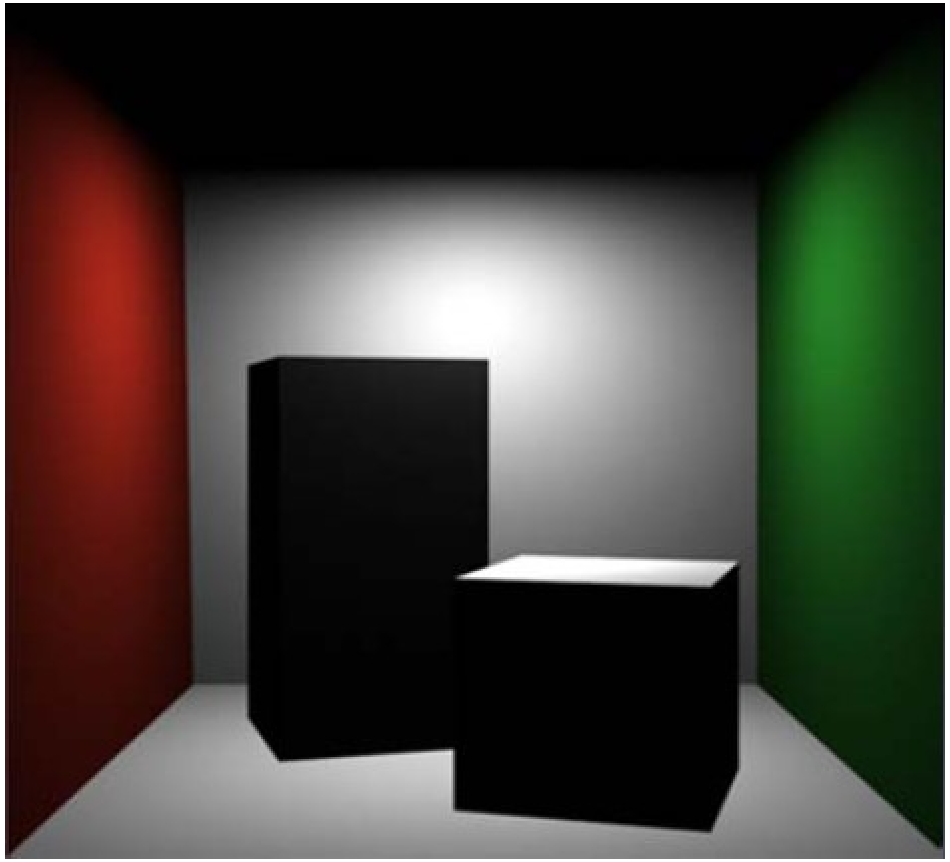


PBR: a (visual) implementation guide



Basic raytracing framework (without shading)
[**primary/eye rays**]

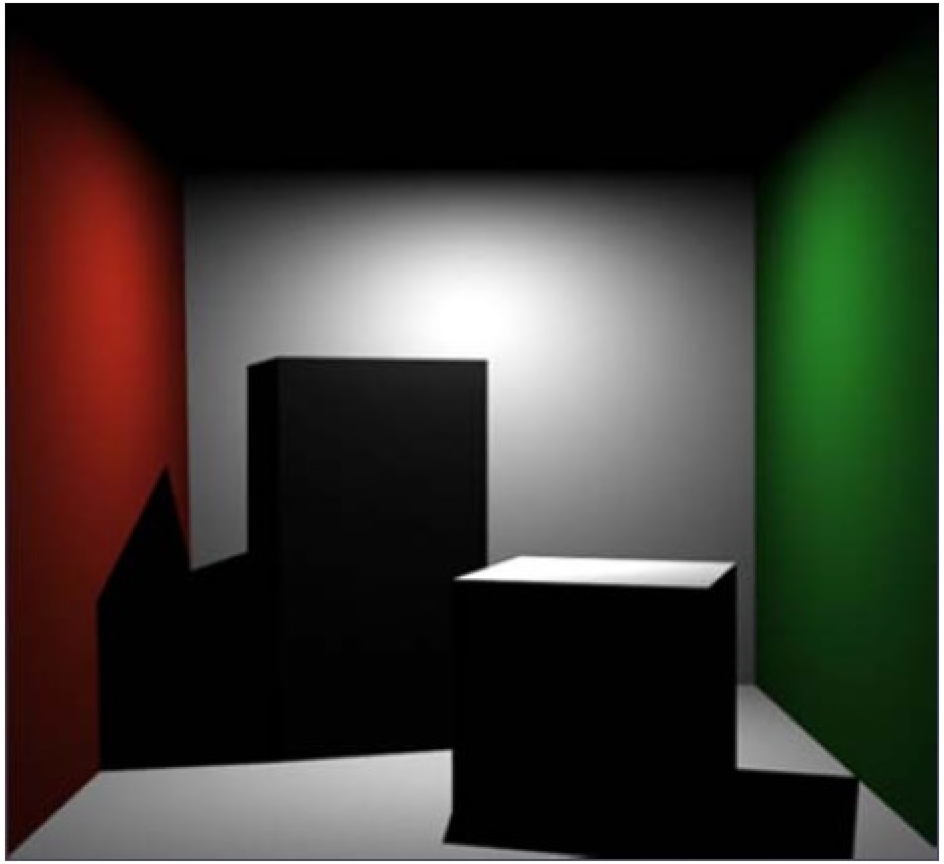
PBR: a (visual) implementation guide



Simple shading calculations

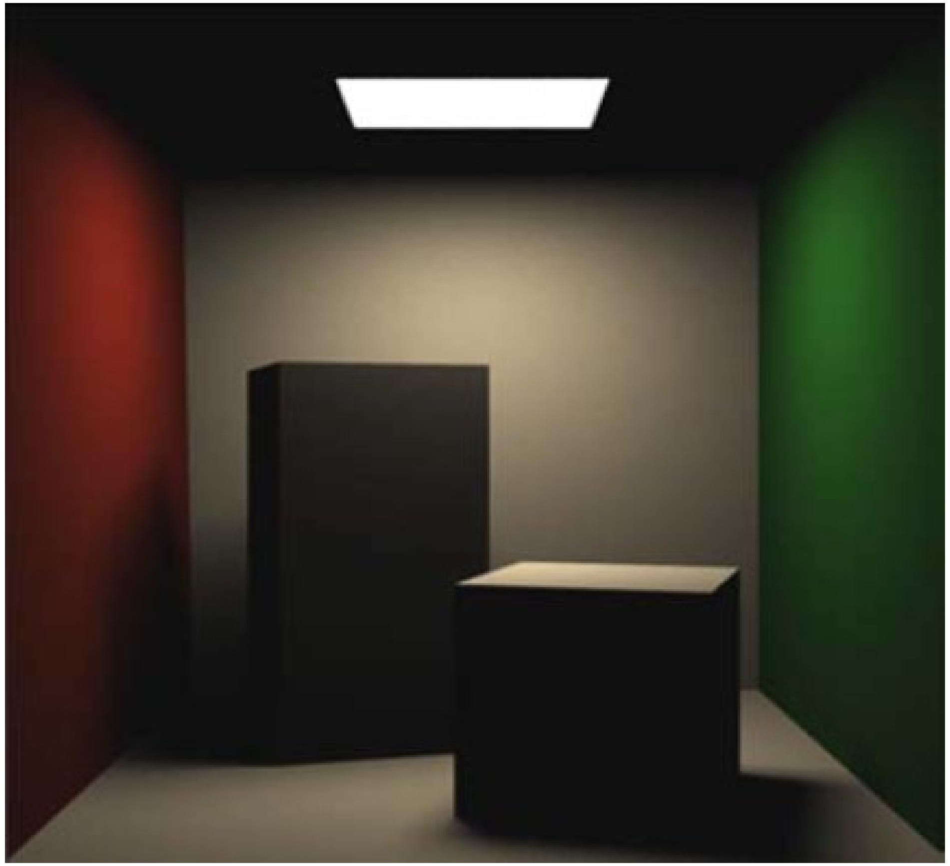
[**still just primary rays**]

PBR: a (visual) implementation guide



Hard shadows [**(one) secondary ray**]

PBR: a (visual) implementation guide



Soft shadows [**many secondary rays**]

PBR: a (visual) implementation guide

Global illumination [**recursive secondary rays**]

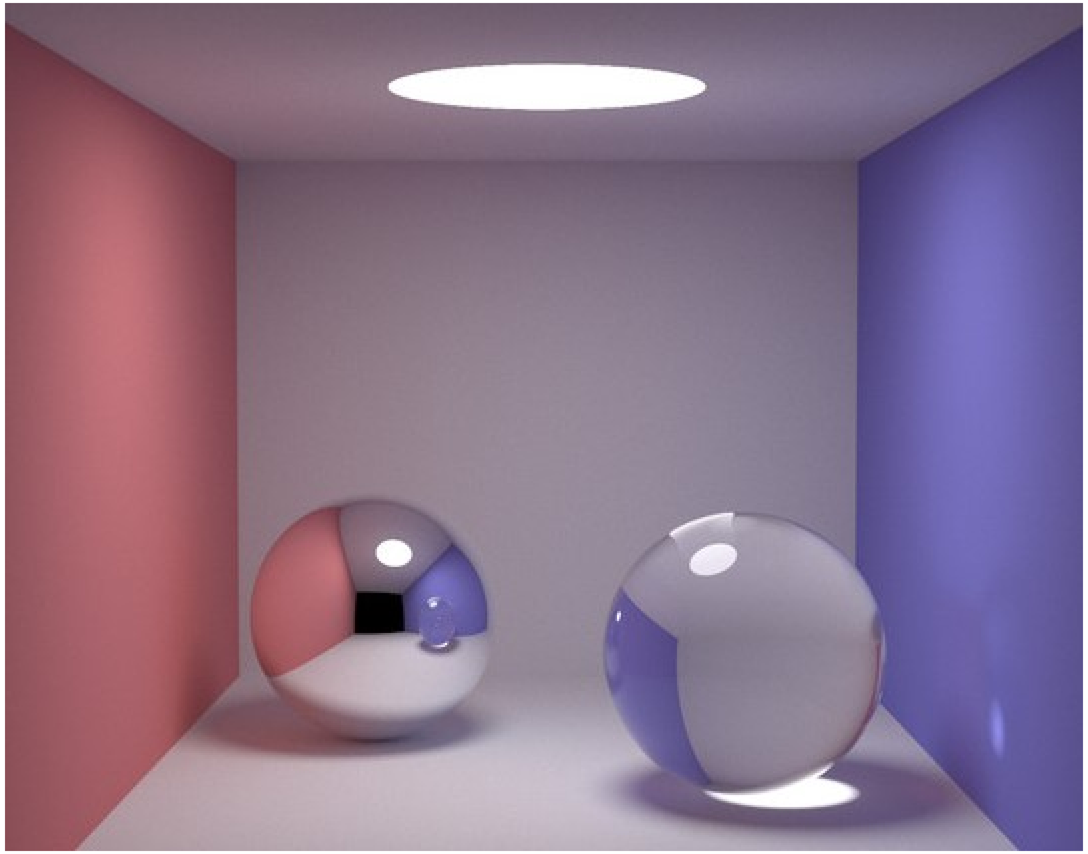
http://www.graphics.cornell.edu/online/box/compare.html
http://area.autodesk.com/fakeorfoto (I got 84%)

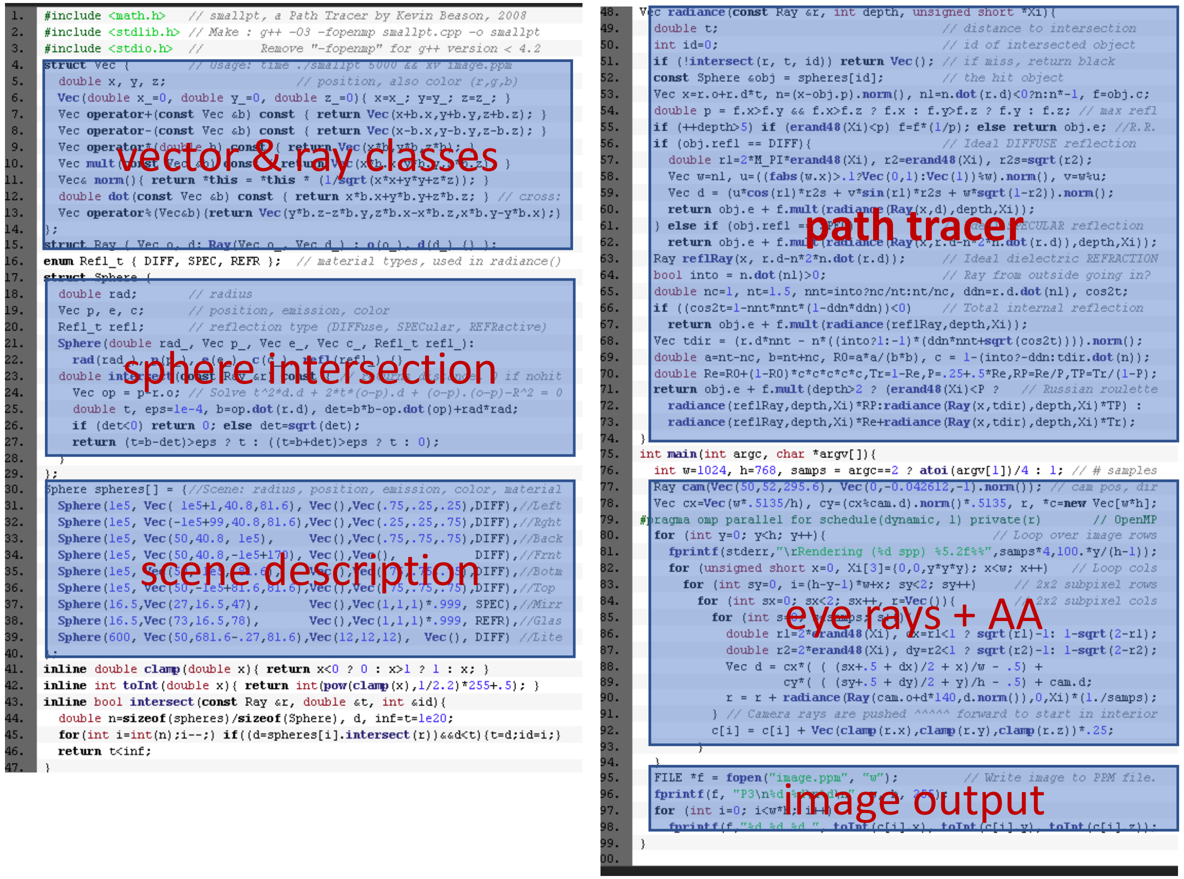
PBR: a (visual) implementation guide

- Devil’s in the details:
  - Where do the secondary rays start from?
  - Where do they go to?
  - What values do they “carry” and “propagate”
- We will bridge the answers to these questions using the theory of realistic image synthesis
  - Good news: in the end, the code isn’t that bad!
- For example: see smallpt (a 99 LoC path tracer!)

PBR: a (visual) implementation guide

- For example: see smallpt (a 99 LoC path tracer!)





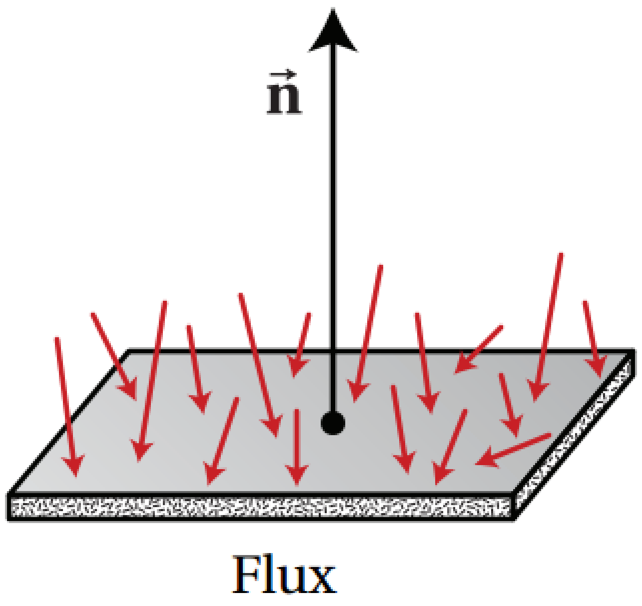
An Introduction to Radiometry

Radiometry

- Definition: the study of electromagnetic radiation, including visible light
- What are the most fundamental physical quantities used when measuring “light”?
- Which are the most important for physically-based rendering?

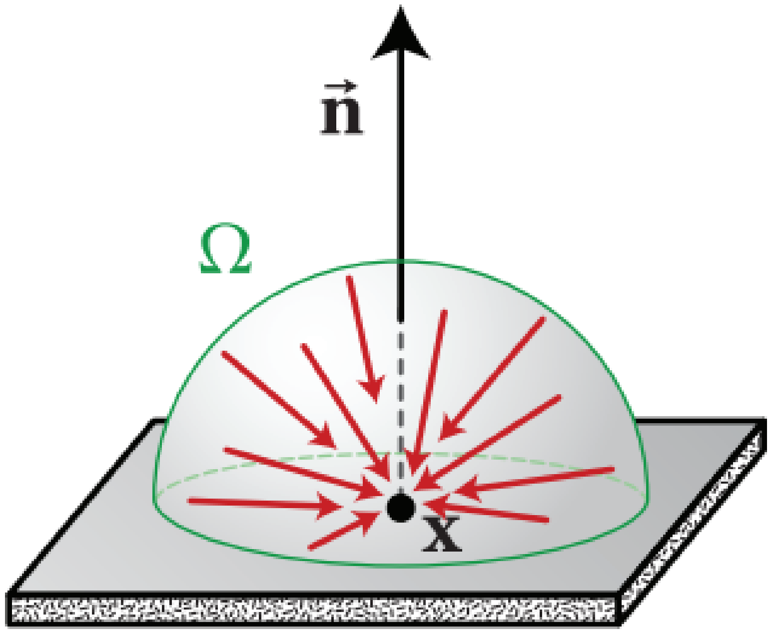
Flux Φ

- The amount of energy per unit time, a.k.a. power, hitting (or passing through) a surface
  - Measured in Watts $\left(W \equiv \frac{J}{s}\right)$



Irradiance $E = \frac{d\Phi}{dA}$

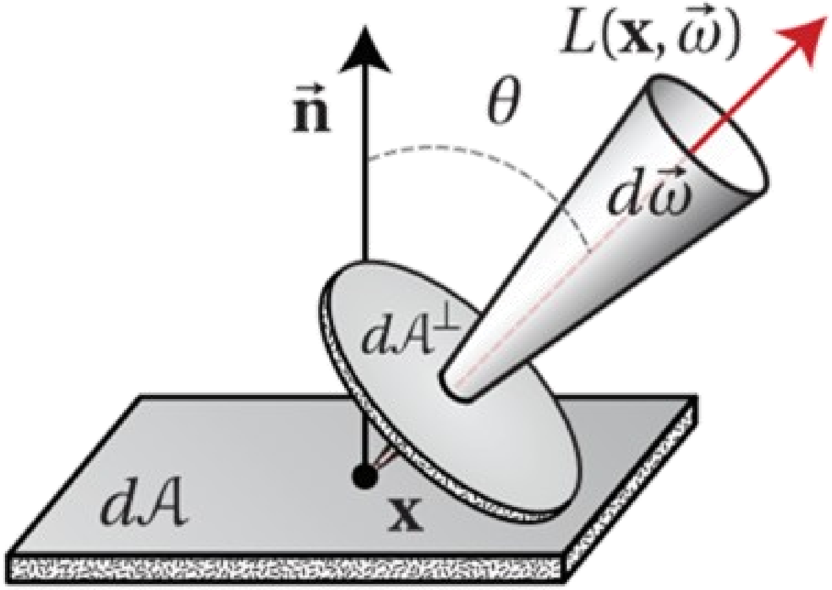
- The power (Flux) per unit (differential) surface area
  - Watts per meter squared $\left(E \equiv \frac{W}{m^2}\right)$
- Always measured at a surface point x with normal $\vec{n}$
- Irradiance leaving a surface area element is often called emitance M, radiant exitance or radiosity B



Irradiance

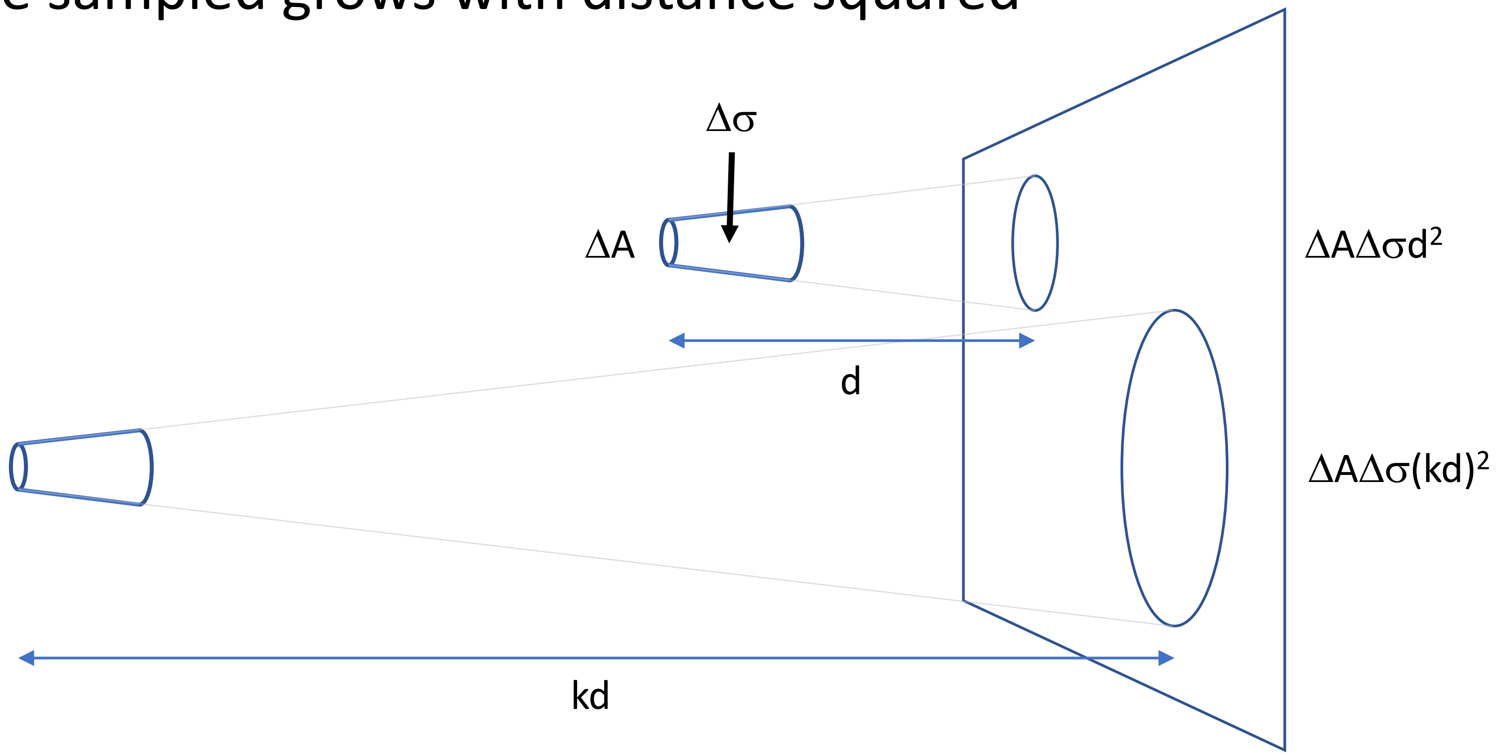
Radiance $L = \frac{d^2\Phi}{dA^{\perp} d\vec{\omega}} = \frac{d^2\Phi}{dA\,(\vec{n} \cdot \vec{\omega})\,d\vec{\omega}}$

- Power hitting (differential) area perpendicularly from a (differential) cone of directions;
  - Watts per meter squared per steradian $\left(L \equiv \frac{W}{sr \cdot m^2} \equiv \frac{E}{sr}\right)$
- The most important quantity in rendering! This is the “color” or pixel intensity you observe in a photo
- Note the cosine of the angle between $dA^{\perp}$ and dA



Radiance

- Radiance does not vary along a line in space
  - Light inversely proportional to distance squared
  - Area of surface sampled grows with distance squared

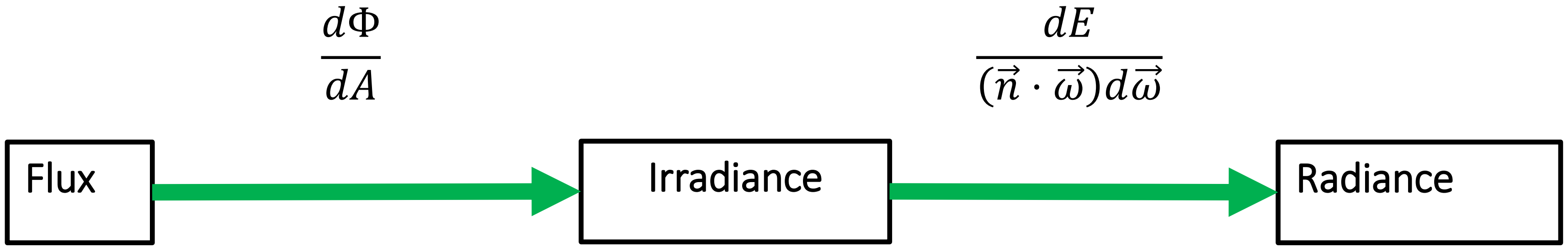


Radiometry
Summary of Terminology

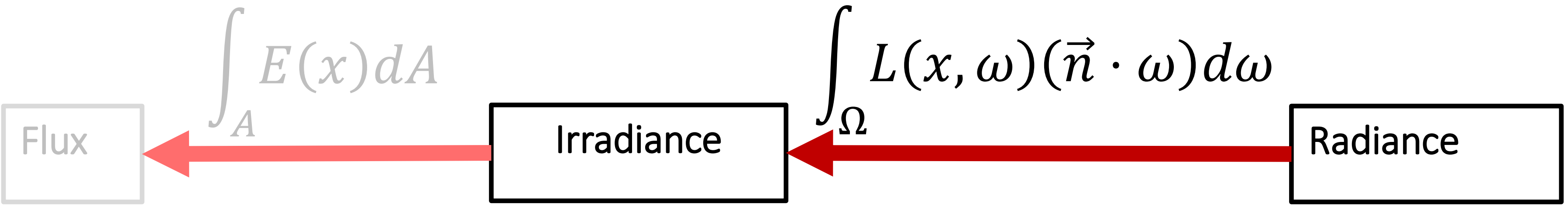
Symbol	Units	Name
$\boldsymbol{\Phi}$	Watts $\left(\mathrm{W} \equiv \frac{J}{s}\right)$	Flux
$\boldsymbol{E}$	$E \equiv \frac{W}{m^2}$	Irradiance
$\boldsymbol{M}$	$E \equiv \frac{W}{m^2}$	Radiant exitance
$\boldsymbol{B}$	$E \equiv \frac{W}{m^2}$	Radiosity
$\boldsymbol{L}$	$L \equiv \frac{W}{sr \cdot m^2} \equiv \frac{E}{sr}$	Radiance

From Radiometry to Rendering

- In radiometry we typically move between these quantities, from left to right, through differentiation

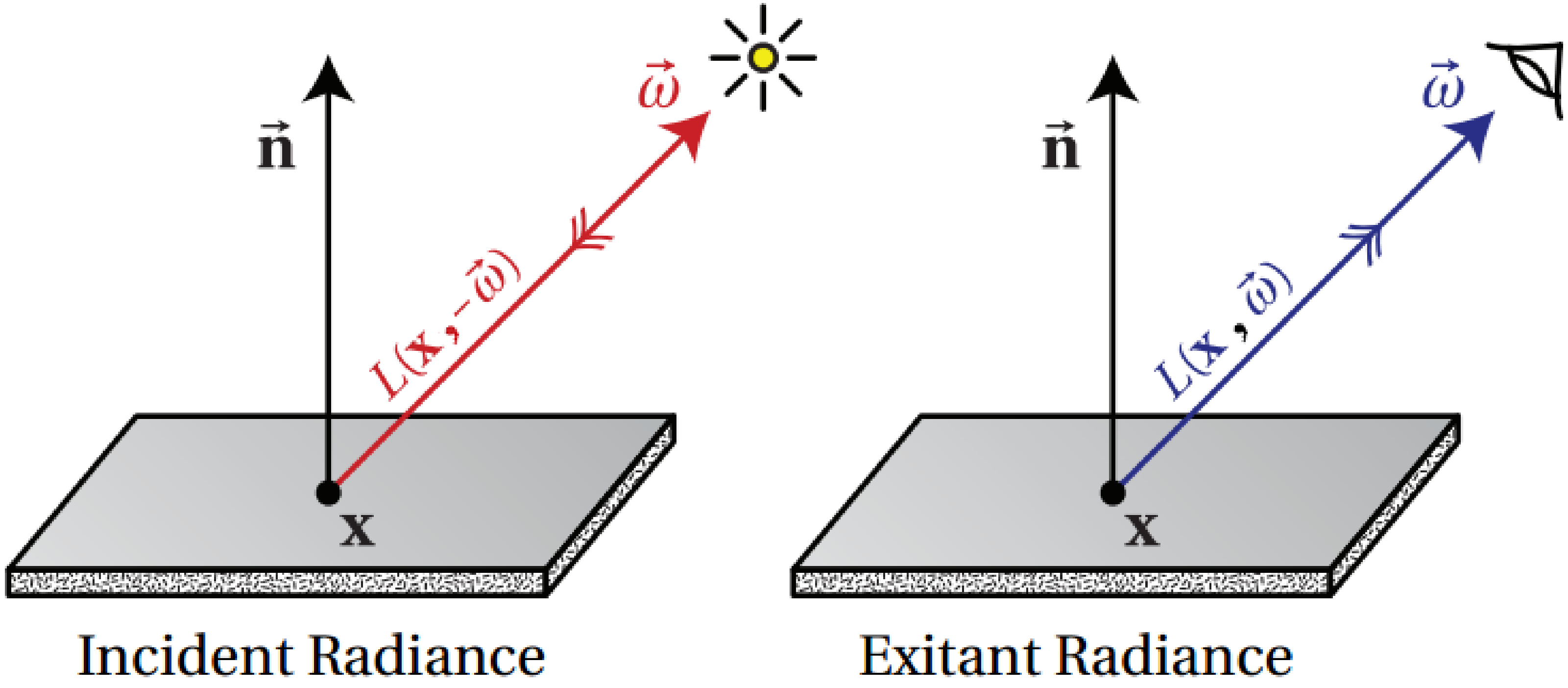


- In rendering, we usually integrate, moving from right to left



Radiance

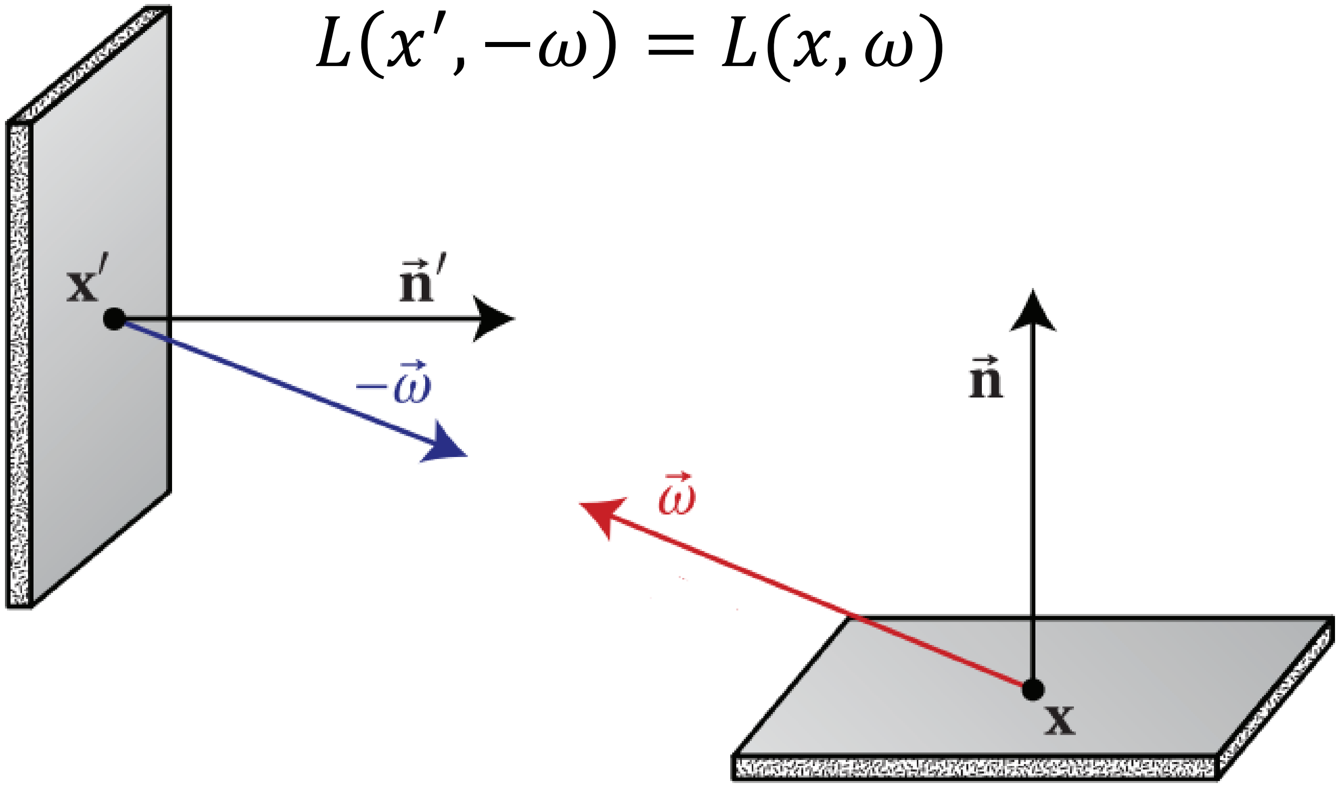
- As with irradiance, radiance can be used to represent both incident and exitant quantities



Radiance

- In a vacuum, radiance remains constant along (unoccluded) rays

$$L(x', -\omega) = L(x, \omega)$$

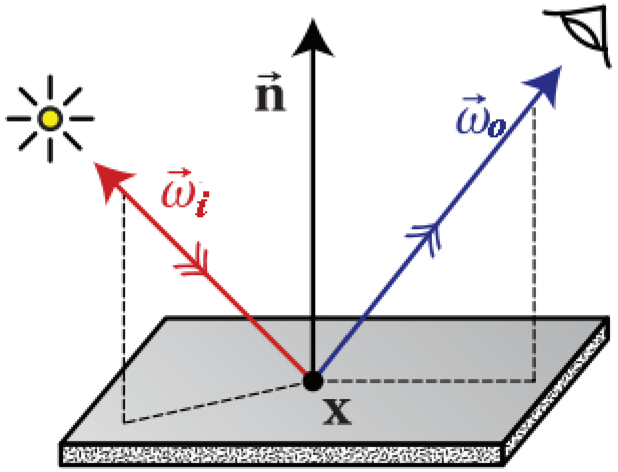


Bi-directional Reflectance Distribution Function (BRDF)

- Incident radiance at x warped to exitant/reflected (differential) radiance according to the BRDF, f_r

$$f_r(x, \omega_i, \omega_o) = \frac{dL_r(x, \omega_o)}{dE(x, \omega_i)} = \frac{dL_r(x, \omega_o)}{L(x, \omega_i)(\vec{n} \cdot \omega_i)d\omega_i}$$

- The BRDF is responsible for the appearance of different materials
- For example, the BRDF of wood is different than that of metal
- The BRDF has units sr^{-1}



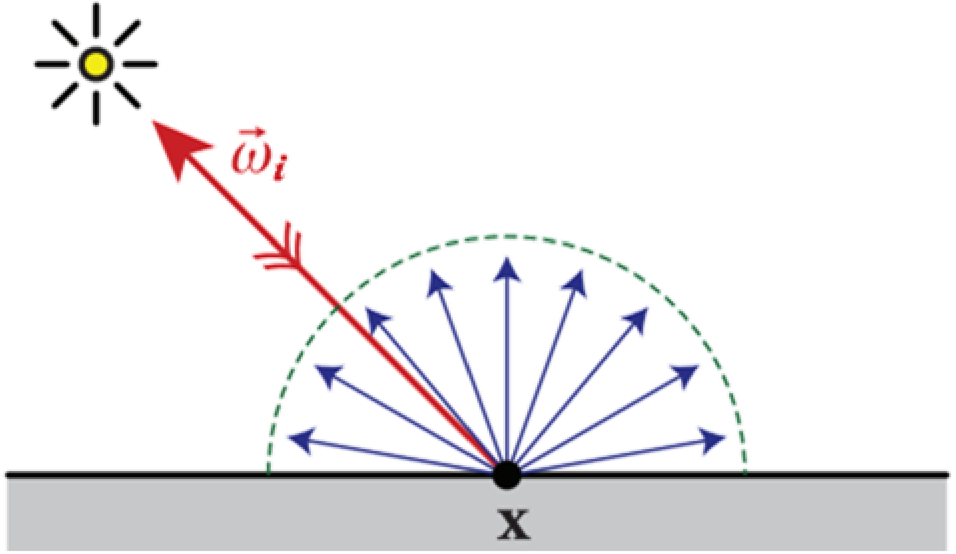
BRDF

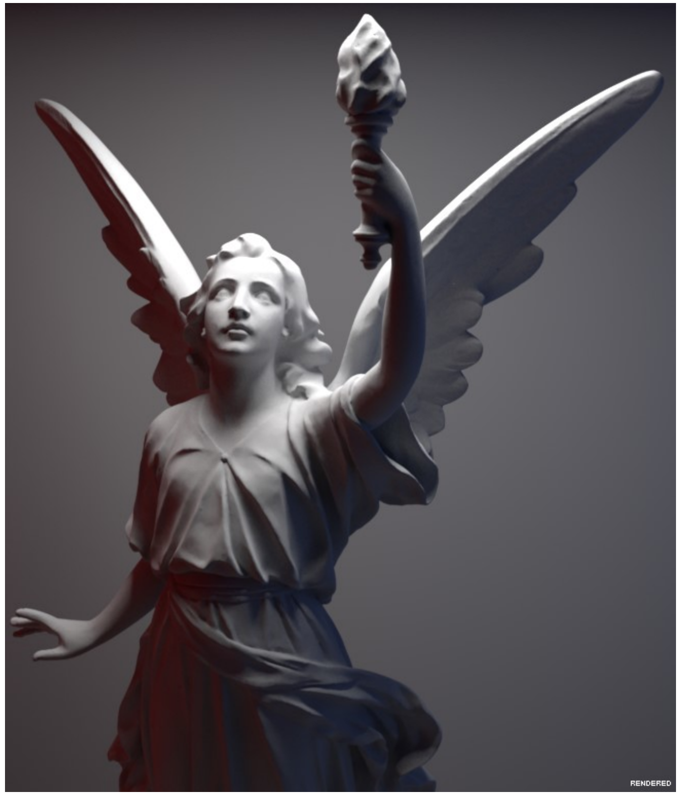
Important BRDF Properties

- At each x, the BRDF f_r is a 4 dimensional function
  - Two for incident lighting directions, and
  - Two for exitant (reflected) lighting directions
- The most important mathematical/physical properties of a BRDF are:
  - Reciprocity: $f_r(x, \omega_i, \omega_o) = f_r(x, \omega_o, \omega_i)$
  - Energy conservation: $\int_\Omega f_r(x, \omega_i, \omega_o)\,(\vec{n} \cdot \omega_i)\, d\omega_i \leq 1, \forall \omega_o$
  - Positivity: $f_r(x, \omega_i, \omega_o) \geq 0, \forall \omega_i \ and\ \forall \omega_o$

A few Examples of BRDFs

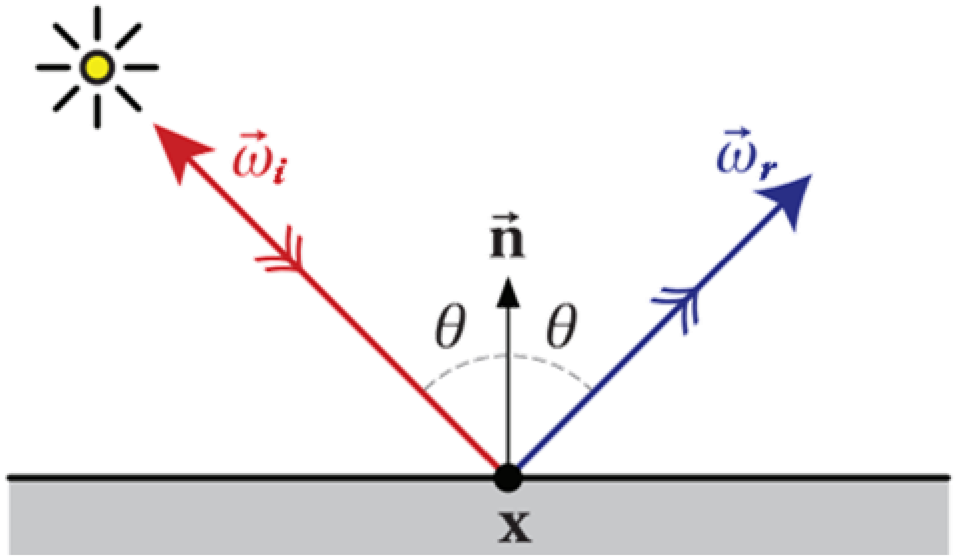
- Special case #1: Diffuse/Lambertian
  - A material that reflects light (radiance) equally in all directions
  - View-independent reflection
  - $f_r(x, \omega_i, \omega_o) = \frac{\rho}{\pi}$



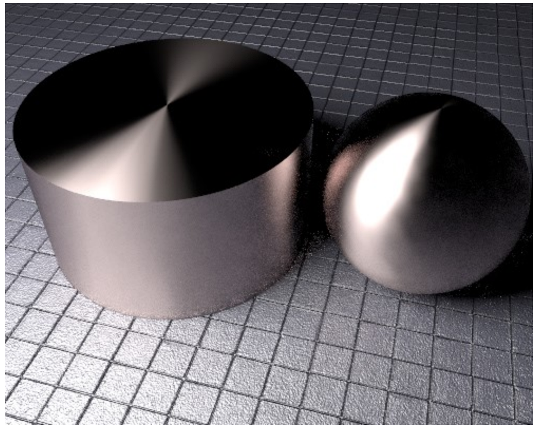


A few Examples of BRDFs

- Special-case #2: perfect specular mirror
  - Only reflects light a single direction: the perfect mirror reflection direction
  - $f_r(x, \omega_i, \omega_o) = \frac{\delta(\omega_r(\omega_i))}{\vec{n} \cdot \omega_i}$

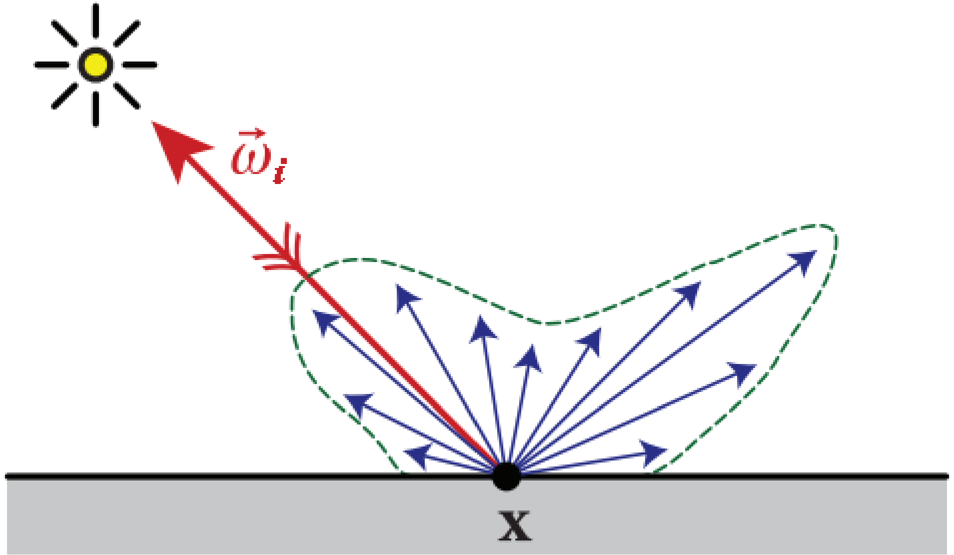


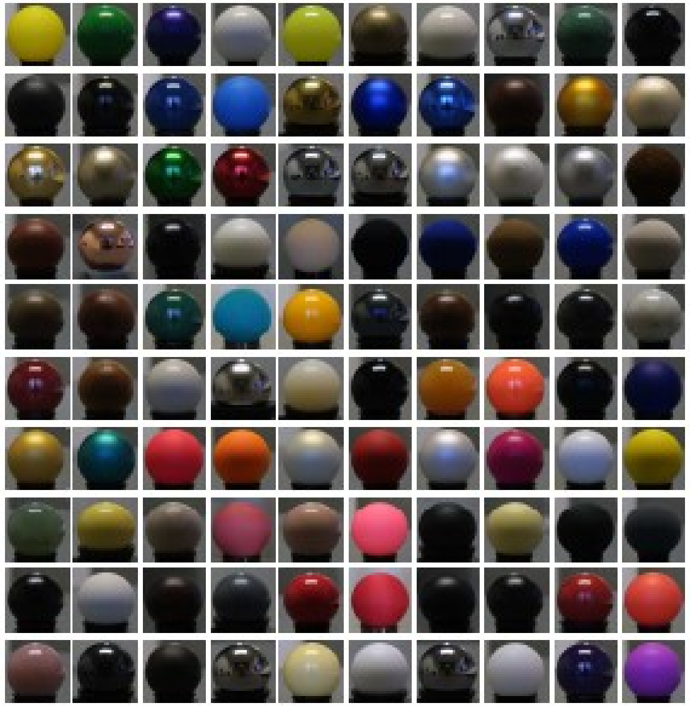
A few Examples of BRDFs



Example of brushed aluminum (anisotropic BRDF)

- General case:
  - For each direction of incident illumination, the BRDF specifies a (spherical) distribution of exitant illumination
    - A ratio of reflected light for any possible view direction





Emitted and Reflected Radiance

- Exitant radiance (from a surface, in a cone of directions) is equal to the sum of emitted (L_e) and reflected (L_r) radiance:

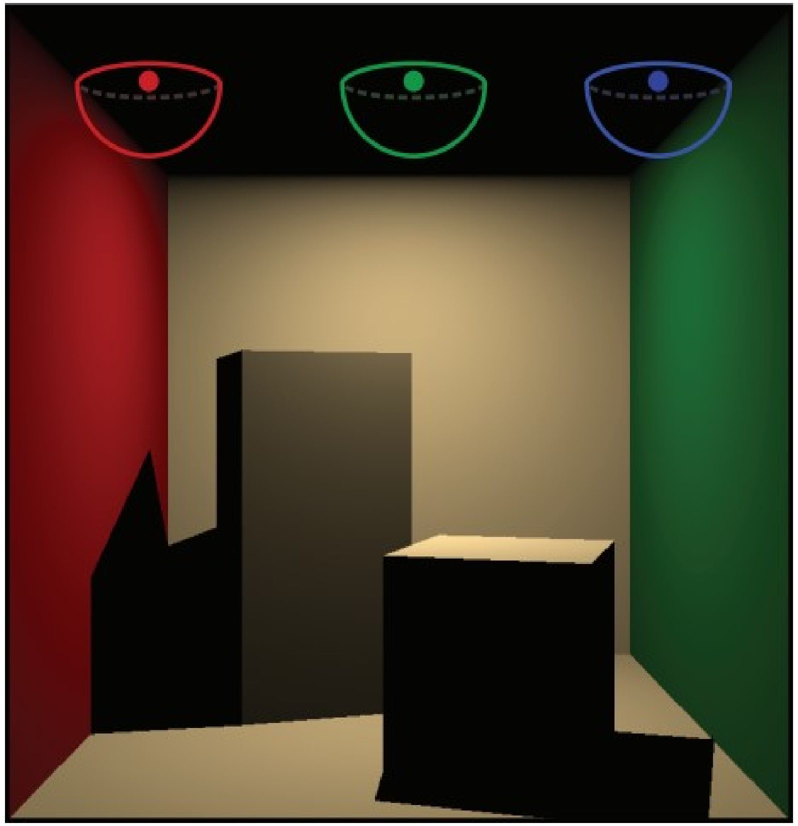
$$L(x, \omega_o) = L_e(x, \omega_o) + L_r(x, \omega_o)$$

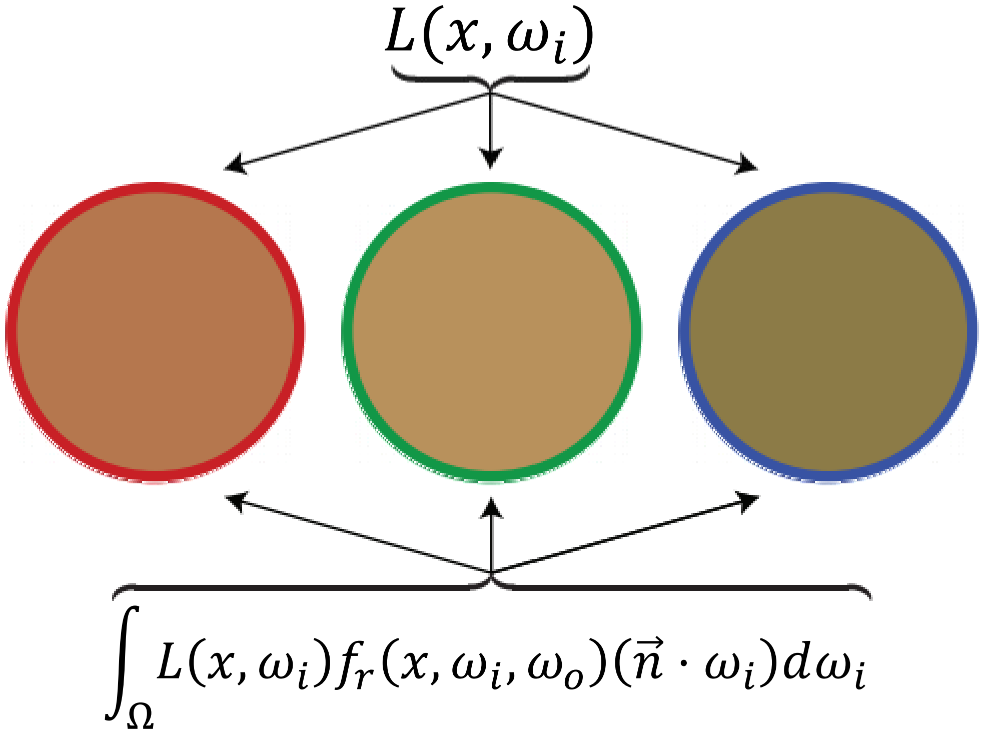
- Only light source surfaces have a non-zero emitted radiance component ($L_e \neq 0$)
  - We typically treat these surfaces (lights) as non-reflective ($f_r = 0$)

Reflected Radiance

a visual example

- L_r is the integral, over (differential) incident directions $d\omega_i$ around $\vec{n}$, of incident radiance weighted by the BRDF and cosine projection





The Rendering Equation

- This finally leads us to the rendering equation

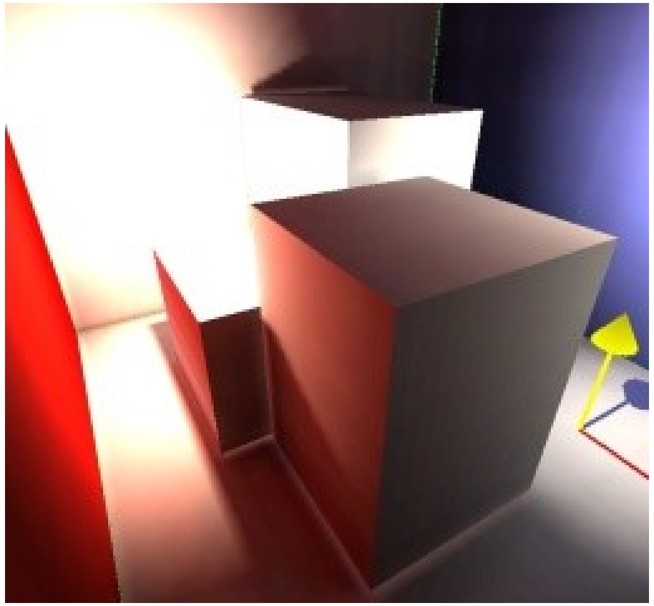
$$L(x, \omega_o) = L_e(x, \omega_o) + \int_{\Omega} L(x', -\omega_i) f_r(x, \omega_i, \omega_o)(\vec{n} \cdot \omega_i) d\omega_i$$

- Note that this general formulation is recursive!
- Almost every shading algorithm you will encounter is based on (or a simplifications of) this equation

Direct vs. Global Illumination

- The rendering equation defines the global illumination of a scene
- This includes direct and indirect illumination
- Direct illumination is light that is directly reflected and/or occluded from light sources
- Indirect illumination is light reflected and/or occluded recursively off the remaining (non-light) surfaces in a scene

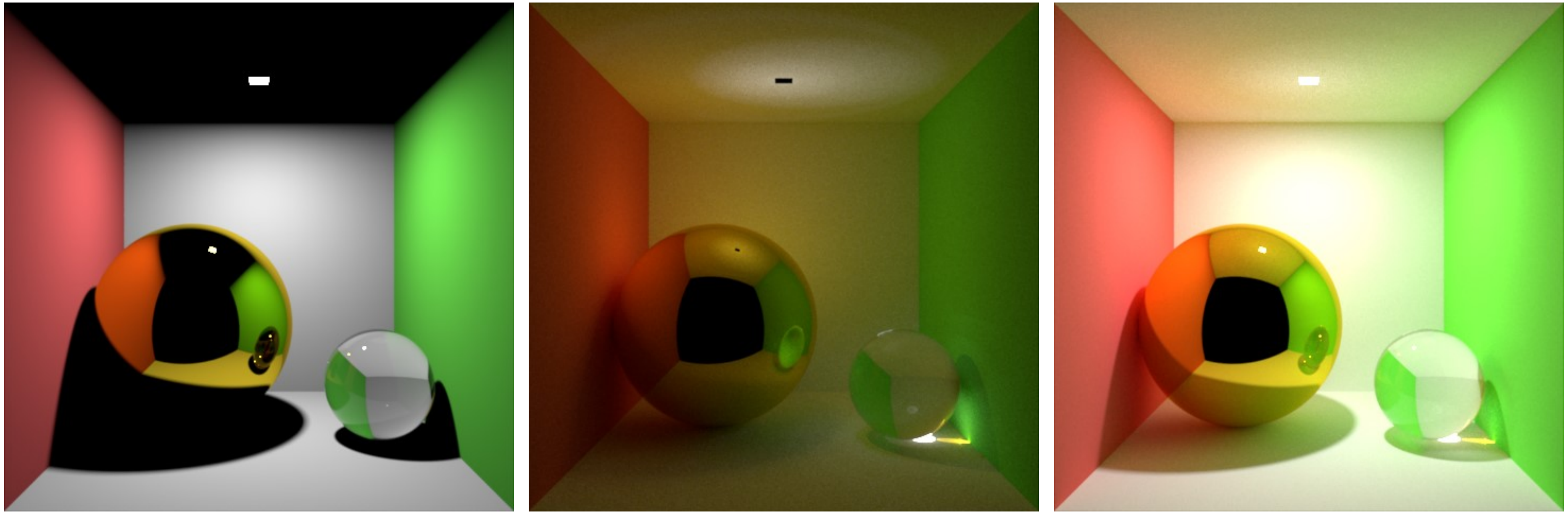
Direct vs. Global Illumination



Direct vs. Global Illumination

Direct vs. Global Illumination

Direct vs. Global Illumination



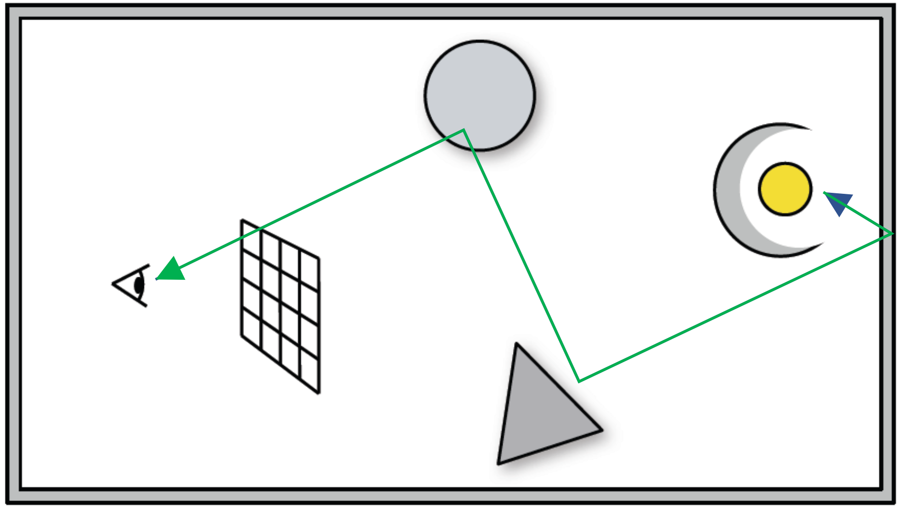
Global Illumination Techniques: Overview

- Monte Carlo Integration
  - Ambient occlusion
  - Direct illumination
  - Indirect illumination
- Path tracing
  - implicit
  - explicit
- Photon mapping

Solving the Rendering Equation

- How to solve a multi-dimensional integral equation?
  - Analytically? Only in trivially simple scenes…
  - Numerically? How to avoid the curse of dimensionality?
- We're going to:
  - use a special type of numerical integration (MC Integration) that completely side steps the curse of dimensionality, and
  - use ray-tracing to sample/evaluate the terms in our integrand
- Conceptually, we want to connect pixels to points on the light sources using paths that light follows

Solving the Rendering Equation



- Several approaches:
  - Do we start at the eye and trace paths?
  - What about starting at the light and tracing paths?
- How do we generate these connecting paths? How can we guarantee the correctness of this process?
- To answer these questions, we first need theory of numerical integration suitable for multi-dimensional integration

An Introduction to Monte Carlo

Monte Carlo Integration

- A Monte Carlo estimator for an integral (of arbitrary dimensionality $\dim(X)$)

$$F = \int_X f(x)\, dx$$

- is defined as

$$\boxed{\bar{F} = \frac{1}{N} \sum_{i=0}^{N-1} \frac{f(X_i)}{pdf(X_i)} \approx \int_X f(x)\, dx}$$

- where the N samples $X_i \in X$ are drawn according to a probability distribution function pdf

$$\bar{F} = \frac{1}{N}\sum_{i=0}^{N-1}\frac{f(X_i)}{pdf(X_i)}$$

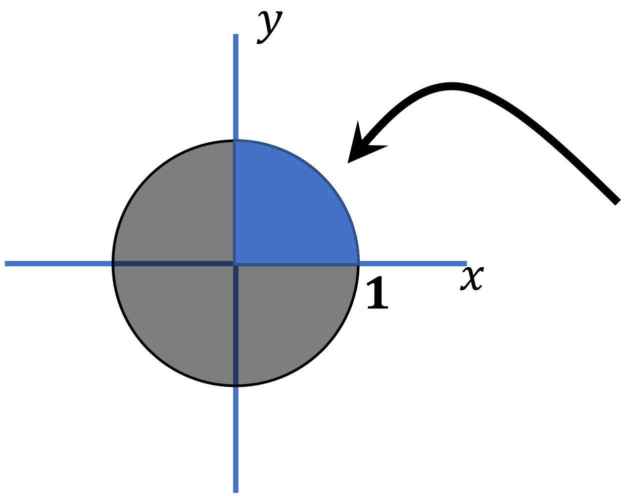
Monte Carlo Integration

- Given the definition of our Monte Carlo estimator, the process of evaluating $\bar{F}$ is divided into 3 steps:
  - Choose* the pdf from which to draw random samples
  - Draw N random samples according to pdf (i.e., the X_i)
  - Evaluate $\bar{F} = \frac{1}{N}\sum_{i=0}^{N-1}\frac{f(X_i)}{pdf(X_i)}$
- Note that the X_i can also be generated on-the-fly in the loop in step 3

Monte Carlo Integration: an example in 1D

$$\bar{F} = \frac{1}{N}\sum_{i=0}^{N-1}\frac{f(X_i)}{pdf(X_i)}$$

- Let’s consider a simple example that we can validate analytically:



$$\int_0^1 \sqrt{1-x^2}\,dx$$

$$= \frac{\pi}{4} \approx 0.785398$$

```
integral = 0
for( i = 0 to N-1 )
    X[i] = rand();
    pdf[i] = 1;
    integral += sqrt(1.-
    X[i]*X[i])/pdf[i]

integral = integral / N
```

1. Choose pdf : for the example, we’ll just use a uniform *pdf*

$$pdf(x) = \frac{1}{b-a} = \frac{1}{1-0} = 1$$

2. Generate the N points X_i
3. Evaluate $\bar{F}$

$$\bar{F} = \frac{1}{N}\sum_{i=0}^{N-1}\frac{f(X_i)}{pdf(X_i)}$$

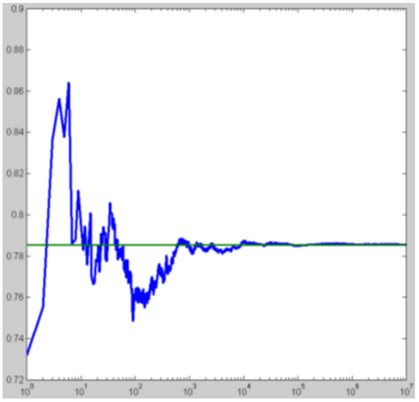
Monte Carlo Integration

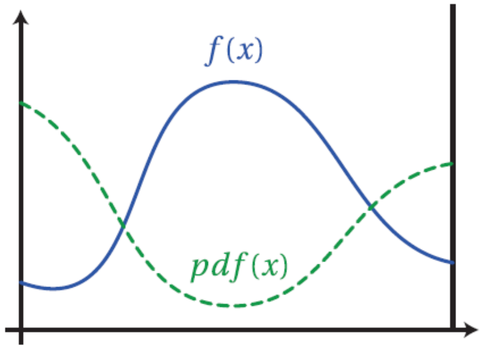
- One design goal for an MC estimator is to sample according to a pdf that reduces the variance in the estimator's convergence
- Two typical variance-reduction strategies are stratification and importance sampling

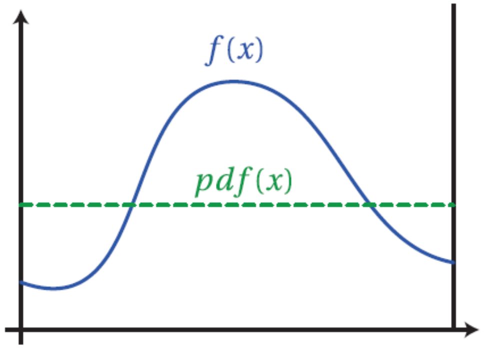
$$\bar{F} = \frac{1}{N}\sum_{i=0}^{N-1}\frac{f(X_i)}{pdf(X_i)}$$

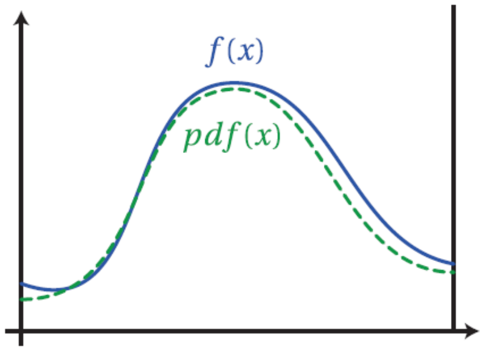
Monte Carlo Integration

- Stratification divides the integration domain into smaller sub-domains and performs MC estimation separately within each sub-domain
- Importance sampling refers to choosing a $pdf(x)$ that samples X_i preferentially where $f(x)$ has higher values







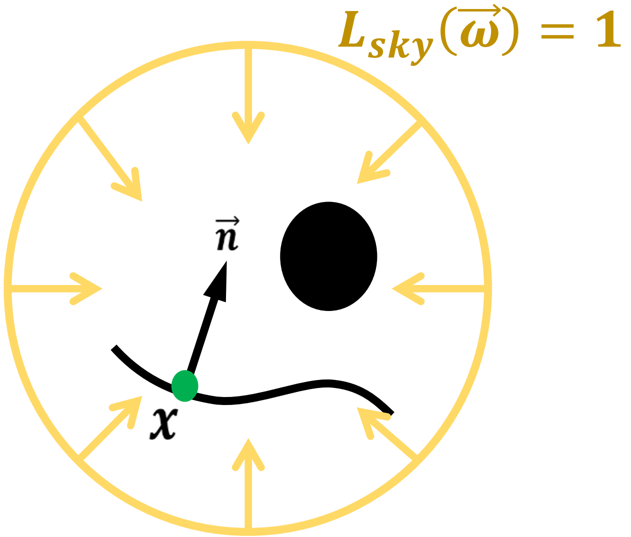


Example 1: Ambient Occlusion

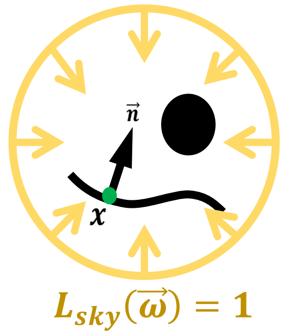
- Consider the following scenario: we want to compute the shading of a scene composed of only **diffuse** objects (each with $\rho = 1$), with **direct illumination** exclusively due to **sky lighting**
  - Here, we treat the sky as an infinitely large sphere surrounding our scene, where all points on the sphere have uniform and unit-intensity emitted radiance $L_{in}(x, \vec{\omega}) = L_{sky}(\vec{\omega}) = 1$

$$L_o(x, \overrightarrow{\omega_o}) = L_{ao}(x) = \int_{\Omega} \frac{\max(\overrightarrow{n_x} \cdot \vec{\omega}, 0)}{\pi} V(x, \vec{\omega}) d\vec{\omega}$$

- We will investigate how to implement a MC estimator for AO using: a naive pdf and an optimal pdf



Example 1: Ambient Occlusion



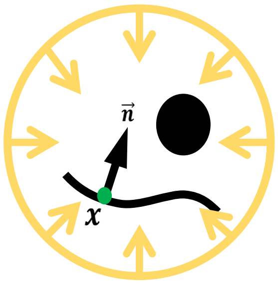
- Let's start by writing the Monte Carlo estimator equation (with a general *pdf* for now) for

$$L_{ao}(x) = \int_{\Omega} \frac{\max(\overrightarrow{n_x} \cdot \vec{\omega}, 0)}{\pi} V(x, \vec{\omega}) d\vec{\omega}$$

as

$$\overline{L_{ao}}(x) = \frac{1}{N} \sum_{i=0}^{N-1} \frac{\max(\overrightarrow{n_x} \cdot \overrightarrow{\omega_i}, 0) V(x, \overrightarrow{\omega_i})}{\pi \, pdf(\overrightarrow{\omega_i})}$$

Example 1: Ambient Occlusion



$$\overline{L_{ao}}(x) = \frac{1}{N}\sum_{i=0}^{N-1}\frac{\max(\overrightarrow{n_x}\cdot\overrightarrow{\omega_i},0)V(x,\overrightarrow{\omega_i})}{\pi\, pdf(\overrightarrow{\omega_i})}$$

- And now, like before, we just need to:
  - Pick a $pdf(\overrightarrow{\omega_i})$, and
  - Distribute the random samples $\overrightarrow{\omega_i}$ according to this pdf
- We'll investigate two choices for $pdf(\overrightarrow{\omega_i})$, how to distribute random samples according to these pdfs, and their effect on the convergence/variance of our estimators

Ambient Occlusion – Estimator 1 (pdf = uniform spherical directions)

$$\overline{L_{ao}}(x) = \frac{1}{N}\sum_{i=0}^{N-1}\frac{\max(\overrightarrow{n_x}\cdot\overrightarrow{\omega_i},0)V(x,\overrightarrow{\omega_i})}{\pi\, pdf(\overrightarrow{\omega_i})}$$

- As with our earlier 1D example, let’s start with the simplest (naive) choice for our pdf: uniform distribution
  - In the 1D case we had $pdf(x_i) = \frac{1}{length(X)} = \frac{1}{\int_X dx} = \frac{1}{b-a}$ but now our integration/sampling domain $\mathbf{\Omega}$ is the set of unit directions
  - This corresponds to the surface (not volume!) of a unit sphere
  - So, for (random) samples distributed uniformly on the surface of the unit sphere, we have $pdf(\overrightarrow{\omega_i}) = \frac{1}{area(\Omega)} = \frac{1}{\int_\Omega d\overrightarrow{\omega}} = ?$

Ambient Occlusion – Estimator 1

(pdf = uniform spherical directions)

$$\overline{L_{ao}}(x) = \frac{1}{N}\sum_{i=0}^{N-1} \frac{\max(\overrightarrow{n_x}\cdot\overrightarrow{\omega_i}, 0)V(x,\overrightarrow{\omega_i})}{\pi\, pdf(\overrightarrow{\omega_i})}$$

$$pdf(\overrightarrow{\omega_i}) = \frac{1}{area(\Omega)} = \frac{1}{\int_\Omega d\vec{\omega}} = \frac{1}{\int_{\theta=0}^{\pi}\int_{\phi=0}^{2\pi} \sin\theta\, d\theta\, d\phi} = \frac{1}{4\pi}$$

- Our estimator for L_{ao}, with this uniform *pdf* , becomes:

$$\overline{L_{ao}}(x) = \frac{4}{N}\sum_{i=0}^{N-1} \max(\overrightarrow{n_x}\cdot\overrightarrow{\omega_i}, 0)V(x,\overrightarrow{\omega_i})$$

- and we only have to figure out how to distribute random samples according to $pdf(\overrightarrow{\omega_i})$ now…

Ambient Occlusion – Estimator 1

(pdf = uniform spherical directions)

$$\overline{L_{ao}}(x) = \frac{4}{N}\sum_{i=0}^{N-1} \max(\overrightarrow{n_x} \cdot \overrightarrow{\omega_i}, 0) V(x, \overrightarrow{\omega_i})$$

- How do we generate random $\overrightarrow{\omega_i}$ directions with a uniform distribution over the sphere?

- <u>Method 1:</u> I show you how to derive it from first principles and we spend the rest of the class on the whiteboard… fun…

- or...

Ambient Occlusion – Estimator 1

(pdf = uniform spherical directions)

- <u>Method 2:</u> we ask Google

```
θ  = acos( 1 - 2 * rand() );
ϕ  = 2 * M_PI * rand();
```

Web   Images   Maps   More ▾   Search tools

About 2,510,000 results (0.26 seconds)

Sphere Point Picking -- from Wolfram MathWorld
mathworld.wolfram.com › ... › Random Point Picking ▾
have a **uniform** distribution on the surface of a unit **sphere**. This method can also be extended to **hypersphere** point picking. The plots above show the ...

#AltDevBlog » Generating **Uniformly** Distributed Points on **Sphere**
www.altdevblogaday.com/.../generating-**uniformly**-distributed-points-on... ▾
May 3, 2012 - An effective **sampling** requires a **uniform** distribution of **samples**. ... shows why this method can generate a **uniform** distribution over a **sphere**.

Uniform Sampling of a **Sphere** - File Exchange - MATLAB Central
www.mathworks.com/.../37004-**uniform-sampling**-of-a-**sphere** ▾
Uniform Sampling of a **Sphere**. by Anton Semechko. 05 Jun 2012 (Updated 23 May 2013). Create an approximately **uniform** triangular tessellation of a unit ...

Ambient Occlusion – Estimator 1

(pdf = uniform spherical directions)

- So, in summary, our first end-to-end MC estimator for AO:

1. Uses a uniform *pdf* of directions on the sphere: $pdf(\overrightarrow{\omega_i}) = \frac{1}{4\pi}$

2. Generates random samples $\overrightarrow{\omega_i} = (\theta_i, \phi_i)$ about this distribution:

```
θi = acos( 1 - 2 * rand() );
φi = 2 * M_PI * rand();
```

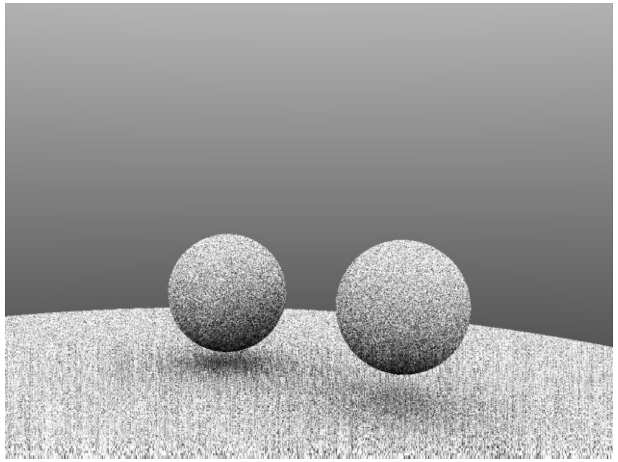
3. And finally calculates ambient occlusion as:

$$\overline{L_{ao}}(x) = \frac{4}{N} \sum_{i=0}^{N-1} \max(\overrightarrow{n_x} \cdot \overrightarrow{\omega_i}, 0) V(x, \overrightarrow{\omega_i})$$

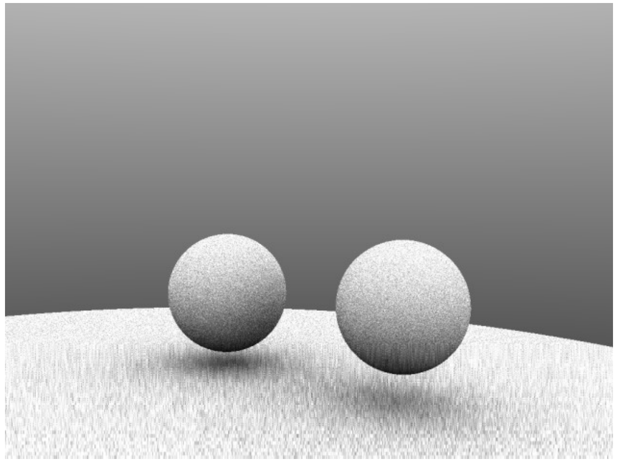
Ambient Occlusion – Estimator 1

(pdf = uniform spherical directions)

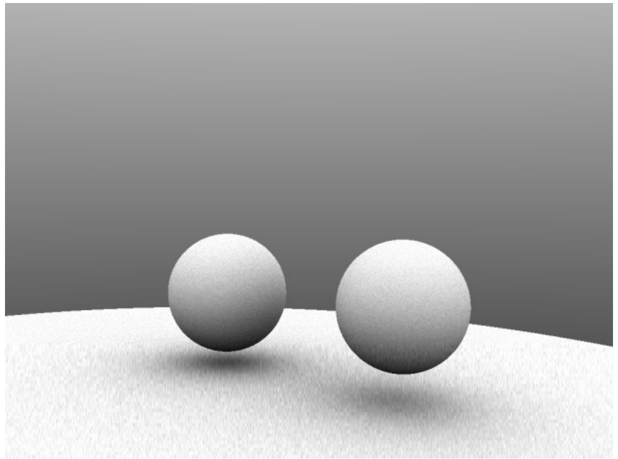
- Results (using **`rand()`** from **`stdlib.h`**):



$N = 10$



$N = 100$

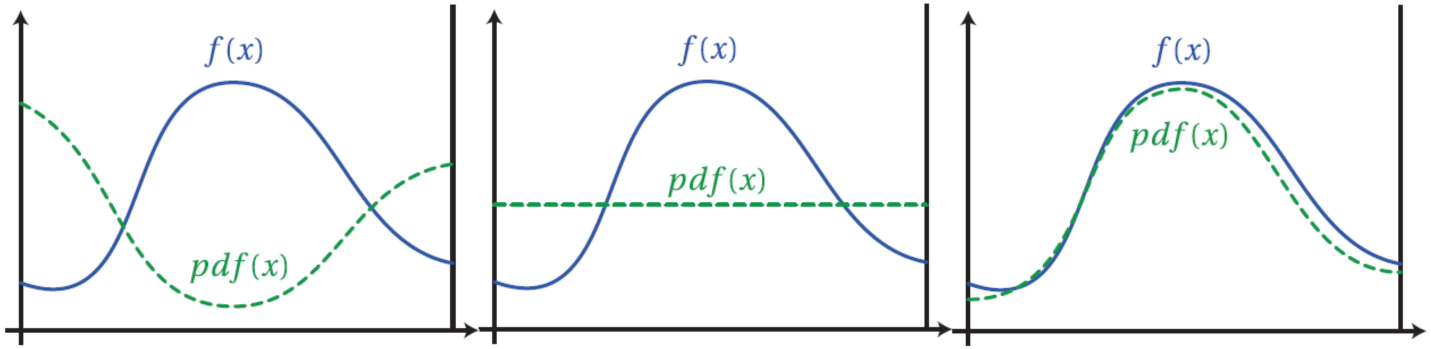


$N = 1000$

Ambient Occlusion: more efficient estimators?

- Recall that we can reduce the variance of an estimator (using importance sampling) by picking a pdf that better matches the profile of our integrand

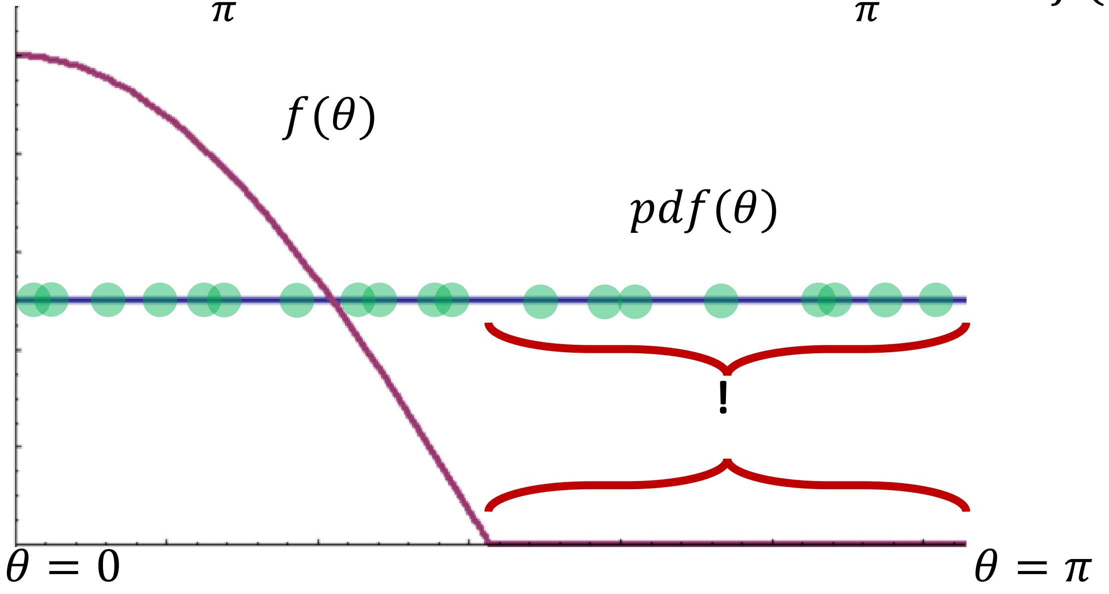
$$\bar{F} = \frac{1}{N}\sum_{i=0}^{N-1}\frac{f(X_i)}{pdf(X_i)}$$



- With our first estimator, the integrand is $f(\omega) = \frac{\max(\overrightarrow{n_x}\cdot\overrightarrow{\omega},0)V(x,\overrightarrow{\omega})}{\pi}$ but our *pdf* is $\frac{1}{4\pi}$

Ambient Occlusion: reducing variance

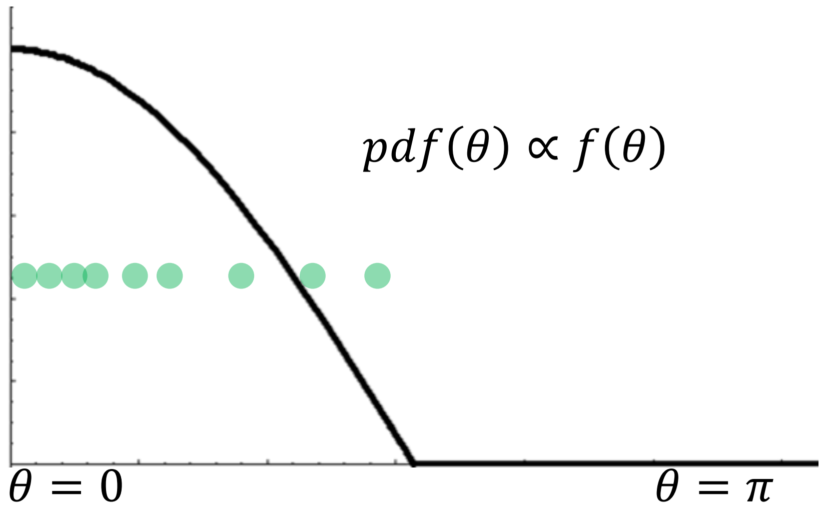
- What does our integrand look like? Visualizing spherical functions on a slide is hard, but if we fix $\overrightarrow{n_x} = \vec{z} = (0,0,1)$ and if we *ignore the visibility,* we can study the *pdf* and integrand visually:
- $f(\omega)|_{\overrightarrow{n_x}=\vec{z}} = \frac{\max((0,0,1)\cdot(\cos\phi\sin\theta, \sin\theta\sin\phi, \cos\theta), 0)}{\pi} = \frac{\max(cos\theta, 0)}{\pi} = f(\theta)$



- Clearly this *pdf* wastes a lot of samples!

Ambient Occlusion: reducing variance

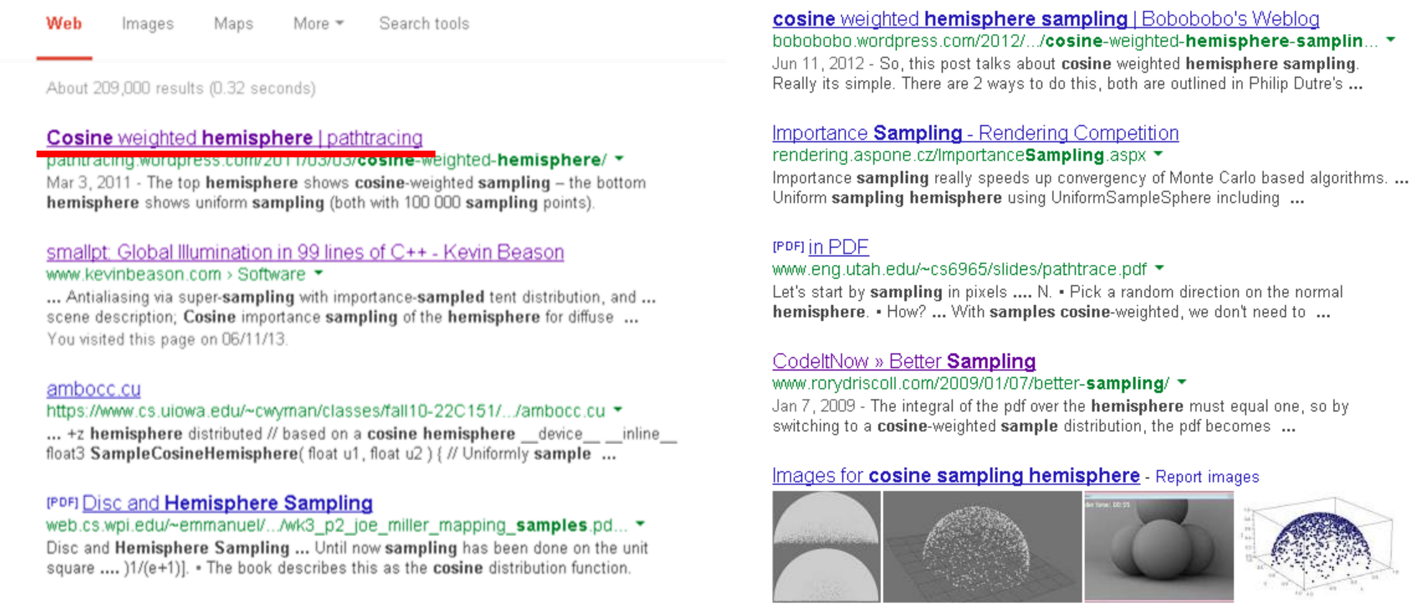
- If we continue to ignore visibility, we know **exactly** what our integrand looks like, $f(\omega) = \frac{\max(\overrightarrow{n_x} \cdot \vec{\omega}, 0)}{\pi}$, and we can adapt our *pdf* to distribute samples according to this profile:



Ambient Occlusion – Estimator 2

(pdf = **hemispherical cosine about the normal**)

- So, following the 3 steps for MC integration again:
  1. Pick a (better) *pdf* : $pdf(\overrightarrow{\omega_i}) = \frac{\max(\overrightarrow{n_x} \cdot \overrightarrow{\omega_i}, 0)}{\pi}$
  2. Generate random samples $\overrightarrow{\omega_i} = (\theta_i, \phi_i)$ according to this *pdf*:



Ambient Occlusion – Estimator 2
(pdf = hemispherical cosine about the normal)

- So, following the 3 steps for MC integration again:

1. Pick a (better) *pdf* : $pdf(\overrightarrow{\omega_i}) = \frac{\max(\overrightarrow{n_x} \cdot \overrightarrow{\omega_i}, 0)}{\pi}$
2. Generate random samples $\overrightarrow{\omega_i} = (\theta_i, \phi_i)$ according to this *pdf*:

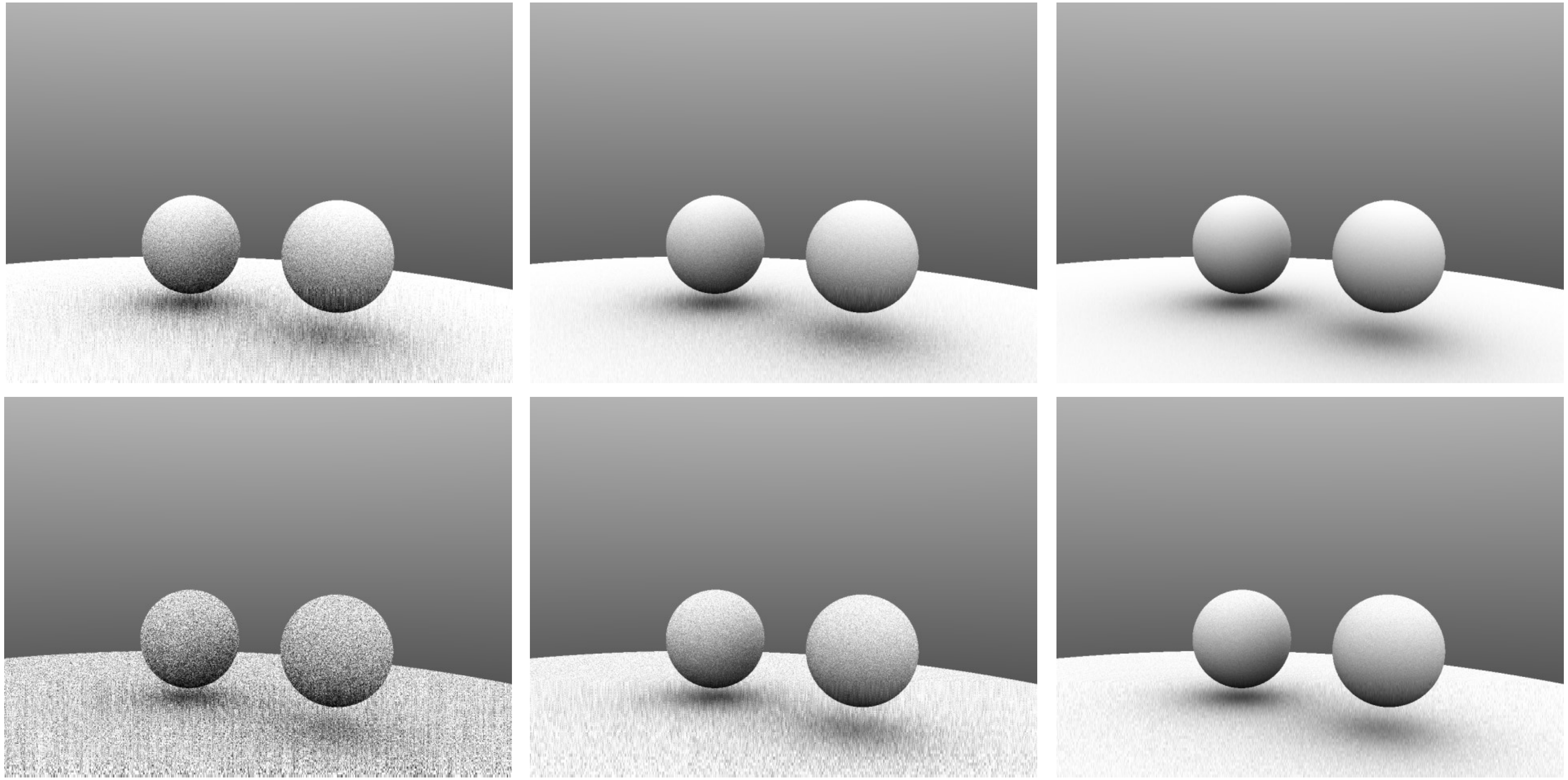
```
θi = acos( sqrt( rand() ) );
φi = 2 * M_PI * rand();
rotate points to a coordinate frame about nx;
```

3. Calculate our estimate of the AO integral:

$$\overline{L_{ao}}(x) = \frac{1}{N} \sum_{i=0}^{N-1} \frac{\max(\overrightarrow{n_x} \cdot \overrightarrow{\omega_i}, 0)\, V(x, \overrightarrow{\omega_i})}{\pi \left(\frac{\max(\overrightarrow{n_x} \cdot \overrightarrow{\omega_i}, 0)}{\pi} \right)}$$

Ambient Occlusion – Estimator 2
(pdf = **hemispherical cosine about the normal**)

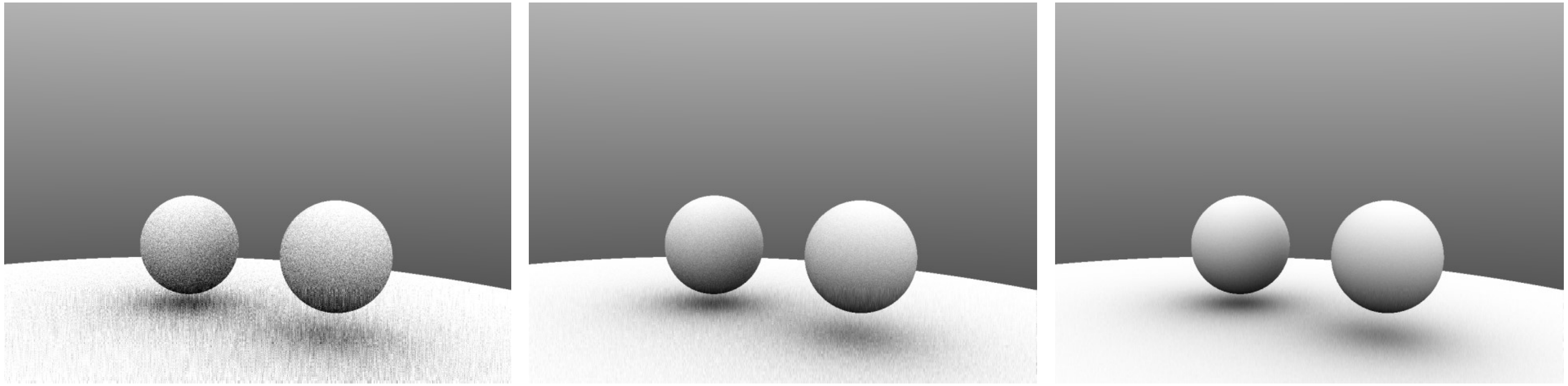
- Results of our new estimator vs. our naive estimator



$N = 10$ $\quad$ $N = 100$ $\quad$ $N = 1000$

Ambient Occlusion – Estimator 2
(pdf = **hemispherical cosine about the normal**)

- Results of our new estimator vs. our naive estimator



$N = 10$ $N = 100$ $N = 1000$

Review and more information

- Fundamentals of Computer Graphics
  - Chapter 20 Light
    - Section 20.1 Radiometry
    - Section 20.2 Transport Equation
    - The rest of the chapter is pretty short
- Physically Based Rendering, from theory to implementation
  - http://www.pbr-book.org/3ed-2018/contents.html
  - https://www.pbrt.org/
  - Chapter 13 covers Monte Carlo Integration