BRDFs and Variance Part 1

- BRDFs
- Materials
- Environment Lighting
- Monte-Carlo Sampling
- Biased vs Unbiased Estimators
- Physically-Based Rendering Methods

Review: The Rendering Equation

$$L_o(\mathbf{p},\omega_o) \;=\; L_e(\mathbf{p},\omega_o) + \int_{\mathcal{H}^2} f_r(\mathbf{p},\omega_i \to \omega_o) L_i(\mathbf{p},\omega_i) \cos\theta \, d\omega_i$$

$L_o(\mathbf{p},\omega_o)$ outgoing radiance at point $\mathbf{p}$ in outgoing direction ω_o

$L_e(\mathbf{p},\omega_o)$ emitted radiance at point $\mathbf{p}$ in outgoing direction ω_o

$f_r(\mathbf{p},\omega_i \to \omega_o)$ scattering function at point $\mathbf{p}$ from incoming direction ω_i to outgoing direction ω_o

$L_i(\mathbf{p},\omega_i)$ incoming radiance to point $\mathbf{p}$ from direction ω_i

Review: The Rendering Equation

$$L_o(\mathbf{p}, \omega_o) = L_e(\mathbf{p}, \omega_o) + \int_{\mathcal{H}^2} f_r(\mathbf{p}, \omega_i \to \omega_o) L_i(\mathbf{p}, \omega_i) \cos\theta \, d\omega_i$$

$L_o(\mathbf{p}, \omega_o)$ outgoing radiance at point $\mathbf{p}$ in outgoing direction ω_o

$L_e(\mathbf{p}, \omega_o)$ emitted radiance at point $\mathbf{p}$ in outgoing direction ω_o

$f_r(\mathbf{p}, \omega_i \to \omega_o)$ scattering function at point $\mathbf{p}$ from incoming direction ω_i to outgoing direction ω_o

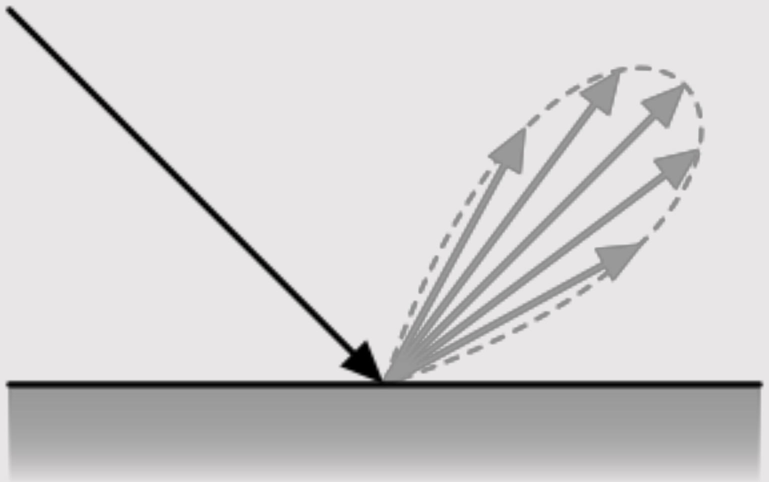
$L_i(\mathbf{p}, \omega_i)$ incoming radiance to point $\mathbf{p}$ from direction ω_i

Reflectance Functions

- **Reflectance Functions** refer to how light reflects off a surface

- **Bidirectional Reflectance Distribution Function (BRDF):**
    - *Bidirectional* – a function of two directions ω_i and ω_o
    - *Reflectance* – light changing directions
    - *Distribution* – likelihood of light changing to a certain direction
    - *Function* – it's a function

- Represented as a **Probability Distribution Function (PDF)**
    - Indicating the likelihood an incident direction ω_i at point $\mathbf{p}$ will reflect to an outgoing direction ω_o



Types of Reflectance Functions

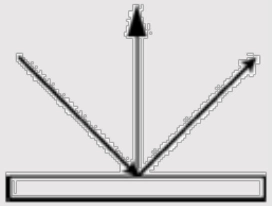
- A BRDF is a passthrough function
  - **Example:** an incoming ray ω_i at incident point **p** reflects 85% of red, 90% of green, and 50% of blue in the outgoing direction ω_o
    - Written as $f_r(\mathbf{p}, \omega_i \rightarrow \omega_o) = < 0.85, 0.90, 0.50 >$
    - Remainder of light gets absorbed
      - Conservation of energy

- Multiply the BRDF function by the incident radiance to get the outgoing radiance:

$$f_r(\mathbf{p}, \omega_i \rightarrow \omega_o) L_i(\mathbf{p}, \omega_i) \cos\theta$$

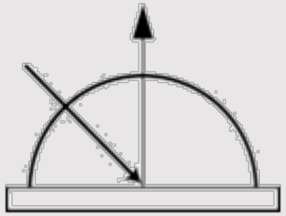
- When people talk about BRDFs, think materials!
  - Graphics is about seeing things
  - How we see a BRDF defines how we see a material

Types of Reflectance Functions

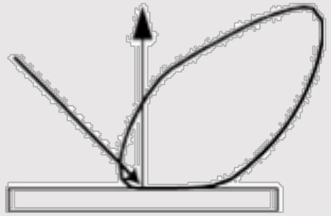
Ideal Specular

- Perfect mirror

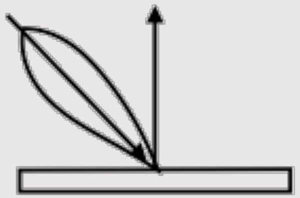
Ideal Diffuse

- Uniform in all directions

Glossy Specular

- Majority of light in reflected direction

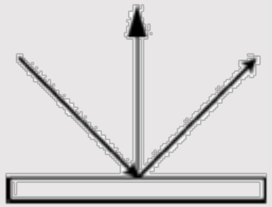
Retroreflective

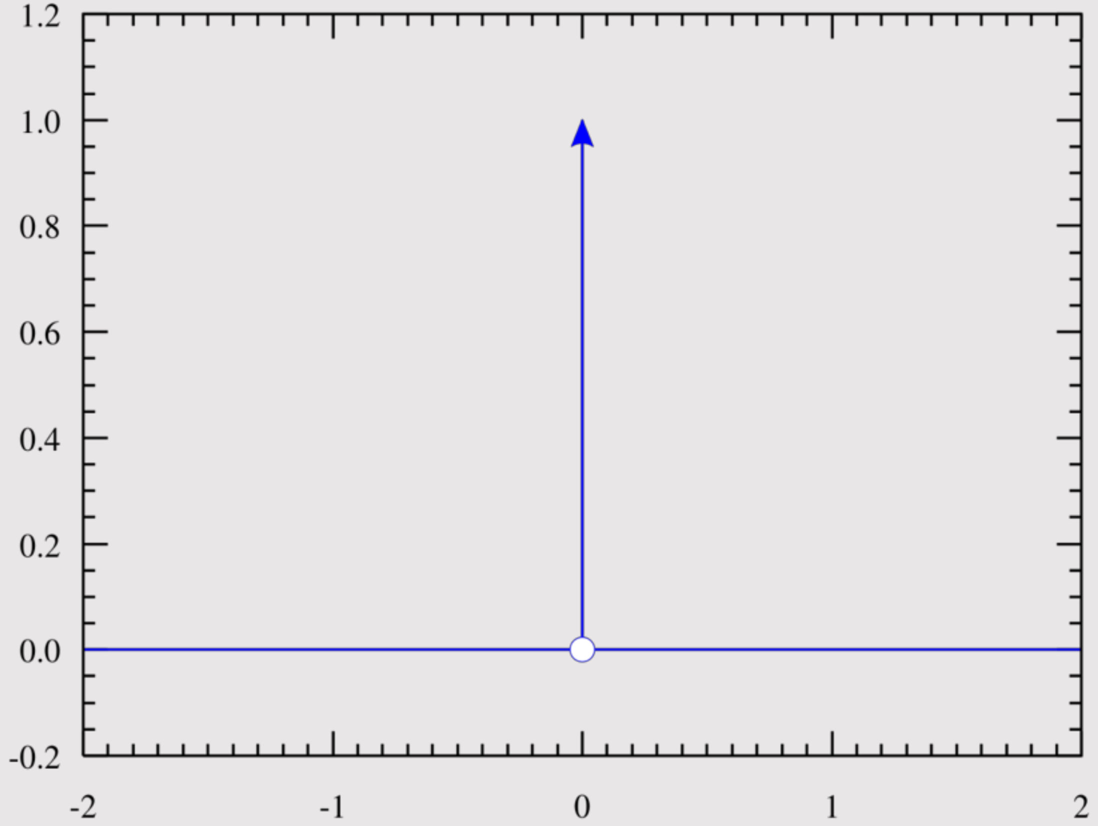
- Reflects light back towards source

Dirac Delta Distribution

- With ideal specular, the BRDF is a constant maximum reflectance (no energy absorbed) in the reflected direction
  - $f_r(\mathbf{p}, \omega_i \rightarrow \omega'_i) = < 1.0, 1.0, 1.0 >$
    - ω'_i is the incoming direction reflected about intersection point **p**’s normal

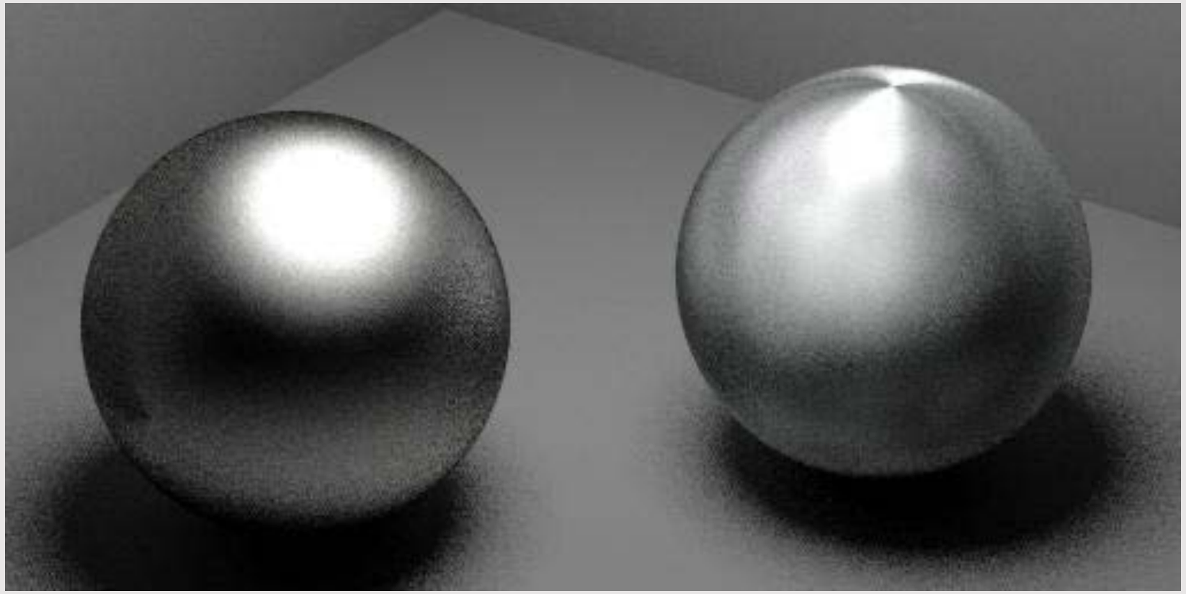
- Can represent the PDF of an ideal specular as a **dirac delta (δ) function**
  - 1 in one place, 0 everywhere else





Reflectance Direction

- **Isotropic BRDFs** are fixed when the incident and exiting directions are rotated about the normal
- **Anisotropic BRDFs** vary when the incident and exiting directions are rotated about the normal

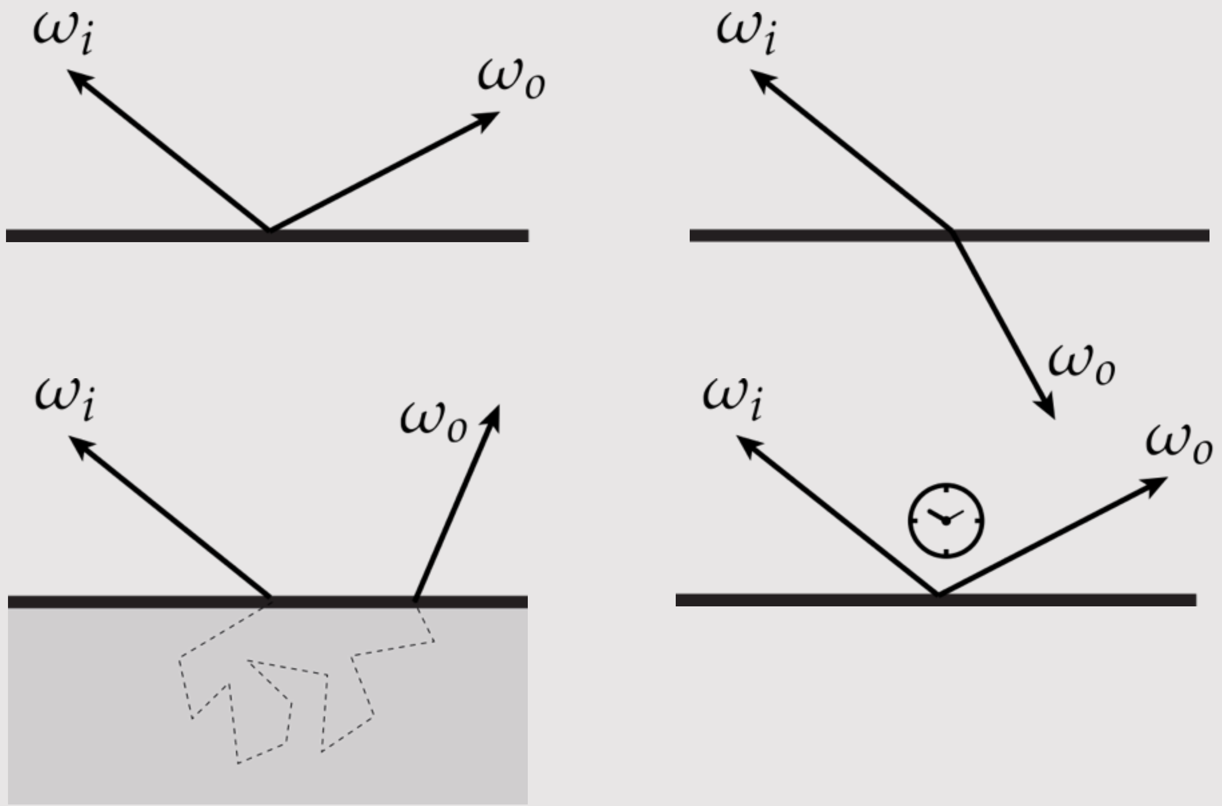


[isotropic] **[anisotropic]**

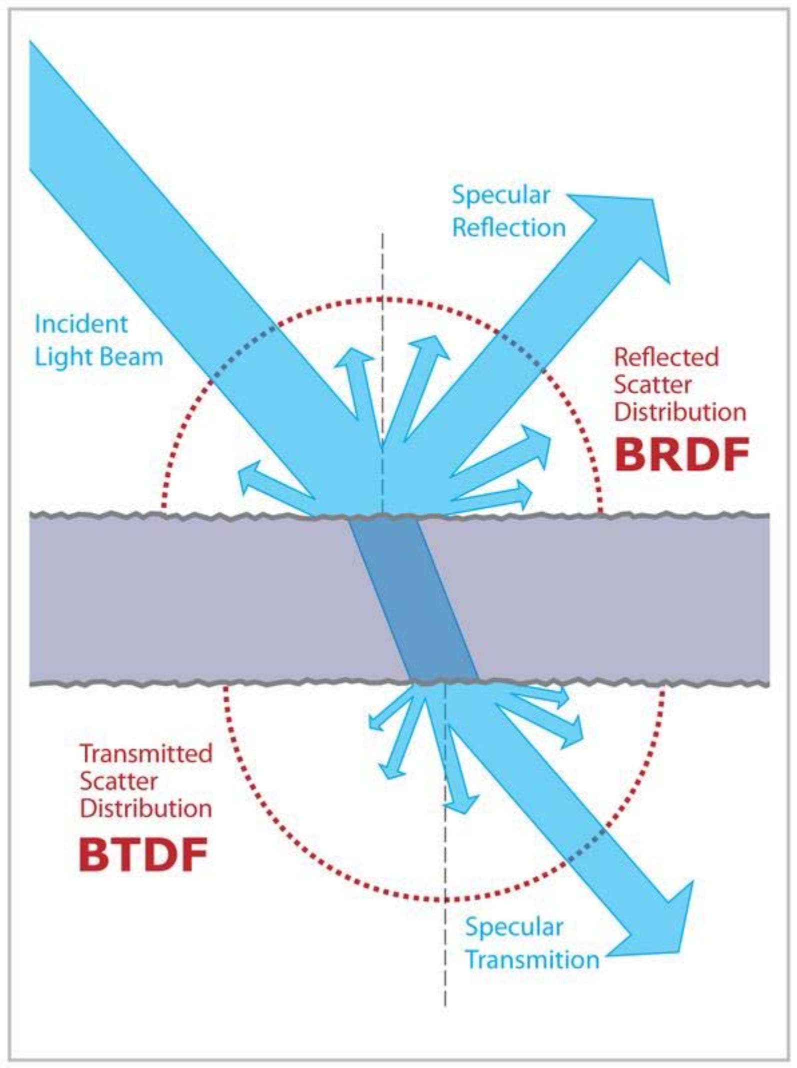
Models Of Scattering

- How can we model “scattering” of light?
- Many different things could happen to a photon:
  - Bounces off surface
  - Transmitted through surface
  - Bounces around inside surface
  - Absorbed and re-emitted

- What goes in must come out!
  - Total energy must be conserved

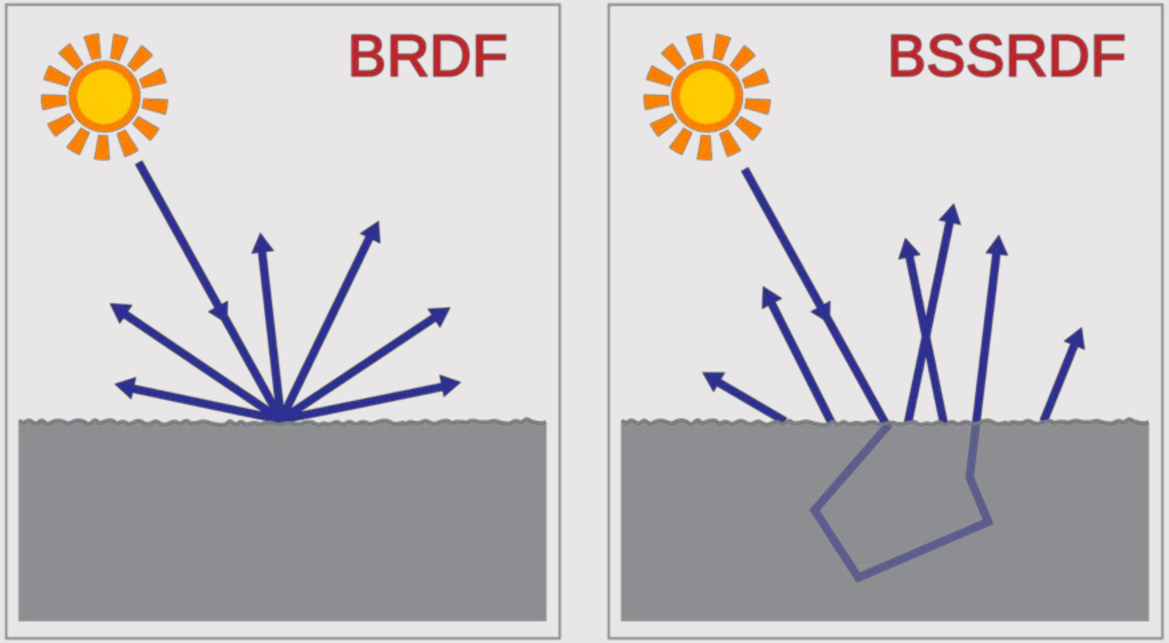


Much More Than Just A BRDF



- **BRDFs** - Bidirectional Reflectance Distribution Function
  - Describes light reflecting without entering the surface
  - Ex: lambertian, mirror
- **BTDFs** - Bidirectional Transmittance Distribution Function
  - Describes light entering the surface
  - Ex: glass
- **BSDFs** - Bidirectional Scattering Distribution Function
  - Encapsulates BRDFs and BTDFs
  - BRDFs are just more common in literature :)

Much Much More Than Just A BRDF

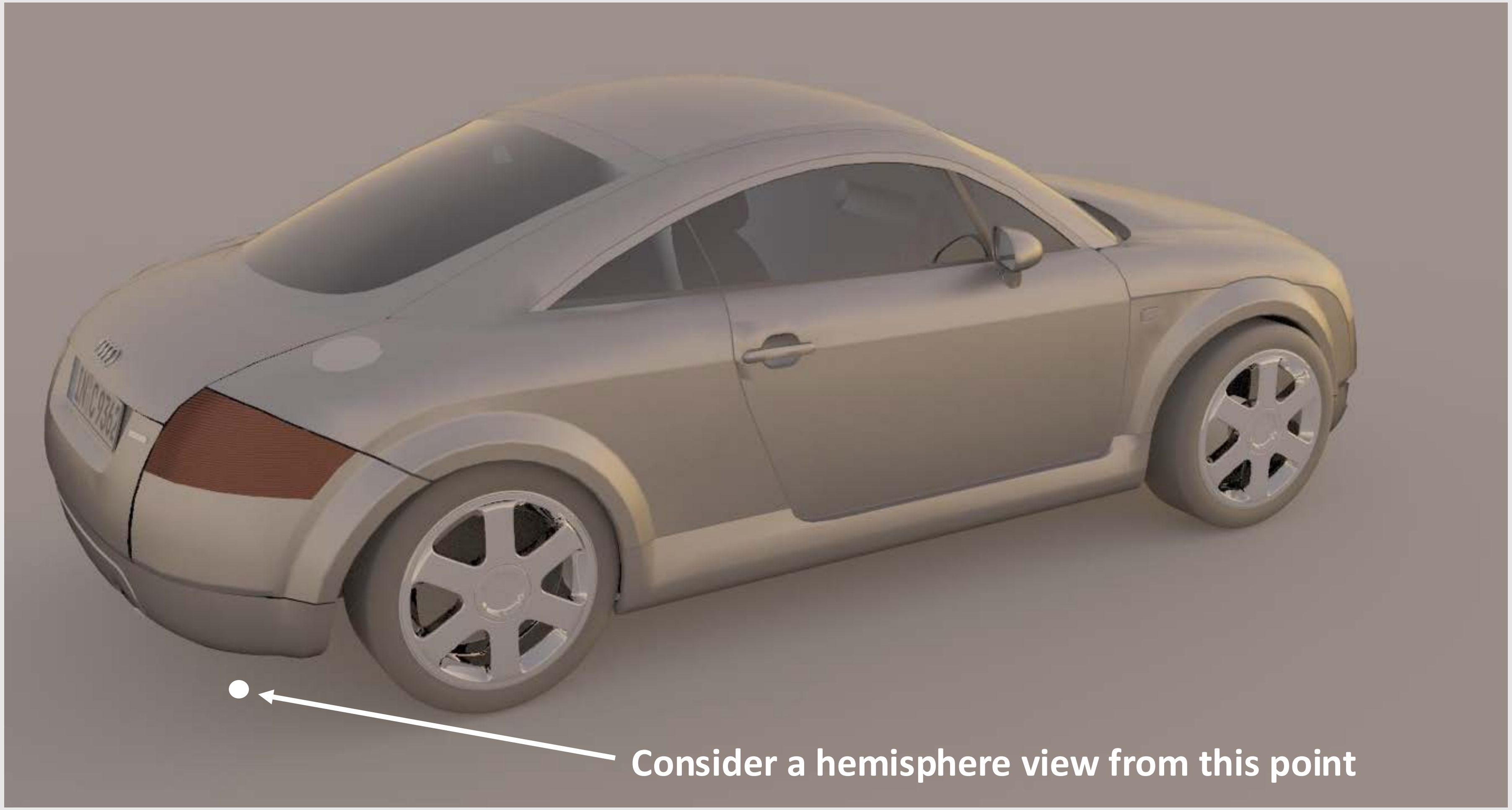


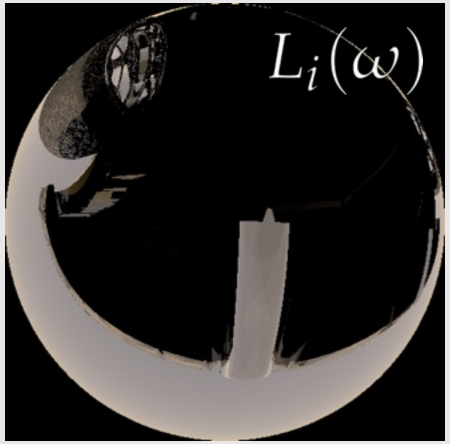
- **BSSRDFs**, *SS - Surface Scattering
  - Describes light entering and scattering the surface before being reflected out
  - Ex: milk
- **BSSTDFs**, *SS - Surface Scattering
  - BTDF but with subsurface scattering
  - Ex: also milk
- **BSSDFs**, *SS - Surface Scattering
  - Encapsulates BSSRDFs and BSSTDFs

BRDF Examples

BRDFs can be a mix of diffuse and specular

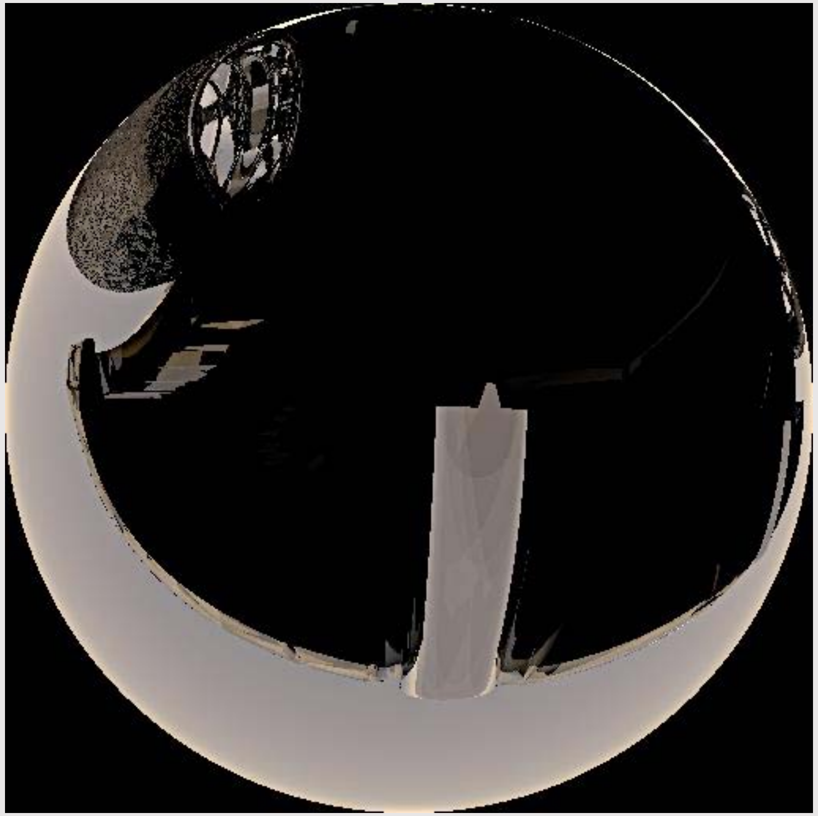
Hemispherical Incident Radiance



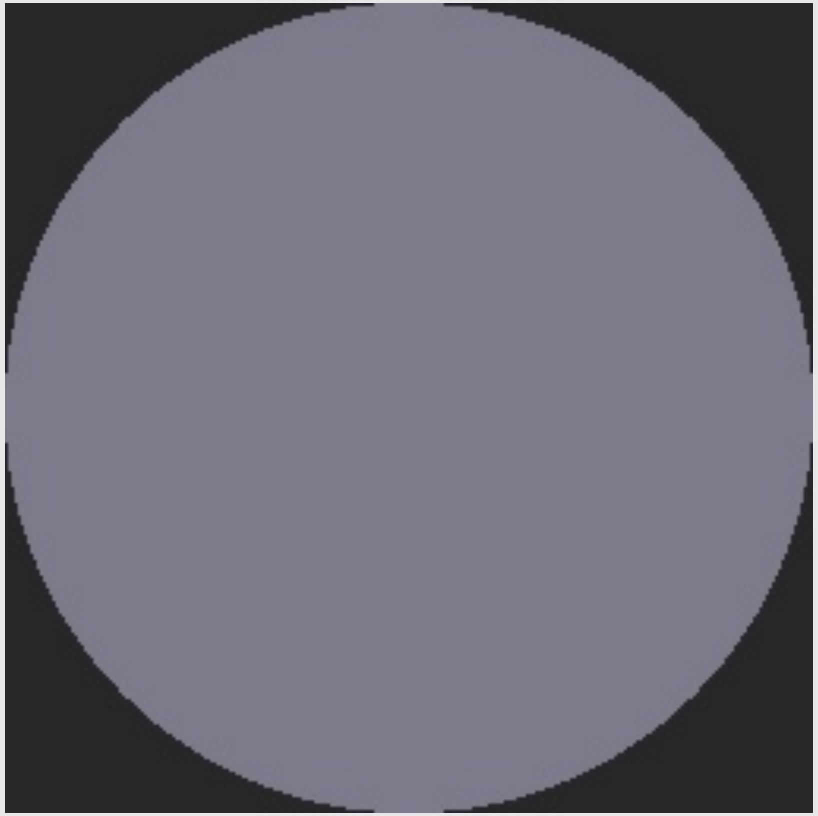


At any point on any surface in the scene, there's an incident radiance field that gives the directional distribution of illumination at the point

Diffuse Exitant Radiance



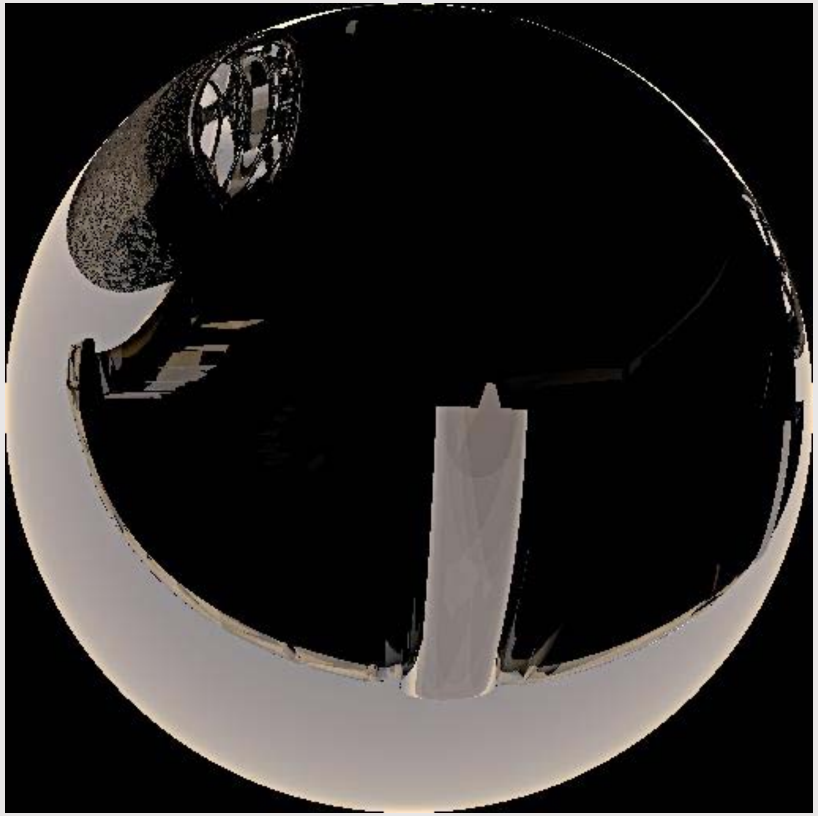
[incident radiance]



[exitant radiance]

Colors sampled from uniform hemisphere blend all colors into one average color.

Ideal Specular Exitant Radiance

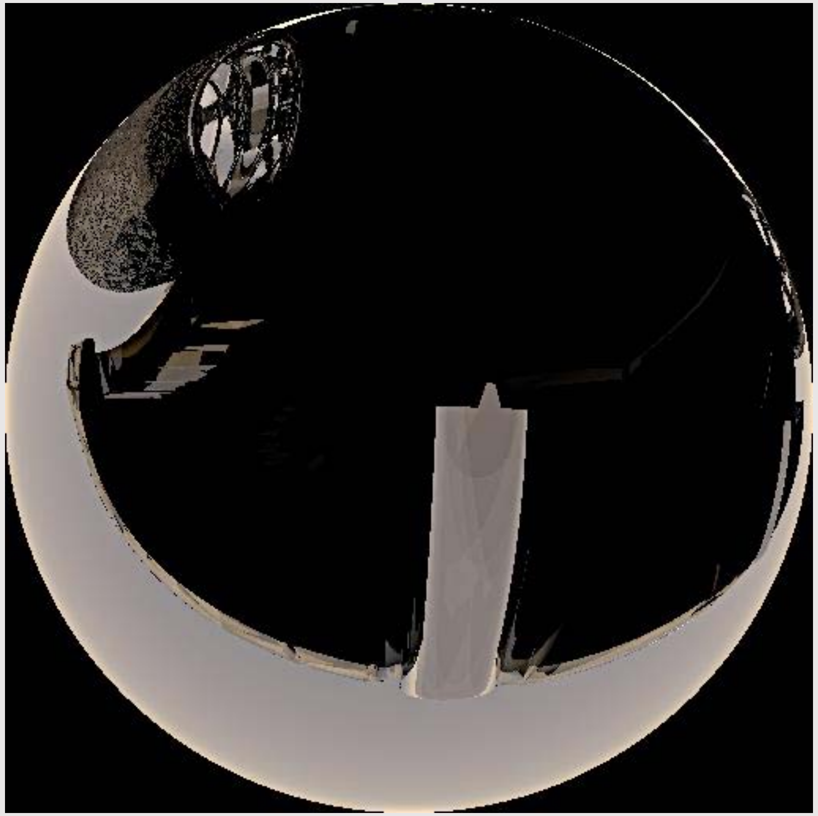


[incident radiance]

[exitant radiance]

Incident radiance is “flipped around normal” to get exitant radiance.

Plastic Exitant Radiance

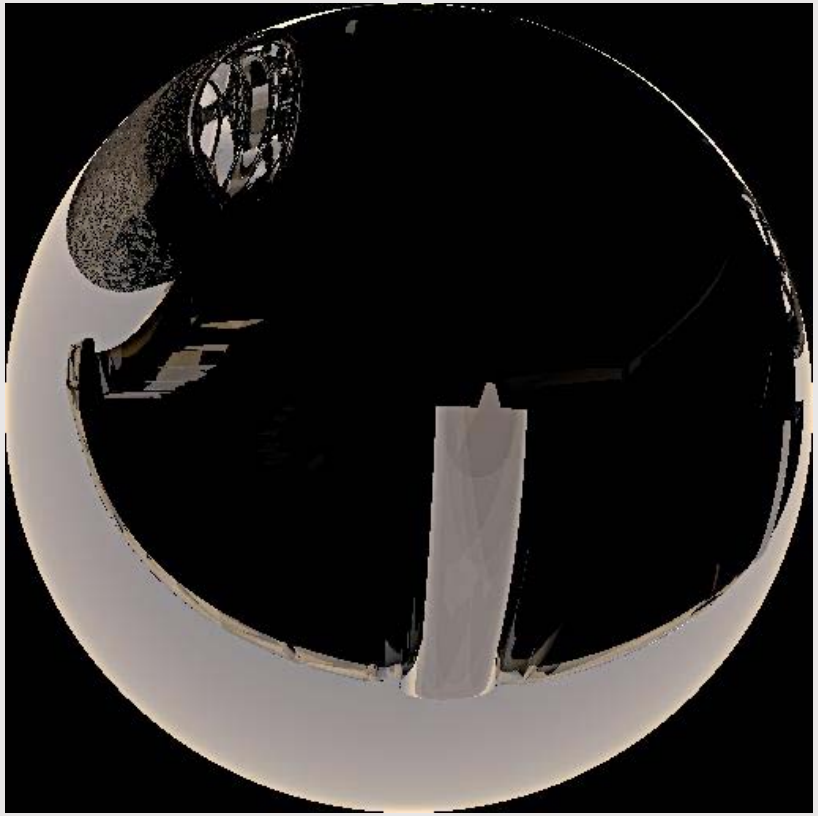


[incident radiance]

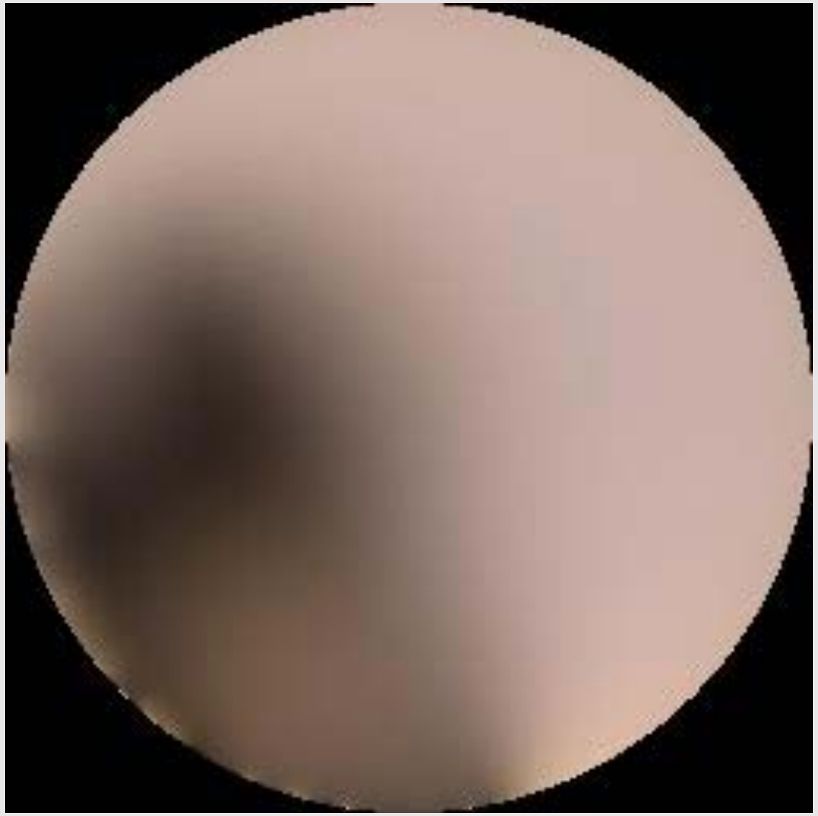
[exitant radiance]

Incident radiance gets flipped and blurred.
Common example of a material that has both diffuse and specular properties.

Copper Exitant Radiance



[incident radiance]



[exitant radiance]

More blurring, plus coloration (nonuniform absorption across frequencies).
Copper absorbs some colors, and emits the rest, giving it a “warm brown” color.

Integration of BRDF

- When integrating the BRDF over the hemisphere, total value will be less than or equal to 1

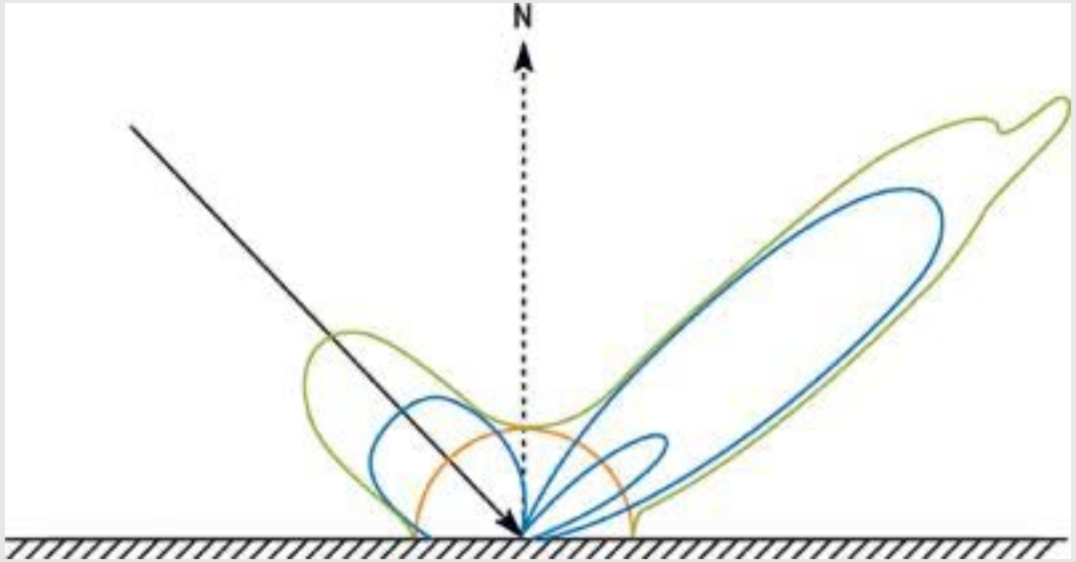
$$\int_{\mathcal{H}^2} f_r(\omega_i \to \omega_o) \; \cos\theta \, d\omega_i \leq 1$$

- Conservation of energy: outgoing energy should be less than or equal to incoming energy
  - Energy should not be created
  - Energy lost is absorbed into the intersected material
    - BRDF helps capture that absorption

- BRDF can never be negative

$$f_r(\omega_i \to \omega_o) \geq 0$$

  - A negative BRDF would imply negative energy???



Radiometric Description of BRDF

- **Recall:** differential irradiance landing on surface from differential cone of directions ω_i

$$dE(\omega_i) = dL(\omega_i) cos\,\theta_i$$

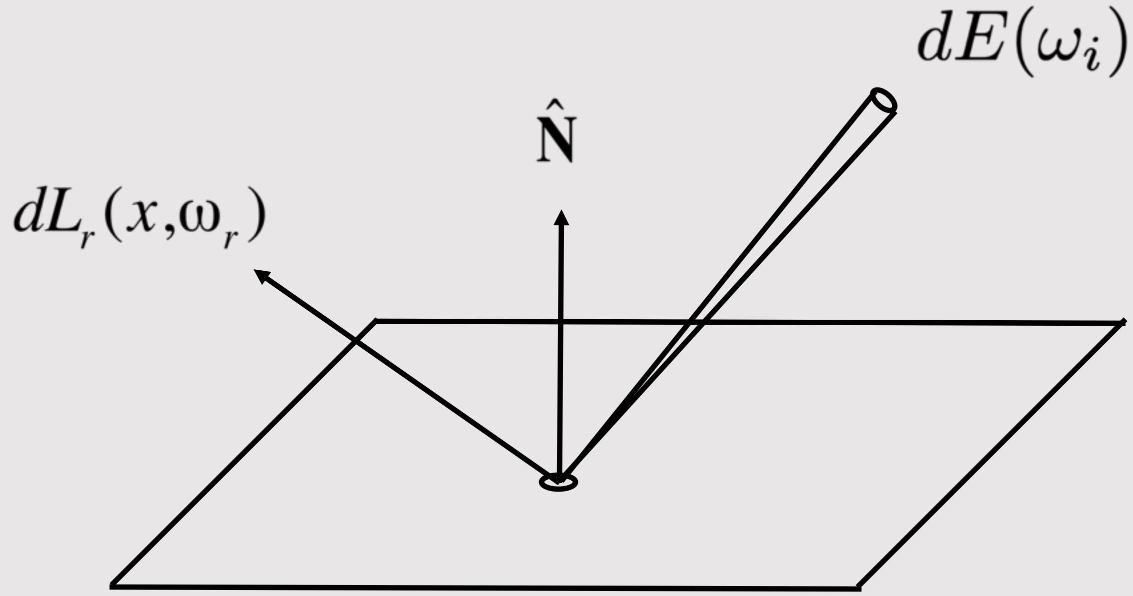
- **Recall:** differential radiance reflected in direction ω_r (due to differential irradiance from ω_i)

$$dL_r(\omega_r)$$

- BRDF captures the ratio between the incoming irradiance and the outgoing radiance

$$f_r(\omega_i \rightarrow \omega_o) = \frac{dL_o(\omega_o)}{dE_i(\omega_i)} = \frac{dL_o(\omega_o)}{dL_i(\omega_i)\cos\theta_i} \left[\frac{1}{sr}\right]$$

  - Given the incoming irradiance, computes the outgoing radiance



$$dL_r(\omega_r) \propto dE_i(\omega_i)$$

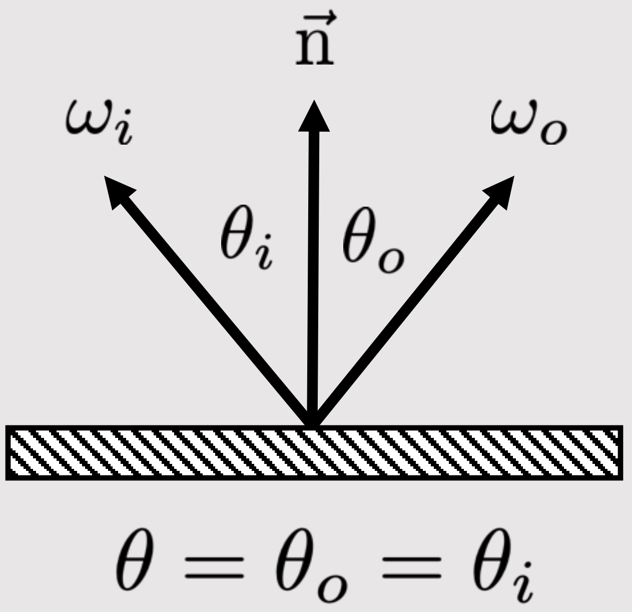
measured in steradians

- ~~BRDFs~~
- Materials
- Environment Lighting

Change Of Syntax

- **Surface-local space**
  - Normal is $n = < 0, 1, 0 >$
  - Unit directions w_i and w_o point away from intersection point p

- All material interactions will occur in surface-local space
  - Transform w_i to surface-local space
  - Compute new outgoing ray w_o
  - Transform w_o back to world space



Lambertian Material

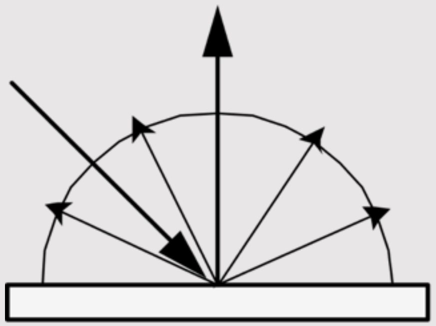
- Also known as diffuse
- Light is equally likely to be reflected in each output direction
  - BRDF is a constant, relying on **albedo** (ρ)

$$f_r = \frac{\rho}{\pi}$$

  - BRDF can be pulled out of the integral

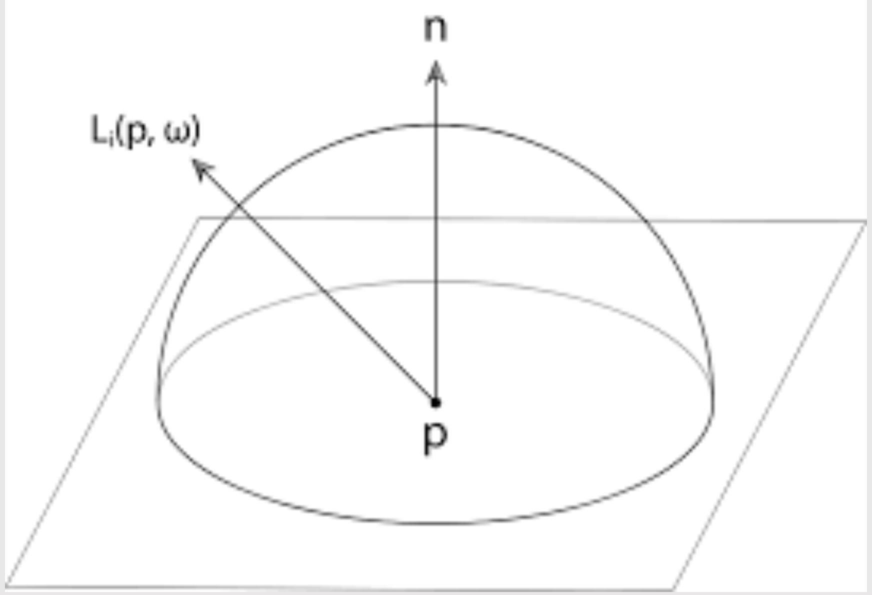
$$\begin{aligned} L_o(\omega_o) &= \int_{H^2} f_r \, L_i(\omega_i) \, \cos\theta_i \, \mathrm{d}\omega_i \\ &= f_r \int_{H^2} L_i(\omega_i) \, \cos\theta_i \, \mathrm{d}\omega_i \\ &= f_r E \end{aligned}$$

- Easy! Pick any outgoing ray w_o



Minions (2015) Illumination Entertainment

Lambertian Material



- The **albedo** (ρ) describes how much of each color is is reflected

- **Why does the Lambertian PDF divide by π?**
  - Consider our irradiance integral:

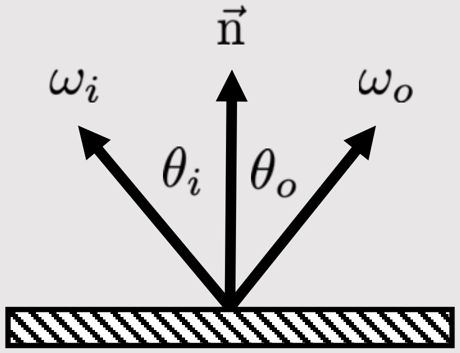
$$\int_{\mathcal{H}^2} f_r(\omega_i \to \omega_o) \cos\theta \, d\omega_i \leq 1$$

  - If the albedo is 1, then the integral is greater than 1 (cosine integral over hemisphere is π)
    - Divide the albedo by π to normalize the irradiance so it is less than or equal to 1

$$f_r = \frac{\rho}{\pi}$$

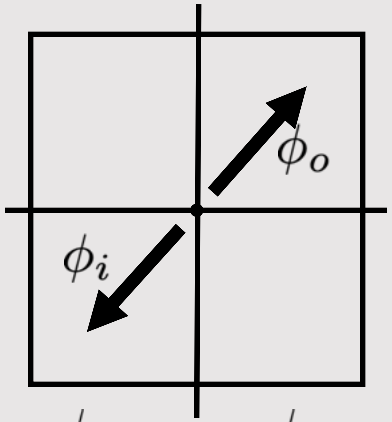
Reflective Material

[side view]



$$\theta = \theta_o = \theta_i$$

[top view]



$$\phi_o = -\phi_i$$

- Reflectance equation described as:

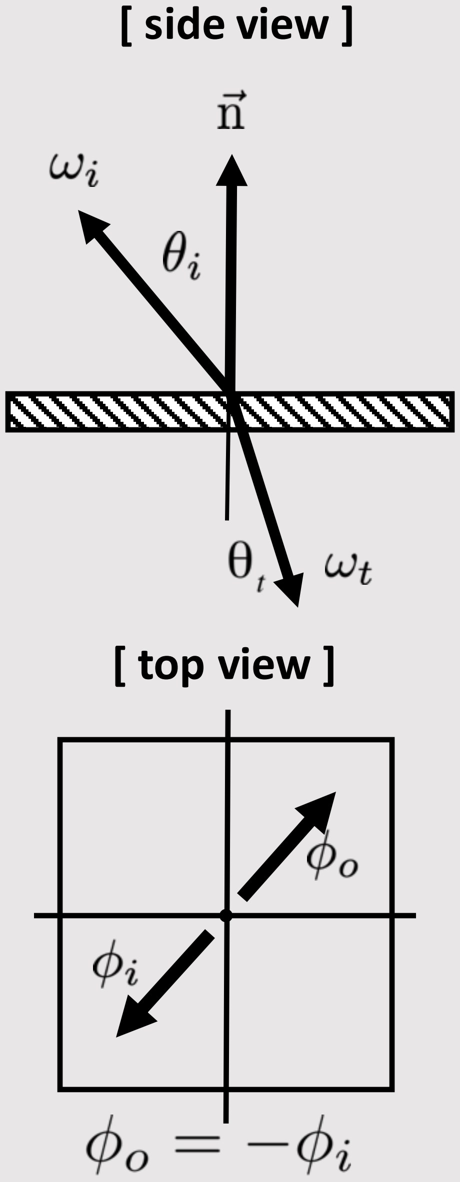
$$\omega_o = -\omega_i + 2(\omega_i \cdot \vec{\mathrm{n}})\vec{\mathrm{n}}$$

  - Recall incoming and outgoing rays share same origin point **p**

- BRDF represented by dirac delta (δ) function
  - 1 when ray is perfect reflection, 0 everywhere else
  - All radiance gets reflected, nothing absorbed

- In practice, no hope of finding reflected direction via random sampling
  - Simply pick the reflected direction!

Refractive Material



- Refractive equation described as:

$$\eta_i \sin\theta_i = \eta_t \sin\theta_t$$

  - Also known as Snell's Law

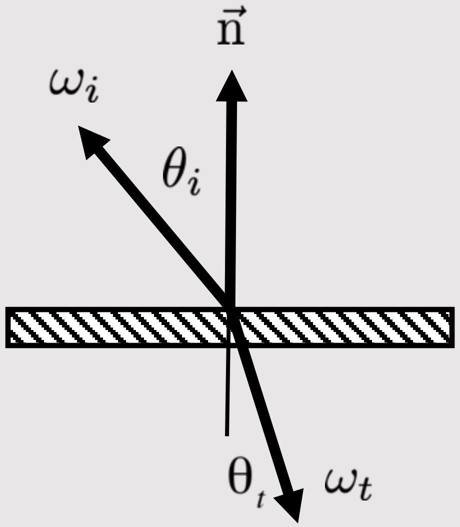
- η_i and η_t describe the index of refraction of the incoming and outgoing mediums
  - Example: η_i is air, η_t is water

Medium	$\boldsymbol{\eta}$
Vacuum	1.0
Air (sea level)	1.00029
Water (20°C)	1.333
Glass	1.5-1.6
Diamond	2.42

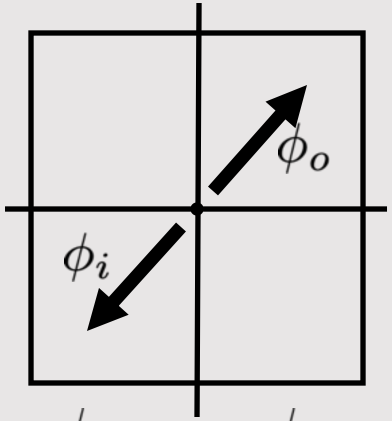
  - η is the ratio of the speed of light in a vacuum to that in a second medium of greater density
    - The larger the η, the denser the material

Refractive Material

[side view]



[top view]



$$\phi_o = -\phi_i$$

- Refractive equation described as:

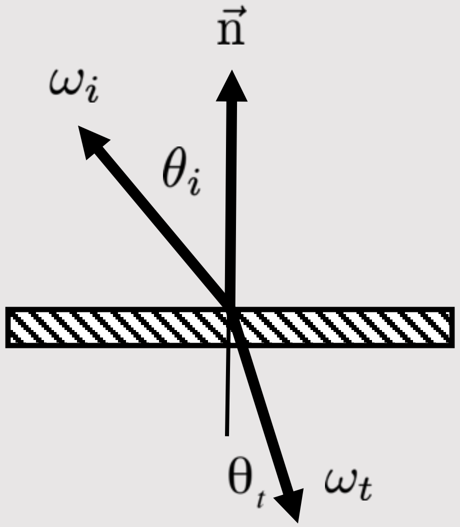
$$\eta_i \sin\theta_i = \eta_t \sin\theta_t$$

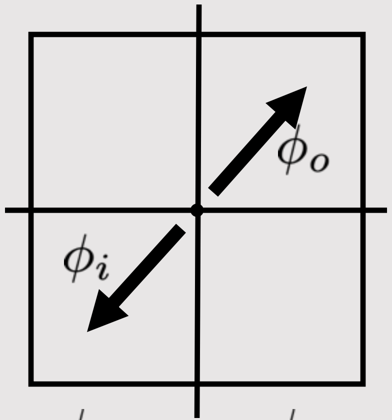
  - Also known as Snell's Law

- Can rewrite the equation as:

$$\begin{aligned}\cos\theta_t &= \sqrt{1-\sin^2\theta_t} \\ &= \sqrt{1-\left(\frac{\eta_i}{\eta_t}\right)^2 \sin^2\theta_i} \\ &= \sqrt{1-\left(\frac{\eta_i}{\eta_t}\right)^2 (1-\cos^2\ \theta_i)}\end{aligned}$$

Refractive Material





$$\phi_o = -\phi_i$$

- Refractive equation described as:

$$\eta_i \sin\theta_i = \eta_t \sin\theta_t$$

  - Also known as Snell's Law

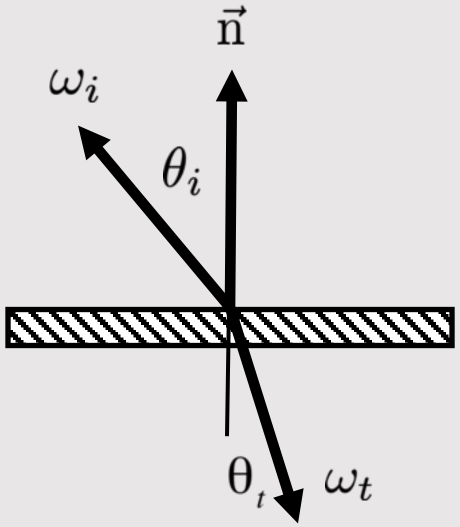
- Can rewrite the equation as:

$$\cos\theta_t = \sqrt{1 - \sin^2\theta_t}$$
$$= \sqrt{1 - \left(\frac{\eta_i}{\eta_t}\right)^2 \sin^2\theta_i}$$
$$= \sqrt{1 - \left(\frac{\eta_i}{\eta_t}\right)^2 (1 - \cos^2\theta_i)}$$

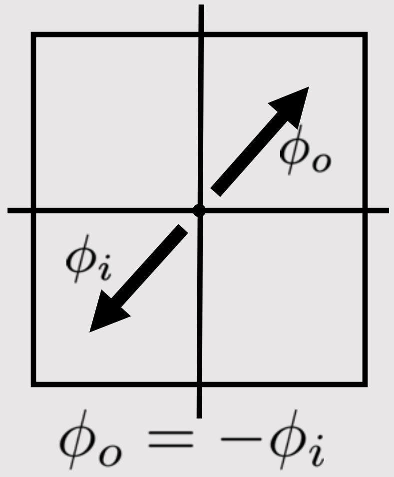
what if the term in the square root is negative?

Refractive Material

[side view]



[top view]



We know that:

$$0 < cos^2\theta < 1$$

And so:

$$0 < 1 - (1 - cos^2\theta) < 1$$

But if $\eta_i / \eta_t > 1$ then it is possible that:

$$1 - \left(\frac{\eta_i}{\eta_t}\right)^2 (1 - \cos^2\theta_i) < 0$$

This is known as **total internal reflection**, and happens when the incoming index η_i is denser than the outgoing index η_t

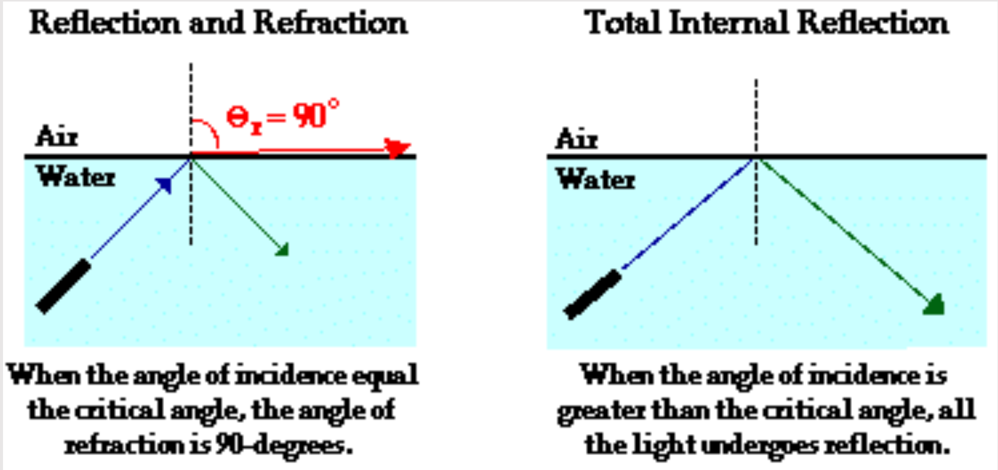
Hence $\eta_i / \eta_t > 1$

Total Internal Reflection

- When going from a more dense (i.e water) to less dense (i.e air) material, light will bend more towards the horizon
    - The incident angle that causes an outgoing 90deg angle is the **critical angle**
    - Can solve for critical angle by solving for θ:

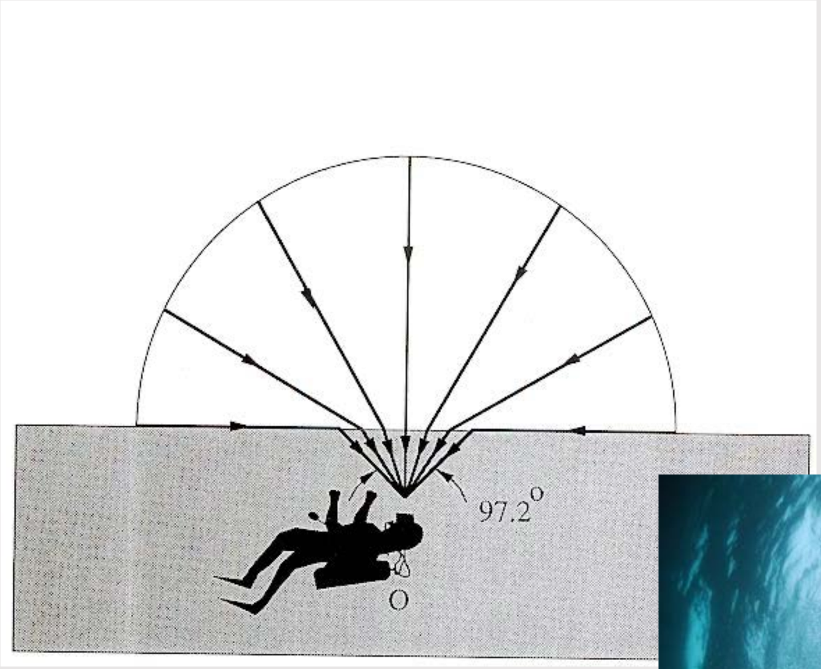
$$1 - \frac{{\eta_i}^2}{{\eta_t}^2}(1 - cos^2\theta) = 0$$

    - When the critical angle is exceeded, the ray is reflected back into the surface



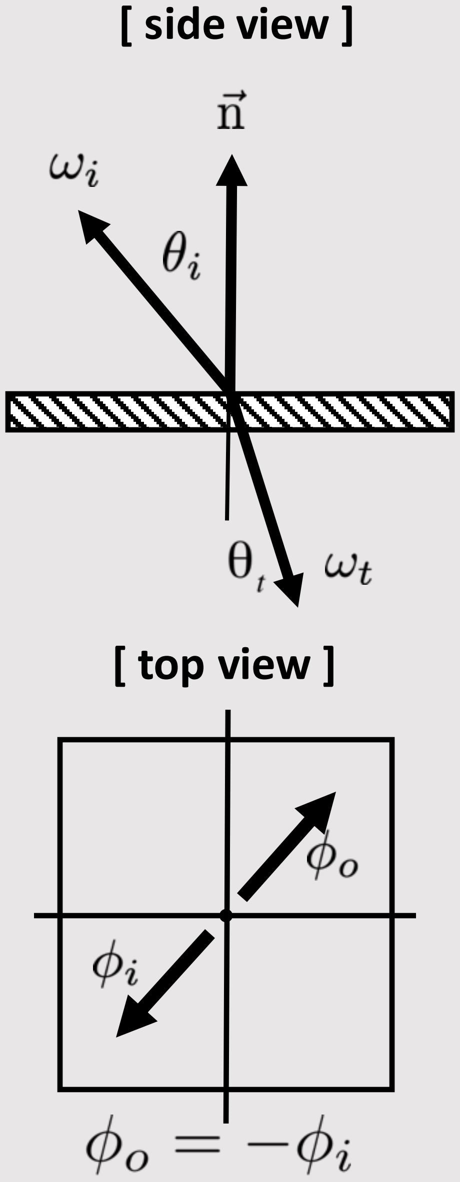
reflections at surface of water viewing from under the surface

Optical Manhole



- Works the other direction too
  - Light rays from air entering water bend themselves into a smaller solid angle
    - Pitch black in surrounding areas
  - Gives the illusion that light is a small cone around user

Refractive Material



- Refractive equation described as:

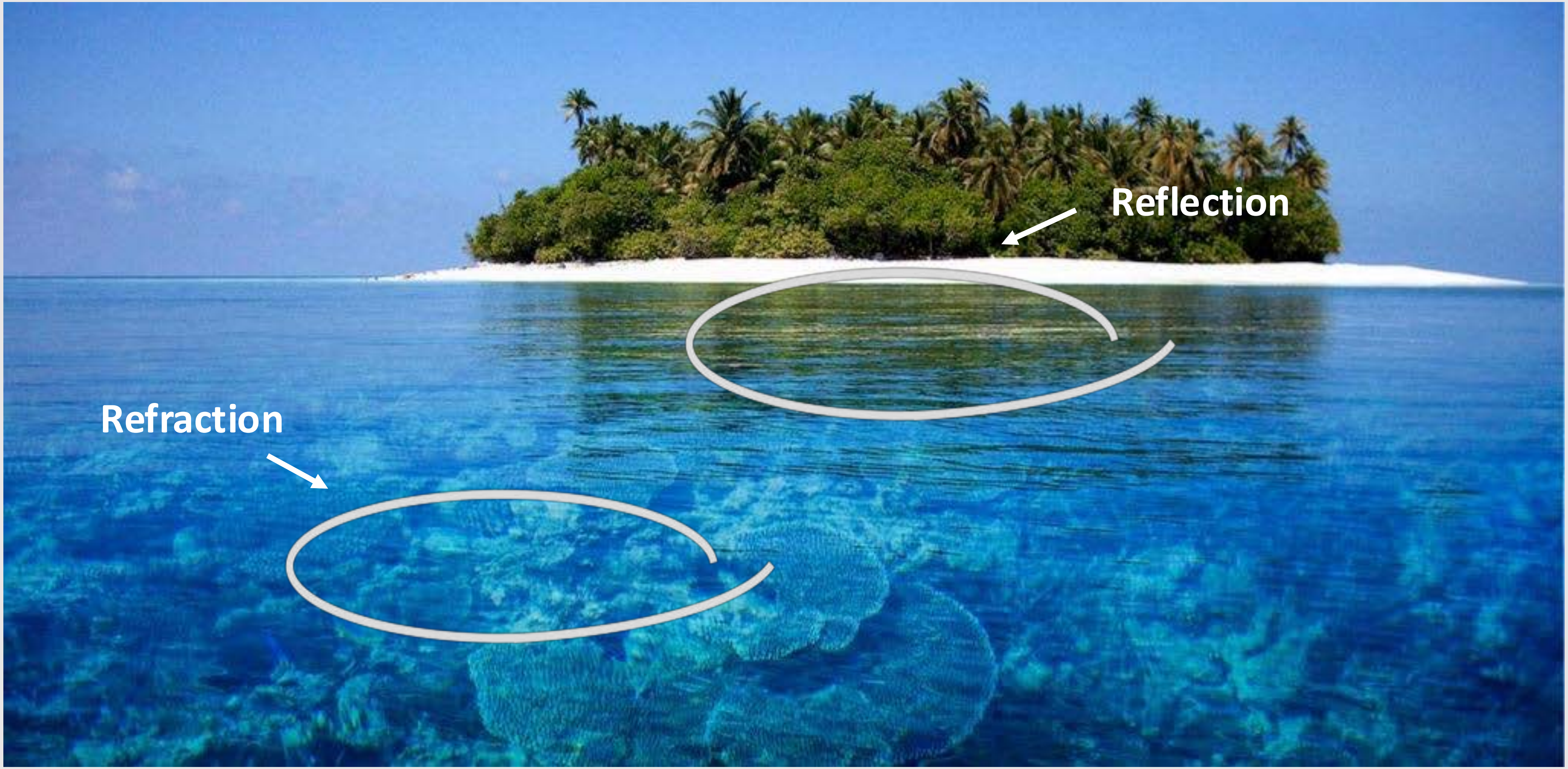
$$\eta_i \sin\theta_i = \eta_t \sin\theta_t$$

  - Also known as Snell's Law

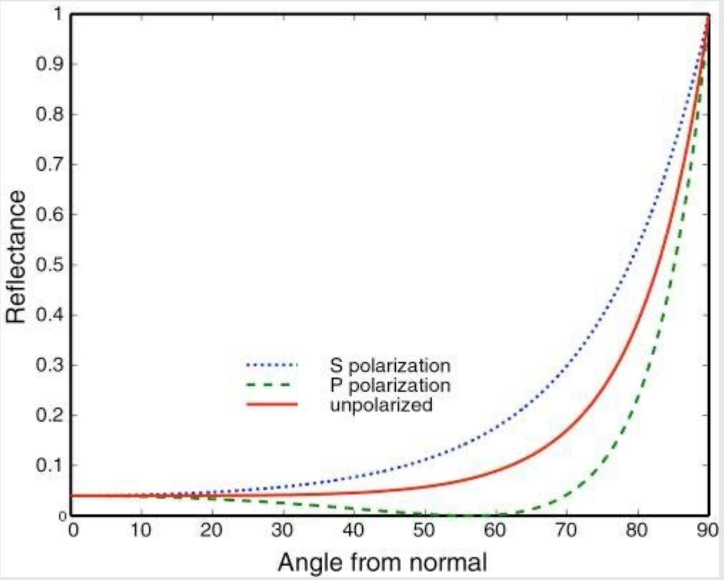
- BRDF represented by dirac delta (δ) function
  - 1 when ray is perfect refraction, 0 everywhere else
  - **Edge Case:** 1 when ray is total internal reflection
  - All radiance gets reflected, nothing absorbed

- In practice, no hope of finding refracted direction via random sampling
  - Simply pick the refracted direction!

Refractive Isn't Just Refractive



Fresnel Reflectance



- The amount of reflectance increases for refractive material as the angle from the normal increases
  - i.e the angle gets steeper
- Known as the **Fresnel coefficient**

Lafortune et al. (1997)

Fresnel Coefficient

Computing the Fresnel coefficient is kinda hard…

The reflectance for s-polarized light is

$$R_s = \left| \frac{Z_2 \cos\theta_i - Z_1 \cos\theta_t}{Z_2 \cos\theta_i + Z_1 \cos\theta_t} \right|^2,$$

while the reflectance for p-polarized light is

$$R_p = \left| \frac{Z_2 \cos\theta_t - Z_1 \cos\theta_i}{Z_2 \cos\theta_t + Z_1 \cos\theta_i} \right|^2,$$

where Z_1 and Z_2 are the wave impedances of media 1 and 2, respectively.

We assume that the media are non-magnetic (i.e., $\mu_1 = \mu_2 = \mu_0$), which is typically a good approximation at optical frequencies (and for transparent media at other frequencies).[3] Then the wave impedances are determined solely by the refractive indices n_1 and n_2:

$$Z_i = \frac{Z_0}{n_i},$$

where Z_0 is the impedance of free space and $i = 1,2$. Making this substitution, we obtain equations using the refractive indices:

$$R_s = \left| \frac{n_1 \cos\theta_i - n_2 \cos\theta_t}{n_1 \cos\theta_i + n_2 \cos\theta_t} \right|^2 = \left| \frac{n_1 \cos\theta_i - n_2 \sqrt{1 - \left(\frac{n_1}{n_2} \sin\theta_i\right)^2}}{n_1 \cos\theta_i + n_2 \sqrt{1 - \left(\frac{n_1}{n_2} \sin\theta_i\right)^2}} \right|^2,$$

$$R_p = \left| \frac{n_1 \cos\theta_t - n_2 \cos\theta_i}{n_1 \cos\theta_t + n_2 \cos\theta_i} \right|^2 = \left| \frac{n_1 \sqrt{1 - \left(\frac{n_1}{n_2} \sin\theta_i\right)^2} - n_2 \cos\theta_i}{n_1 \sqrt{1 - \left(\frac{n_1}{n_2} \sin\theta_i\right)^2} + n_2 \cos\theta_i} \right|^2.$$

Schlick's Approximation

Easier to compute :)

Harder to spell : (

According to Schlick's model, the specular reflection coefficient R can be approximated by:

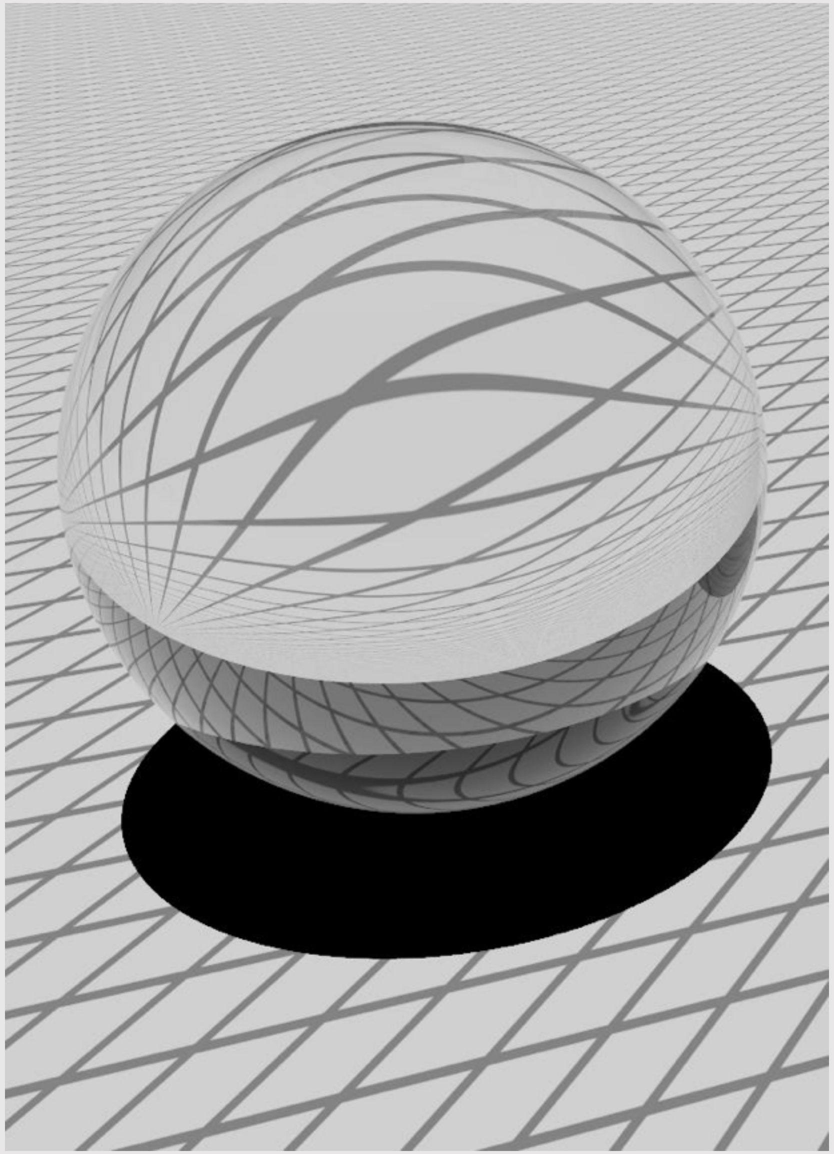
$$R(\theta) = R_0 + (1 - R_0)(1 - \cos\theta)^5$$

where

$$R_0 = \left(\frac{n_1 - n_2}{n_1 + n_2}\right)^2$$

$cos\theta$ is the same as $n \cdot \omega$ for normal n and ray ω

Glass

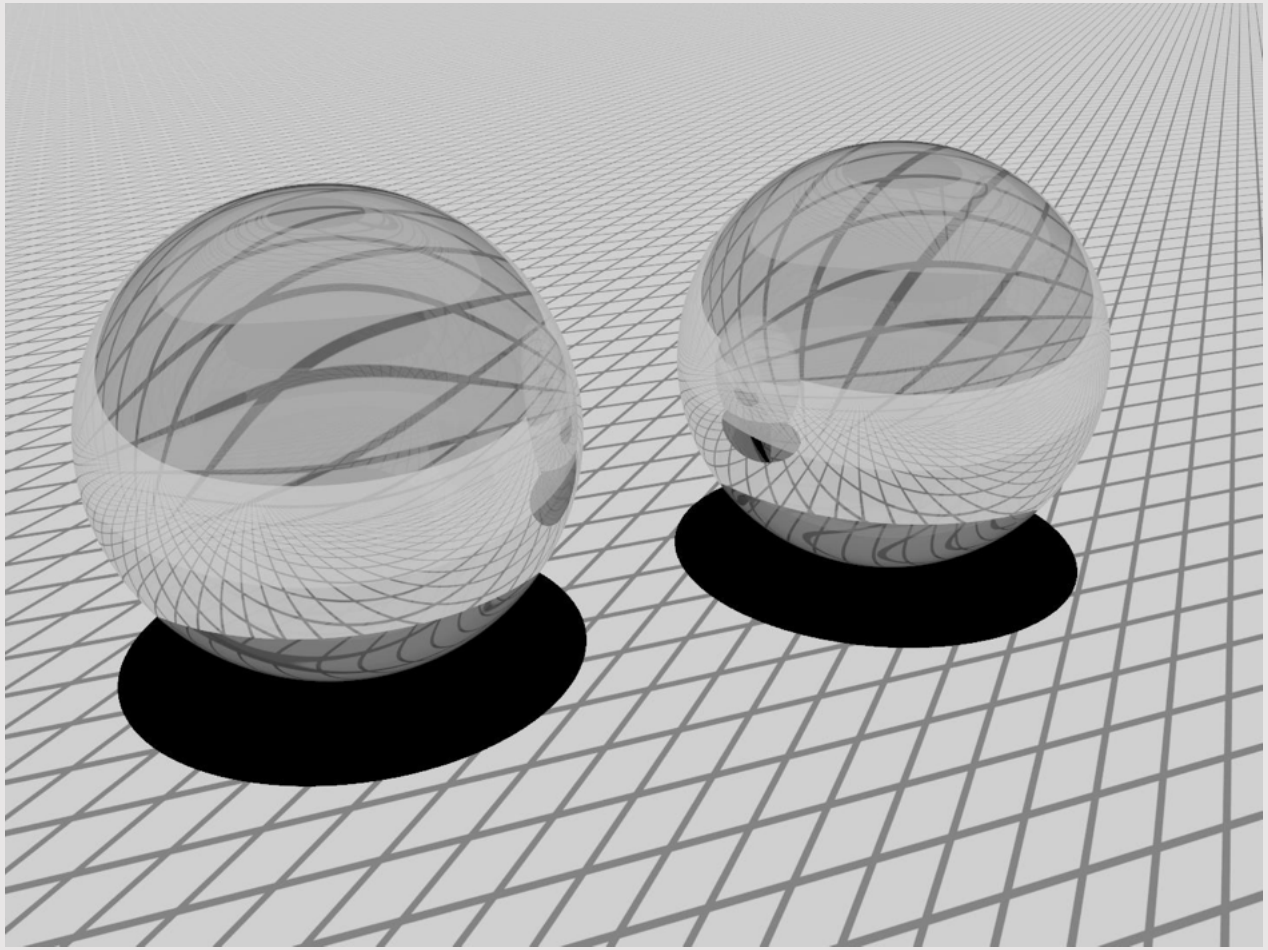


- Comprised of both reflection (Fresnel) and refraction (Snell)

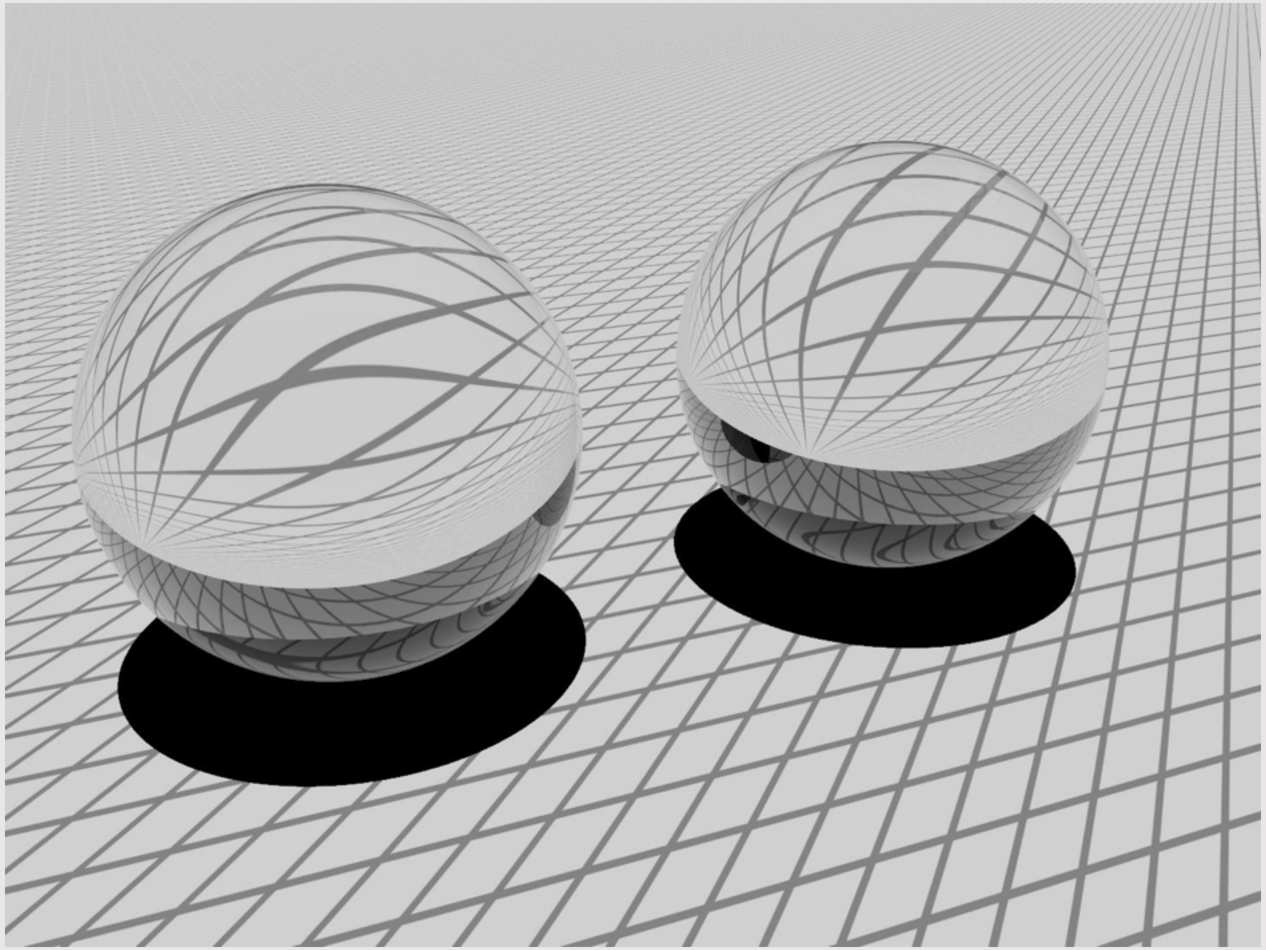
```
void glass(ni, nf, ray ri, ray *rf)
{
  bool internal_reflect = refract(ni, nf, ri, rf);
  if(internal _reflect) {
    // if refraction fails, reflect
    reflect(ri, rf);
    return;
  }

  // compute Fresnel for probability split
  float fr = fresnel(ni, nf, *rf);
  float p = rand();
  if (p < fr) {
    // fr% chance of reflecting
    reflect(ri, rf);
  }
  else {
    // 1 - fr% chance of refracting
    // already refracted, nothing left to do
  }
}
```

Glass



[without Fresnel]
constant reflection



[with Fresnel]
varying reflection

- ~~BRDFs~~
- ~~Materials~~
- Environment Lighting

Recall: Environment Light

- Sample light directly from an image
- No intensity falloff. Image distance is at infinity
- Very easy to check for visibility
  - Every point in active area

- We'll learn how to build this ~~in a future lecture~~ now

Uncharted 4 (2016) Naughty Dog

Environment As A Light

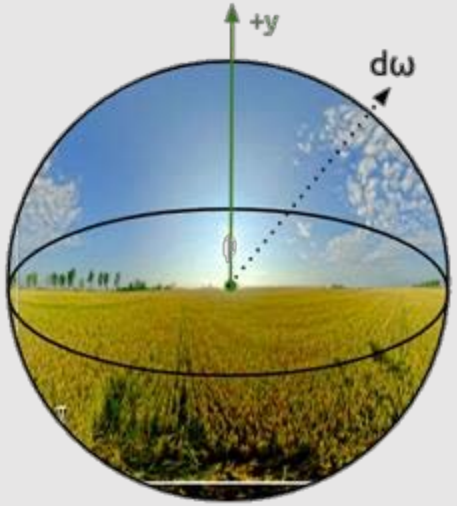
Monster's University (2013) Pixar

- Environment lighting is more than just placing a background image in the scene
  - Scene elements can use the background as a light source, sampling emitted colors the same way we would sample from regular lights
- Saves heavily on compute costs
  - No need to create complex background geometry
  - Think of it as baking diffuse information into a texture and then using that texture as a light
- **Best part:** any image can be used as an environment light!

Polar Coordinates

- Normally refer to coordinates on an image by cartesian $[x, y]$ coordinates

- Since we ”wrap” an image around a scene as a sphere, more intuitive to refer to coordinates on an image by polar $[\theta, \varphi]$ coordinates
  - Easy to convert back to cartesian

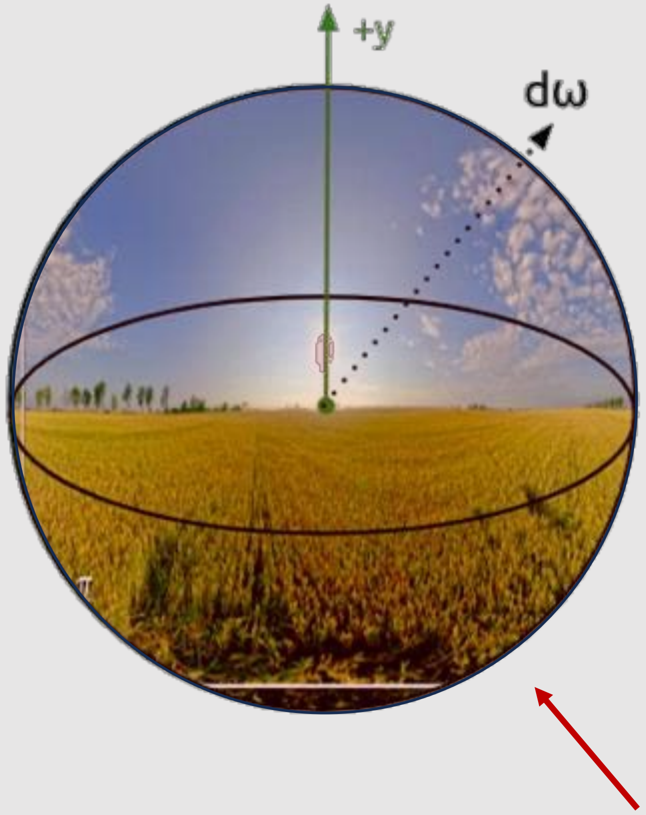


Uniform Sampling

- When our ray terminates, we randomly sample a light source
  - If the light source we pick is the environment map, where on the image do we sample?

- **Idea:** randomly sample a direction on the unit sphere, trace ray to environment map
  - Surface area of unit sphere is 4π, pdf is $\frac{1}{4\pi}$

- Scotty3D has a hemisphere sampler, how can we extend that to a sphere sampler?
  - Flip a coin, flip the sign
  - Cut the pdf in half: $\frac{1}{2\pi} * \frac{1}{2} = \frac{1}{4\pi}$



focus on all areas equally

Uniform Sampling

```
void env_lighting(ray ri)
{
  // generate ray uniformly
  ray rf = hemisphere::sampler();
  // half chance of flipping ray
  // our "clever" sphere sampler
  float p = rand();
  if (p > 0.5) {
    rf.y = -rf.y;
  }

  // double the options, half the pdf
  float pdf = hemisphere::pdf() * 0.5;

  // trace ray into environment map
  trace_ray(rf);
}
```

- Why do we need to trace the environment lighting ray? Can just sample image pixel
  - Environment lighting ray may still be occluded by scene geometry!

Uniform Sampling

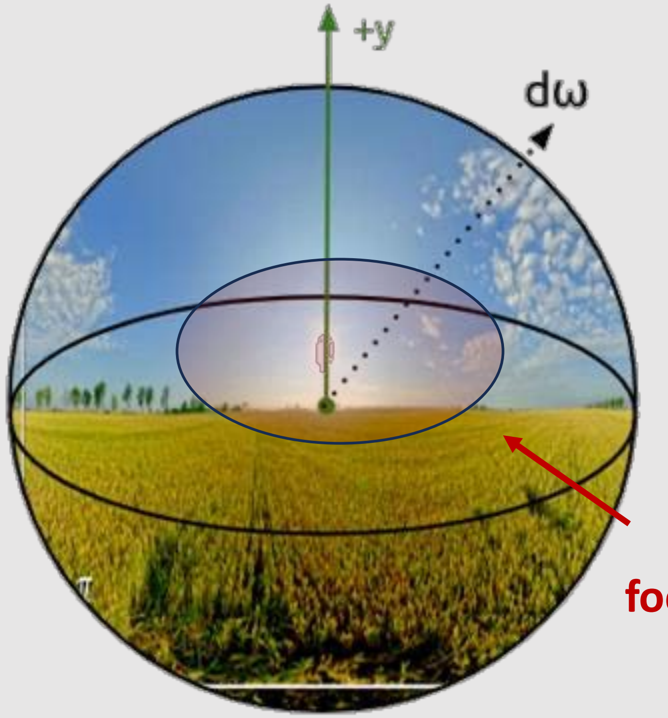
- **Issue:** uniform sampling takes a long time to converge
  - Mixing dark regions of the image with light regions
    - Gives appearance of high variance
  - Will converge with enough samples, but needs a lot of samples

- Is there another approach we can use that prioritizes areas with high info (light)?

Importance Sampling

- **Idea:** sample a direction on the unit sphere proportional to the luminance at that pixel
    - Brighter areas are more important
- **Algorithm:**
    - Assign a probability to each pixel proportional to its luminance
    - Use inversion sampling to pick a sample based on the new probability distribution
    - Create and trace a ray to pixel
    - Divide contribution by PDF of selected pixel
        - Division helps keep sampler unbiased

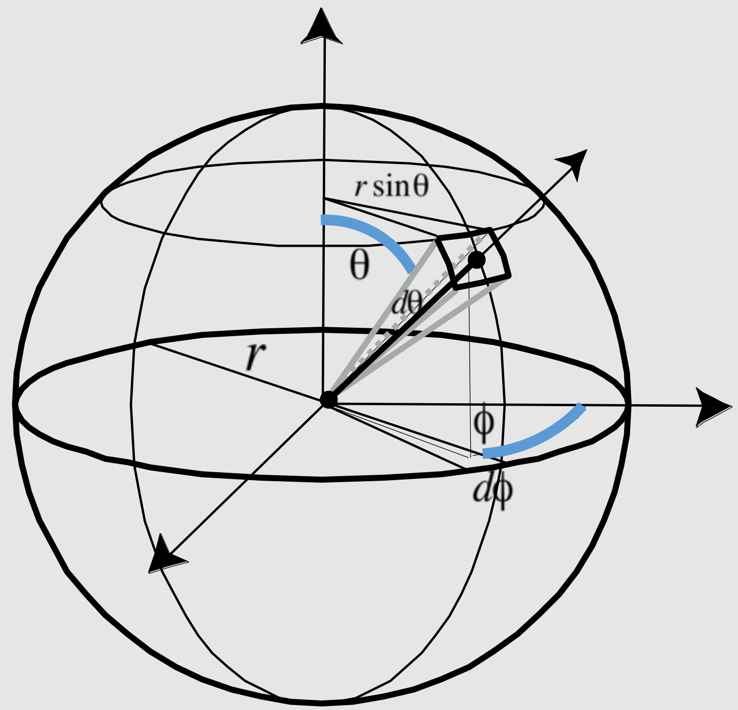
Will learn more about this shortly



focus on bright regions

Creating A PDF

- PDF of a pixel should be proportional to its flux
  - Flux = luminance $*$ solid angle
  - Luminance is L
  - Solid angle is $sin\theta d\theta d\varphi$
    - Area for each pixel is the same
    - Simplifies to $Lsin\theta$
- Already have a mapping from $[x, y]$ to $[\theta, \varphi]$
- To make sure distribution is valid, need to normalize PDFs
  - Divide every PDF by the sum of all PDFs
- How can we use that info to sample pixel with a discrete probability distribution?



Inversion Sampling

- Convert PDF to CDF:

$$cdf(i) = pdf(i) + cdf(i-1)$$

  - Image is 2D, CDF is 1D
  - Flatten image into 1D array
    - Recall images are 1D in memory

- Generate random number r between 0 and 1
  - Find index i such that:

$$cdf(i-1) < r < cdf(i)$$

    - Convert i to polar coordinates $[\theta, \varphi]$
    - Construct and trace ray from polar coordinates

Importance Sampling

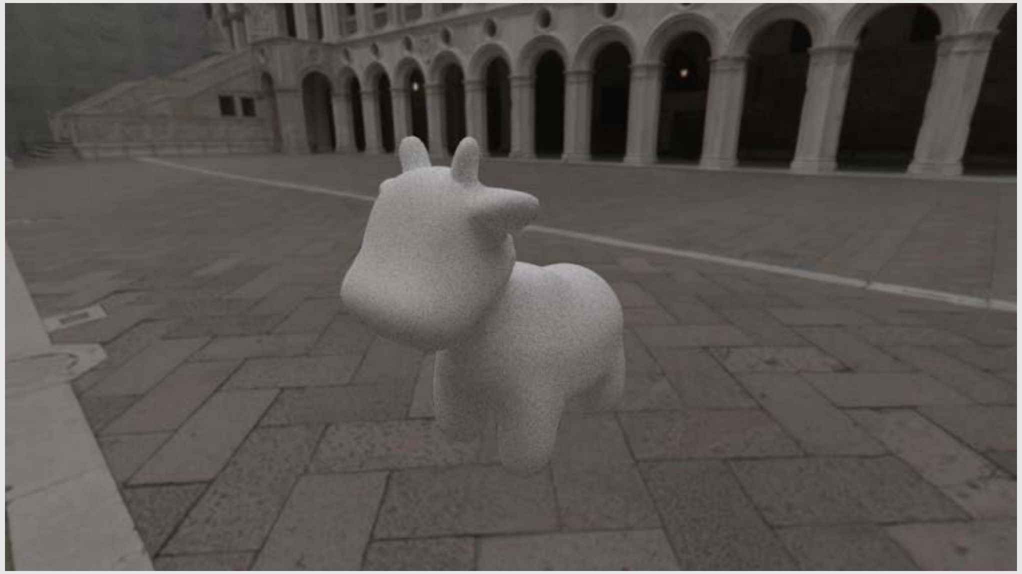
```
void env_lighting(ray ri)
{
  // generate pdf and cdf
  vector<float> pdf = Image::pdf();
  vector<float> cdf = Image::cdf(pdf);

  // inversion sampling
  float p = rand();
  auto i = upper_bound(cdf.begin(),
                       cdf.end(), p);

  // create ray from target pixel
  ray rf = ray_from_index(i);

  // trace ray into environment map
  trace_ray(rf);
}
```

- Notice how we do not even use the incoming ray
  - Both uniform and importance ignore incident directions



Uniform vs. Importance

32 Uniform samples

32 Importance samples

Importance sampling is better able to capture directional light

Uniform vs. Importance

32 Uniform samples

32 Importance samples

Importance sampling is better able to capture sparse lights

Variance Reduction

- Monte-Carlo Sampling
- Biased vs Unbiased Estimators
- Physically-Based Rendering Methods

What Makes A Render Expensive

- **Number of Rays**
  - How many rays traced into the scene
    - Measured as samples (rays) per pixel [spp]
- **Number of Ray Bounces**
  - How ray bounces before termination
    - Measured as ray bounce/depth
- Choosing the right number is difficult
  - Similar to **sample theory**

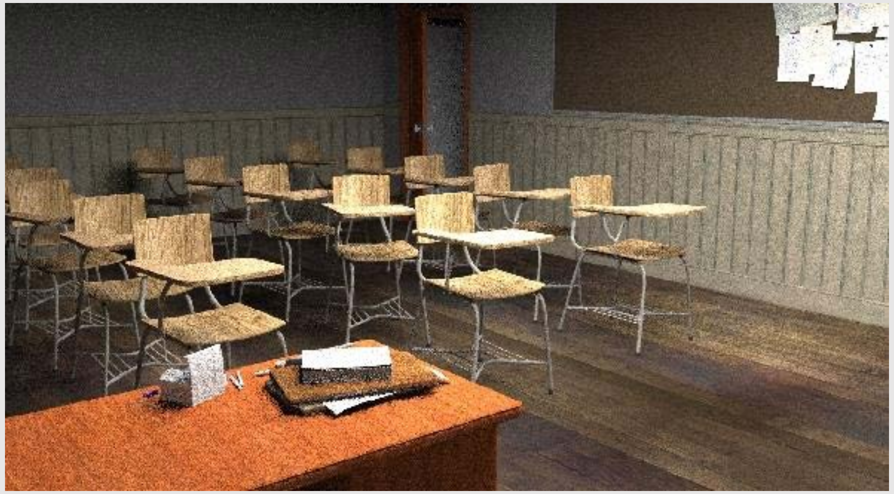
Star Wars VII: The Force Awakens (2015) Lucasfilm

Number Of Ray Samples

- **Number of Rays**
  - How many rays traced into the scene
    - Measured as samples (rays) per pixel [spp]

- Increasing number of rays increases image quality
  - Anti-aliasing
  - Reduces black spots from terminating emission occlusion

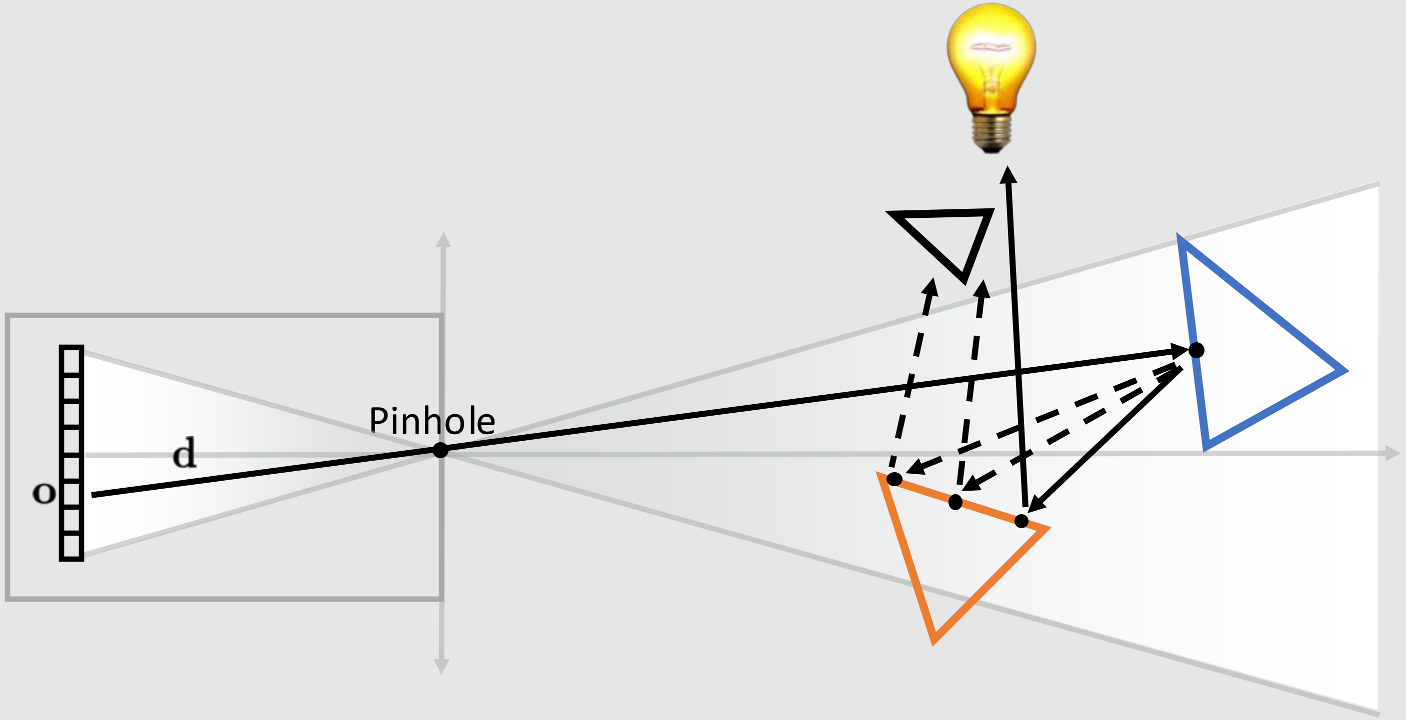
[1 spp]



[16 spp]

Number Of Ray Samples

- Having more rays similar to taking more samples in rasterization
  - Samples taken in a larger sample buffer and resolved into smaller output buffer
- More likely to find terminating ray that reaches light source/not be occluded



Number Of Ray Bounces

- **Number of Ray Bounces**
  - How ray bounces before termination
    - Measured as ray bounce/depth

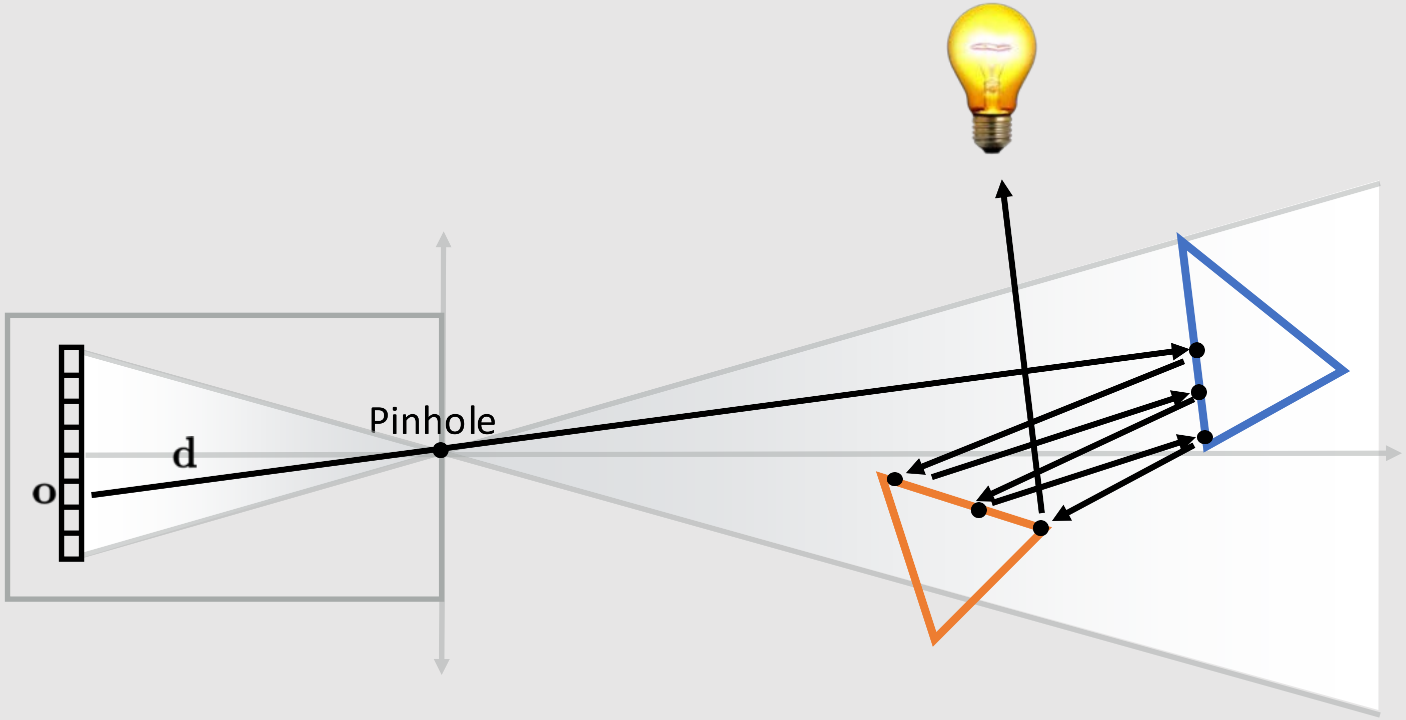
- Increasing ray bounces increases image quality
  - Better color blending around images
  - More details reflected in specular images

[2 depth]

[8 depth]

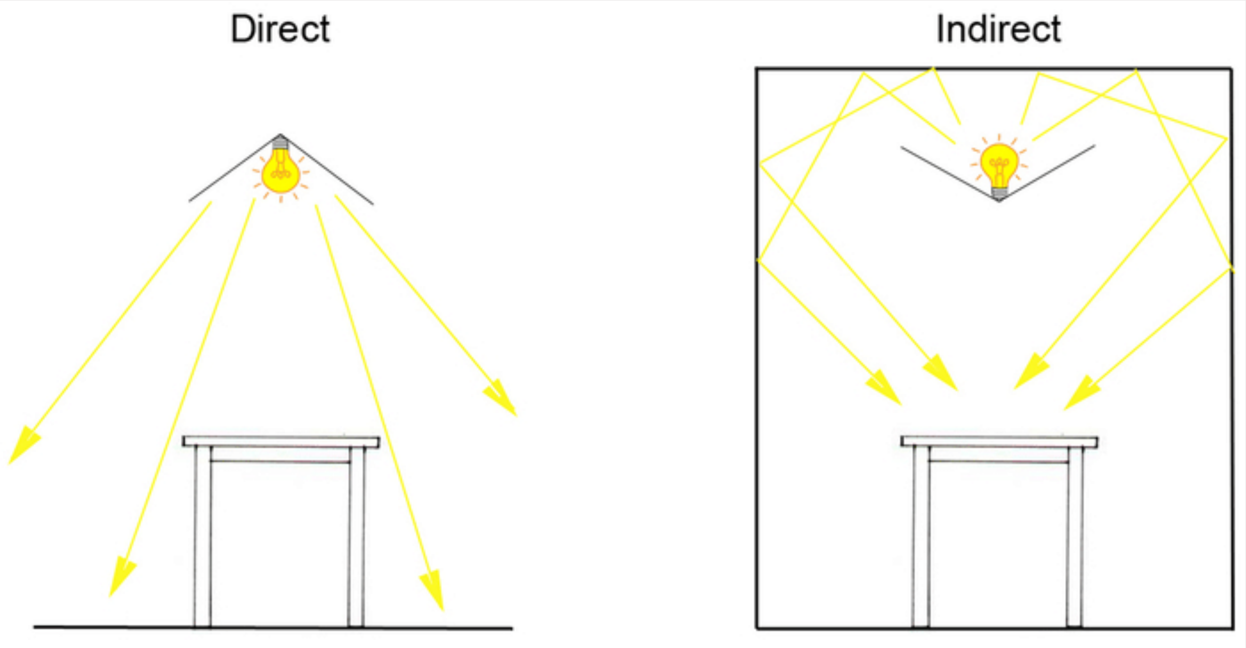
Number Of Ray Bounces

- Having more ray bouncing allows for better color blending
  - Final ray will be a larger mix of blue/orange than the original yellow
- Can render more interesting reflective and refractive paths with more bounces

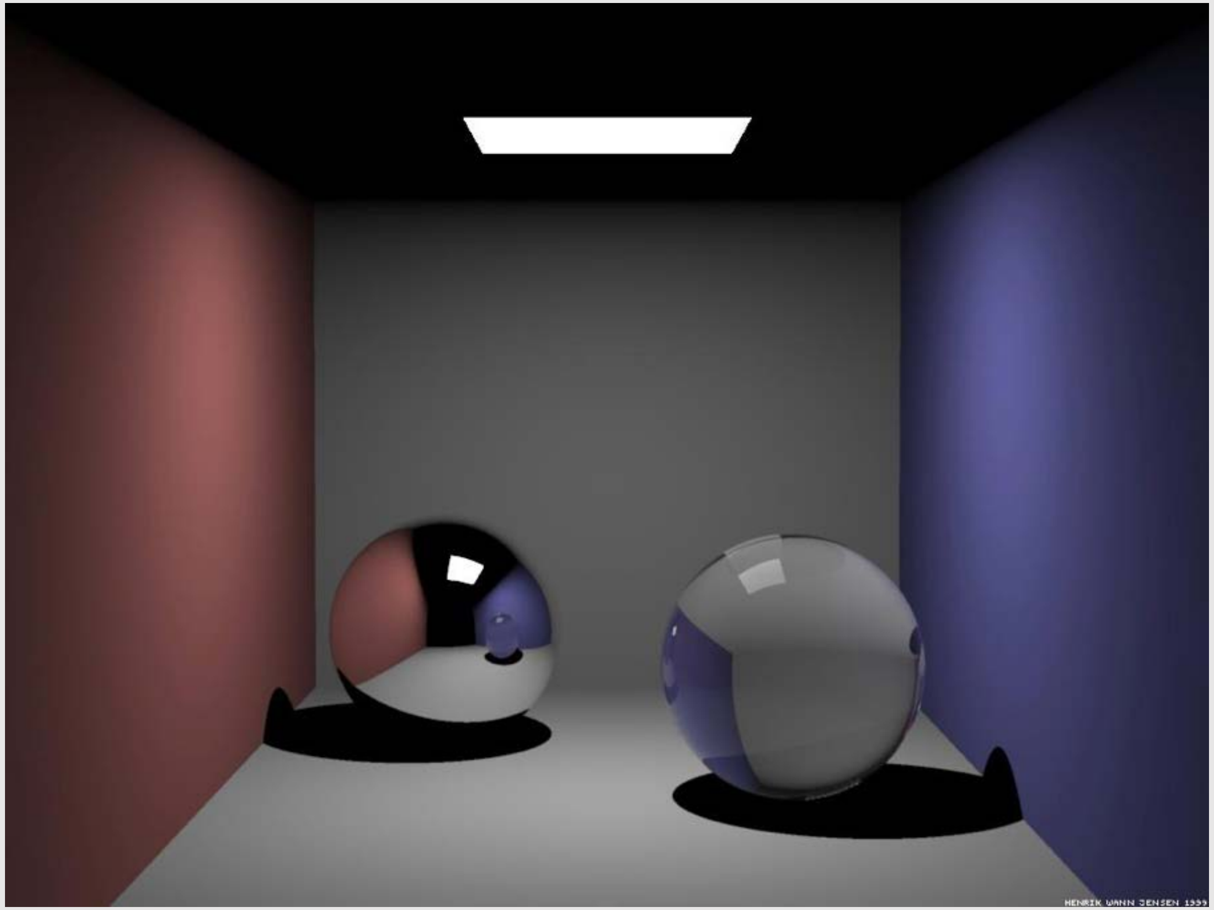


Direct VS Indirect Illumination

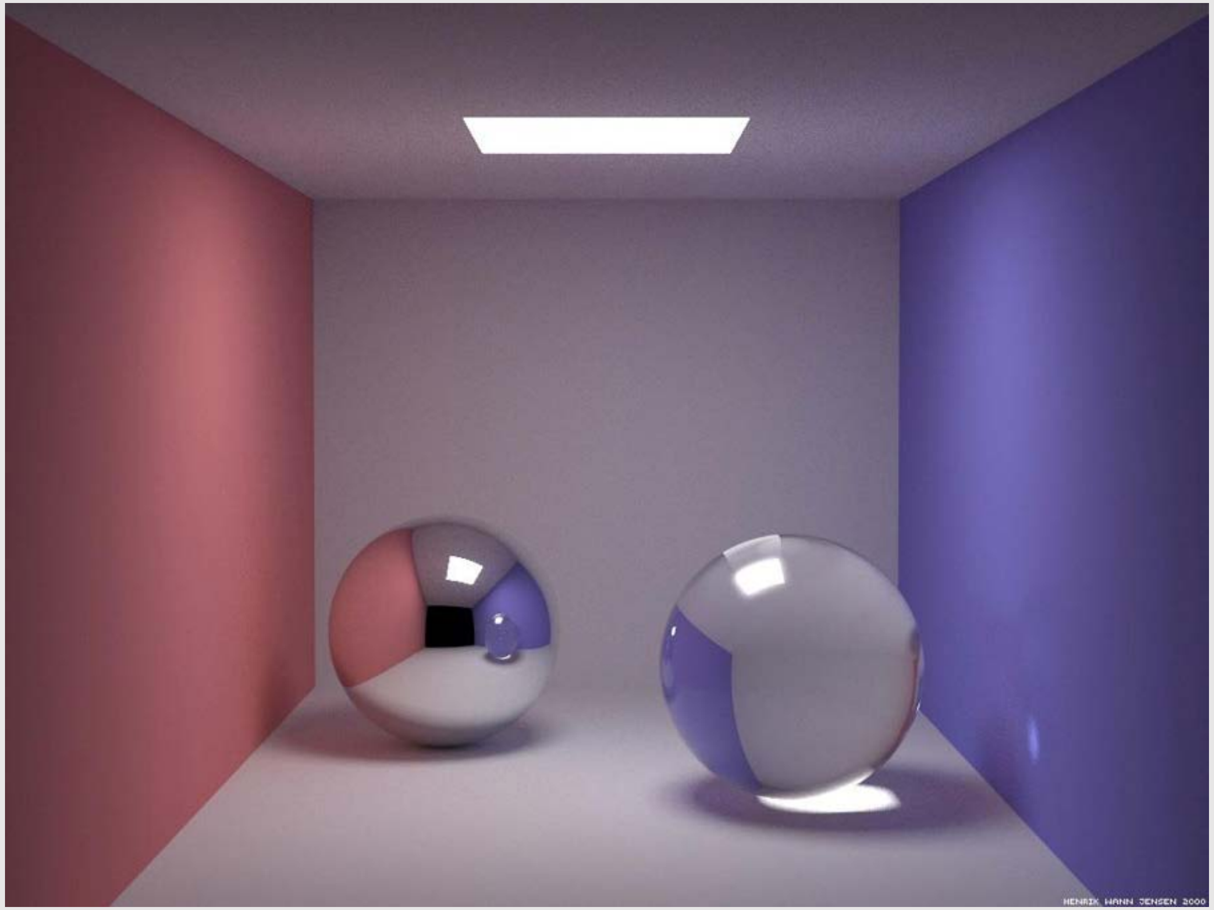
- **Direct Illumination:** Direct path from emitter to point
- **Indirect Illumination:** Multi-bounce path from emitter to point
- **Bounce** describes how many piecewise linear rays we can stich together to form a path
  - Direct is 1-bounce
  - Indirect is N-bounce
    - Some authors say Direct is 0-bounce [index at 0]



Direct VS Indirect Illumination



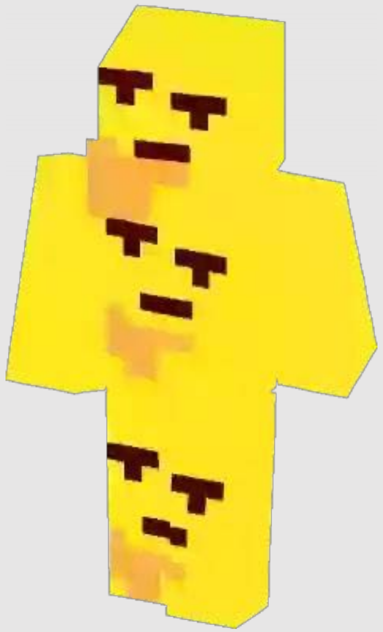
[Direct + Reflection + Refraction]**



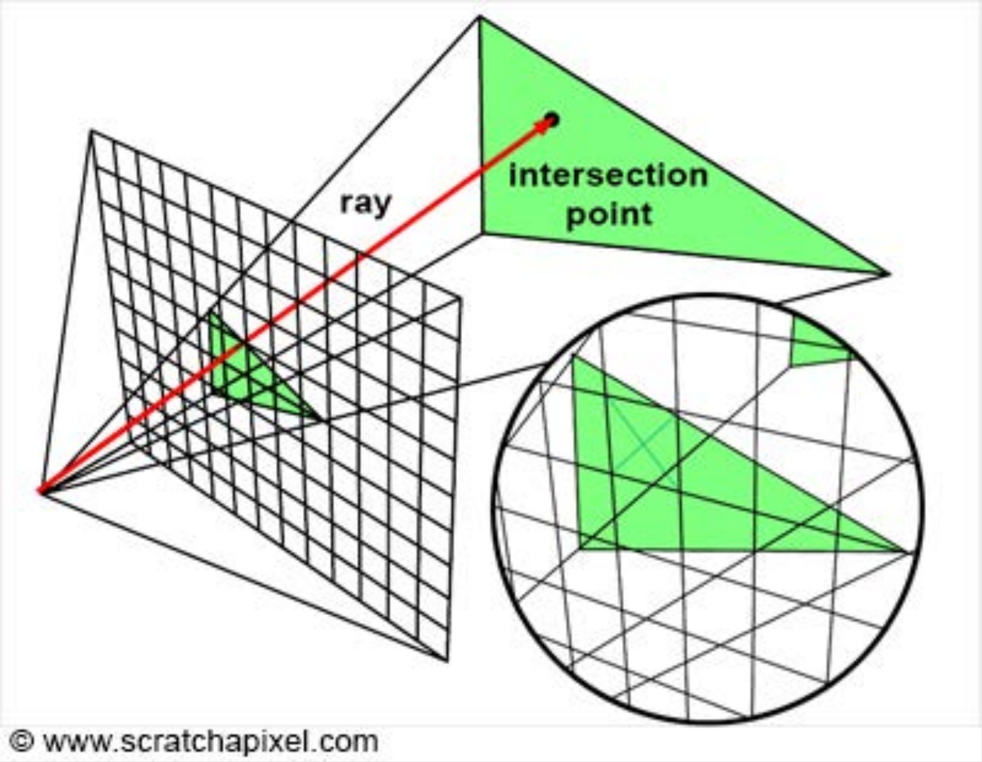
[Global]

**Normally can’t do reflection & refraction in direct illumination

Wait a minute…
direct illumination looks like rasterization



Direct VS Rasterization



- **Food for thought:** rasterization traces rays from a point in the output buffer to a shape in the scene
  - Even in rasterization, shapes have depth
  - We only care about the closest object we see (transparency disabled)
- Both rasterization and direct illumination only ever trace **one ray**!

Direct VS Indirect Illumination

Minecraft (2020) Microsoft

- Direct Illumination gives you **efficiency**
  - Easy to render
  - Straightforward complexity
  - Comparable to rasterization in difficulty
  - Amendable to ray packeting
  - Easy real-time performance
- Indirect (Global) Illumination gives you **quality**
  - Some materials require multi-bounces
    - Ex: refraction
  - Ambient occlusion
  - Higher contrast
  - Samples converge to true values
- **More bounces = ↓ efficiency, ↑ quality**

So how do we take multiple samples?

Continuous Vs. Discrete

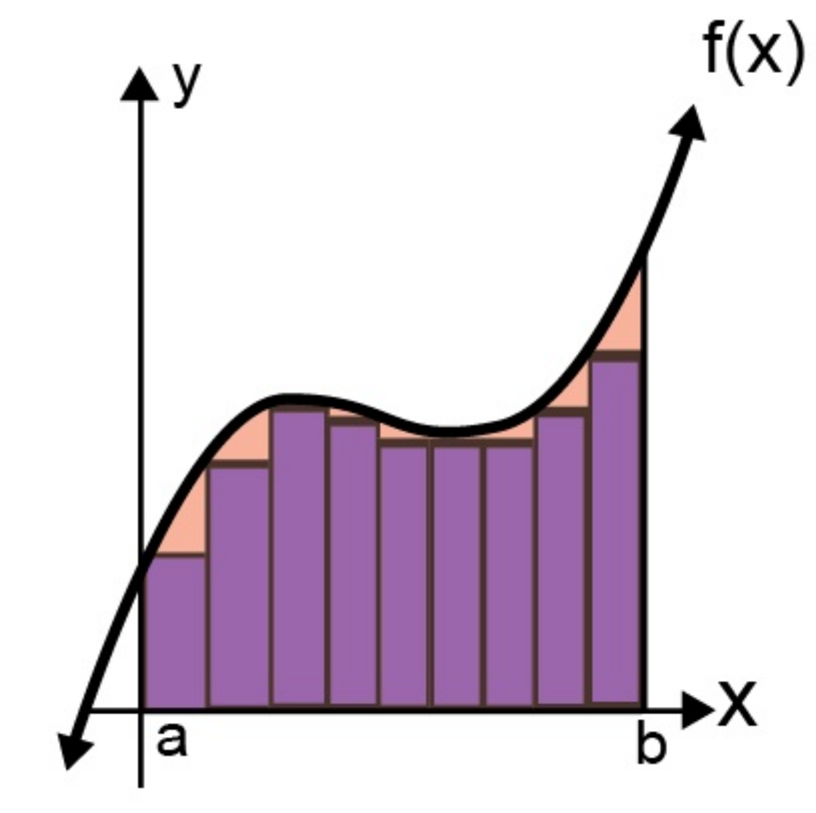
- Our eyes see a **continuous** signal of energy
- Our digital cameras see a **discrete** signal of energy
  - Computers process discrete values
- Let the following integral be the true continuous signal of the scene:

$$\int_{\mathcal{H}^2} f_r(\mathbf{p}, \omega_i \rightarrow \omega_o) L_i(\mathbf{p}, \omega_i) \cos\theta \, d\omega_i$$

- Approximate the integral by taking multiple samples of our discrete scene function:

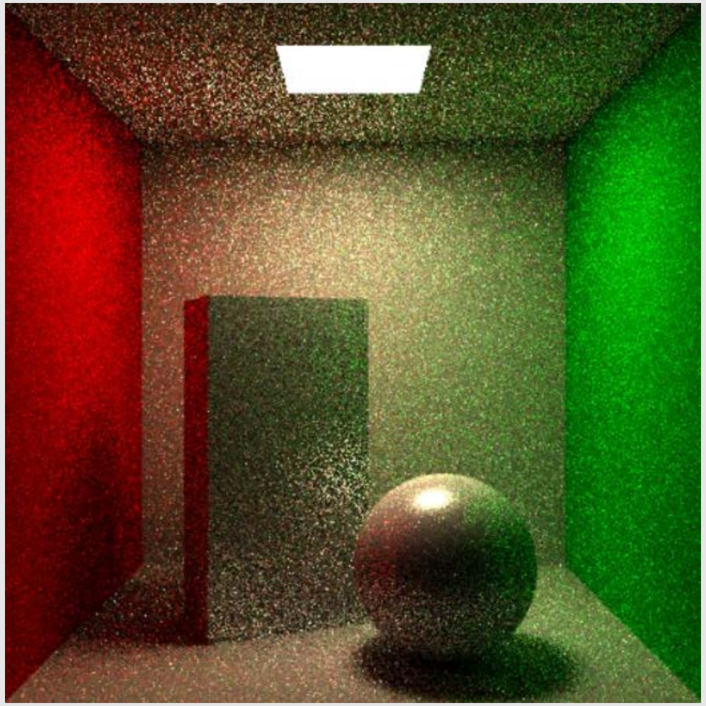
$$\frac{2\pi}{N} \sum_{i=1}^{N} f_r(\mathbf{p}, \omega_i \rightarrow \omega_o) L_i(\mathbf{p}, \omega_i) \cos\theta$$

- We compute the average sample by dividing by N
- We multiply by the size of the domain, which is 2pi

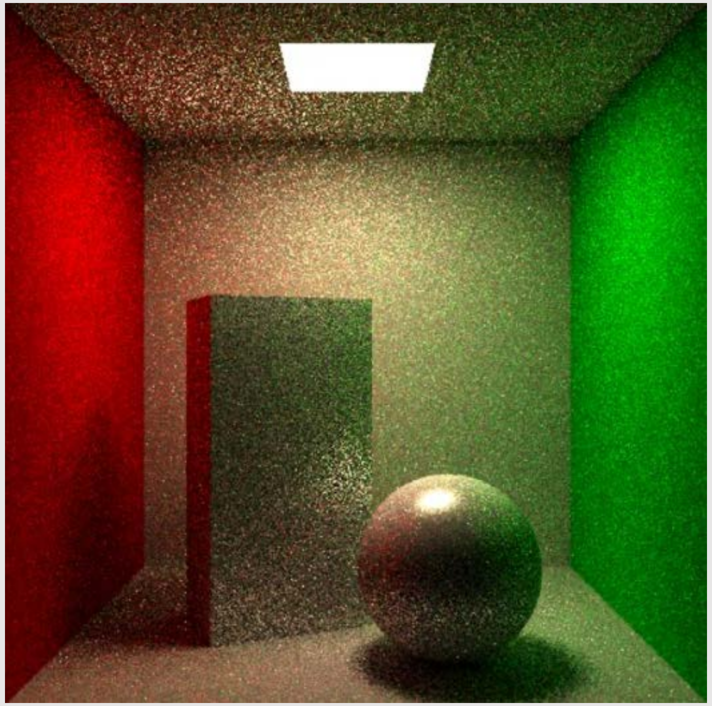


Sampling Rays

- **Issue:** Responsible for picking rays since we are no longer integrating over every possible ray direction in hemisphere
  - Some rays will be better than others
  - Again, similar to **sample theory**
- **Idea:** pick rays from a PDF
  - **Uniform PDF:** ray sampled in uniformly random direction in hemisphere
  - **Cos-weighted PDF:** rays are more likely to be sampled in the direction of the normal



[uniform sampling]

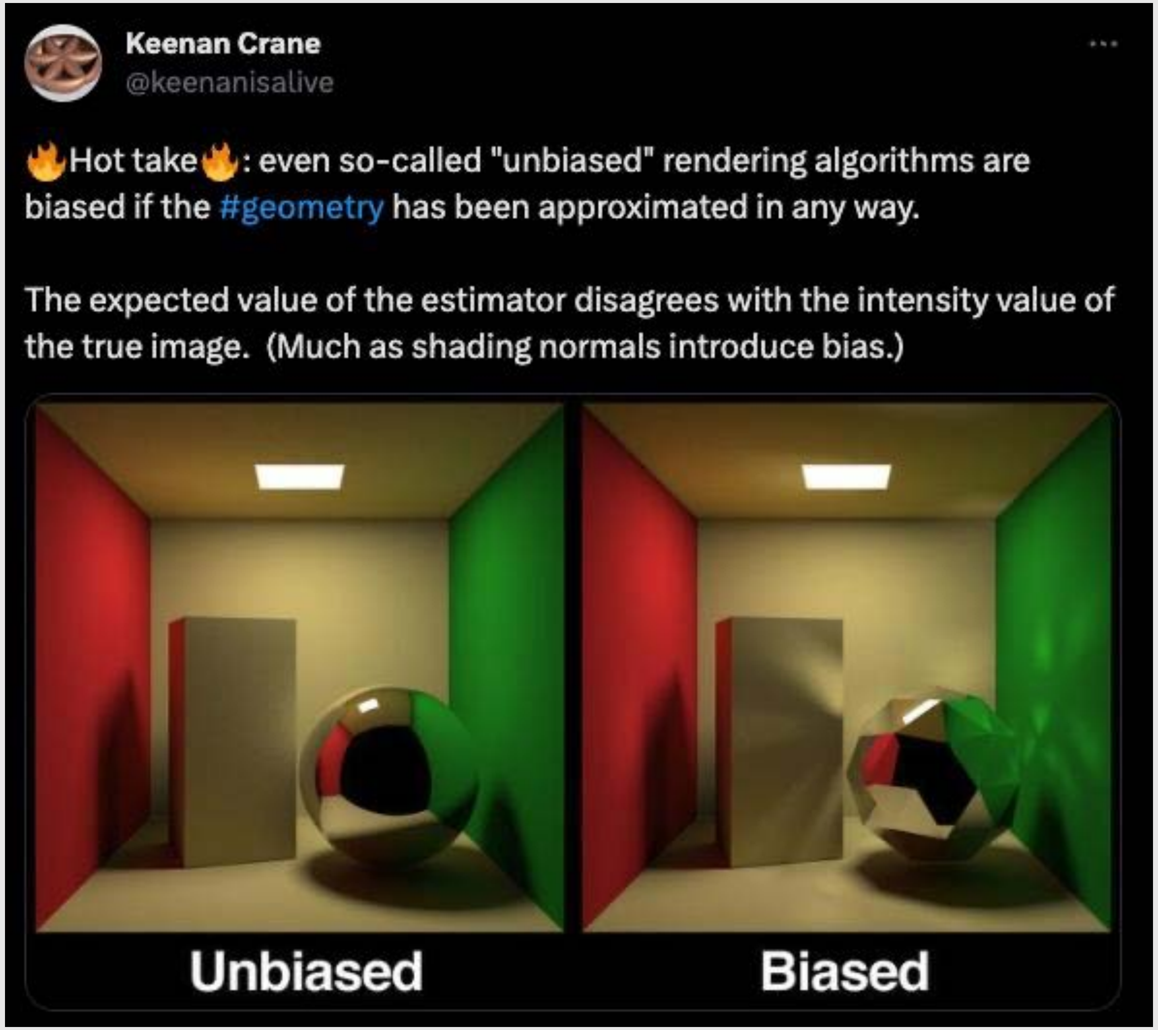


[cosine sampling]

But wait,
Isn’t taking non-uniform samples biased?

- ~~Monte-Carlo Sampling~~
- Biased vs Unbiased Estimators
- Physically-Based Rendering Methods

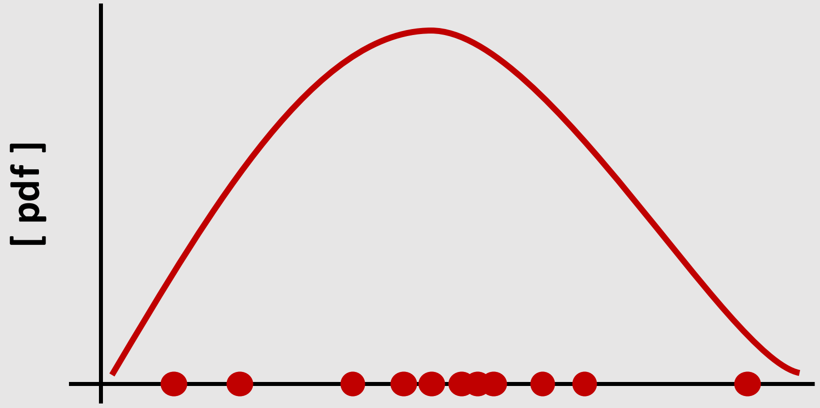
Biased vs. Unbiased Renderer

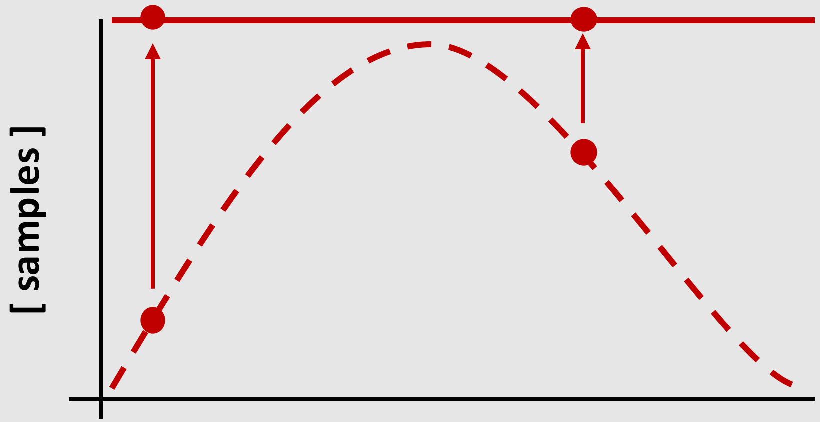


- An **unbiased** renderer tries to mimic the uniformity of real life
  - Does not introduce systematic bias
  - Taking more samples will reduce error
  - Approaches ground truth with infinite sampling
- A **biased** renderer will take shortcuts to make renders look better
  - Taking more samples may introduce even more signal than the original image
  - Usually faster rendering/less samples
  - Can seek out more difficult paths
- When comparing render methods, makes more sense to compare **unbiased methods**

Biased vs. Unbiased Example

- In a biased estimator, draw samples proportional to the PDF
  - More samples drawn where PDF is high
  - Under-sampling where PDF is low
- The good news is that it is easy to turn this biased estimator into an unbiased one!
- To make this biased estimator unbiased, simply divide by the PDF of the sample
  - Samples with a **high PDF** are divided by a high value, **not increasing its contribution much**
  - Samples with a **low PDF** are divided by a low value, **increasing its contribution a lot**
    - Produces an unbiased sample set





The Monte Carlo Estimator

- Named **Monte Carlo** after the famous gambling location in Monaco
  - Shares the same random characteristic as a roulette game

- **Algorithm:**
  - Sample a direction based on the PDF $p(w_j)$
  - Compute the incident radiance of the direction
  - Divide by the PDF $p(w_j)$ to make unbiased
  - Repeat, averaging the samples together

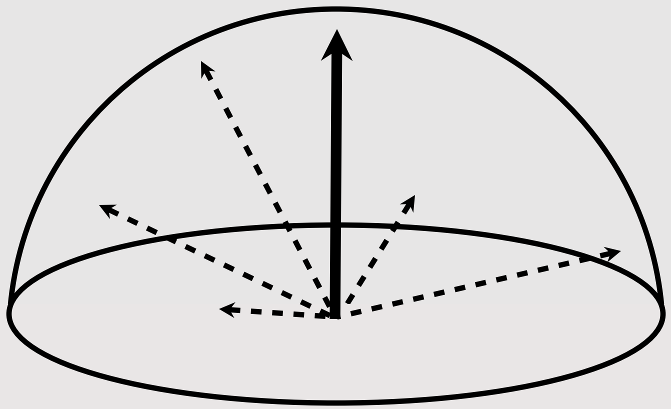
$$\frac{1}{N}\sum_{j=1}^{N}\frac{f_r(\mathrm{p},\omega_j \to \omega_r)\, L_i(\mathrm{p},\omega_j)\, \cos\theta_j}{p(\omega_j)}$$

Note! We no longer multiply our average by the size of the domain. Dividing by the PDF takes care of that for us. Why? Because the PDF must integrate to 1 over the entire domain.

Monte Carlo Uniform Sampling

- Let $f(w)$ be the incident radiance [ignoring BRDF]
- Let $p(w)$ be the PDF of the sampled direction w

$$f(\omega) = L_i(\omega)\cos\theta \qquad\qquad p(\omega) = \frac{1}{2\pi}$$



- Taking random samples leads to:

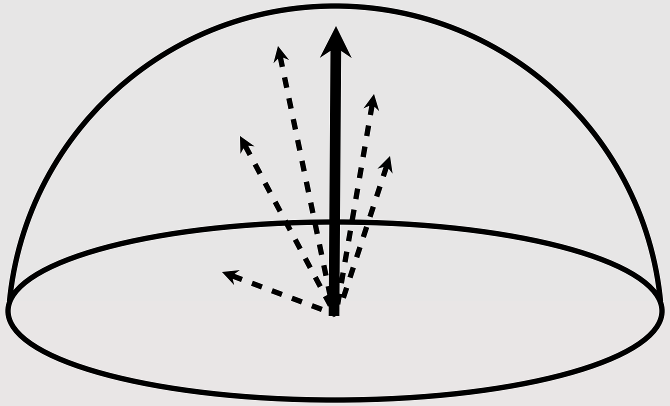
$$\int_\Omega f(\omega)\,\mathrm{d}\omega \approx \frac{1}{N}\sum_i^N \frac{f(\omega)}{p(\omega)} = \frac{1}{N}\sum_i^N \frac{L_i(\omega)\,\cos\theta}{1/2\pi} = \boxed{\frac{2\pi}{N}\sum_i^N L_i(\omega)\cos\theta}$$

- PDF is constant in all directions, just multiply by scalar 2π

Monte Carlo Cosine Sampling

- Let $f(w)$ be the incident radiance [ignoring BRDF]
- Let $p(w)$ be the PDF of the sampled direction w

$$f(\omega) = L_i(\omega)\cos\theta \qquad\qquad p(\omega) = \frac{\cos\theta}{\pi}$$



- Taking random samples leads to:

$$\int_\Omega f(\omega)\,\mathrm{d}\omega \approx \frac{1}{N}\sum_i^N \frac{f(\omega)}{p(\omega)} = \frac{1}{N}\sum_i^N \frac{L_i(\omega)\,\cos\theta}{\cos\theta/\pi} = \boxed{\frac{\pi}{N}\sum_i^N L_i(\omega)}$$

- PDF removes the cosine term, we now get more radiance per sample!

How do we get a good sense of “how well” we did?

Variance

- **Variance** is how far we are from the average, on average

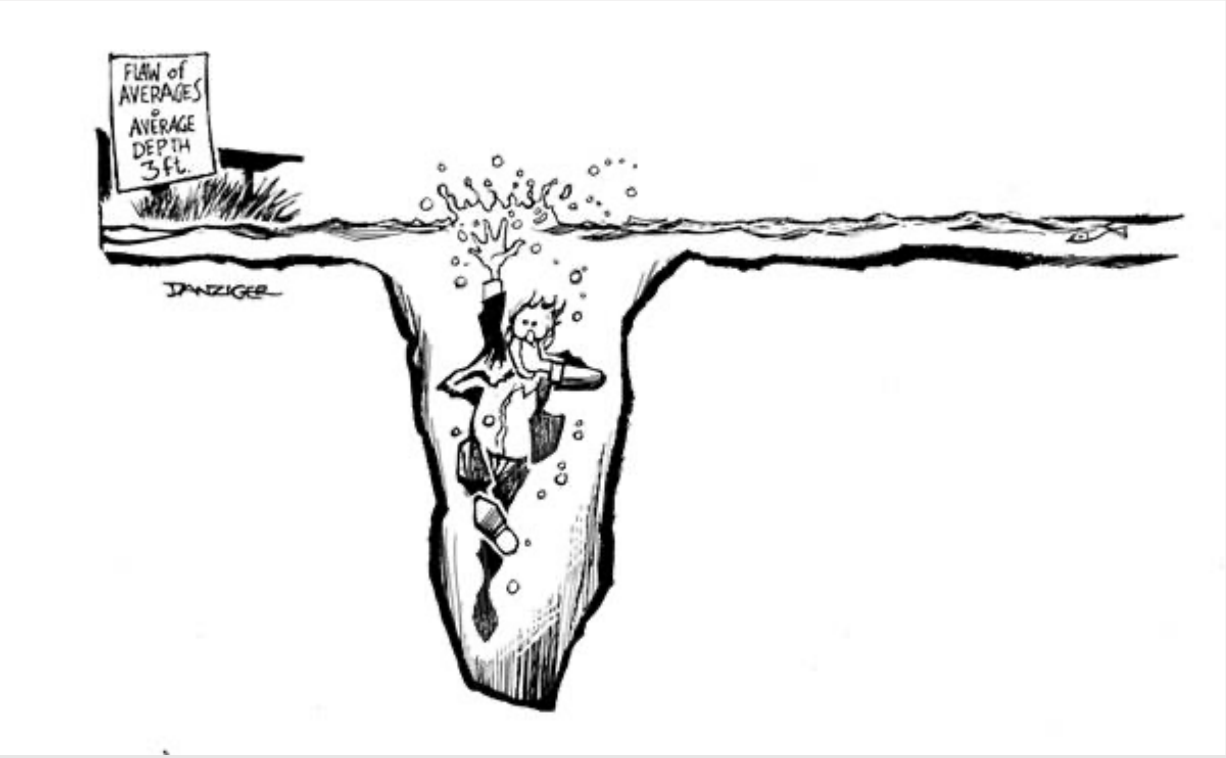
$$\mathrm{Var}(X) := E[(X - E[X])^2]$$

- Discrete:

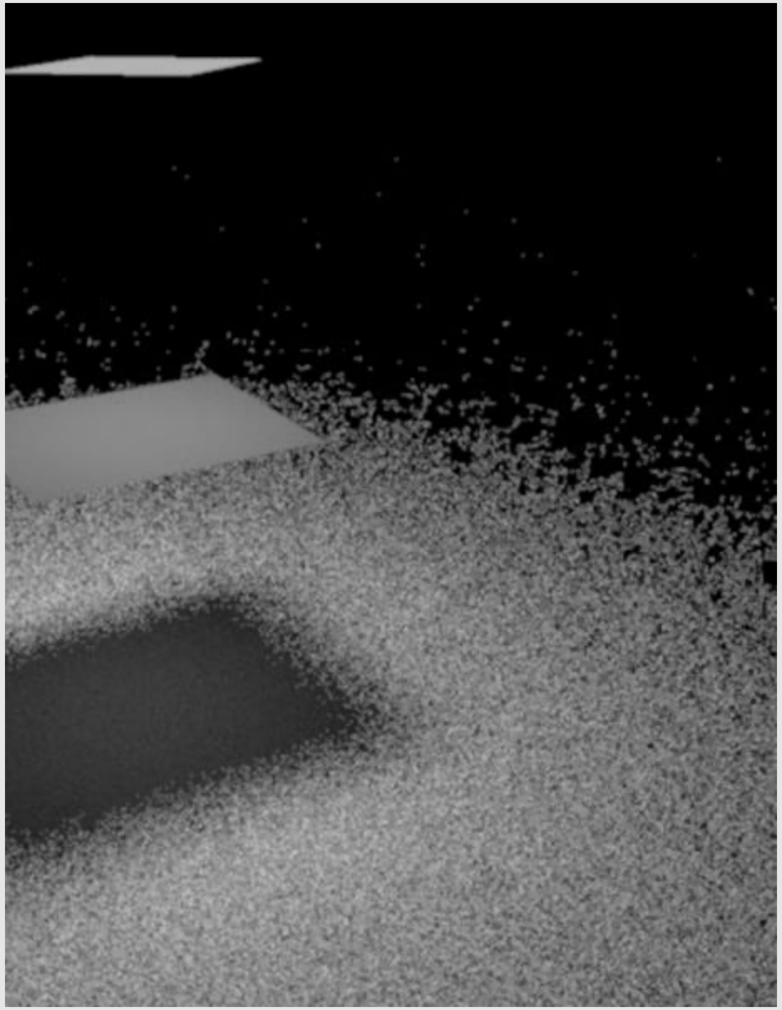
$$\sum_{i=1}^{n} p_i (x_i - \sum_j p_j x_j)^2$$

- Continuous:

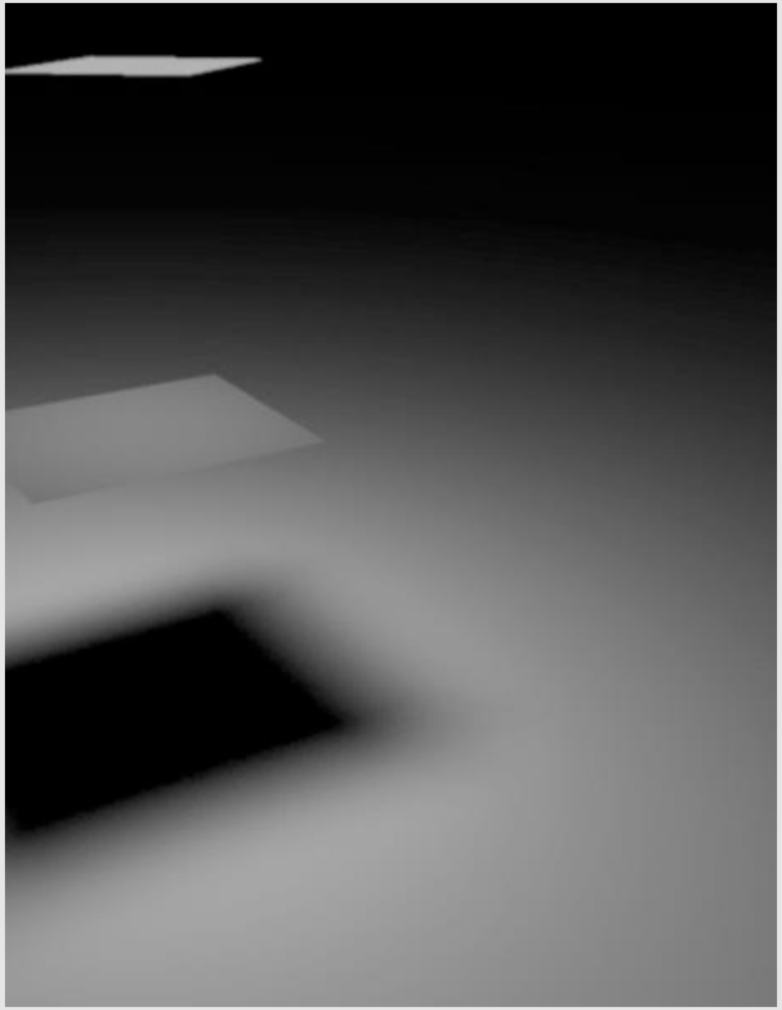
$$\int_\Omega p(x)(x - \int_\Omega y p(y)\ dy)^2\ dx$$



Variance In Rendering



[high variance]



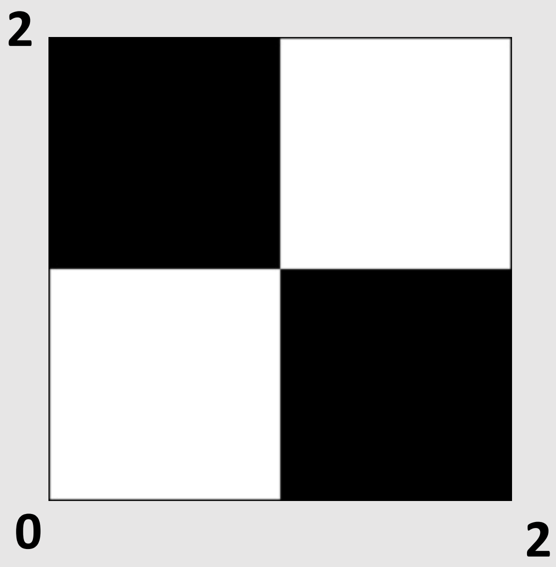
[low variance]

Variance Reduction Example

$$\Omega := [0,2] \times [0,2]$$

$$f(x,y) := \begin{cases} 1 & \lfloor x \rfloor + \lfloor y \rfloor \text{ is even,} \\ 0 & \text{otherwise} \end{cases}$$

$$I := \int_\Omega f(x,y)\ dxdy$$



- What's the expected value of the integrand?
  - Just by inspection: 1/2 (half black, half white)
- What's the variance?
  - $(1/2)(0-1/2)^2 + (1/2)(1-1/2)^2 = (1/2)(1/4) + (1/2)(1/4) = 1/4$
- How do we reduce the variance?

Trick question!
You can’t reduce the variance of an integrand.
Can only reduce variance of an estimator.

Bias & Consistency

- An estimator is **consistent** if it converges to the correct answer:

$$\lim_{n \to \infty} P(|I - \hat{I}_n| > 0) = 0$$

near infinite # of samples

- An estimator is **unbiased** if it is correct on average:

$$E[I - \hat{I}_n] = 0$$

even if just 1 sample

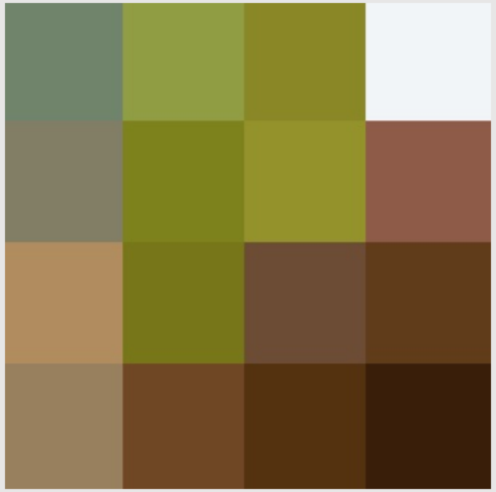
- **consistent != unbiased**

[biased] **[unbiased]**

Consistent Or Unbiased?

- Estimator for the integral over an image:
  - Take n = m x m samples at fixed grid points
  - Sum the contributions of each box
  - Let m go to ∞

- Is the estimator:
  - Consistent?
  - Unbiased?



[m = 4]

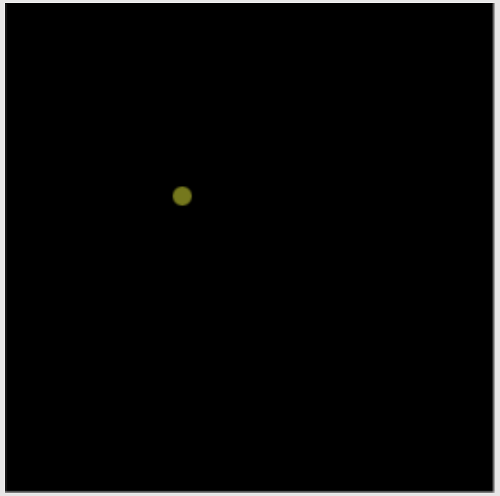
[m = 16]

[m = 64]

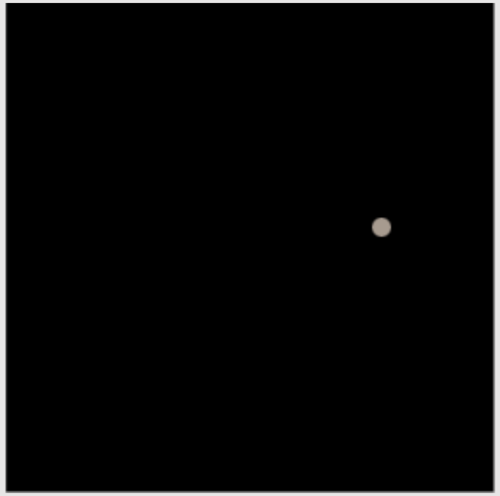
[m = ∞]

Consistent Or Unbiased?

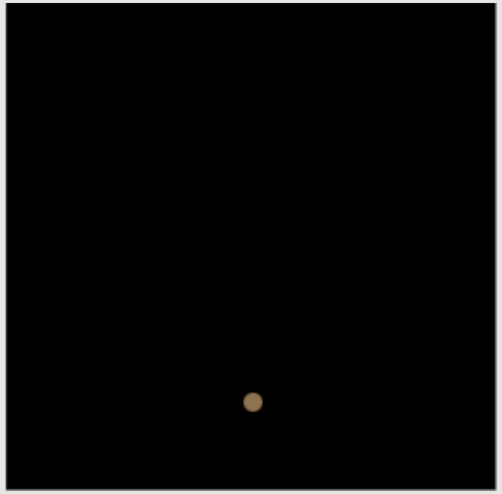
- Estimator for the integral over an image:
  - Take only a single random sample of the image (n=1)
  - Multiply it by the image area
  - Use this value as my estimate
- Is the estimator:
  - Consistent?
  - Unbiased?
- What if I let my estimator go to ∞



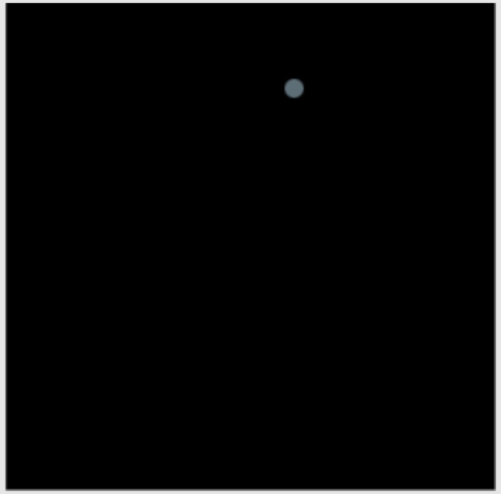
[m = 1]



[m = 1]



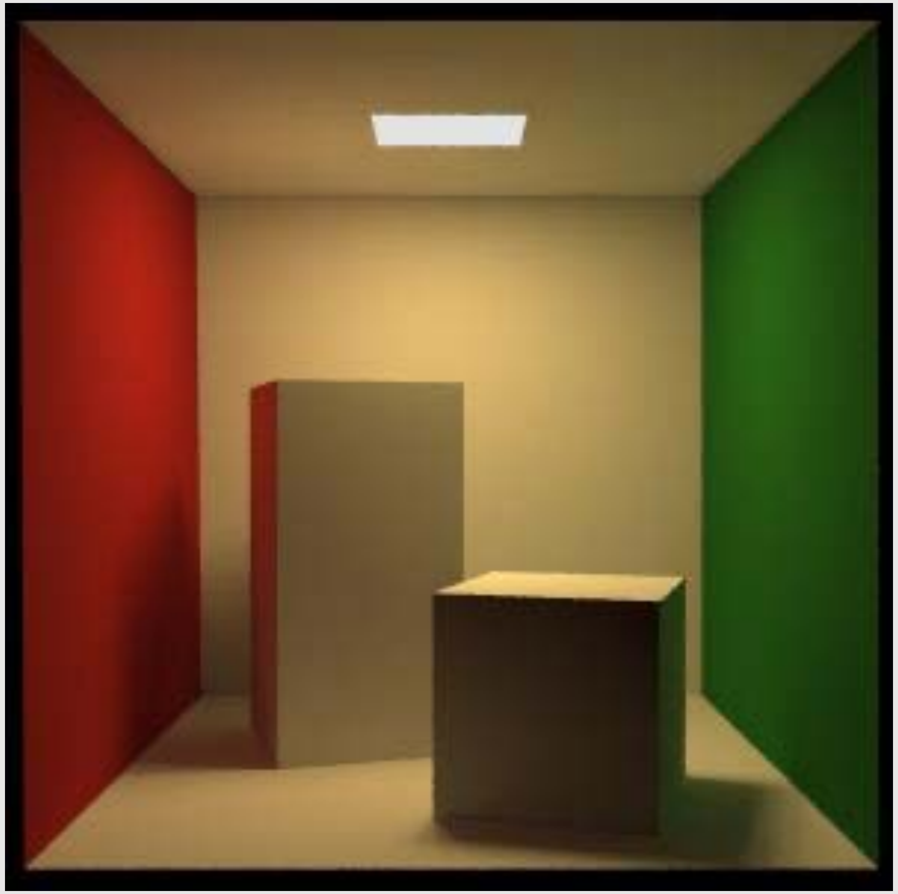
[m = 1]



[m = 1]

What is my true image?

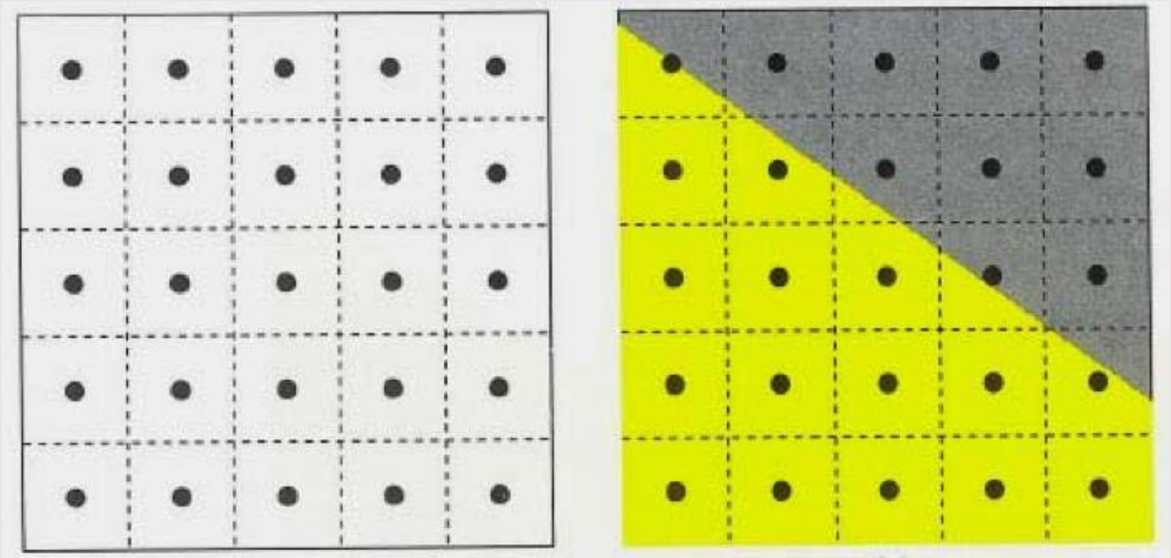
The Cornell Box

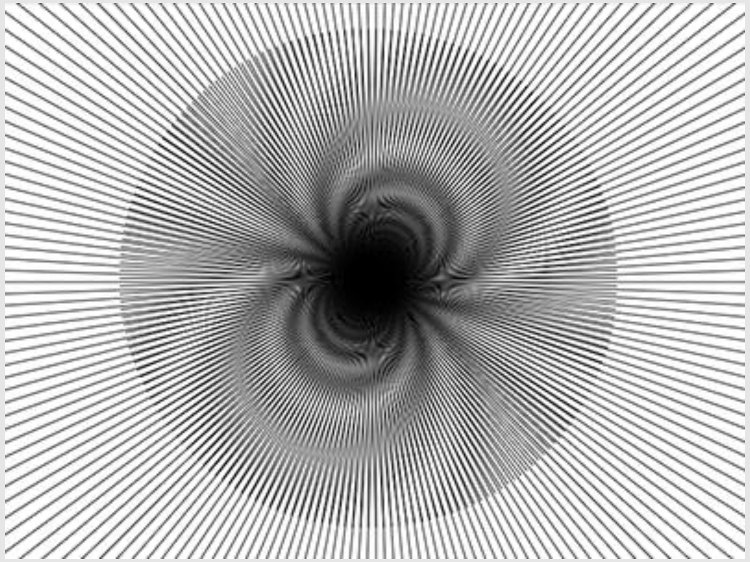


How do we take good samples?

Uniform Sampling

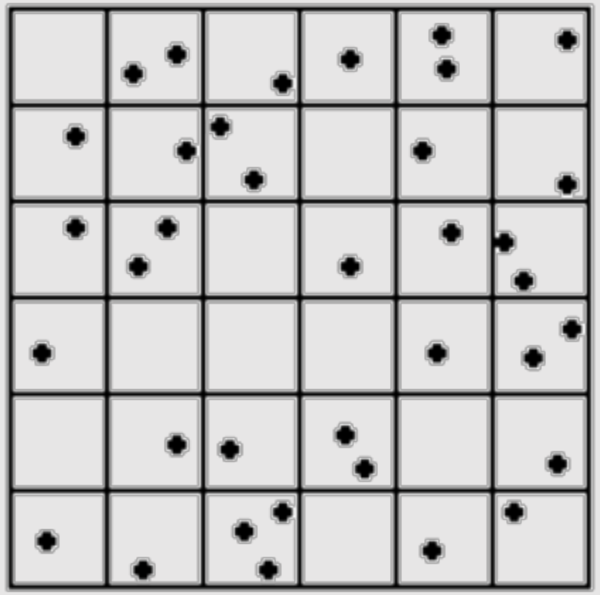
- Place samples uniformly apart in grid fashion
  - [+] Easy to compute
  - [-] We still have jagged edges, just at higher resolutions
  - [-] More samples needed
  - [-] Does not fix moiré pattern

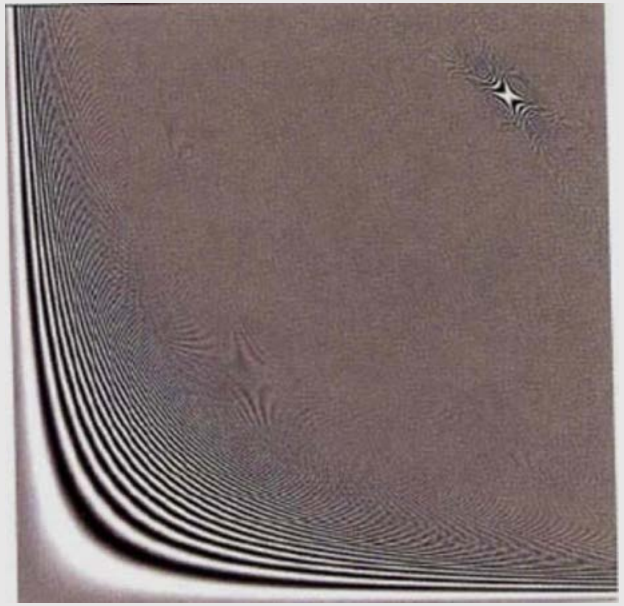


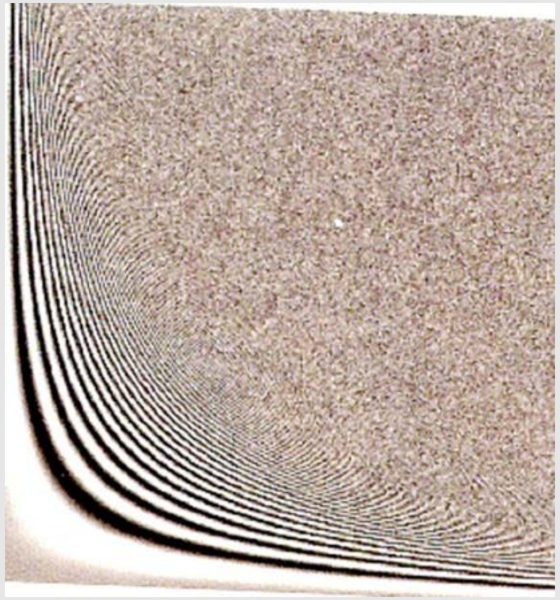


Random Sampling

- Place samples randomly
  - [+] Easy to compute
  - [-] Introduces noise, noticeable at low resolutions
  - [-] Lack of distance between samples

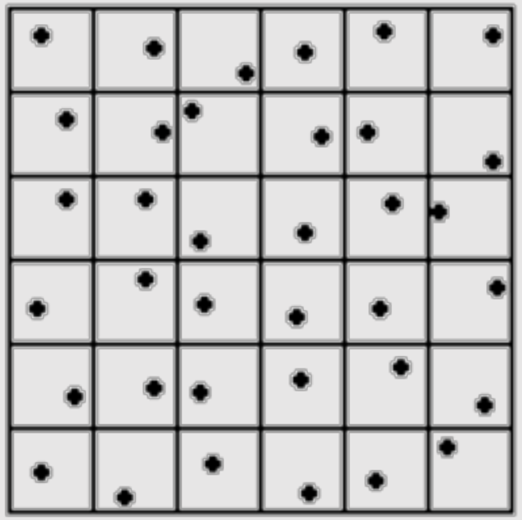


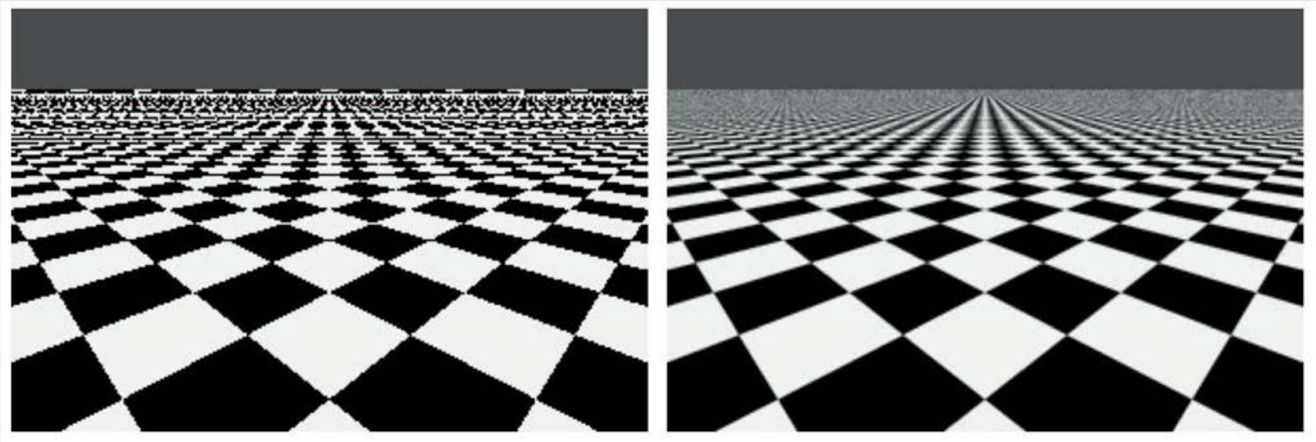




Jittered Sampling

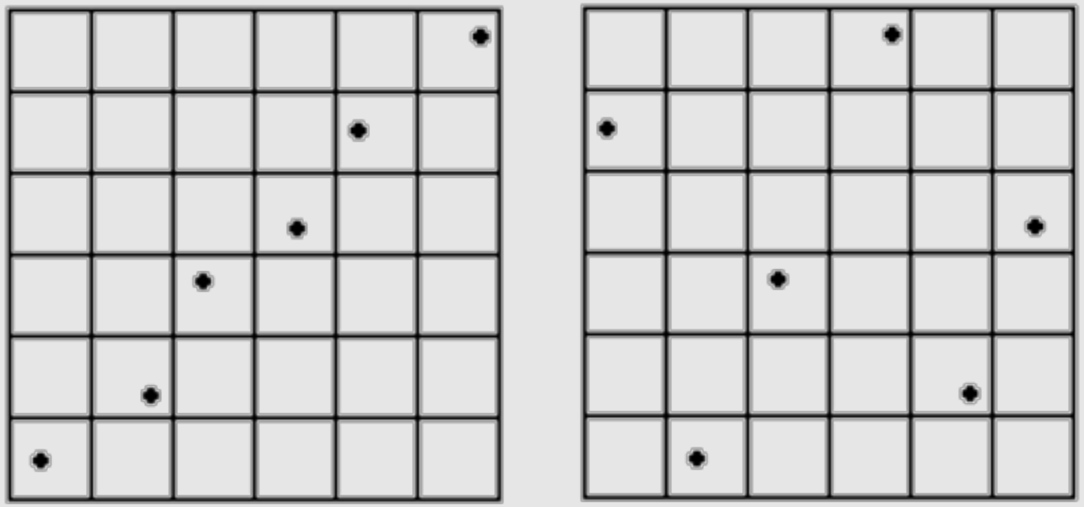
- Divide into N x N grid, place a sample randomly per grid cell
  - [+] Easy to compute
  - [+] A more constrained version of random sampling
  - [-] Ensures distance between samples, but not enough!





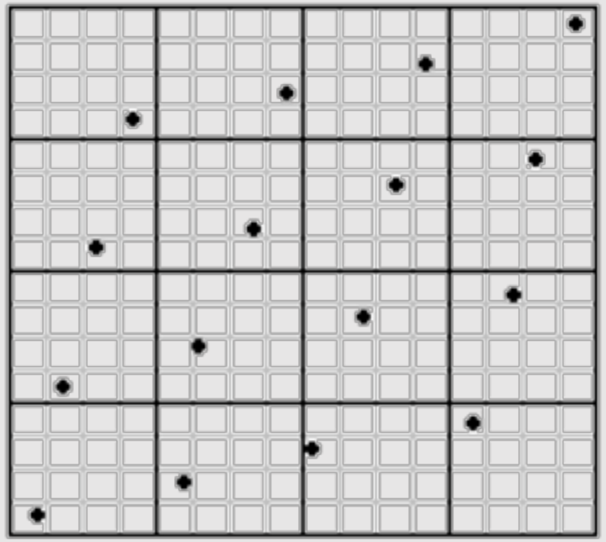
N-Rooks Sampling

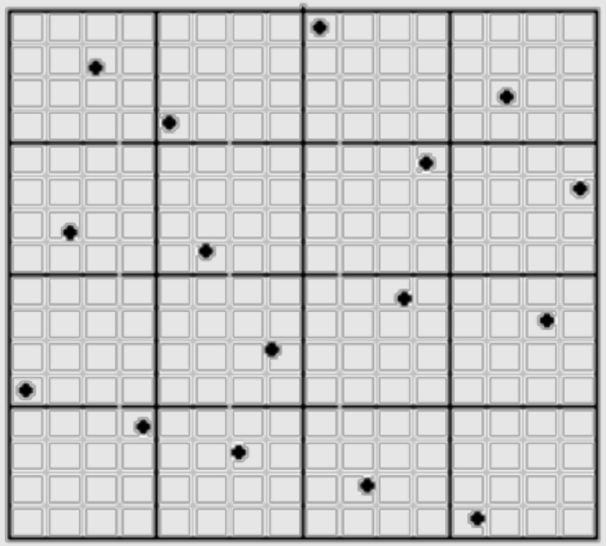
- All samples start on the diagonal, randomly shuffle (x, y) coordinates until rooks condition satisfied (no 2 samples lie on the same column or row)
  - [+] Provides good sample sparsity
  - [-] Expensive to compute
  - [-] Possibility of not terminating



Multi-Jittered Sampling

- Jittering + n-rook sampling
  - [+] Provides good sample sparsity
  - [+] Easier to satisfy rook condition
  - [-] Expensive to compute





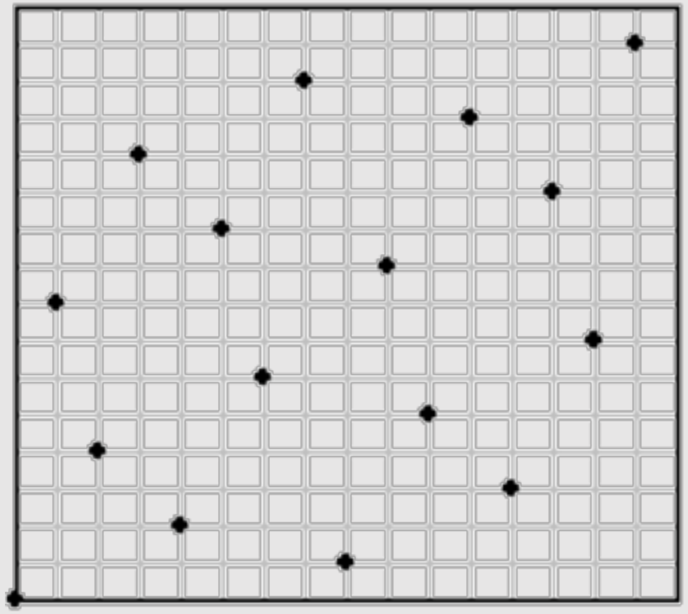
Hammersley Sampling

- Sample according to a fixed, well formed distribution
  - [+] Can pre-compute results
  - [+] Evenly distributed in 2D space
  - [-] No randomness in results

$$\Phi_2(i) \in [0,1] = \sum_{j=0}^{n} a_j(i) 2^{-j-1} = a_0 2^{-1} + a_1 2^{-2} + \ldots$$

$$1101 \Rightarrow 0.1011 = 1/2 + 1/8 + 1/16 = 11/16 = 0.6975$$

$$p_j = (x_i, y_i) = [\frac{i}{n}, \Phi_2(i)]$$



Low-Discrepancy Sampling

- In general, number of samples should be **proportional to area**
- **Discrepancy** measures deviation from this ideal

discrepancy of sample points X in a region S

# of samples in S

$$d_S(X) := \left| A(S) - \frac{n(S)}{|X|} \right|$$

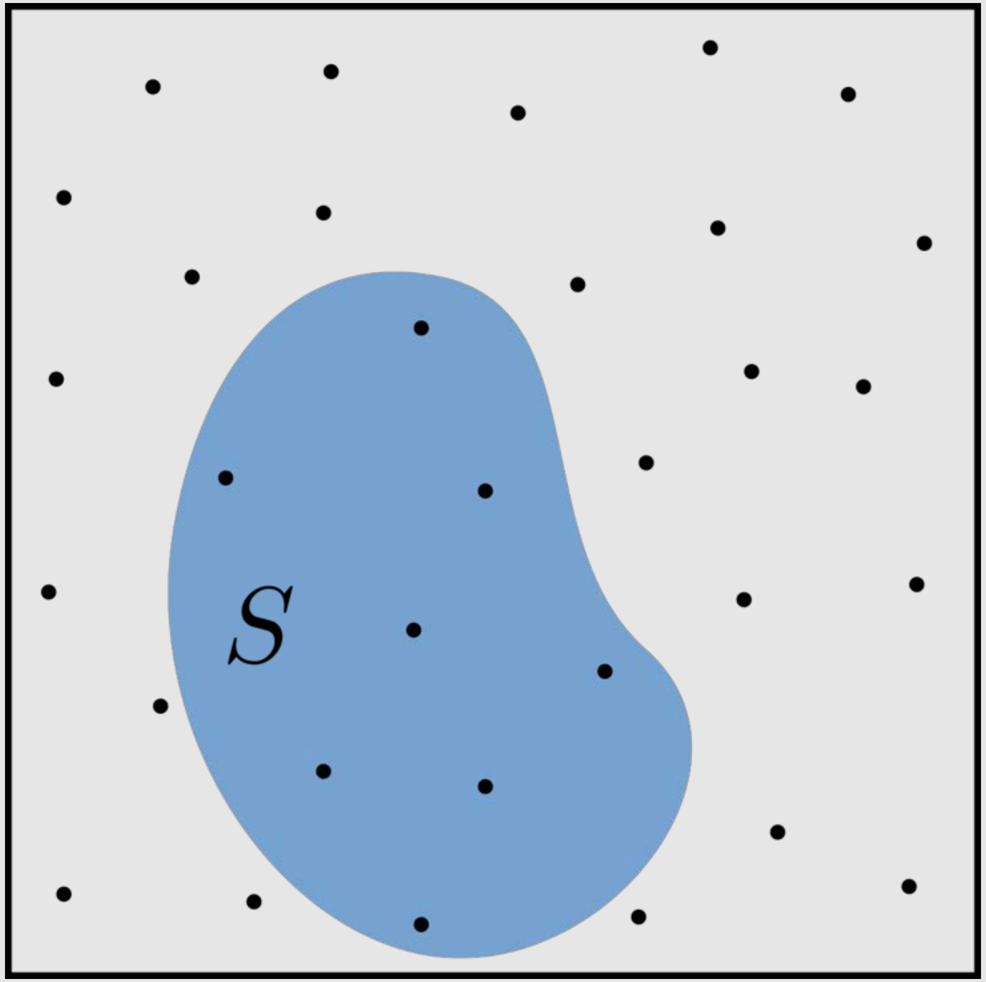
area of S

total # of samples in X

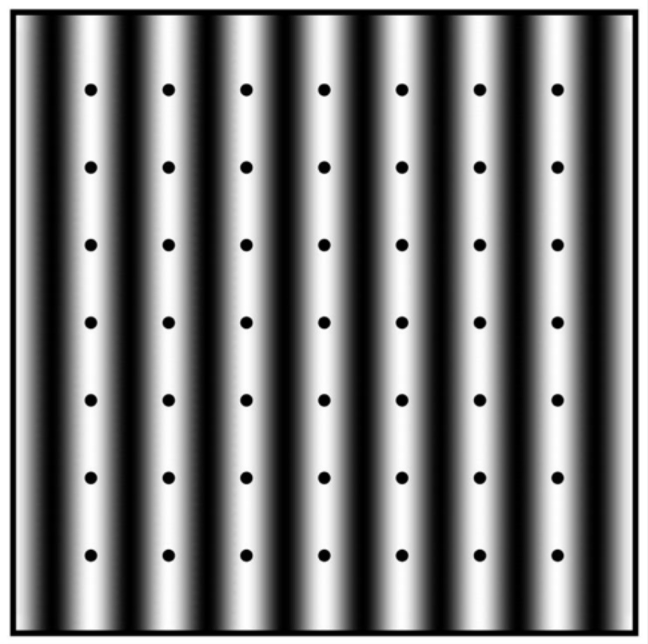
$$D(X) := \max_{S \in F} d_S(X)$$

overall discrepancy

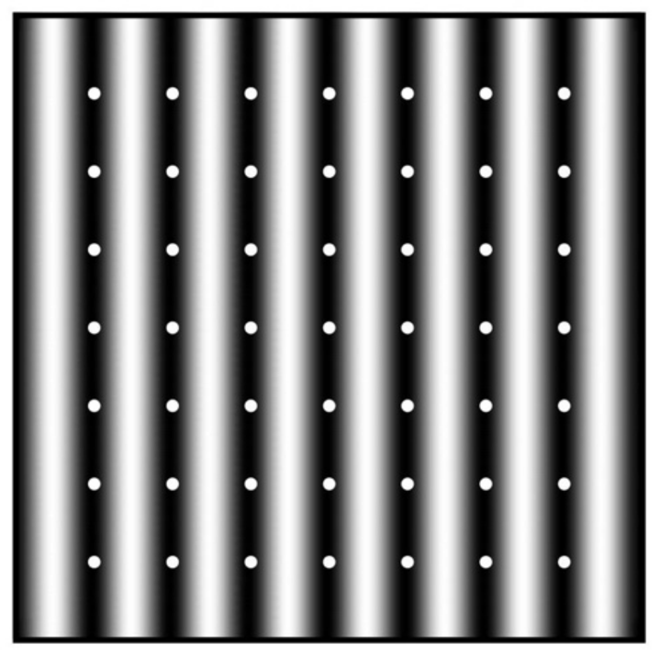
some family of regions S (box, disk, etc...)



Low-Discrepancy Sampling



$$\frac{1}{n}\sum_{i=1}^{n} f(x_i) = 1$$



$$\frac{1}{n}\sum_{i=1}^{n} f(x_i) = 0$$

- A uniform grid has the lowest discrepancy
  - But even low-discrepancy patterns can exhibit poor behavior
  - We want patterns to be **anisotropic** (no preferred direction)

Blue Noise

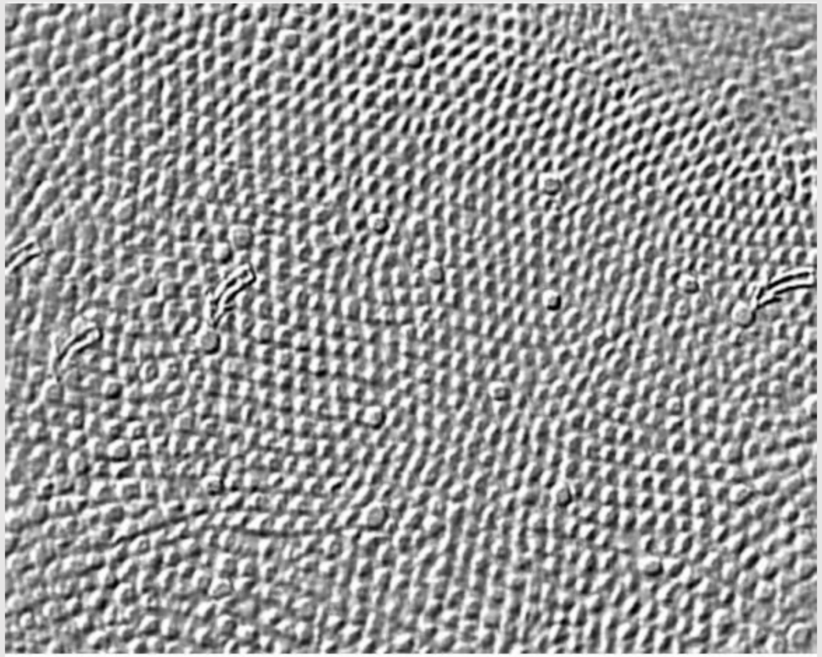
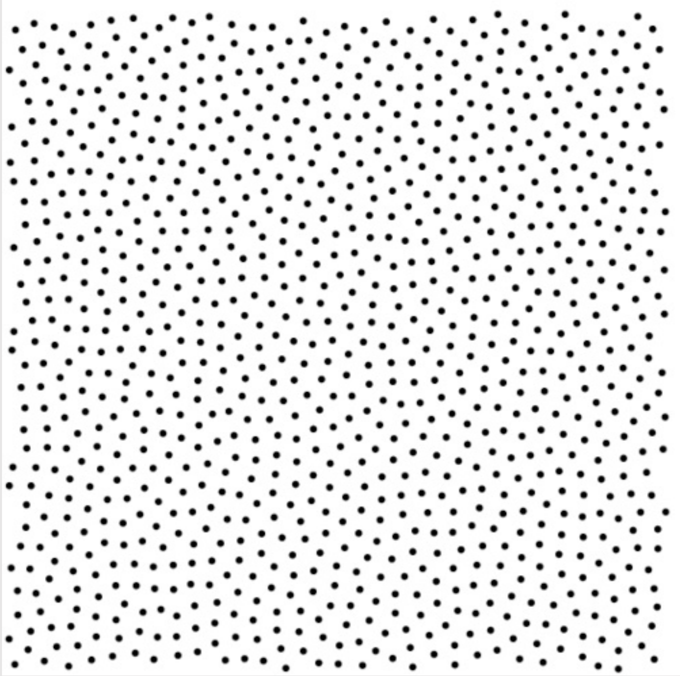


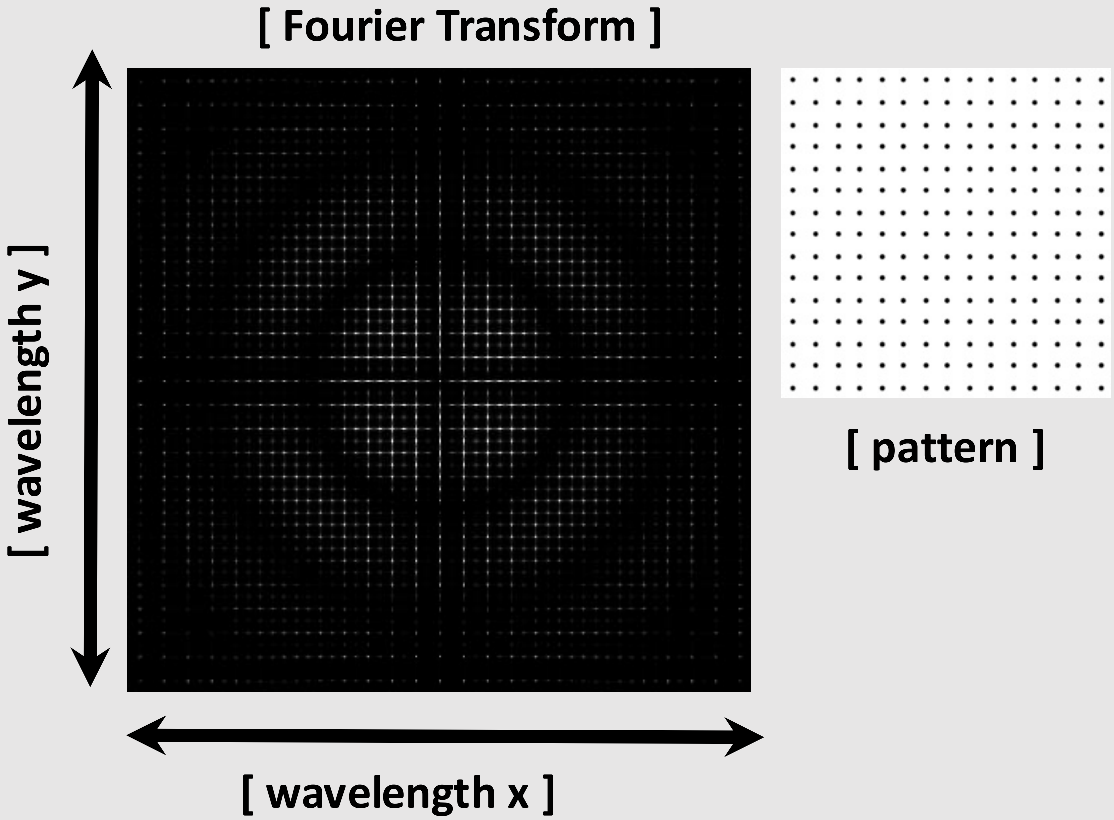
Fig. 13. Tangential section through the human fovea. Larger cones (arrows) are blue cones. From Ahnelt et al. 1987.

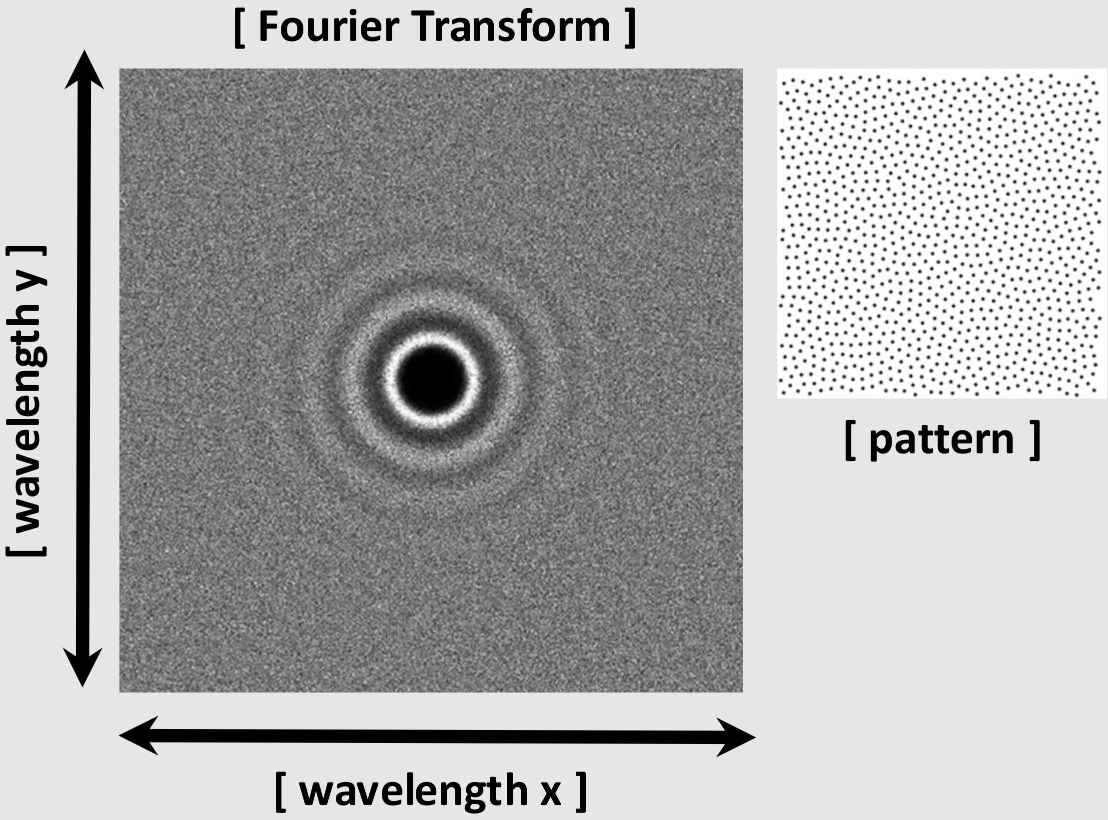


["blue noise"]

- Monkey retina exhibits **blue noise** pattern [Yellott 1983]
    - No preferred directions **(anisotropic)**

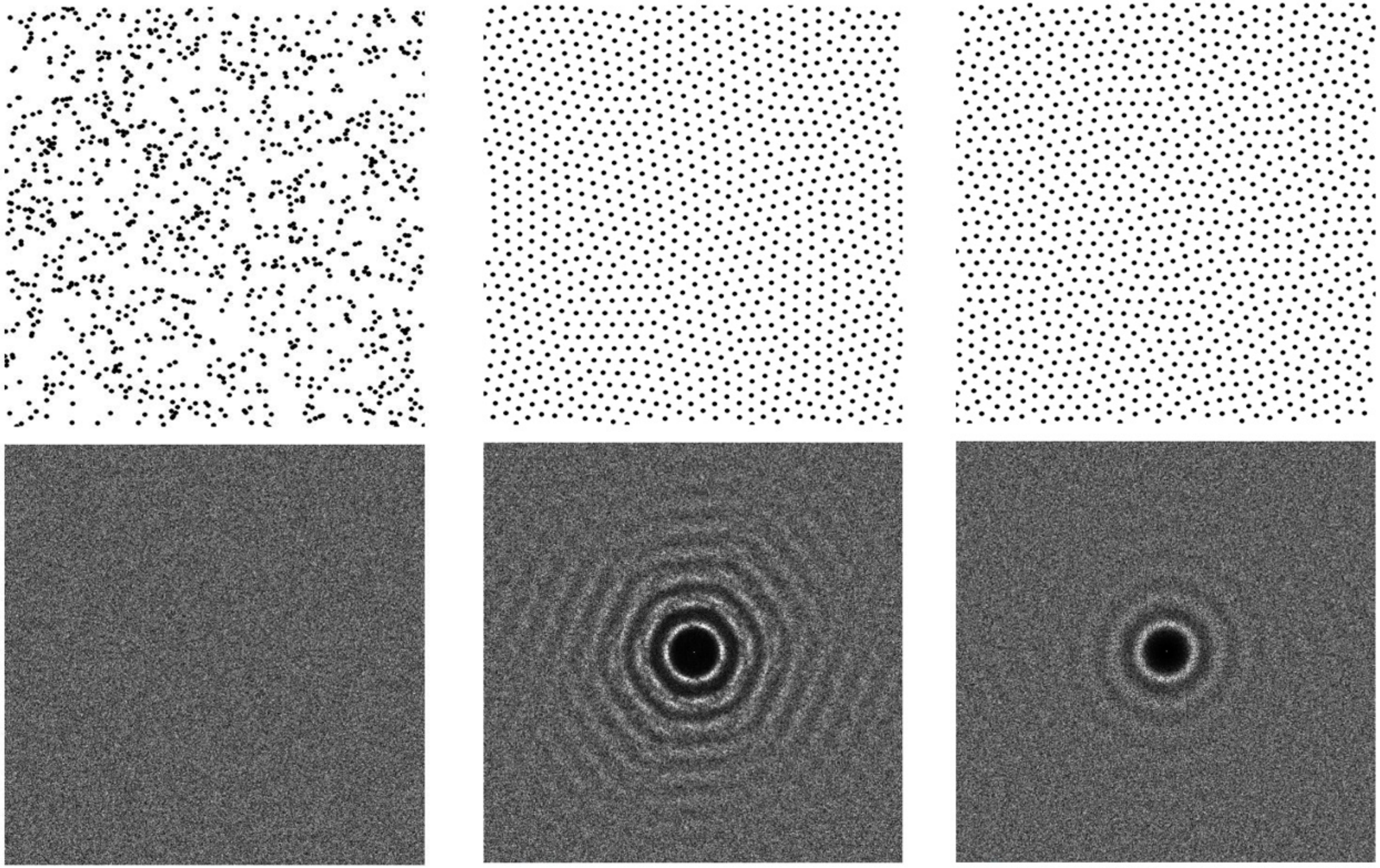
Blue Noise Fourier Transform





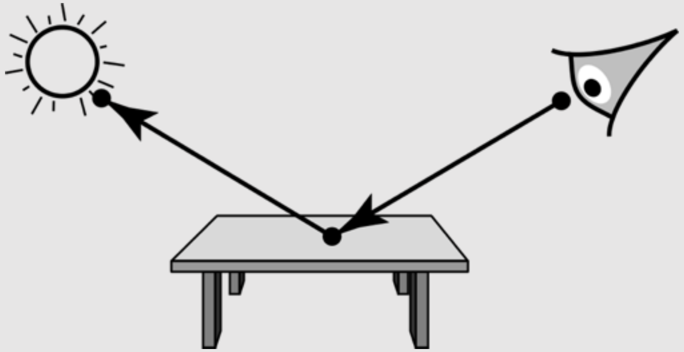
- Regular pattern has "spikes" at regular intervals
- Blue noise is spread evenly over all frequencies in all directions
  - Bright center "ring" corresponds to sample spacing

Blue Noise Fourier Transform



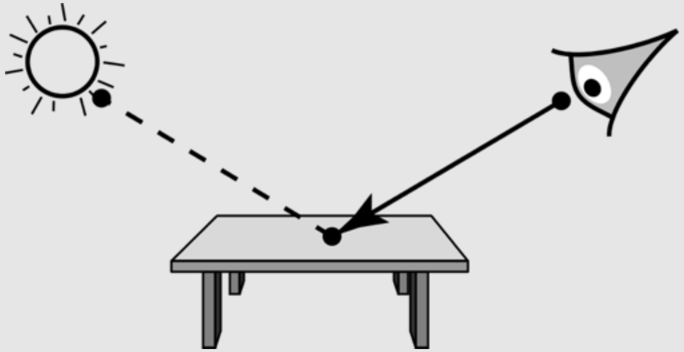
- ~~Monte-Carlo Sampling~~
- ~~Biased vs Unbiased Estimators~~
- Physically-Based Rendering Methods

Previous Methods



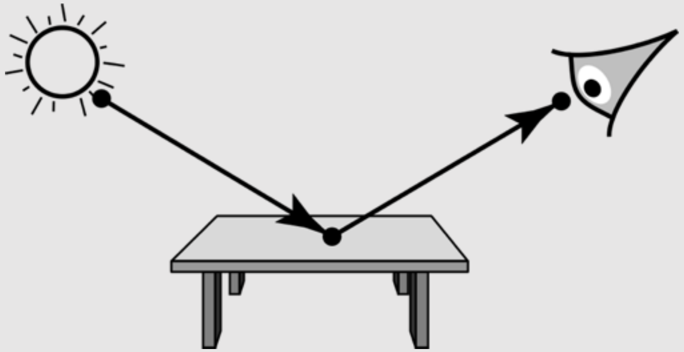
[backward path tracing]

Fails: cannot intersect point lights



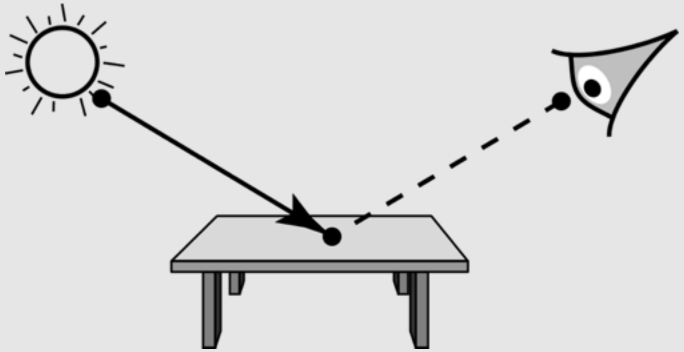
[backward path tracing + connect to light]

Works: reaches point lights



[forward path tracing]

Fails: cannot intersect pinhole camera



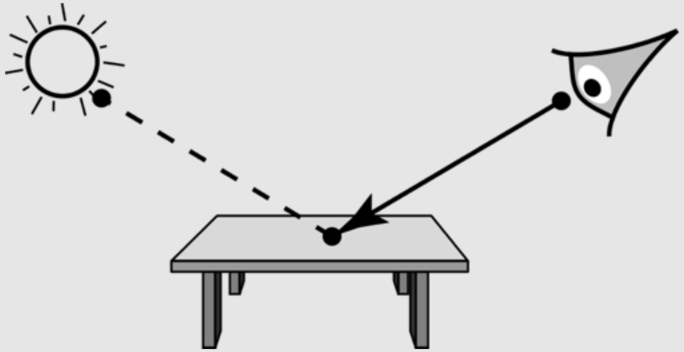
[forward path tracing + connect to camera]

Works: reaches pinhole camera

Path Tracing Can Be Biased

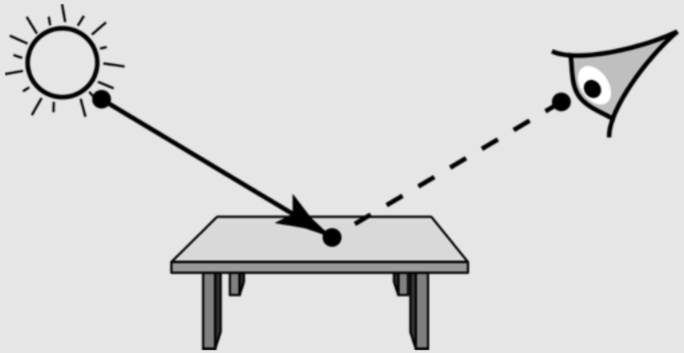
- Deliberately connect terminating rays to light (forward) or camera (backward)

- Probability of sampling a ray that hits a non-volume source (point light, pinhole camera) is 0
  - We bias our renderer by choosing those rays



[backward path tracing + connect to light]

works: reaches point lights



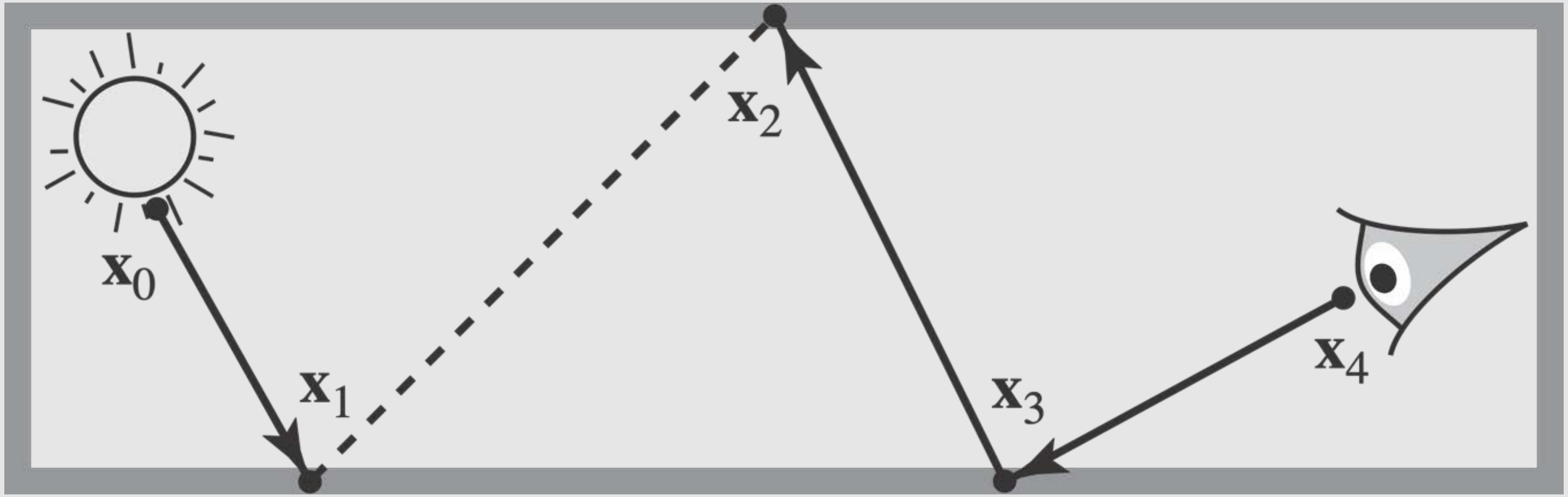
[forward path tracing + connect to camera]

works: reaches pinhole camera

Bidirectional Path Tracing

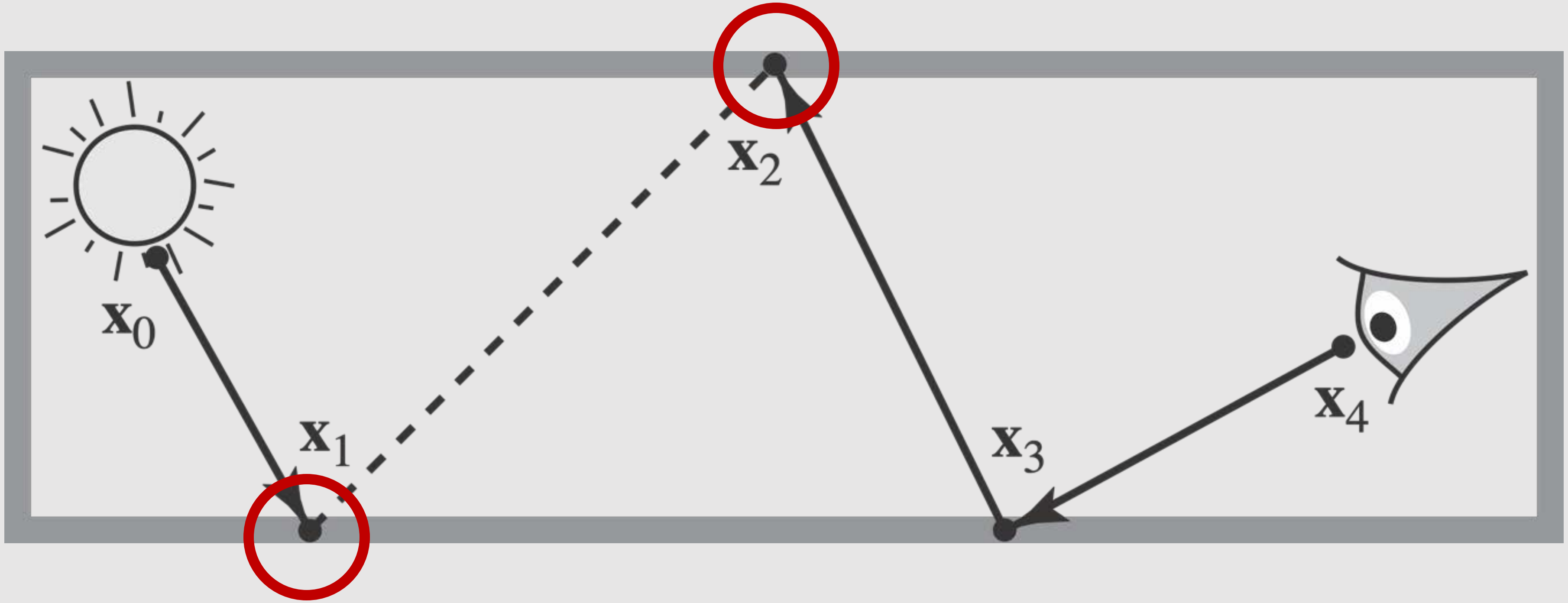
- If path-tracing is so great, why not do it **twice?**
  - Main idea of bidirectional!
- Trace a ray from the camera into the scene
- Trace a ray from the light into the scene
  - Connect the rays at the end

- Unbiased algorithm
  - No longer trying to connect rays through non-volume sources
- Can set different lengths per ray
  - Example: Forward m = 2, Backward m = 1



Bidirectional Path Tracing

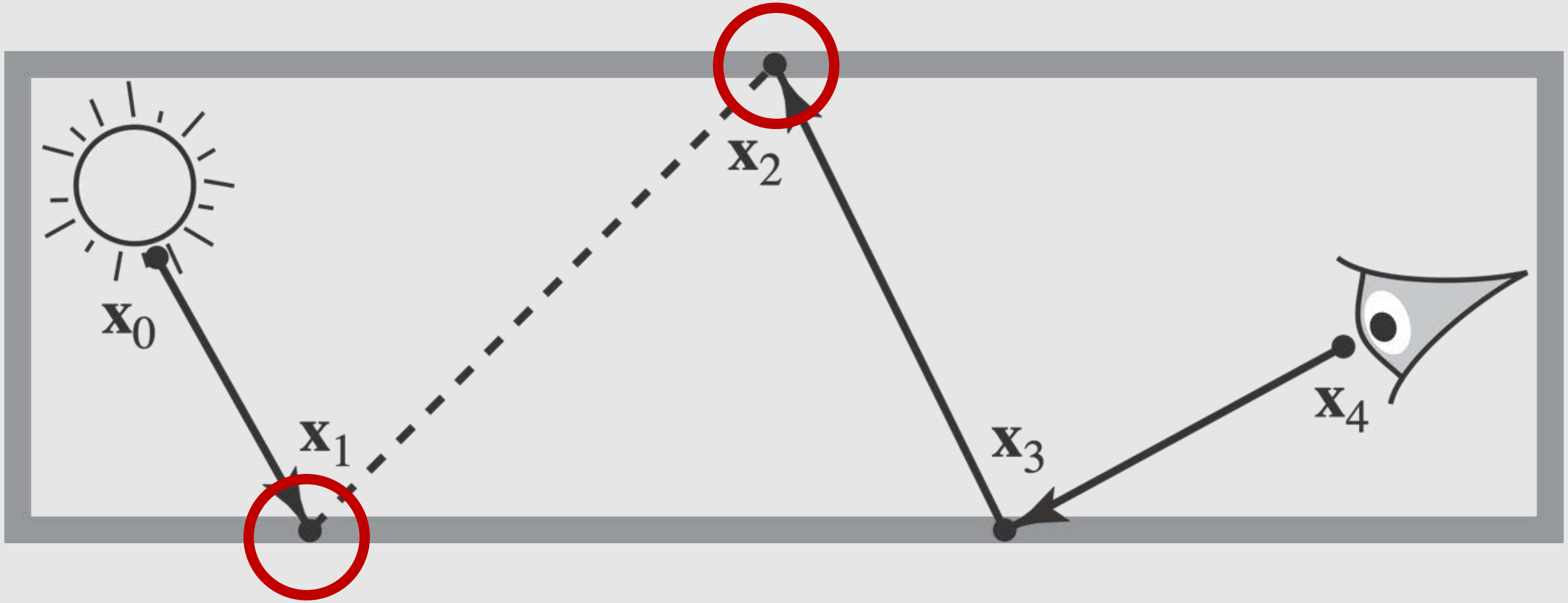
Issue: what if these are mirrors!



Bidirectional Path Tracing

- In cases of mirrors, we cannot choose any ray path
- Instead, continue tracing rays until diffuse surfaces are reached on both rays

Issue: what if these are mirrors!

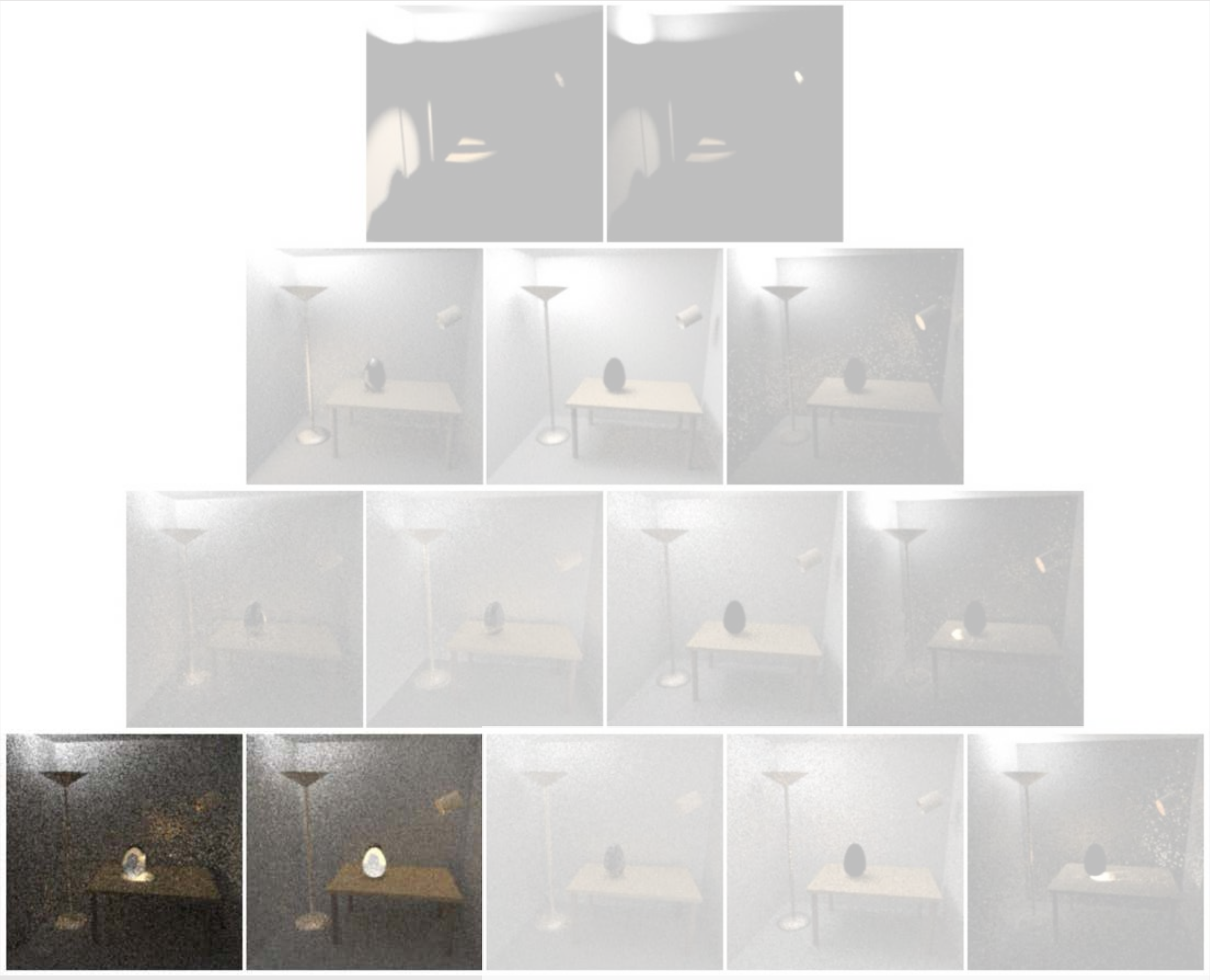


Bidirectional Path Tracing

[final image]

- Each row shows path length
- As we move over images in a row, we decrease forward ray depth and increase a backward ray depth
  - Overall length kept constant per row

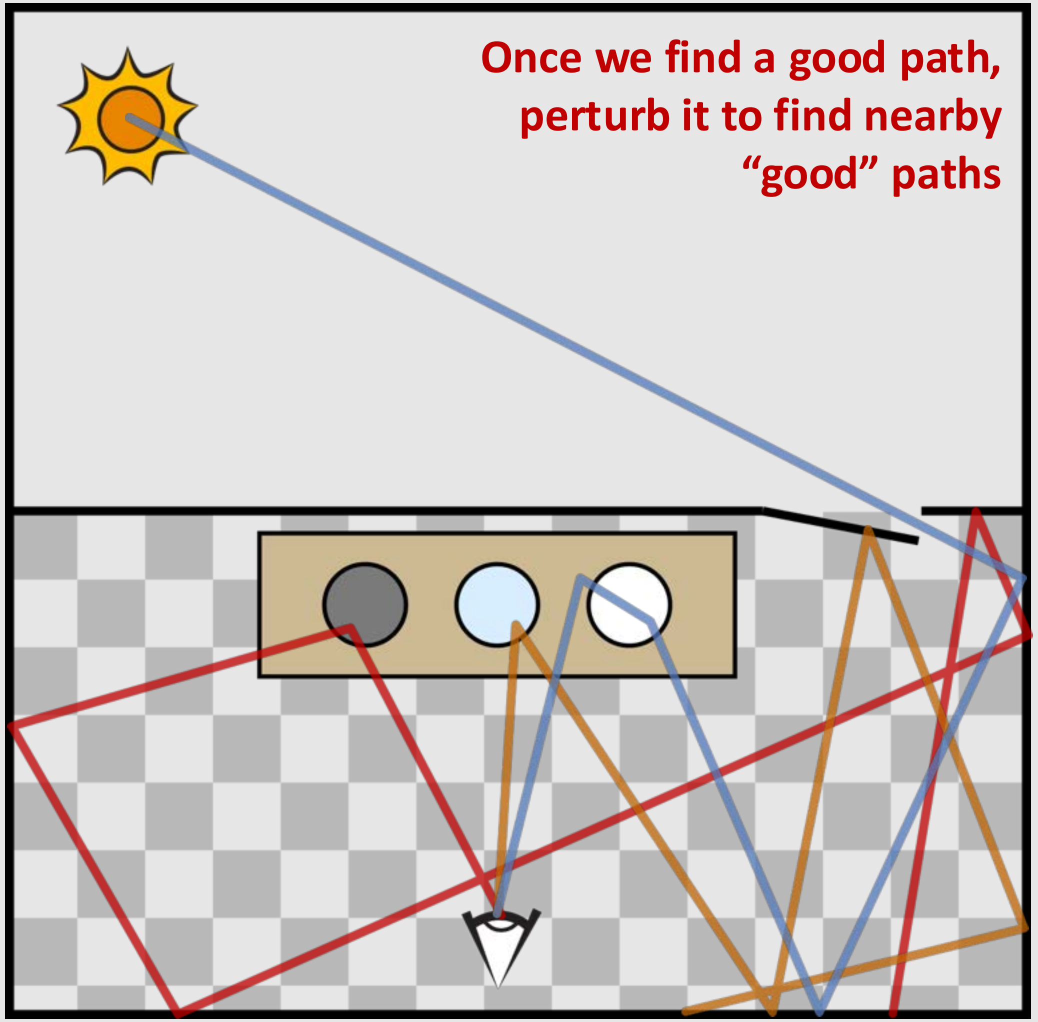
Bidirectional Path Tracing



[final image]

- Not easy to tell which path lengths work well for a scene!
  - The glass egg is illuminated at specific path lengths for forward and backward rays

Good Paths Are Hard To Find



[Bidirectional Path Tracing]

[Metropolis Light Transport]

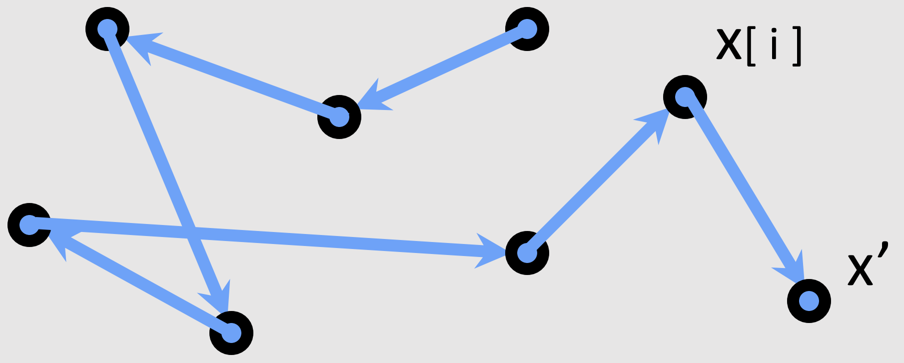
Metropolis Hasting Algorithm

- *"Once we find a good path, perturb it to find nearby 'good' paths"* – previous slide

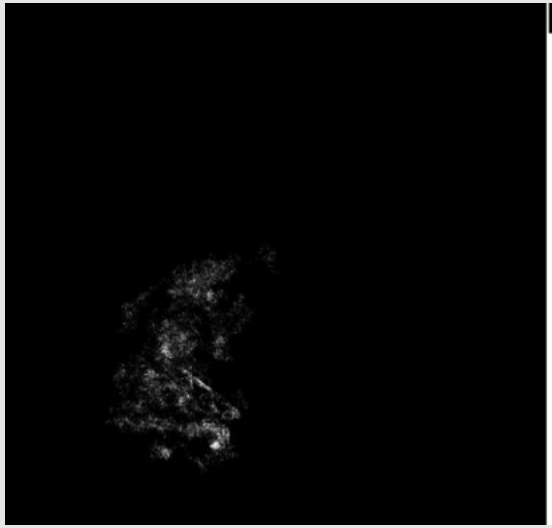
- **Algorithm:** take random walk of dependent samples
  - If in an area where sampling yields high values, stay in or near the area
    - Otherwise move far away

- Sample distribution should be proportional to integrand
  - Make sure mutations are "ergodic" (reach whole space)
  - Need to take a long walk, so initial point doesn't matter

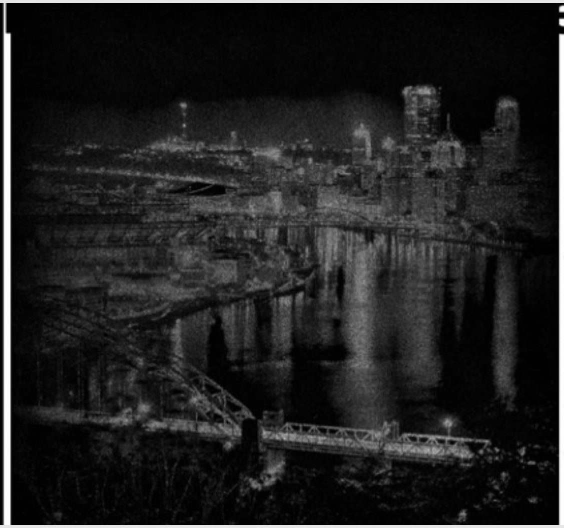
```
float r = rand();
// if f(x') >> f(x[i]), then a is large
// and we increase chances of moving to x'
// if f(x') << f(x[i]), then a is small
// and we increase chances of staying at x
float a = f(x')/f(x[i]);
if (r < a)
  x[i+1] = x';
else
  x[i+1] = x;
```



Metropolis Hasting: Sampling An Image



[short walk]



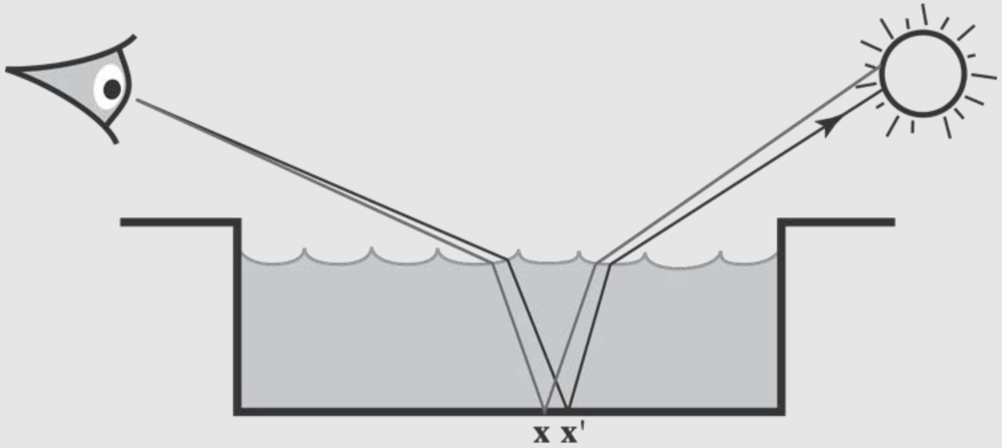
[long walk]

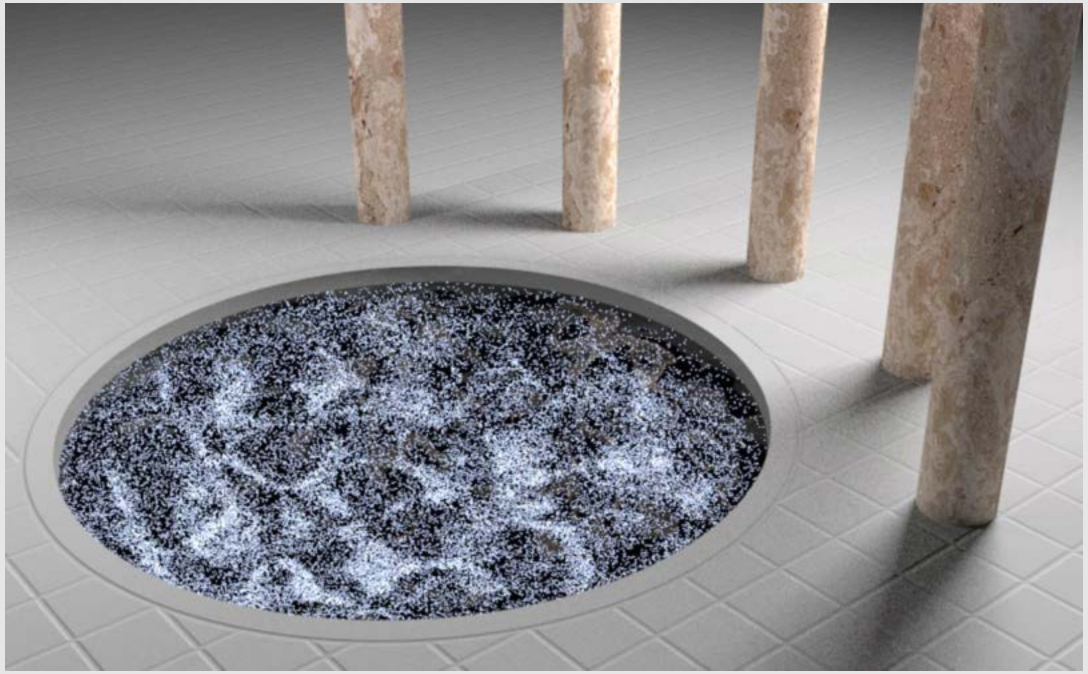
[original image]

- Want to take samples proportional to image density f
- Occasionally jump to a random point (ergodicity)
- Transition probability is 'relative darkness'
  - $f(x')/f(x_i)$

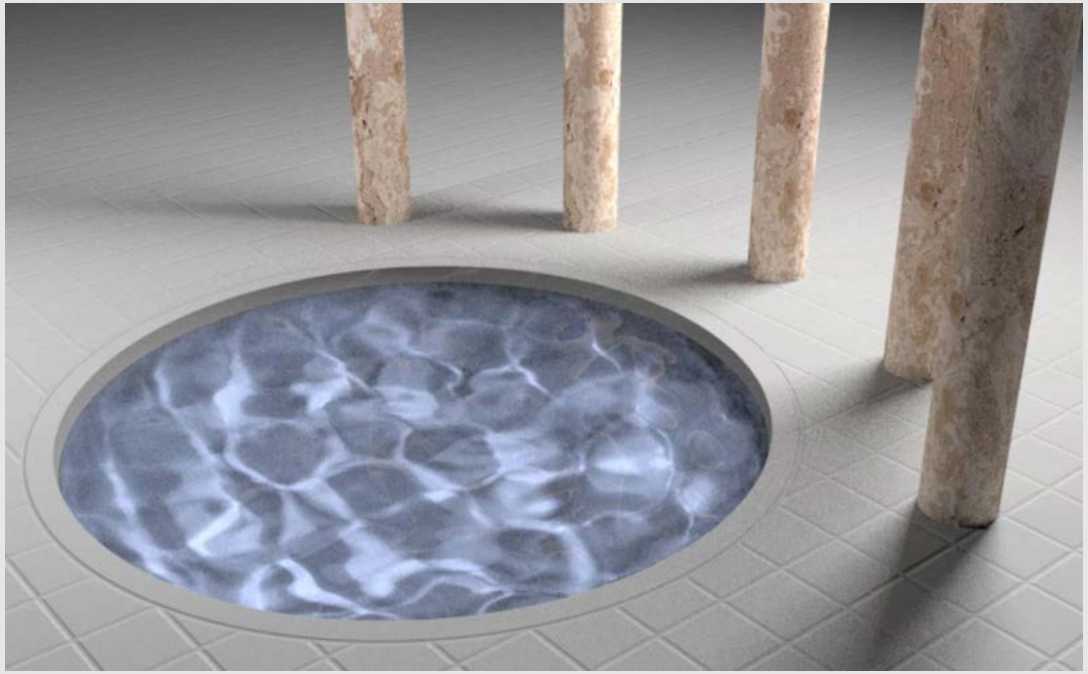
Metropolis Light Transport

- **Similar idea:** mutate good paths
- Water causes paths to refract a lot
  - Small mutations allows renderer to find contributions faster
- Path Tracing and MLT rendered in the same time





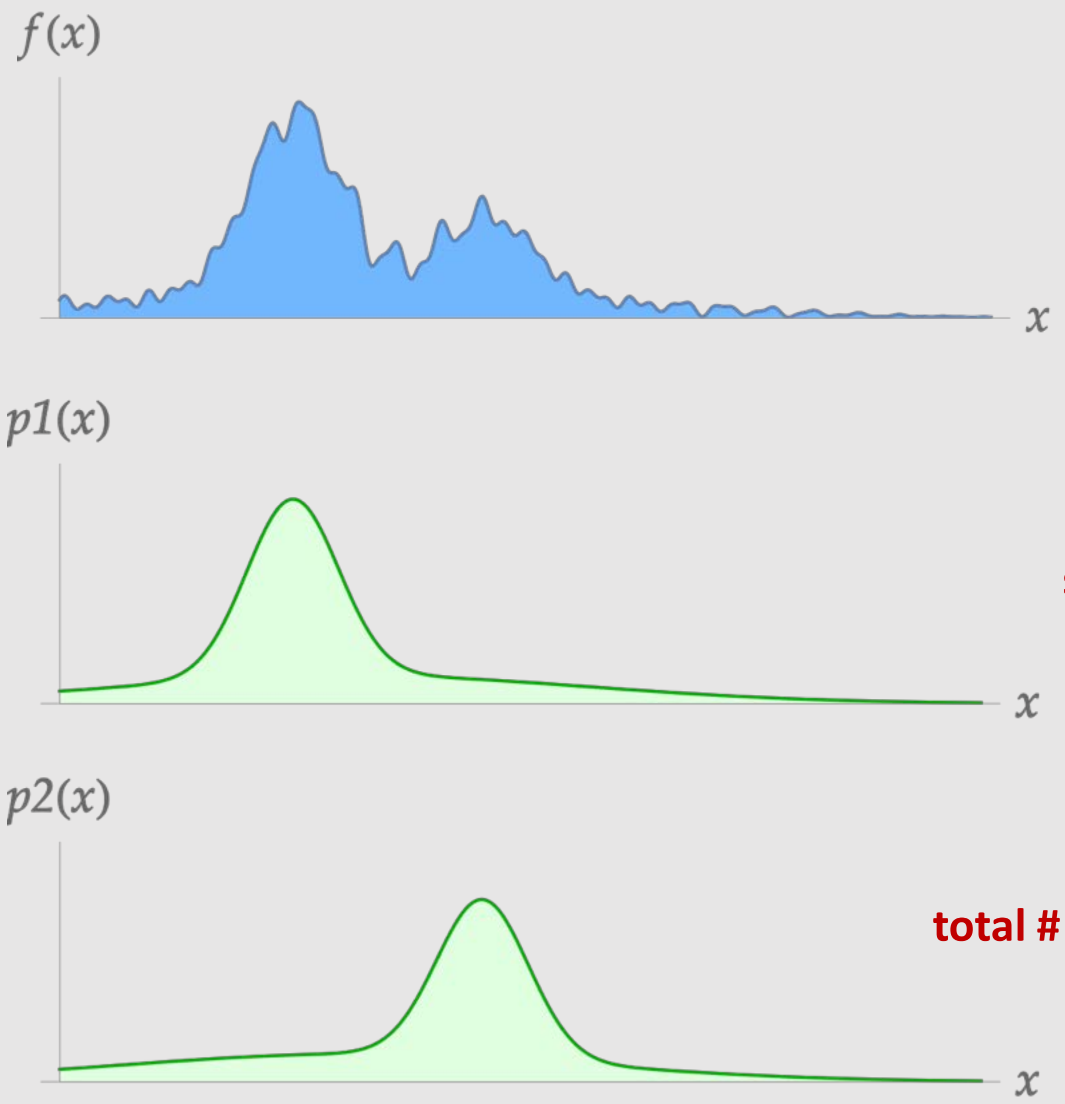
[Path Tracing]



[Metropolis Light Transport]

If there are so many good sampling methods,
why not combine them?

Multiple Importance Sampling

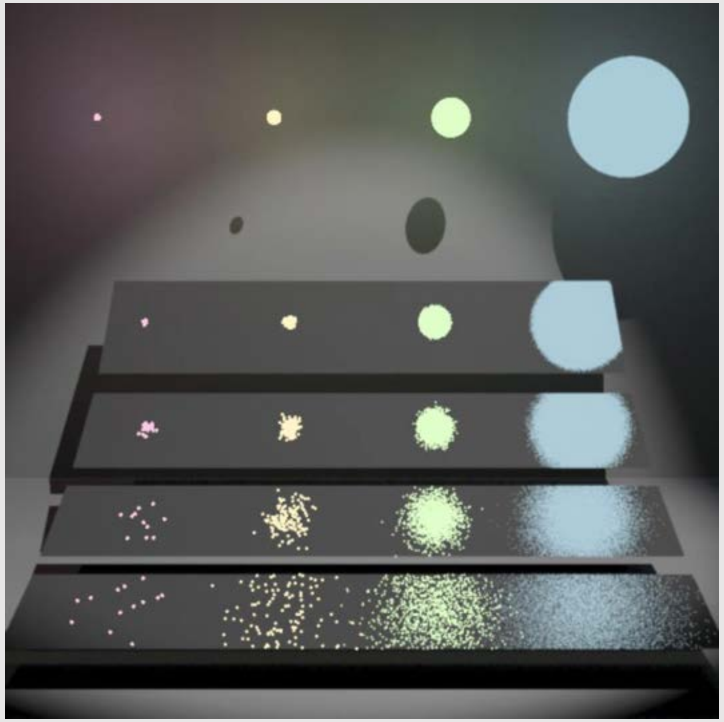


- **Multiple Importance Sampling:** combine strategies to preserve strengths of all of them
  - Think of it as taking multiple rays/samples at each bounce

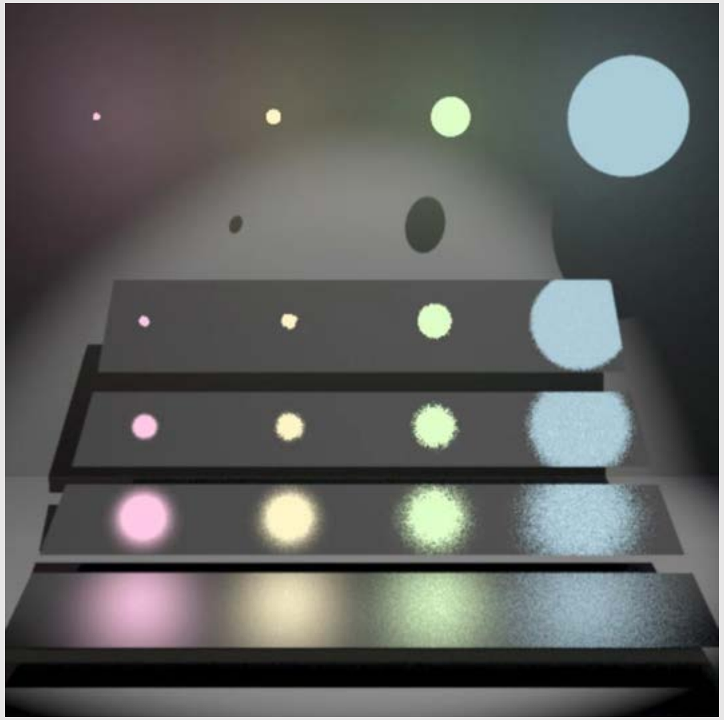
$$\frac{1}{N}\sum_{i=1}^{n}\sum_{j=1}^{n_i}\frac{f(x_{ij})}{\sum_k c_k p_k(x_{ij})}$$

- sum over strategies
- sum over samples
- j^{th} sample taken with i^{th} strategy
- total # of samples
- fraction of samples taken with k^{th} strategy
- k^{th} strategy PDF

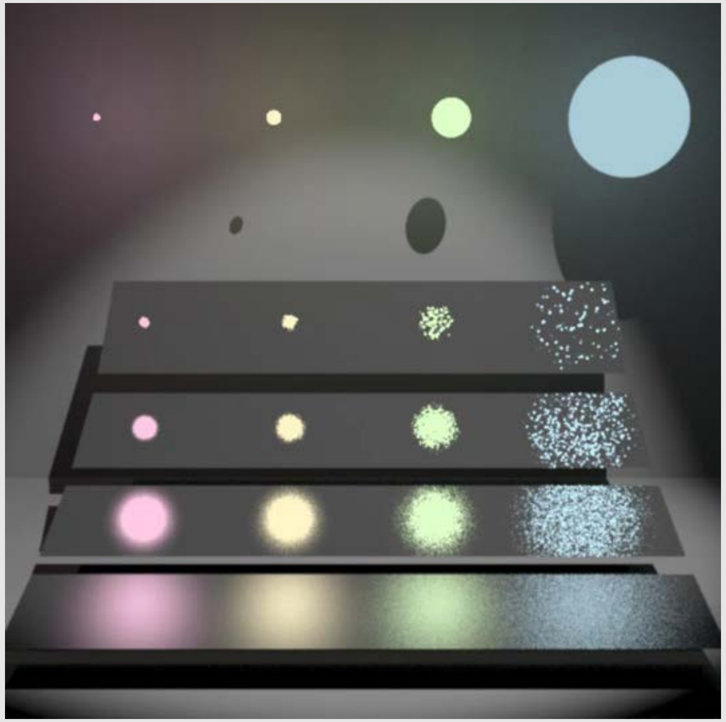
Multiple Importance Sampling



[sample materials]



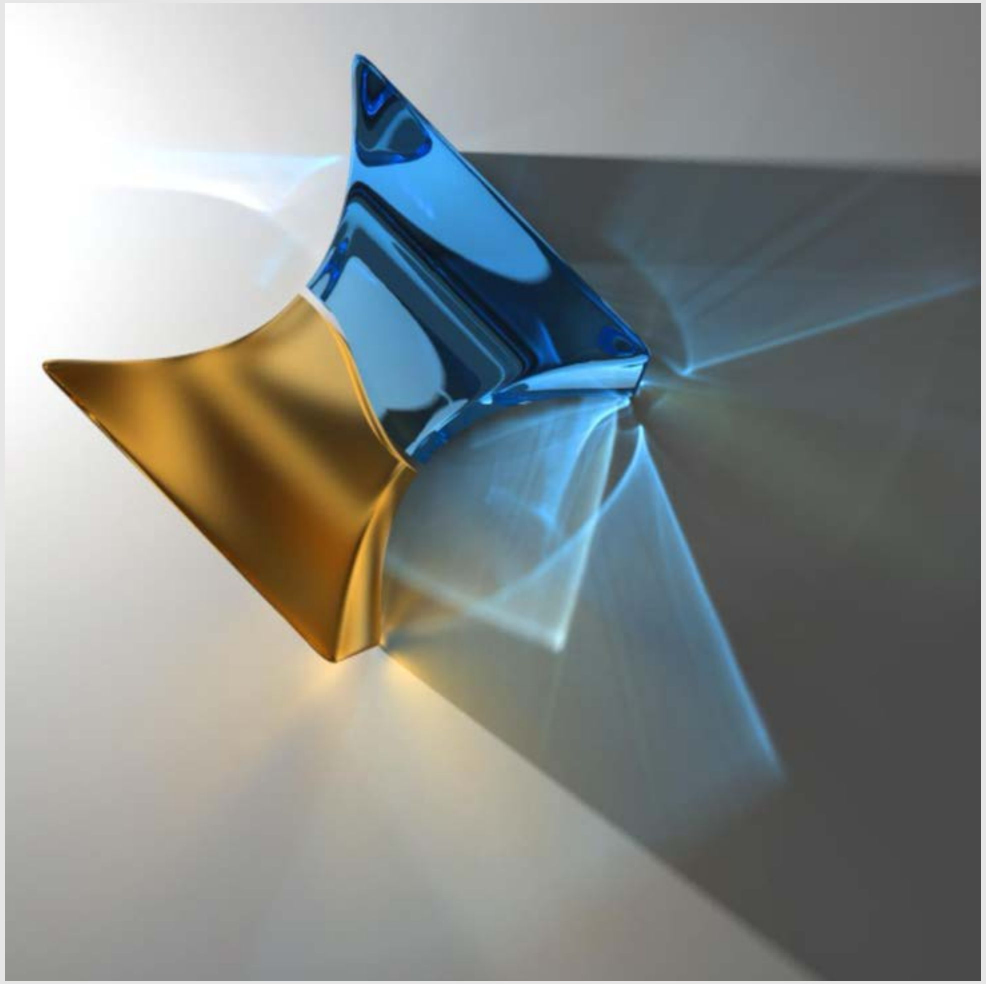
[sample both]



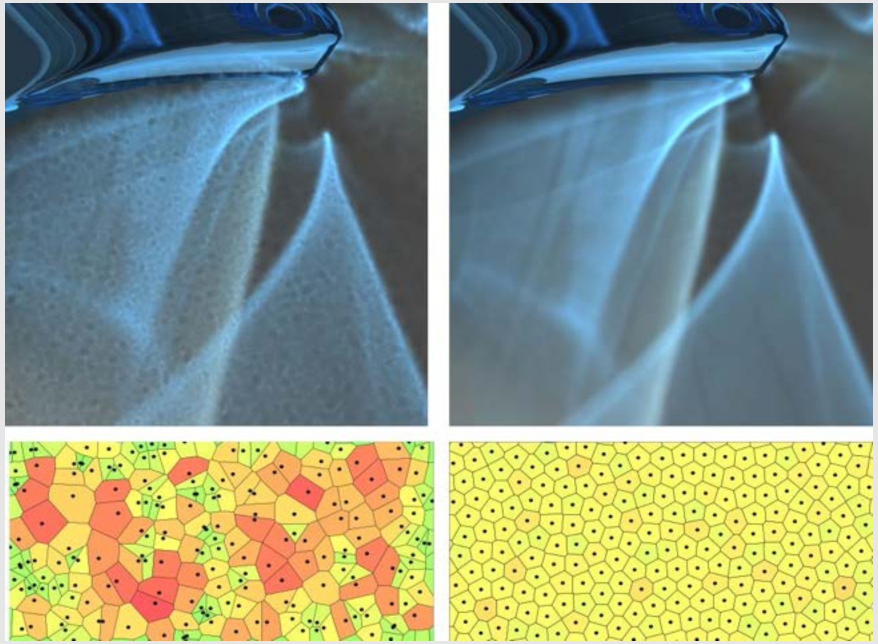
[sample lights]

- Normally need to pick next ray bounce as hitting a material or hitting light
    - MIS allows us to take both rays and average them together
    - At each bounce, trace a ray as normal, and another ray to the light

Photon Mapping

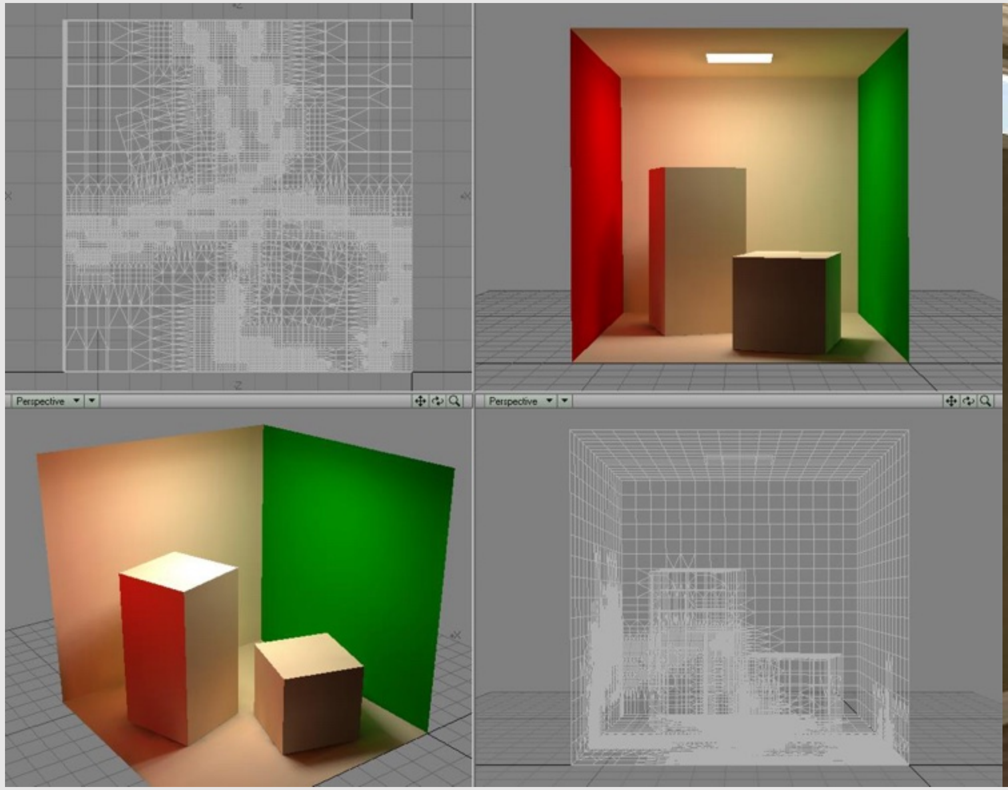


- Trace particles from light, deposit "photons" in KD-tree
  - Useful for, e.g., caustics, fog
- Voronoi diagrams can improve photon distribution
  - **Careful:** poor Voronoi resolution causes aliasing!



Finite Element Radiosity

- Transport light between patches in scene
- Solve large linear system for equilibrium distribution
  - Good for diffuse lighting; hard to capture other light paths
    - Light paths travel in groups
    - Difficult when light diverges



Rendering Algorithm Chart

method	consistent?	unbiased?
Rasterization	no	no
Path Tracing	almost	almost
Bidirectional Path Tracing	yes	yes
Metropolis Light Transport	yes	yes
Photon Mapping	yes	no
Finite Element Radiosity	no	no