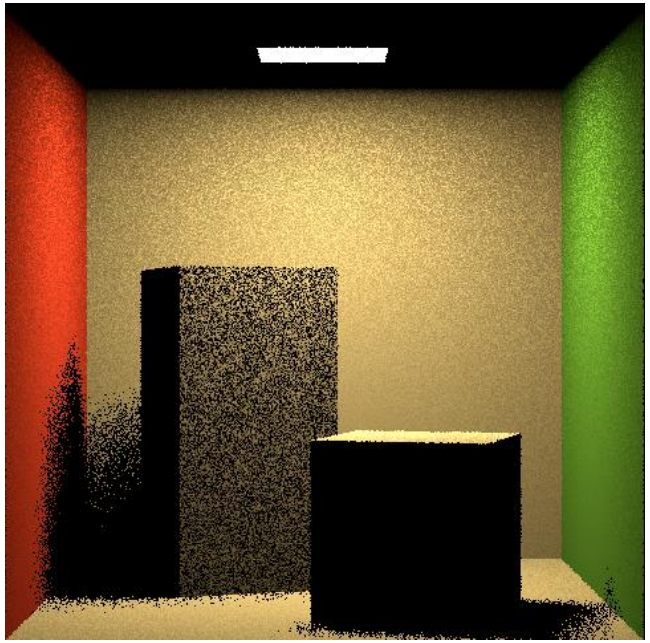
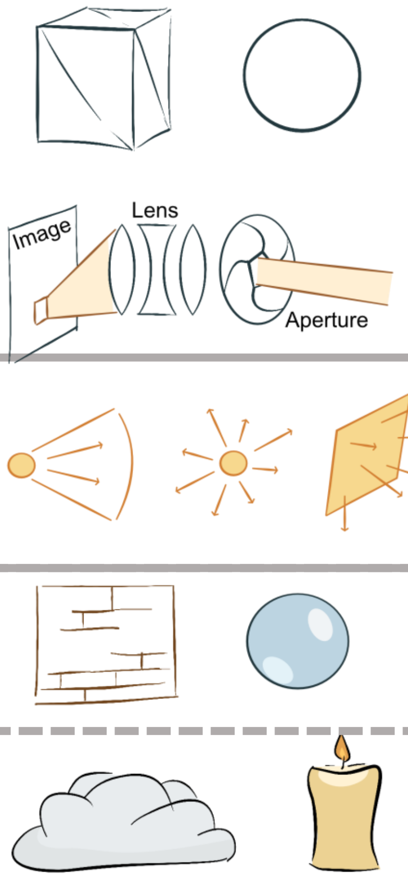


Sampling, Materials, and Volumes

Last time



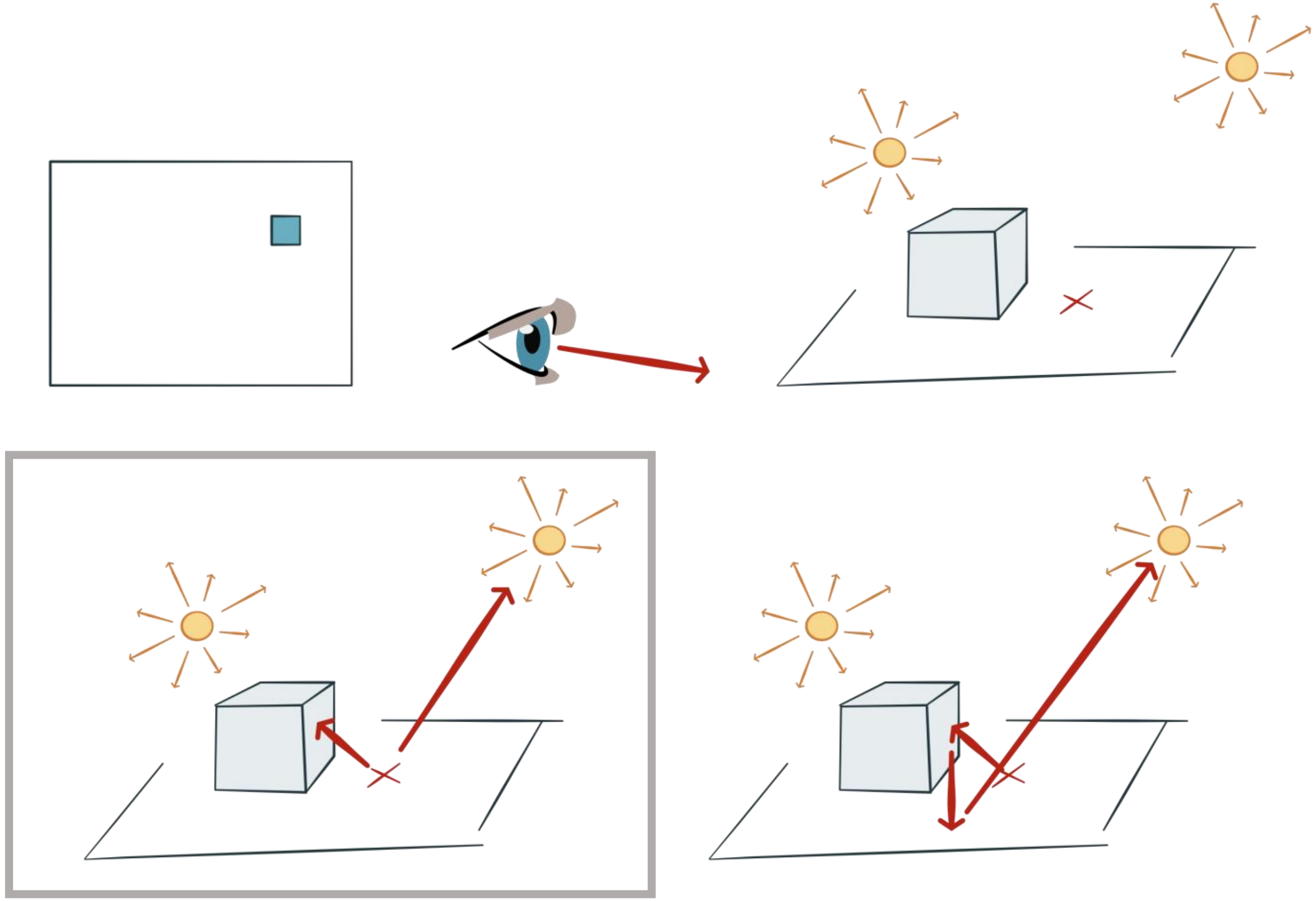
Geometries

Camera model

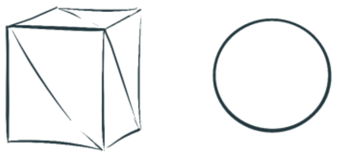
Light sources

Materials & Textures

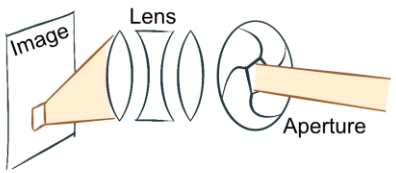
Volumes



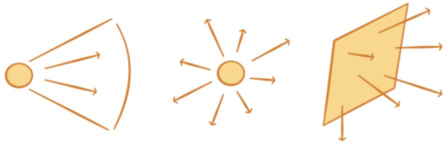
Today



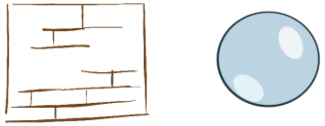
Geometries



Camera model

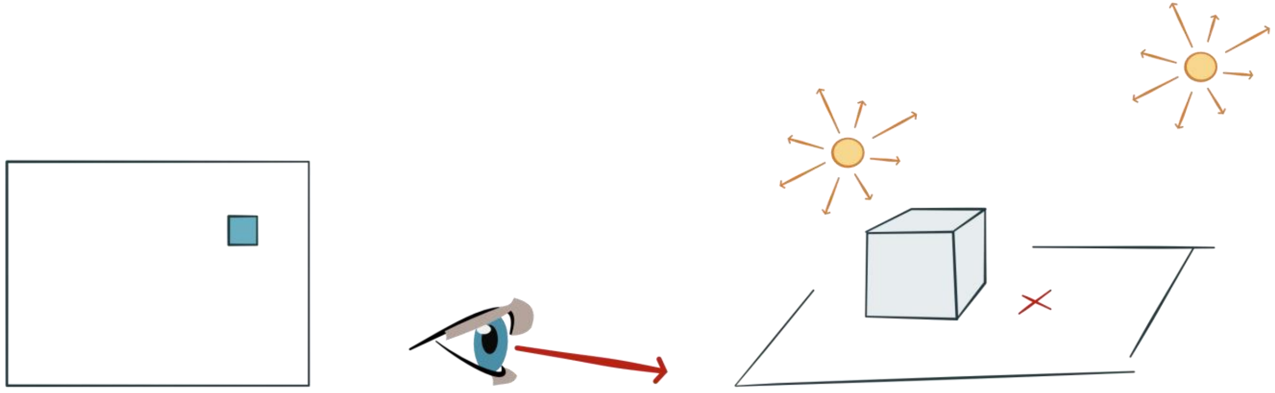


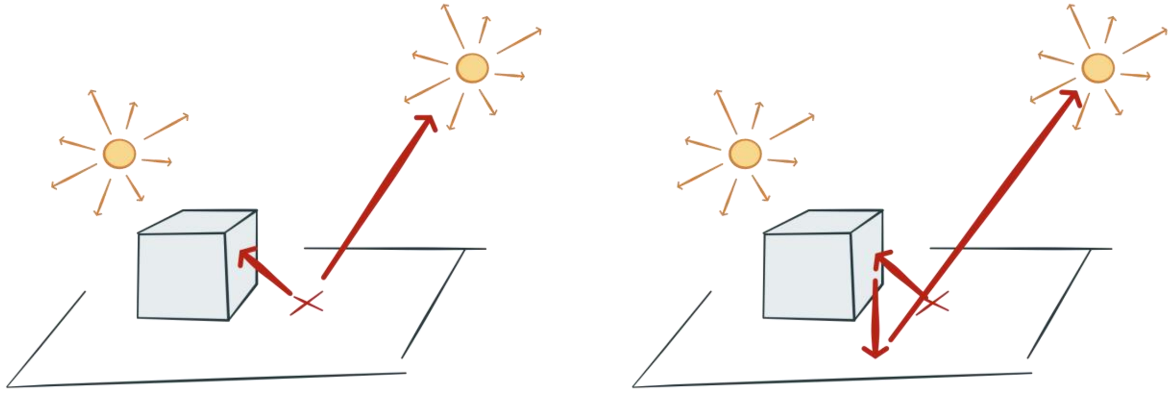
Light sources



Materials & Textures

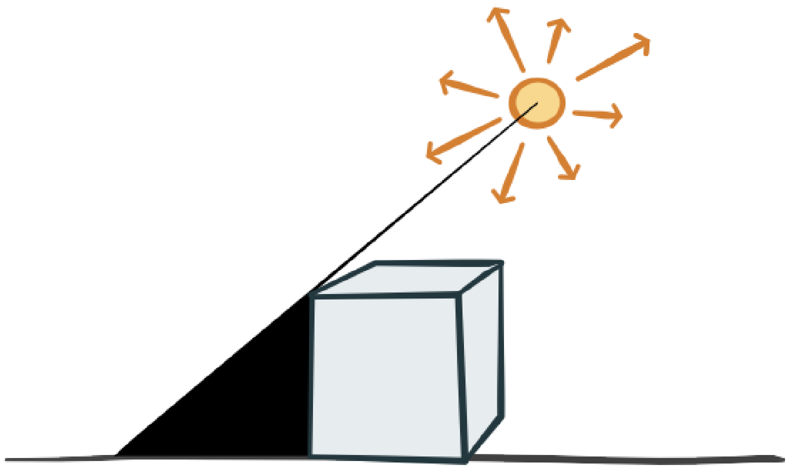
Volumes





What about soft shadows?

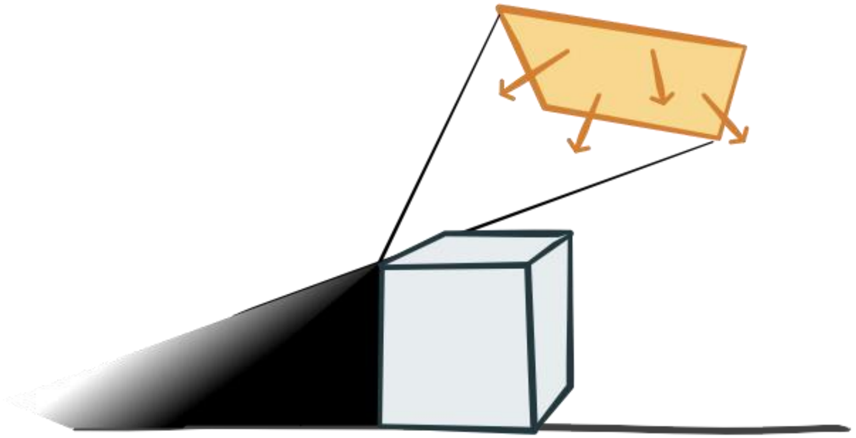
- Point, spot, and directional lights provide hard shadows only
- For soft shadows, we need, e.g., area lights



- Light arrives from all points on the area (infinitely many)

➔ No simple analytic solution for the integral

$$L_{\mathrm{o}}(x,\omega_{\mathrm{o}}) = L_{\mathrm{e}}(x,\omega_{\mathrm{o}}) + \boxed{\int_\Omega L_{\mathrm{i}}(x,\omega_{\mathrm{i}}) f_{\mathrm{r}}(x,\omega_{\mathrm{i}},\omega_{\mathrm{o}}) |\cos\theta_{\mathrm{i}}| d\omega_{\mathrm{i}}}$$

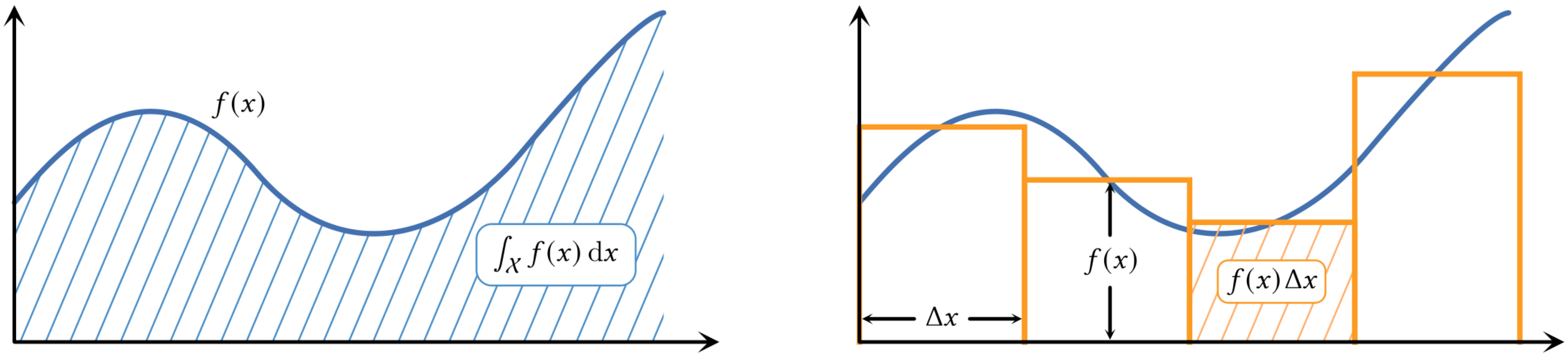


Monte Carlo integration

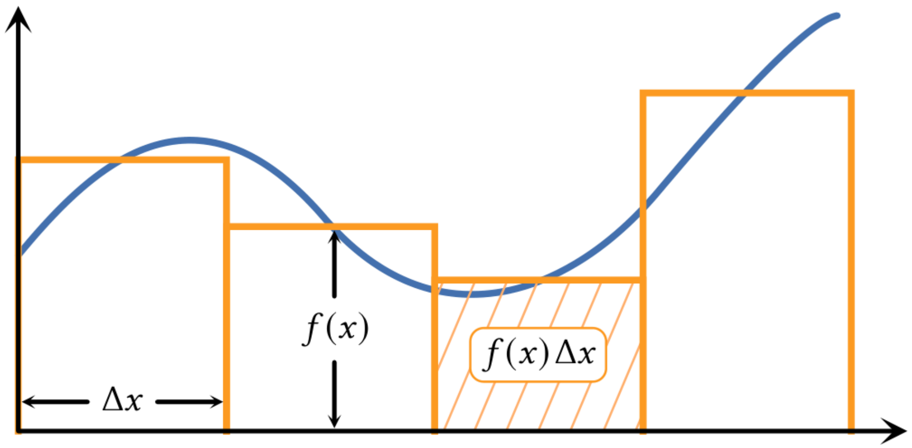
The favored numerical integration method for rendering

Formalizing: First, lets recall the Riemann sum

$$\int_X f(x)dx \coloneqq \lim_{\Delta x \to 0} \sum_i f(x_i)\Delta x$$



Why not use a Riemann sum?



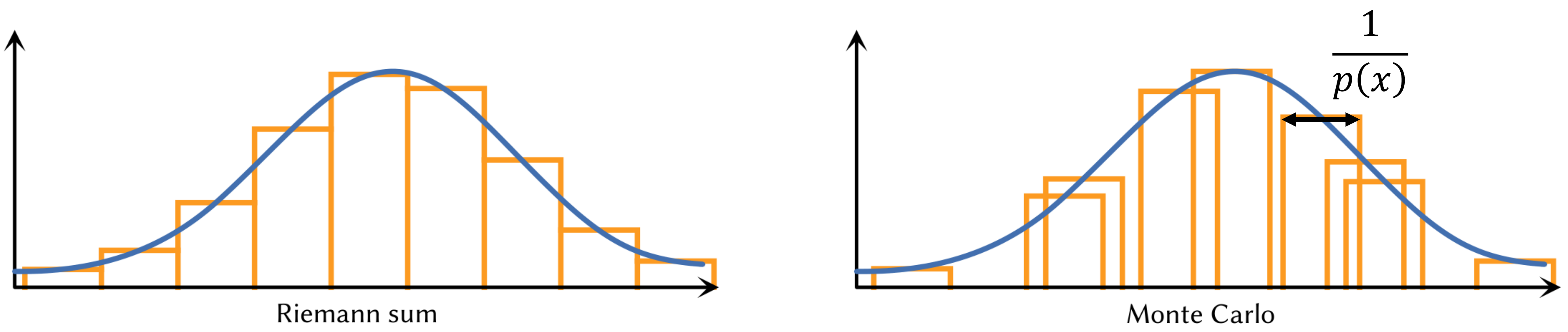
- Rigid structure:
  - Sample count and sample placement are linked
  - Adaptive sample placement is tricky
- Exacerbated in higher dimensions:
  - If we take 4 samples along each dimension

➔ Need to compute 4^d samples with prescribed positions that need careful tracking

Monte Carlo is like a randomized Riemann sum

$$\int_X f(x)dx \approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)}$$

probability density (PDF) of the sample



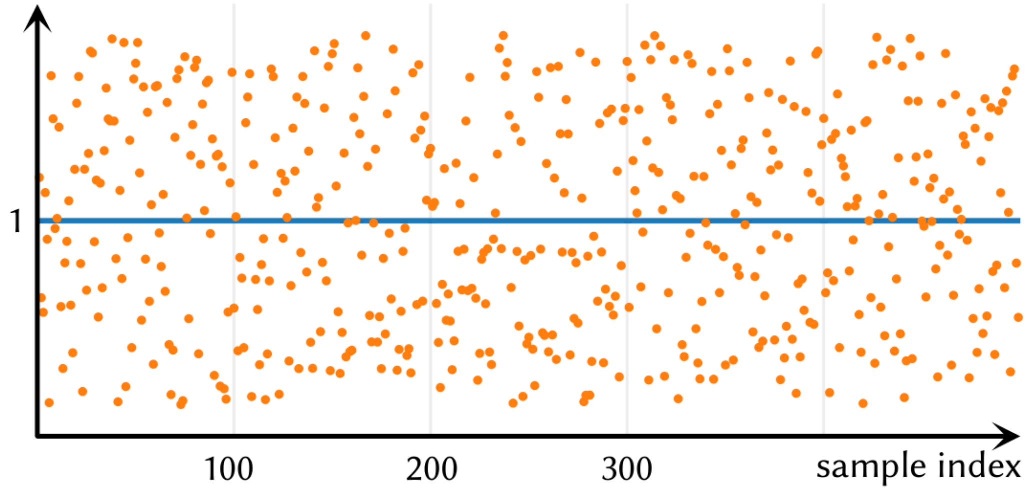
Why does it work? Law of large numbers

- The expected value of the Monte Carlo estimator is the desired integral

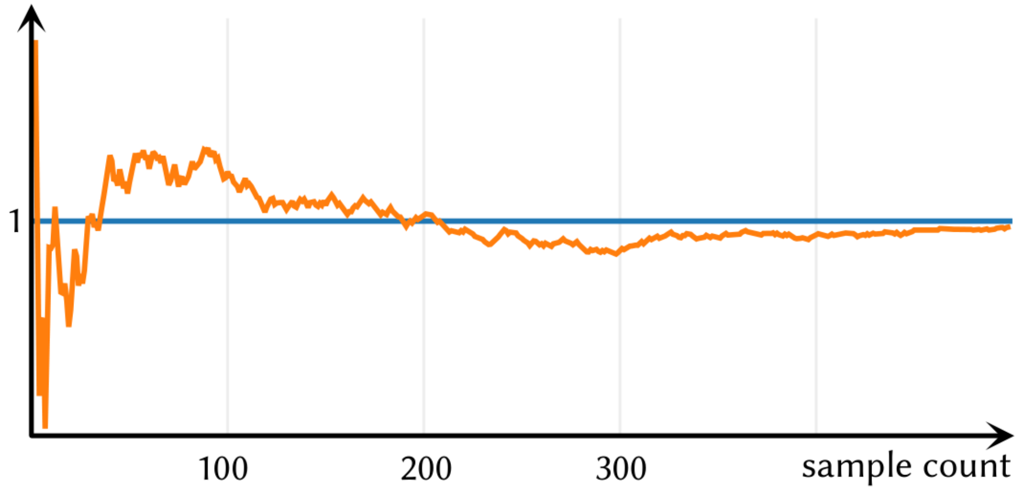
$$E\left[\frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)}\right] = E\left[\frac{f(x)}{p(x)}\right] = \int_X \frac{f(x)}{\cancel{p(x)}}\cancel{p(x)}dx$$

- The more samples we use, the more accurate the estimate (law of large numbers)

Individual sample values



Average after N samples

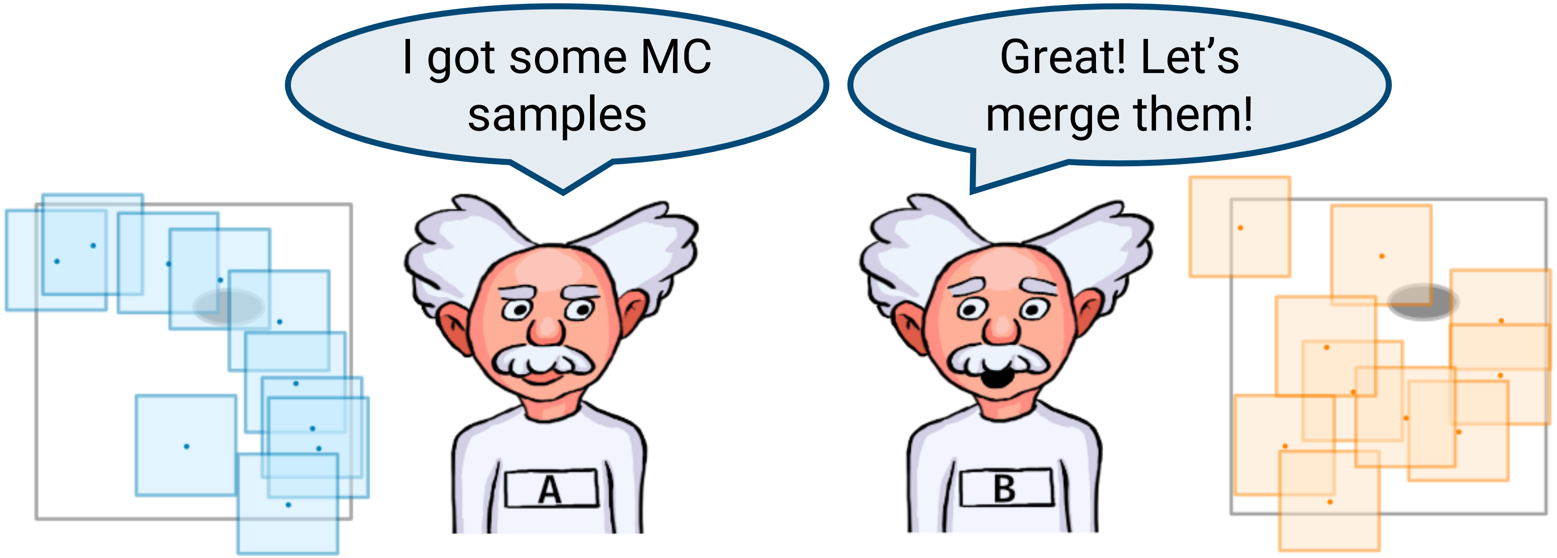


Monte Carlo integration is *flexible*, *scalable*, and *simple*

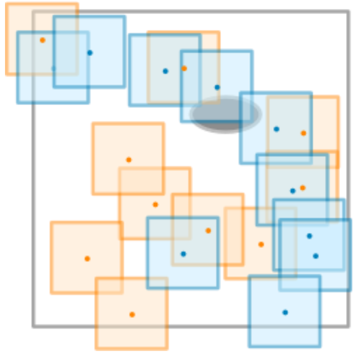
- Can add arbitrary numbers of samples at arbitrary times
- No bookkeeping:
  - Samples are independent
  - Samples processed one at a time

➔ Trivial to parallelize

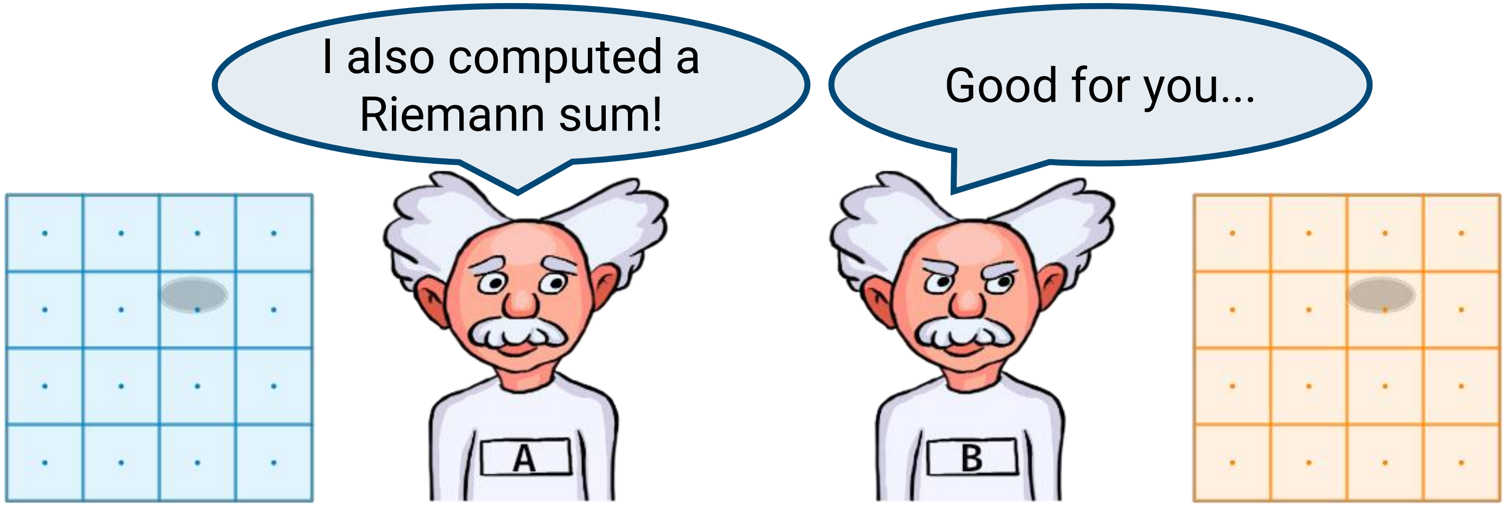
➔ Trivially extends to arbitrary number of dimensions



A trivial average of the results

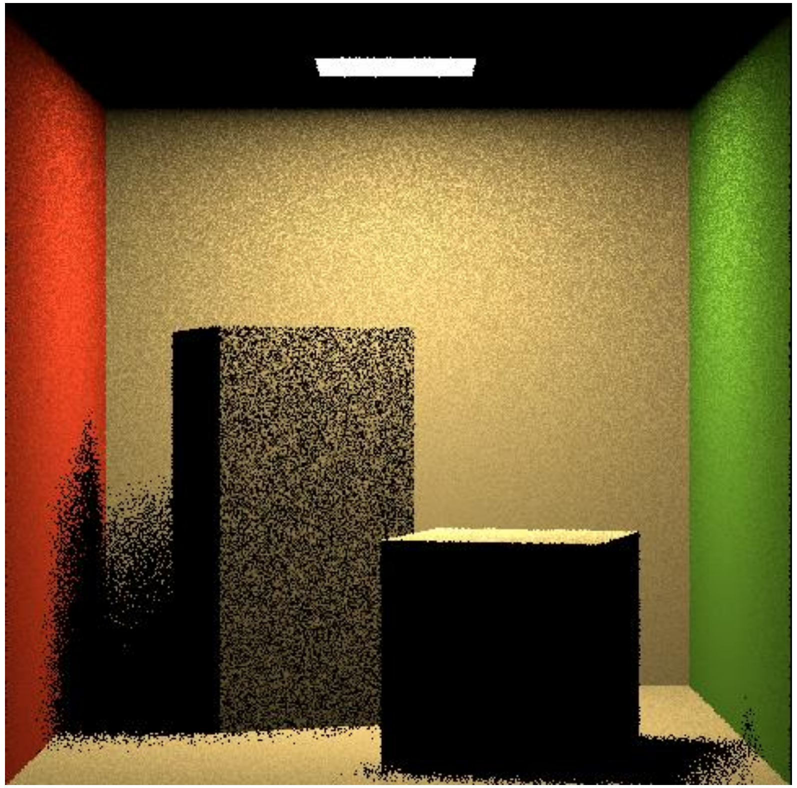


$$\frac{A+B}{2}$$

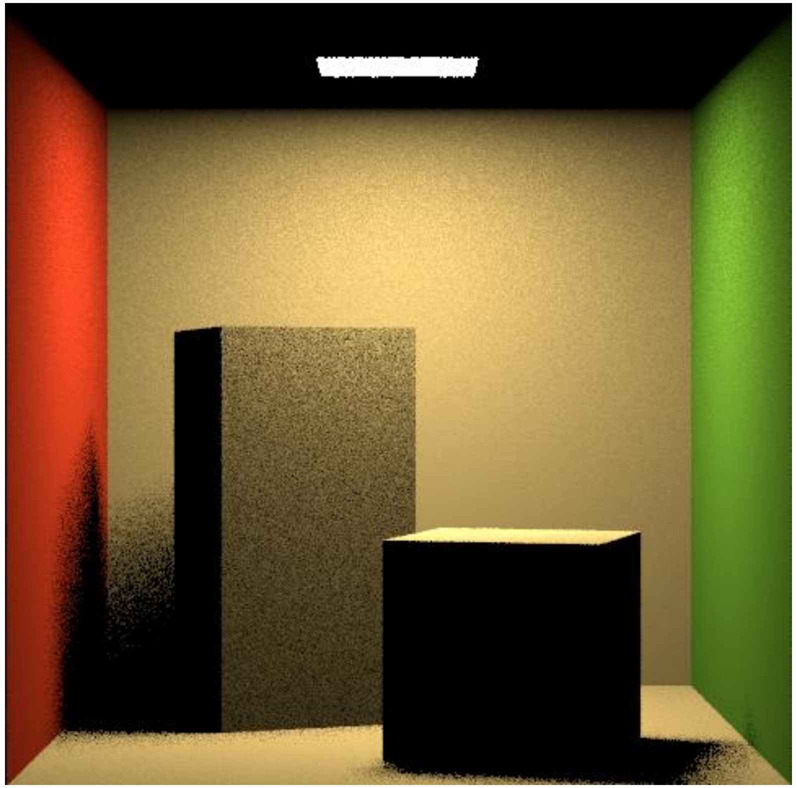


A waste of time.
(Without excessive coordination)

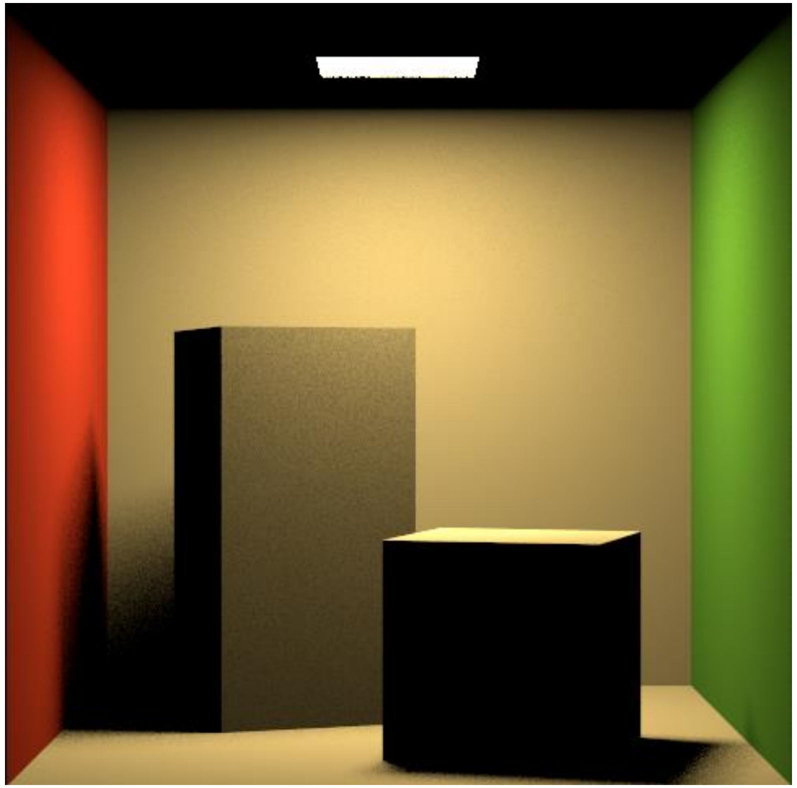
Error manifests as noise, reduces at $O\left(\frac{1}{\sqrt{n}}\right)$



1 sample per pixel

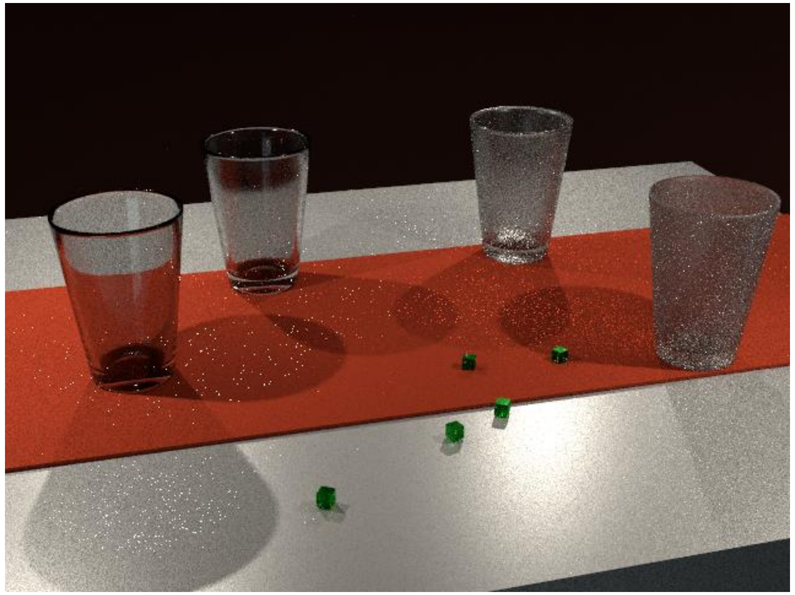


8 samples per pixel

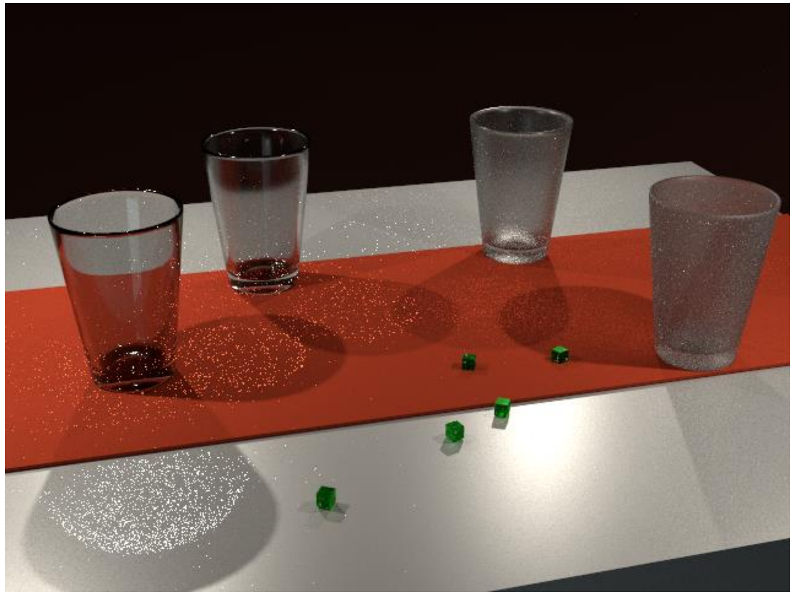


100 samples per pixel

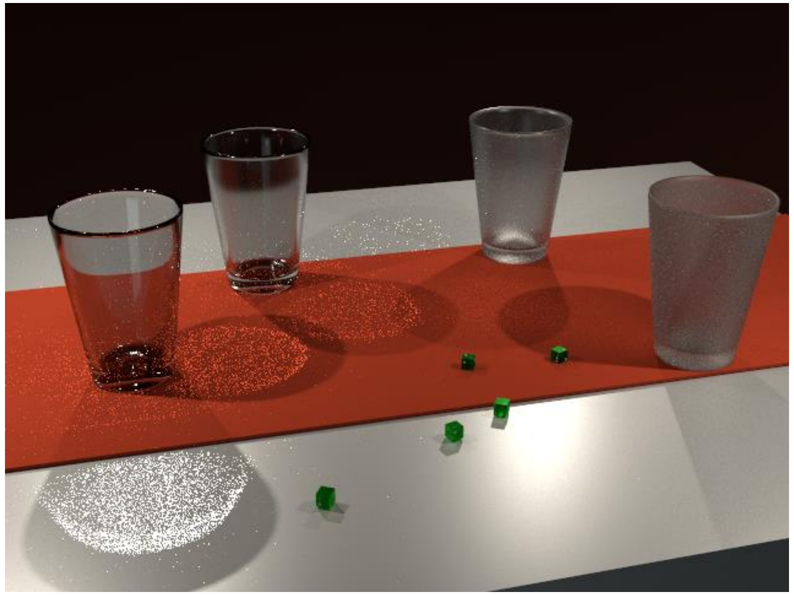
If our samples are not focused on important paths, we'll need more



100 samples per pixel



1000 samples per pixel

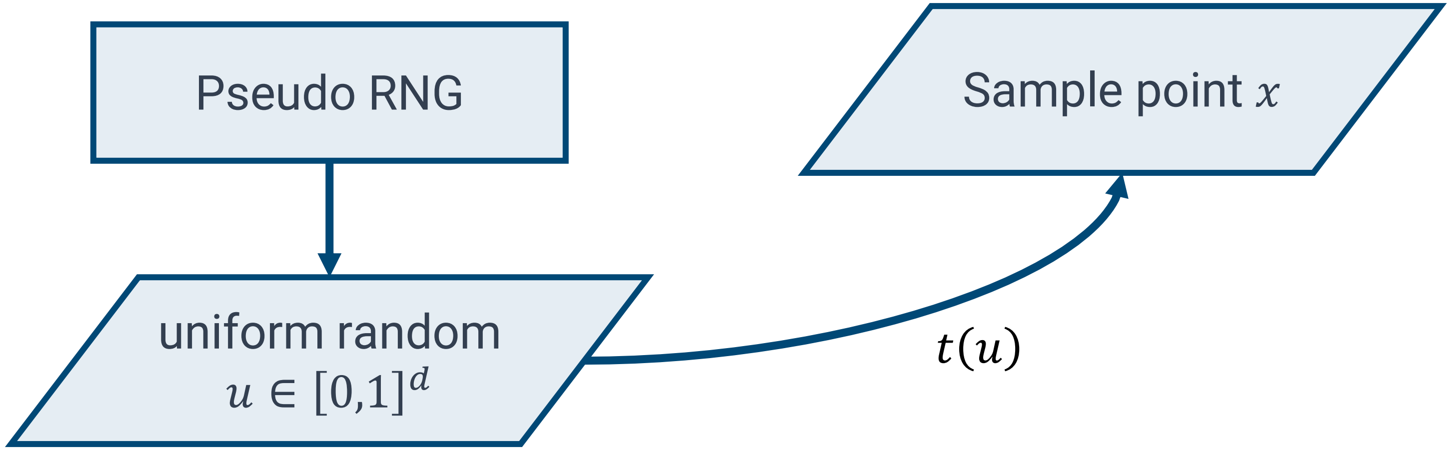


4000 samples per pixel

Here, the caustics are "hard to find" and remain noisy for a long time

How to sample

- Given: uniform random numbers from RNG
- Design a transformation $x = t(u)$
  - Surjective mapping to points in desired domain
- To achieve the desired probability density:
  - Find $t(u)$ such that $\frac{d}{du}t(u) = \frac{1}{p(x)}$
- Multiple methods exist, e.g.,
  - CDF inversion
  - Rejection sampling
  - Box-Muller transform



Example: uniformly sampling a triangle

- Input: uniform random numbers u, v

1. Map to barycentric coordinates
   - $s = 1 - \sqrt{u}$
   - $t = v\sqrt{u}$
2. Compute position from triangle vertices v_1, v_2, v_3
   - $x = sv_1 + tv_2 + (1 - s - t)v_3$

- Output: random point x on triangle with PDF $p(x) = \frac{1}{|A|}$

Example: sampling the cosine hemisphere

- Input: uniform random numbers u, v

1. Map to spherical coordinates
   - $\theta = \cos^{-1}\sqrt{v}$
   - $\phi = 2\pi\, u$
2. Map to cartesian coordinates
   - $\omega = \begin{pmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ \cos\theta \end{pmatrix}$
   - If you're clever: use $\cos\theta = \sqrt{v}$ and $\sin\theta = \sqrt{1-v}$

- Output: random direction ω with PDF $p(\omega) = \frac{\cos\theta}{\pi}$

For more on Monte Carlo: Realistic Image Synthesis lecture

- How to derive these sample transformations
- Improving efficiency, e.g., by
  - Combining multiple PDFs via Multiple Importance Sampling
  - Adapting PDFs on-the-fly while rendering to focus sampling on important parts
  - Adapting sample counts on-the-fly
  - Control variates

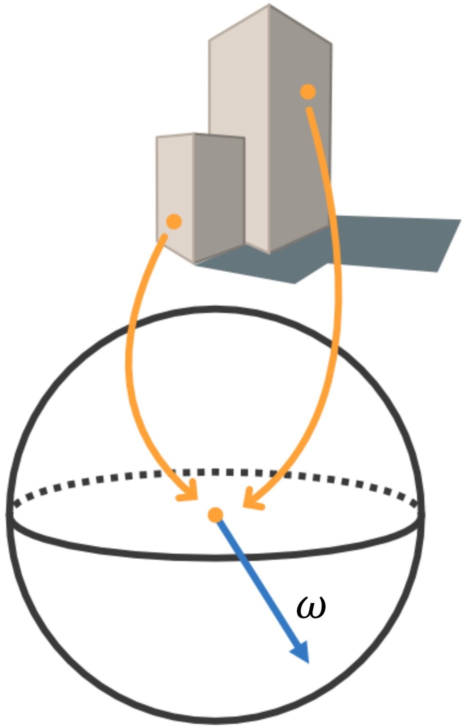
Reading materials

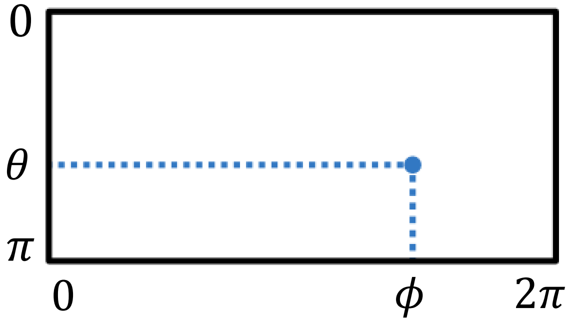
- *Chapter 2 of*: Eric Veach. Robust Monte Carlo methods for light transport simulation. PhD thesis. 1997. https://graphics.stanford.edu/papers/veach_thesis/thesis-bw.pdf
- *Section 2.2 of:* Pascal Grittmann. Rethinking multiple importance sampling for general and efficient Monte Carlo rendering. PhD thesis. 2023. https://randomrays.eu/content/publications/phd/thesis.pdf

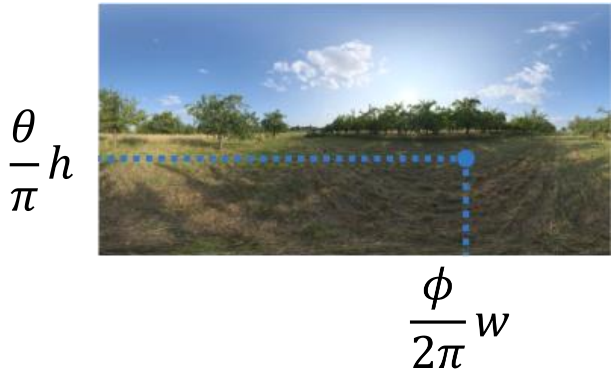
Rendering direct illumination with Monte Carlo

Environment maps query incoming radiance from an image

- $L_i(\omega, x) = I\left[\frac{\theta}{\pi}h,\ \frac{\phi}{2\pi}w\right]$ for all points x in the scene
- I is an image with height h and width w







(other mappings are possible, too, like cube maps)

A simple environment map estimator

- Goal: compute the reflected radiance from the rendering equation:

$$L_\mathrm{r} = \int_\Omega f_r L_i |\cos\theta_i| d\omega_i$$

- where L_i is given by the environment map value for direction ω_i

1. Sample a direction ω_i (e.g., cosine hemisphere or uniform sphere sampling)
2. Compute the Monte Carlo estimate

$$\langle L_\mathrm{r}\rangle = \frac{f_r L_i |\cos\theta_i|}{p(\omega_i)}$$

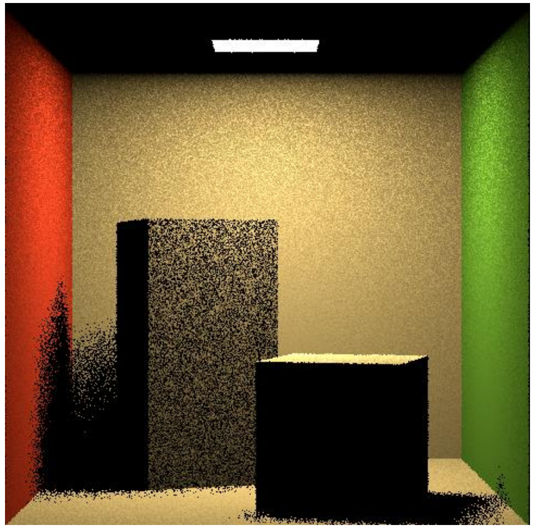
3. Repeat n times and average

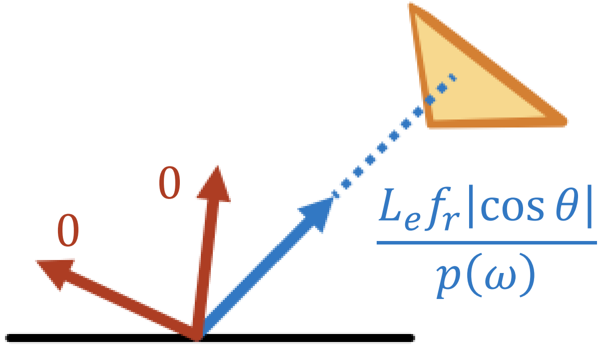
$$L_r \approx \frac{1}{n}\sum_{k=1}^{n} \langle L_\mathrm{r}\rangle_k$$

Area lights can be handled the same way

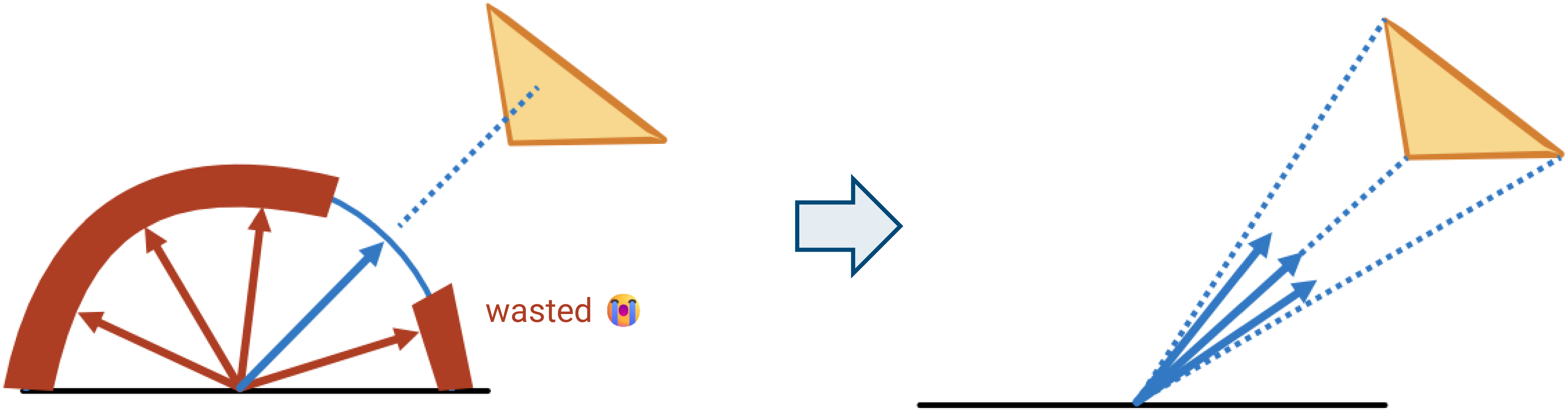
1. Sample a direction ω_i
2. Trace a ray and compute

$$\langle L_\mathrm{r} \rangle = \begin{cases} 0, & \text{if no light hit} \\ \dfrac{L_e(y) f_r |\cos\theta_i|}{p(\omega_i)}, & \text{if } y \text{ is on a light} \end{cases}$$

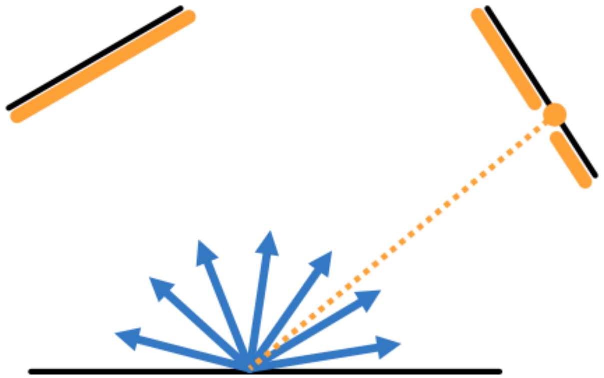




Can we sample only directions towards a light?



Area integral version of the rendering equation

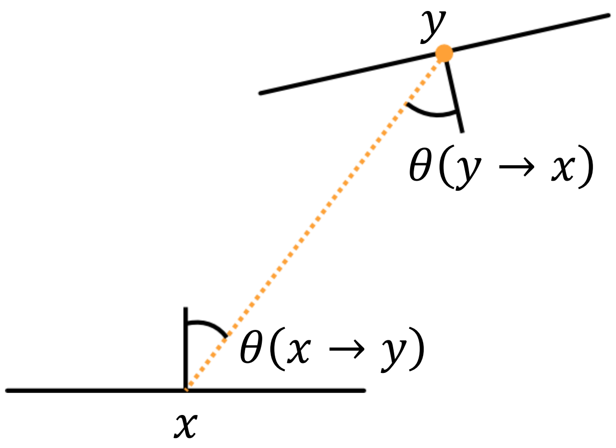


- We can do a change of variables
  - Integral over directions ➔ Integral over visible area
- Substitute $\omega = \overrightarrow{xy} = \frac{y-x}{\|y-x\|}$

$$\int_\Omega f(\omega)d\omega = \int_A f(\overrightarrow{xy})\, V(y)\, \frac{d\omega}{dy} dy$$

- Introduces a Jacobian determinant

$$\frac{d\omega}{dy} = \frac{\cos\theta(y \to x)}{\|x-y\|^2}$$



- Applied to the rendering equation

$$L_r(x,\omega_o) = \int_A L_i(x,\overrightarrow{xy})\, f_r(x,\omega_o,\overrightarrow{xy}) \cos\theta(x \to y)\, V(y)\, \frac{\cos\theta(y \to x)}{\|x-y\|^2} dy$$

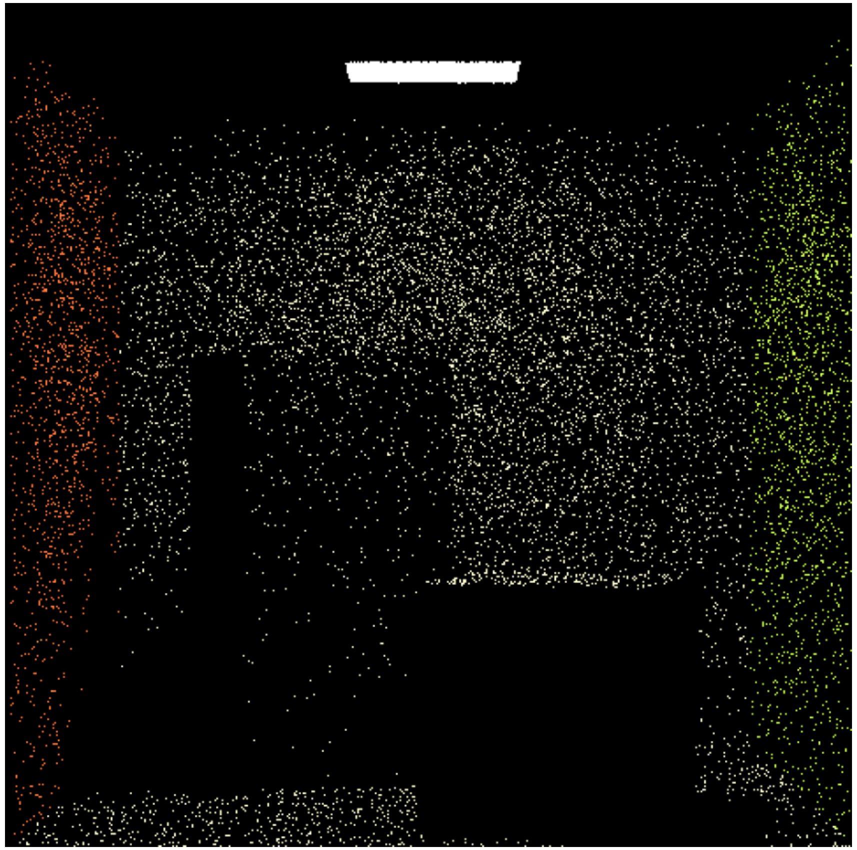
Applying Monte Carlo integration to area lights

1. Uniformly sample point on the light
   - $p(y) = \frac{1}{|A|}$
   - (Because PDFs must integrate to one just like discrete probabilities must sum to one)
2. Trace a shadow ray to evaluate $V(x, y)$
3. Compute the remaining terms in the integrand
4. Divide the result by the PDF
5. Repeat n times to obtain the Monte Carlo estimate

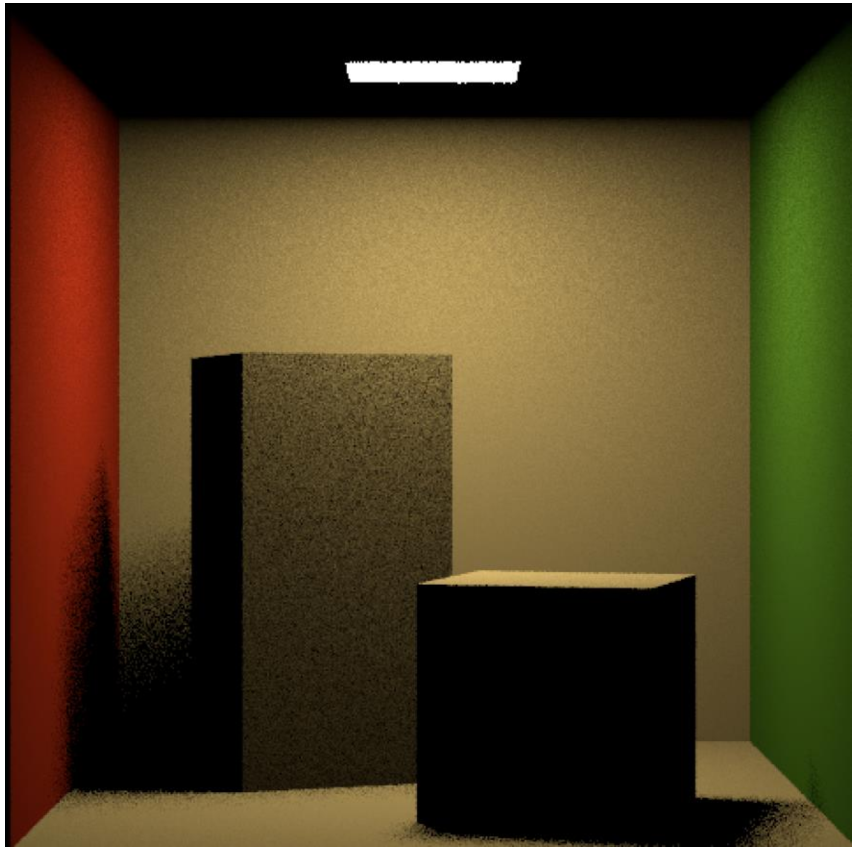
$$\sum_{k=1}^{n} L_e\, f_r \cos\theta(x \to y_k)\, V(x, y_k) \frac{\cos\theta(y_k \to x)}{\|x - y_k\|^2} \frac{|A|}{n}$$

$$|A| = \frac{1}{p(y)}$$

Area sampling is often more effective

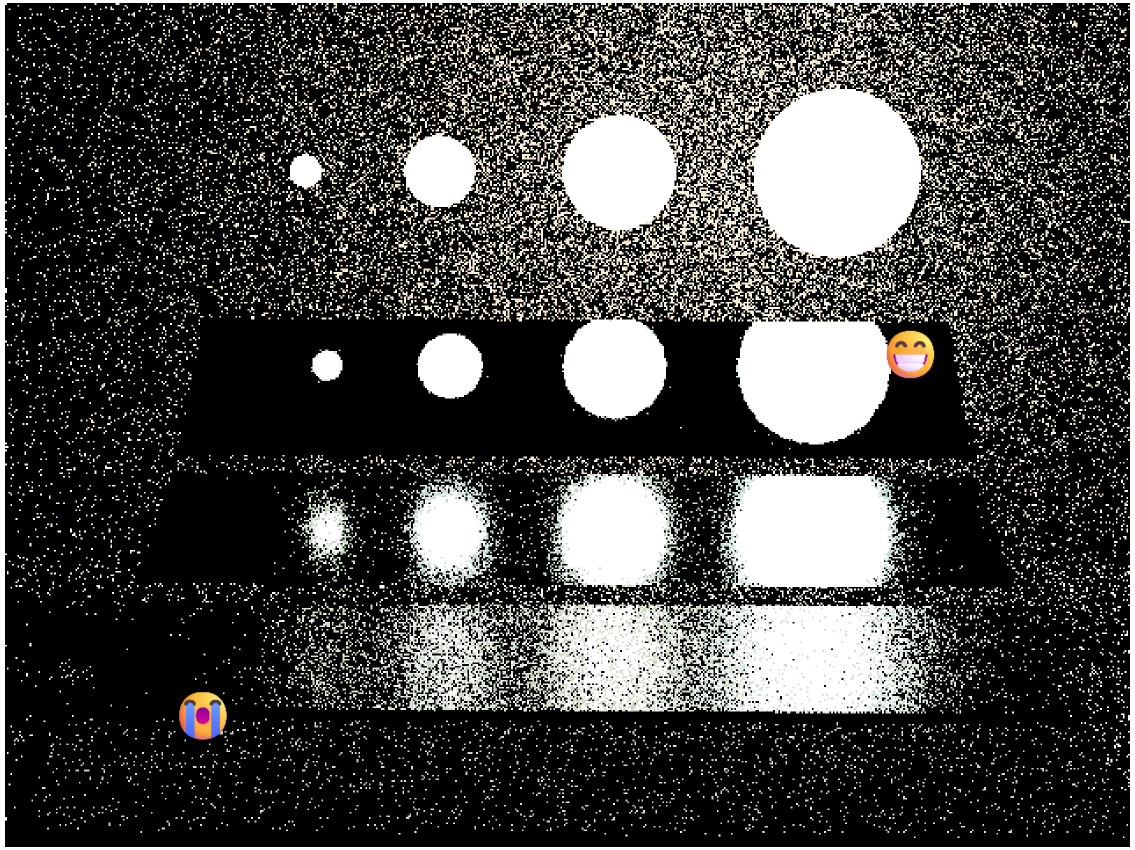


Direction sampling

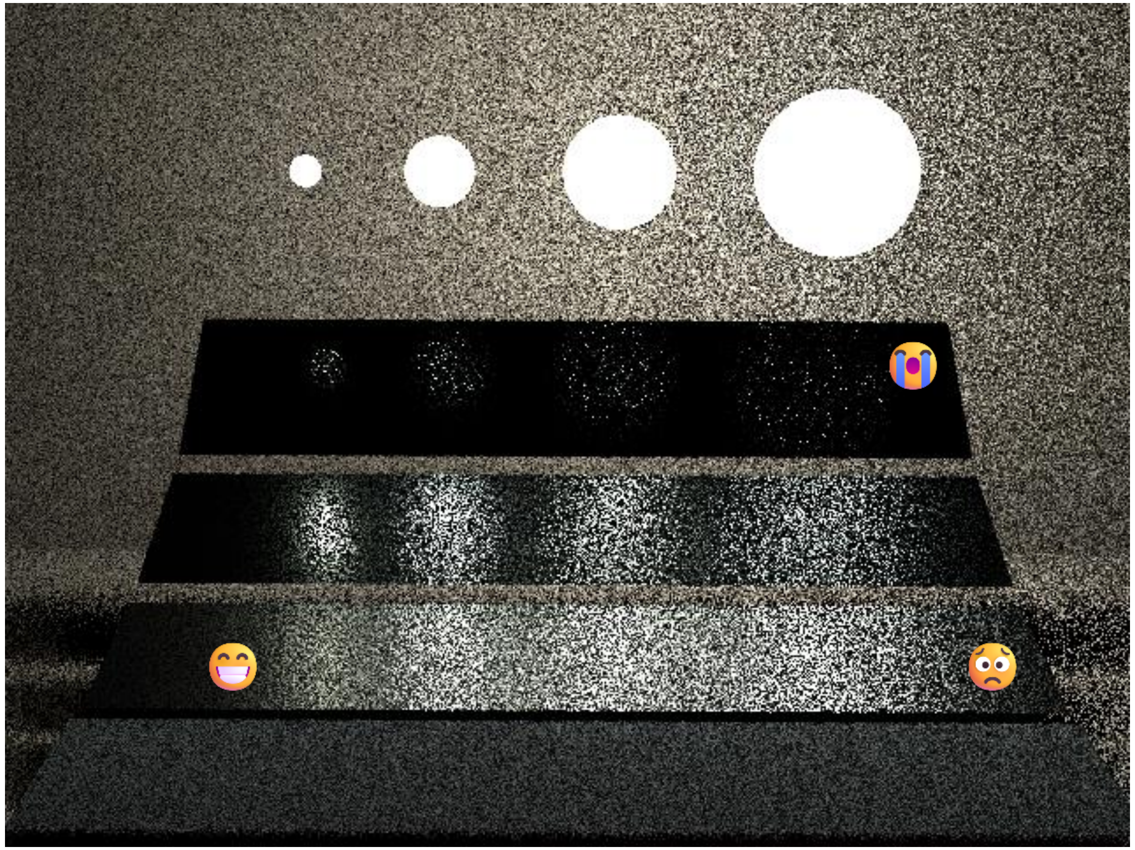


Area sampling

Direction sampling is more effective for large lights and glossy surfaces



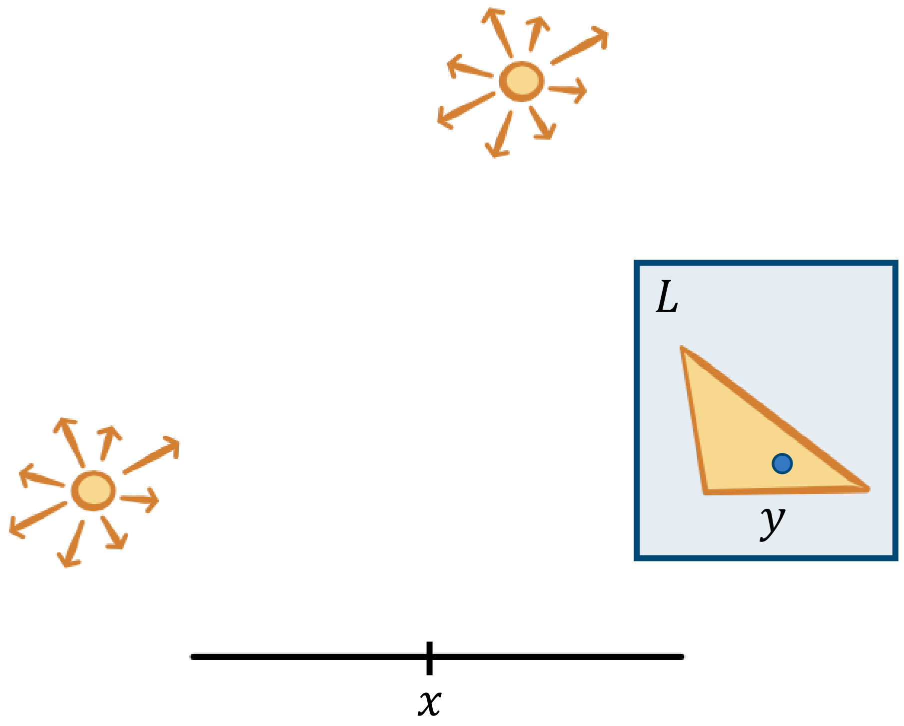
Direction sampling



Area sampling

Handling many light sources

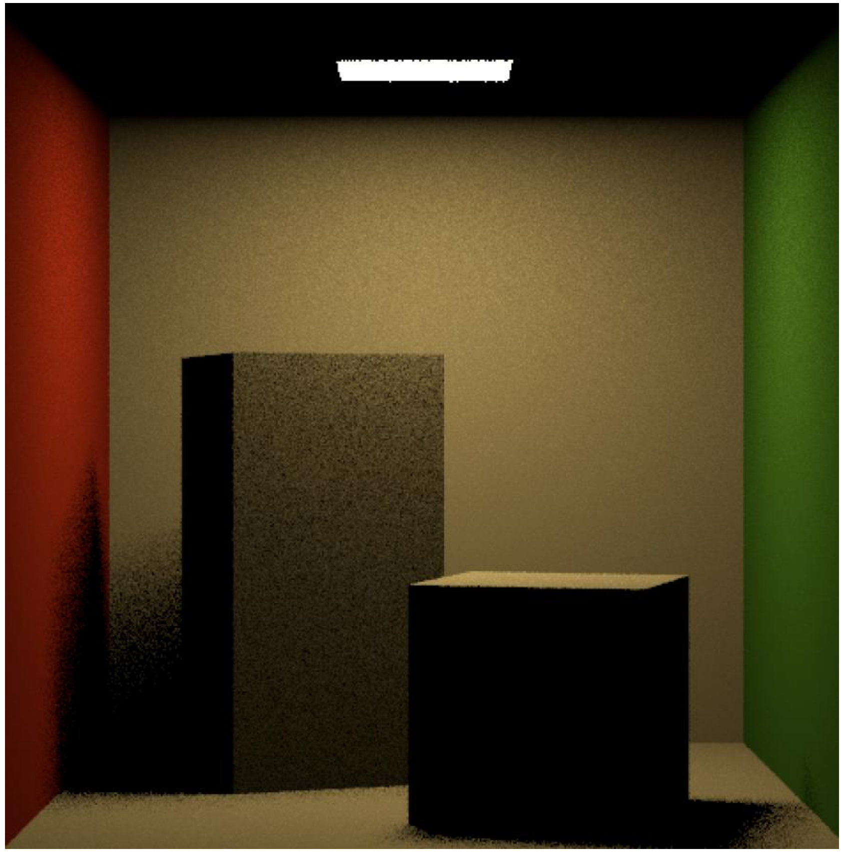
1. Select a random light with discrete probability $P(L)$
   - e.g., uniform, or proportional to flux of the light
2. Sample point $y \in L$
3. Compute estimate as with a single light
4. Divide by $P(L)$

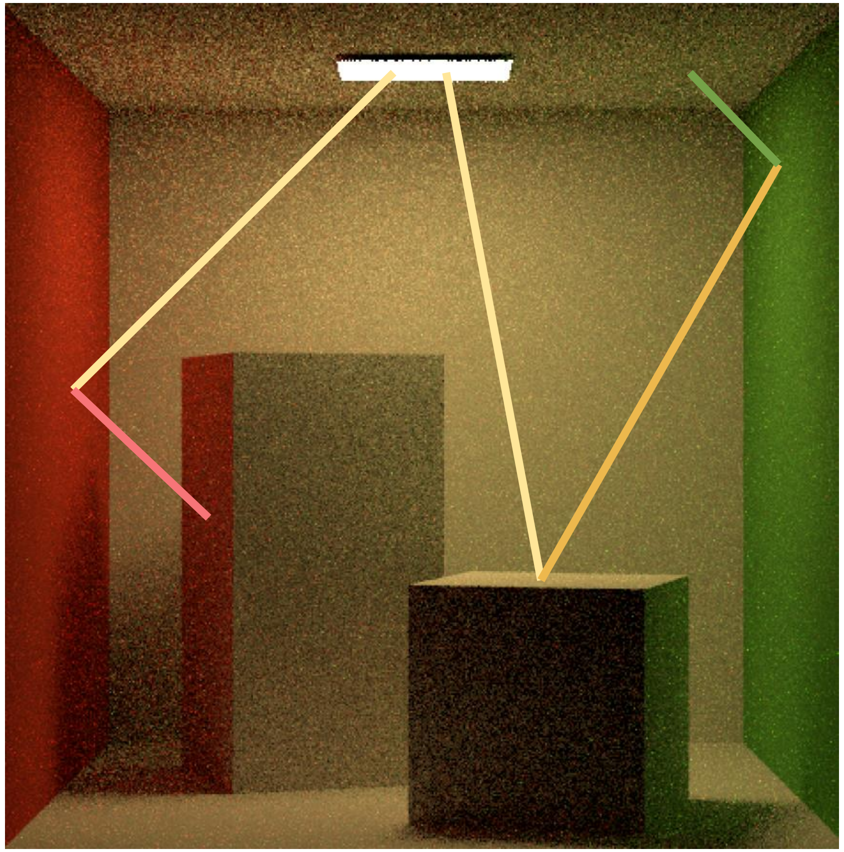


- Many ways to improve on this... But not during this course 😥

Path tracing

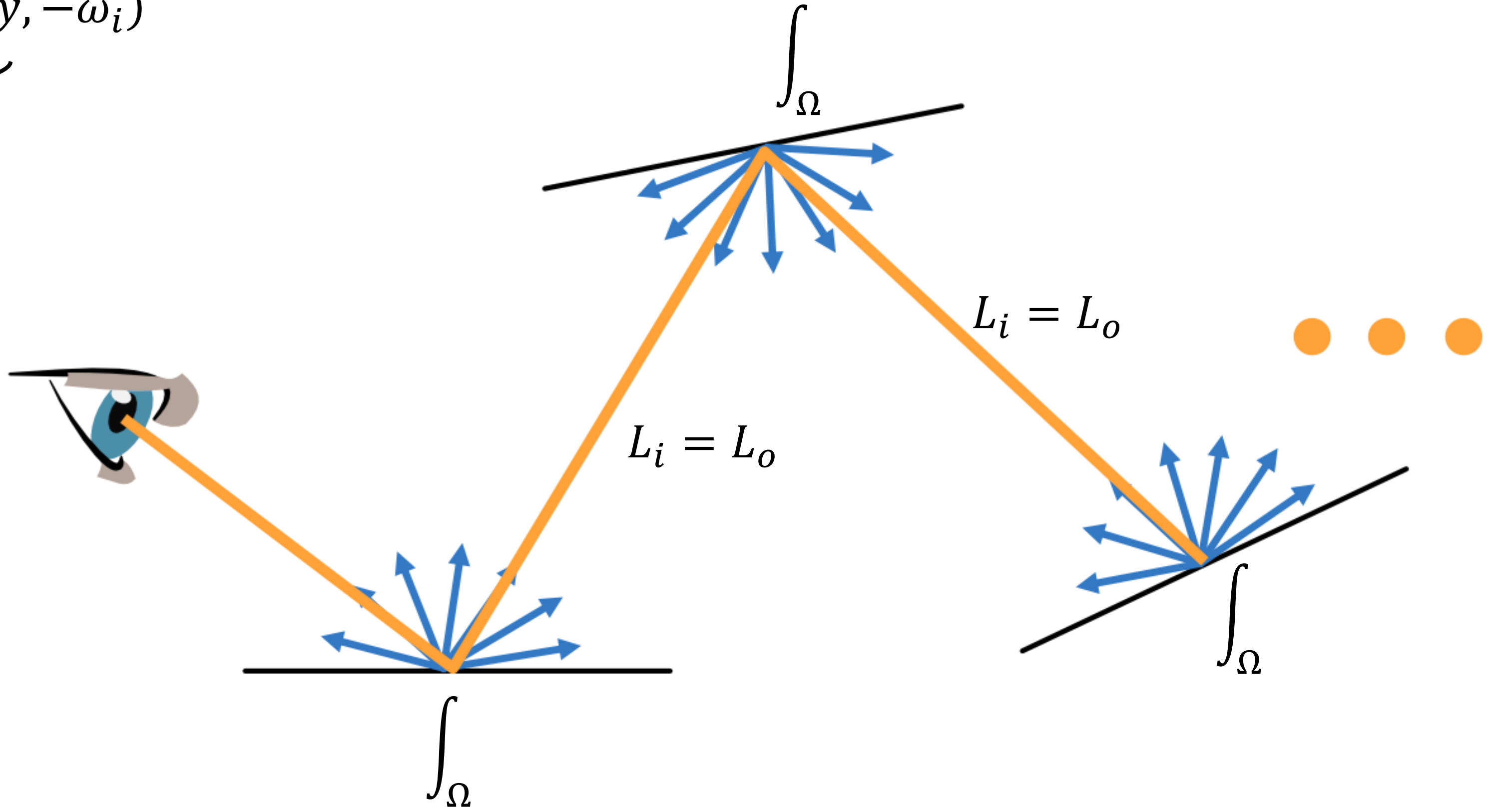
What about indirect (“global”) illumination?





The rendering equation is recursive

$$L_{\mathrm{o}}(x,\omega_{\mathrm{o}}) = L_{\mathrm{e}}(x,\omega_{\mathrm{o}}) + \int_{\Omega} \underbrace{L_{\mathrm{i}}(x,\omega_i)}_{= L_o(y,-\omega_i)} f_{\mathrm{r}}(x,\omega_{\mathrm{i}},\omega_{\mathrm{o}}) \cos\theta_i \, d\omega_{\mathrm{i}}$$

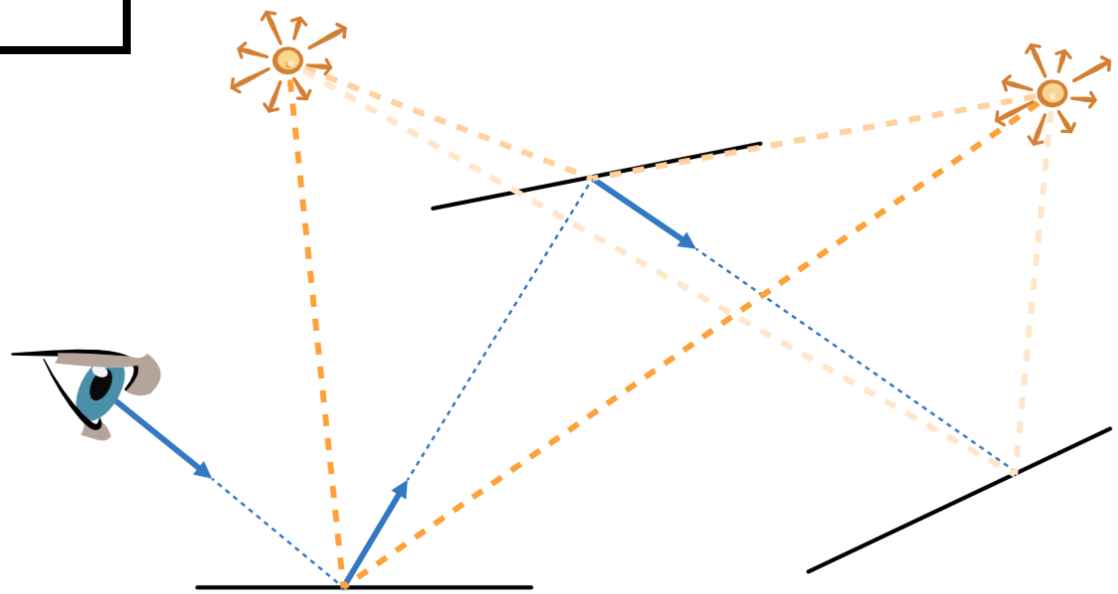


A simple path tracer

1. Trace ray from the camera
2. Add emitted radiance L_e
3. Compute direct illumination via shadow rays
4. Sample direction ω_i to continue the path
5. Trace ray to find the next point

for directly visible lights only, so we don't count it twice!

terminate if maximum depth reached



For more on path tracing...

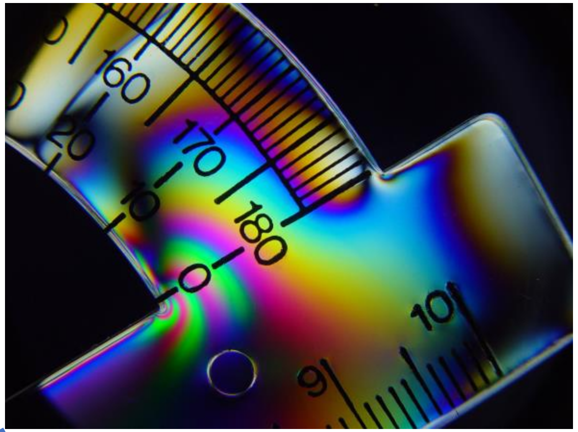
- Like tracing paths from lights and camera,
- better sampling procedures,
- or adaptive sampling

- ... You’ll have to wait for the Realistic Image Synthesis lecture 😞

Material models for glass and metal

The refractive index

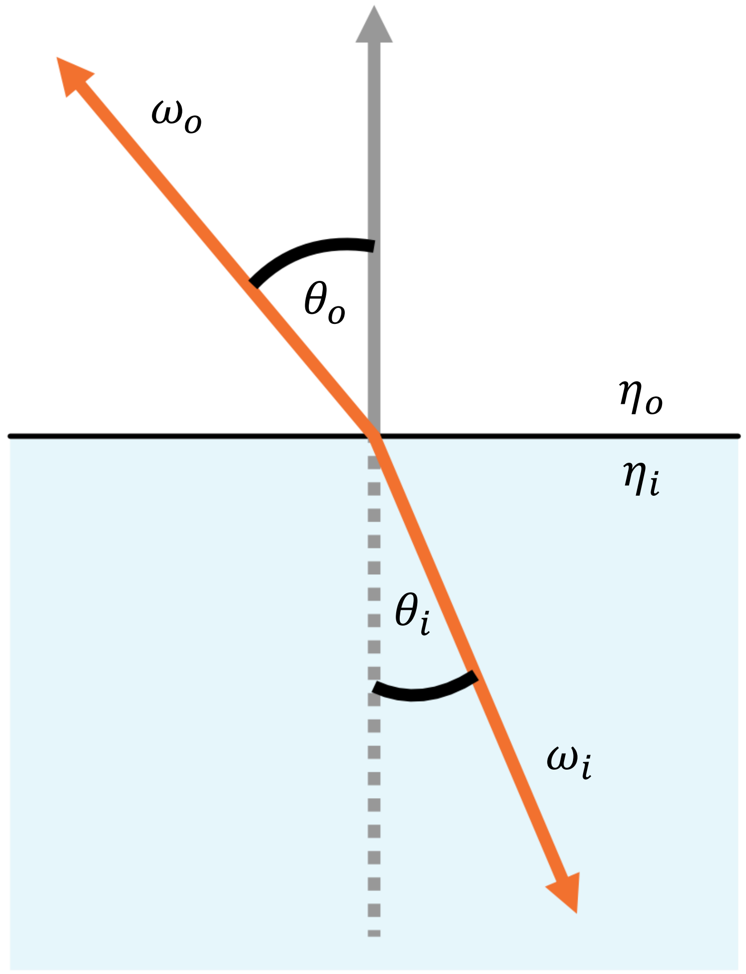
- Speed of light in vacuum $c = 299{,}792{,}458\,\frac{m}{s}$
- Refractive index: relative speed of light inside medium
  - $\eta = \frac{c}{v}$
- Typical values
  - Vacuum: $\eta = 1$
  - Air: $\eta = 1.000293$
  - Water at 20°C: $\eta = 1.33$
  - Window glass: $\eta = 1.52$



- Depends on temperature, wavelength, mechanical stress, polarization, ...

Snell's law – refraction at an optical boundary

$$\frac{\sin\theta_1}{\sin\theta_2} = \frac{\eta_2}{\eta_1}$$

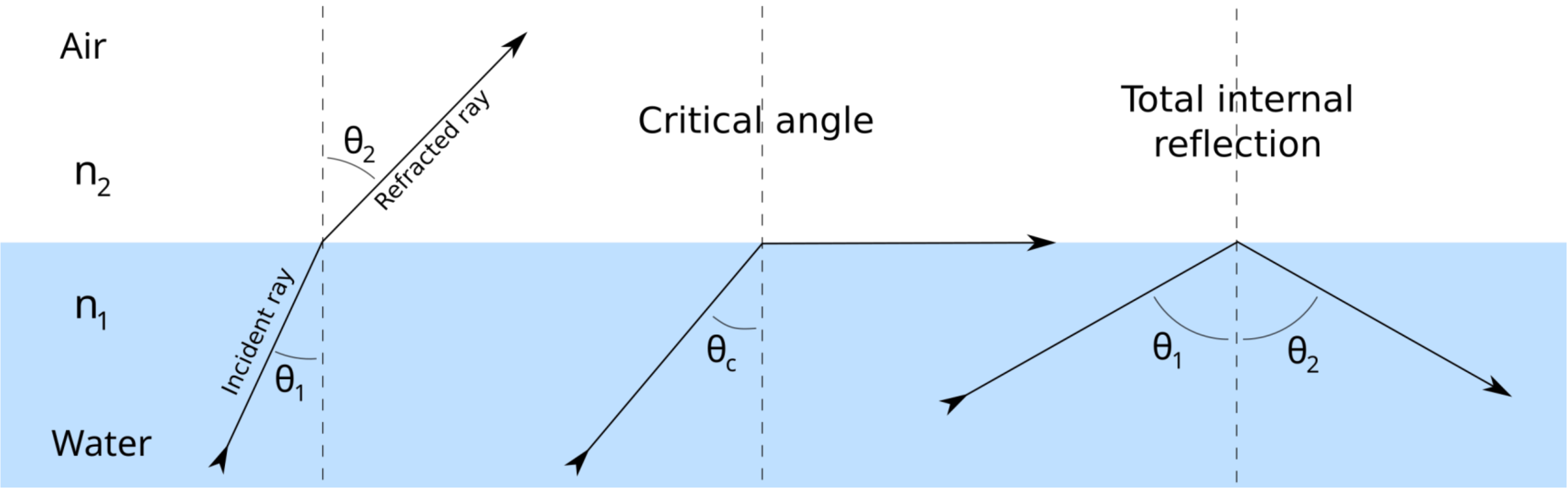


Computing the refracted direction – Step 1

1. $\cos\theta_i = \langle \omega_i, n \rangle$
2. $\sin^2\theta_i = 1 - \cos^2\theta_i$
   - Using: $\cos^2 x + \sin^2 x = 1$
3. $\sin^2\theta_o = \left(\frac{\eta_i}{\eta_o}\right)^2 \sin^2\theta_i$
   - Using Snell's law

Total internal reflection (TIR)

- If $\sin^2\theta_o = \left(\frac{\eta_i}{\eta_o}\right)^2 \sin^2\theta_i \geq 1$
- Refracted direction does not exist, all light is reflected

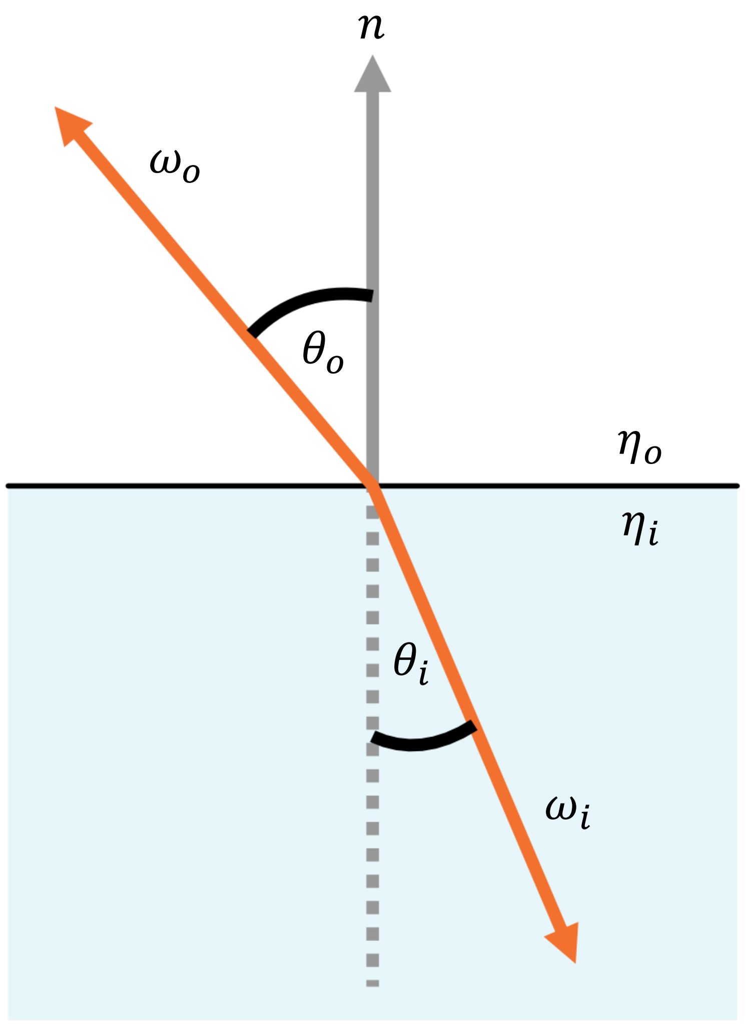


$\left(\frac{\eta_i}{\eta_o}\right)^2 \sin^2\theta_i < 1$ $\left(\frac{\eta_i}{\eta_o}\right)^2 \sin^2\theta_i = 1$ $\left(\frac{\eta_i}{\eta_o}\right)^2 \sin^2\theta_i > 1$

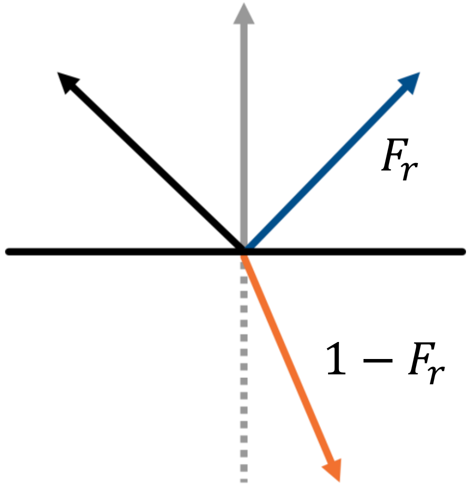
Computing the refracted direction – Step 2

- Assuming TIR does not occur

1. $\cos\theta_o = \sqrt{1 - \sin^2\theta_o}$
   - Using: $\cos^2 x + \sin^2 x = 1$
2. $\omega_o = -\frac{\eta_i}{\eta_o}\omega_i + \left(\frac{\eta_i}{\eta_o}\cos\theta_i - \cos\theta_o\right)n$
   - Using Snell's law again and some trigonometry



Fresnel equations

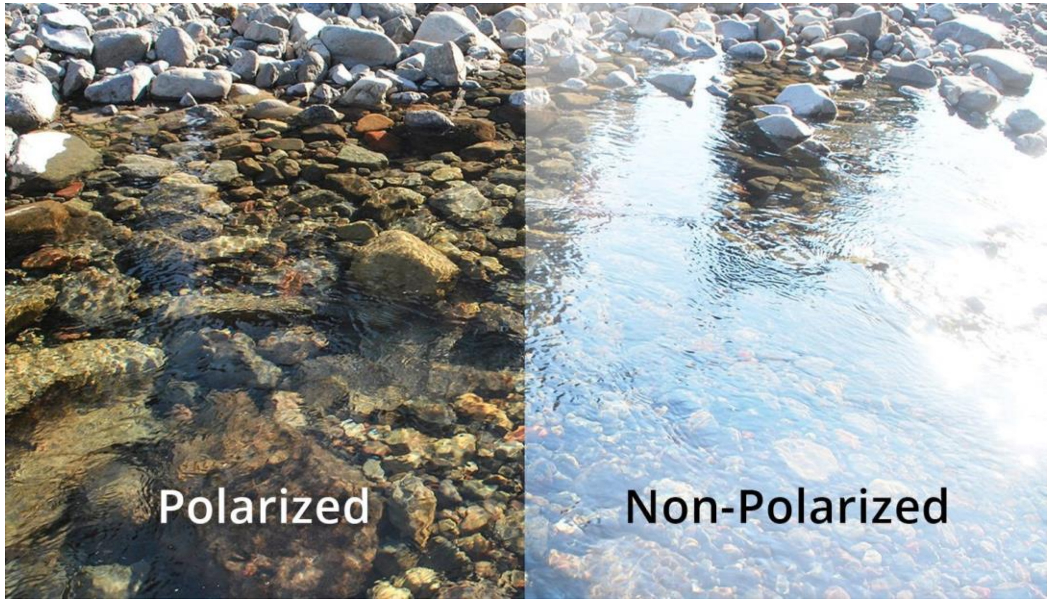


- How much light is reflected? How much refracted?

➔ Quantified by the Fresnel coefficient F_r

- Depends on incident angle and polarization

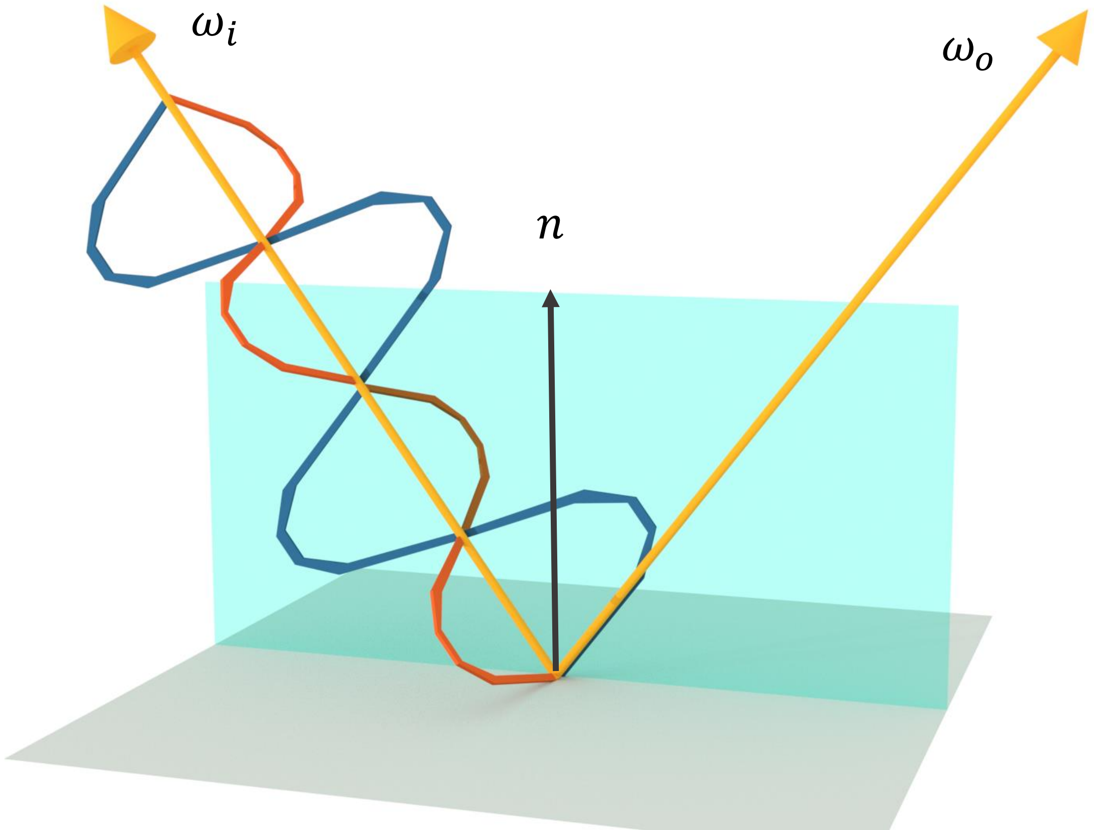
Less reflection when looking straight down



Less reflection through polarized sunglasses

S and P Polarization on the plane of incidence

- Light is a transverse wave (oscillates perpendicularly to direction of travel)
- Polarization is the orientation of this oscillation
- Can be expressed as linear combination of
  - Parallelly polarized light
    - (“P” from German “parallel”)
  - Orthogonally polarized light
    - (“S” from German “senkrecht”)



Fresnel term for dielectrics

- Assuming unpolarized light,

$$F_r = \frac{r_p^2 + r_s^2}{2}$$

- Where

$$r_p = \frac{\eta_o \cos\theta_i - \eta_i \cos\theta_o}{\eta_o \cos\theta_i + \eta_i \cos\theta_o}$$

$$r_s = \frac{\eta_i \cos\theta_i - \eta_o \cos\theta_o}{\eta_i \cos\theta_i + \eta_o \cos\theta_o}$$

Fresnel term for conductors

- Conductors reflect F_r and absorb the rest
- Assuming unpolarized light,

$$F_r = \frac{r_p^2 + r_s^2}{2}$$

- Where

$$r_p^2 = \frac{(\eta^2 + \kappa^2)\cos^2\theta - 2\eta\cos\theta + 1}{(\eta^2 + \kappa^2)\cos^2\theta + 2\eta\cos\theta + 1}$$

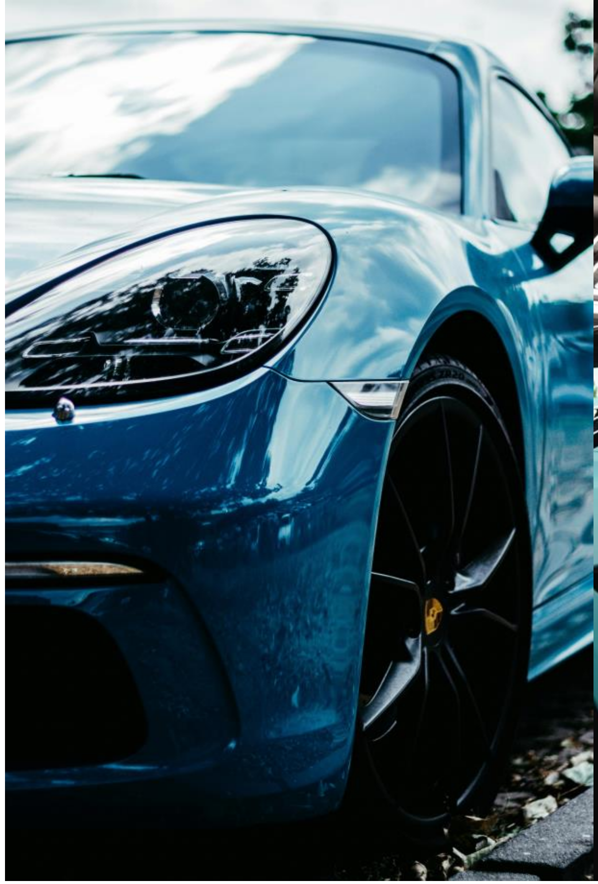
$$r_s^2 = \frac{(\eta^2 + \kappa^2) - 2\eta\cos\theta + \cos^2\theta}{(\eta^2 + \kappa^2) + 2\eta\cos\theta + \cos^2\theta}$$

- κ is the absorption coefficient and η the refractive index

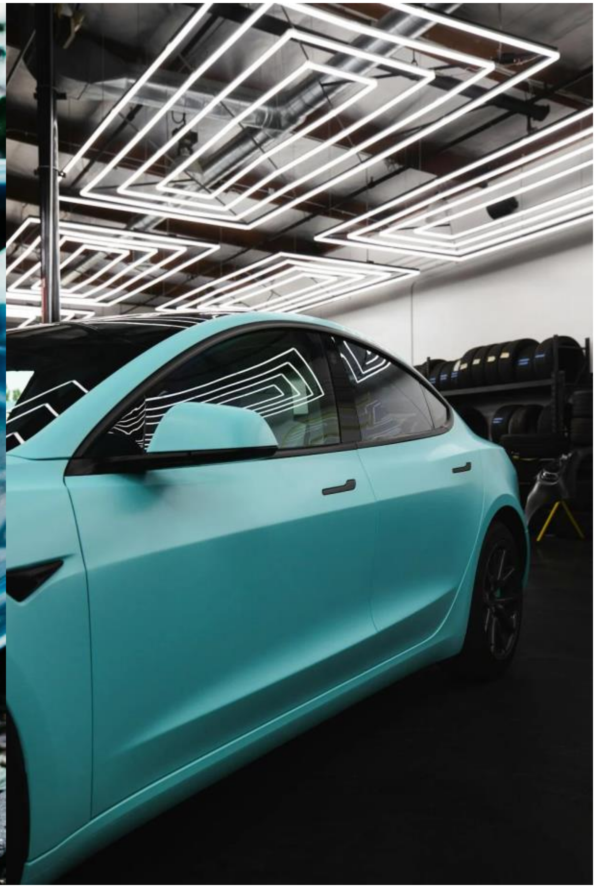
	η	κ
Gold	0.370	2.820
Silver	0.177	3.638
Copper	0.617	2.630
Steel	2.485	3.433

Microfacet BSDFs

Smooth versus rough surface appearances

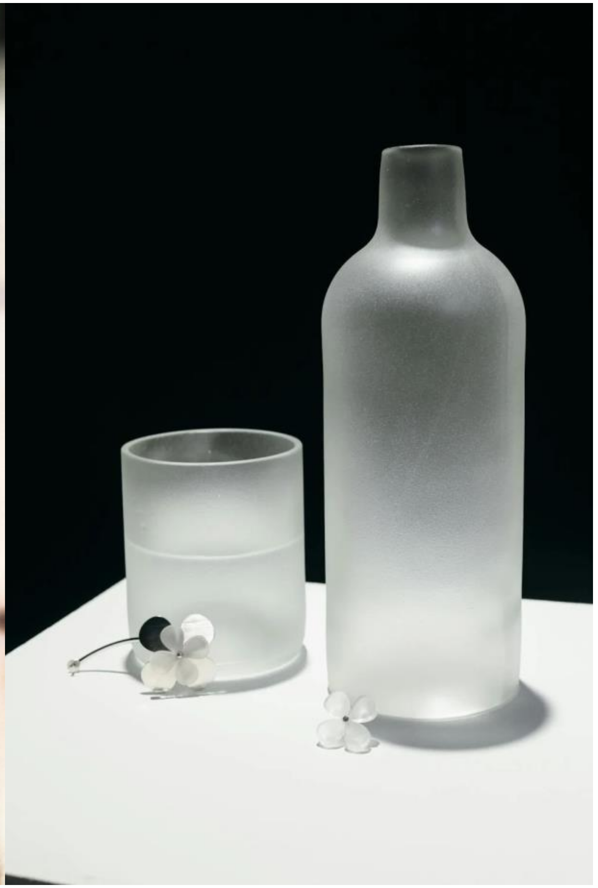


Smooth



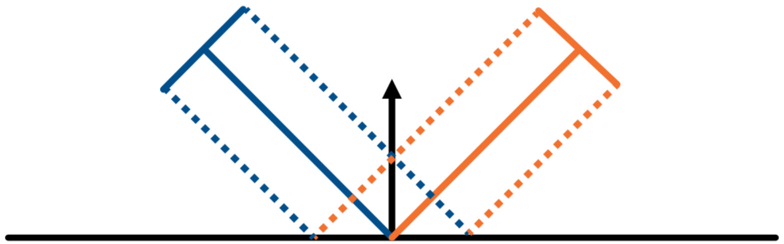
Rough

Smooth

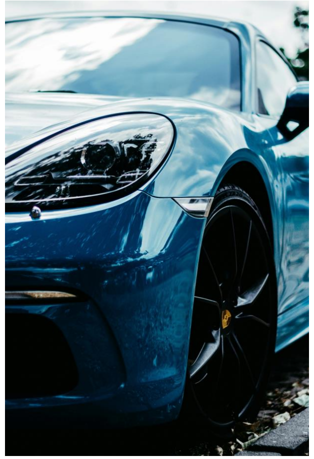


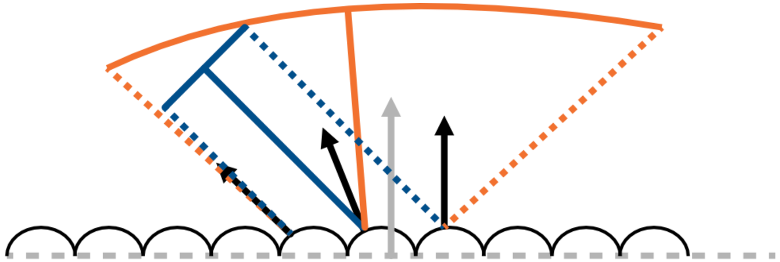
Rough

Microscopic detail – The microfacet model

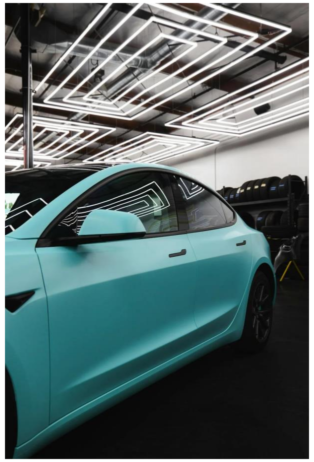


Smooth surface – perfect reflection





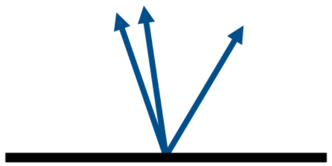
Rough surface – “wider” reflection



- These “microfacets” are too small to be seen individually
- We don’t model them explicitly (too expensive)
- Instead: Model their average, statistical effect as part of the BSDF

We describe microfacet models by the distribution of normals

Uniformly spread normals
➔ Rough surface

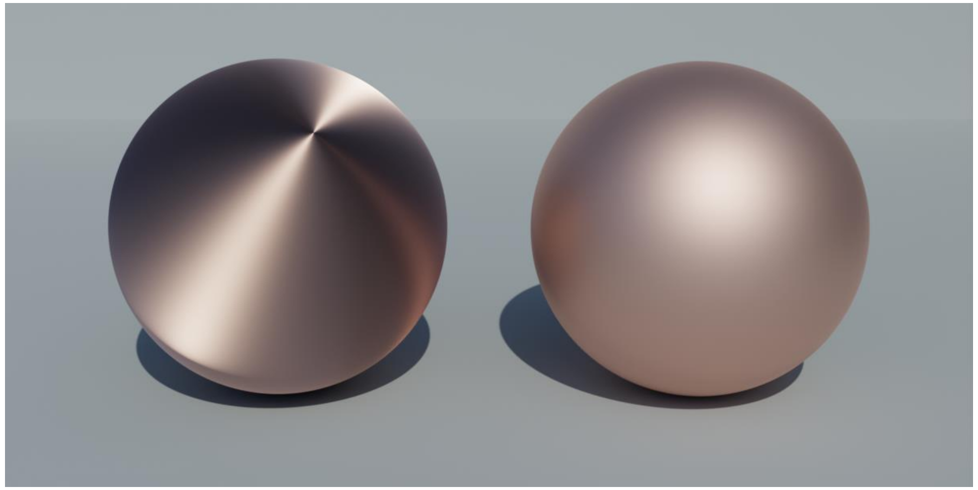


More focused normals
➔ Smooth surface

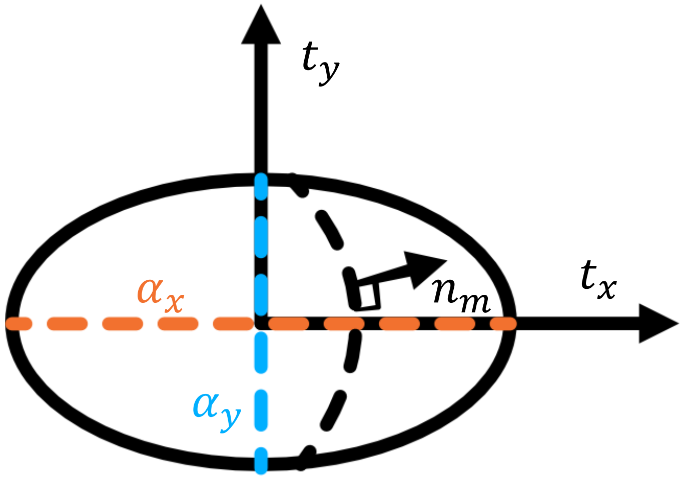
- Many different models possible.
- Differ, e.g., in the choice of *primitive geometry* assumed for the microfacets
  - e.g., V-grooves, spheres, ellipsoids, ...

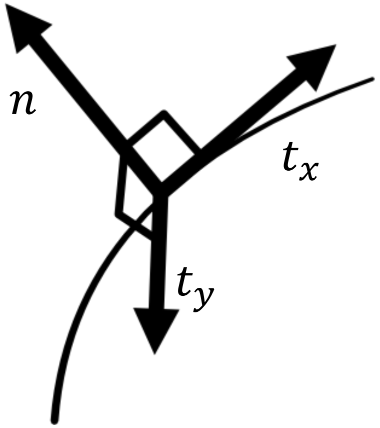
The Trowbridge-Reitz normal distribution function (NDF)

- Assumes ellipsoidal microgeometry
- Parameters α_x and α_y control the ellipsoid shape
  - Reciprocal size of the ellipsoid along each *surface tangent*
  - Larger ellipsoids → smaller α_x / α_y → smoother surface
  - *Anisotropic* appearance if $\alpha_x \neq \alpha_y$



Anisotropic Isotropic





Expressed in *shading space*
i.e., normal n is the z-axis

The Trowbridge-Reitz normal distribution function (NDF)

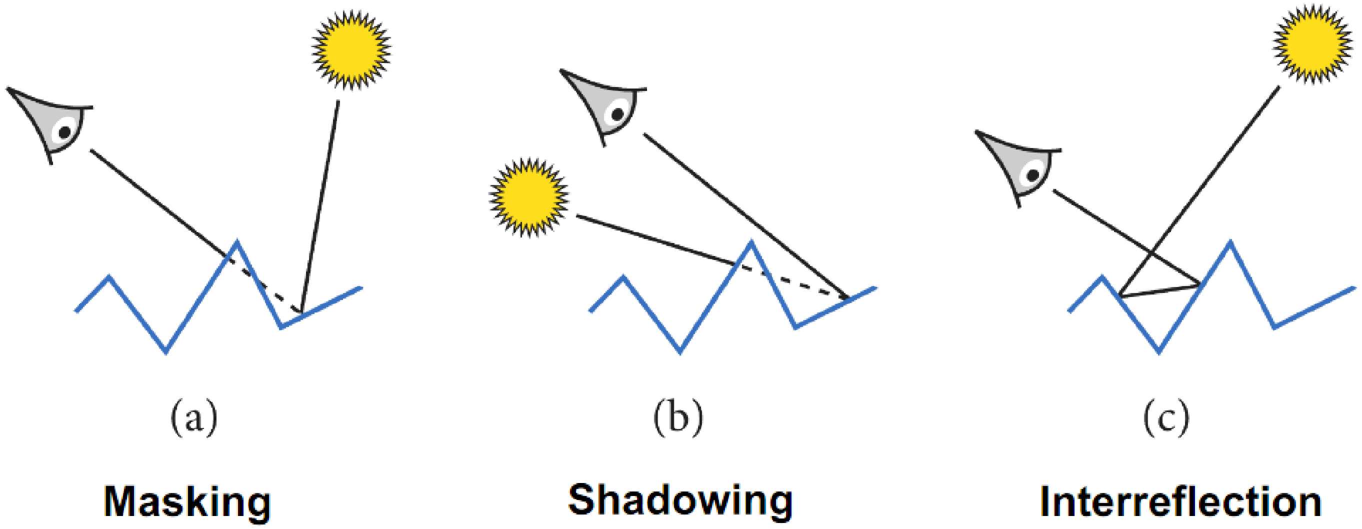
- The probability density of microfacet normal n_m

$$D(\omega_\mathrm{m}) = \frac{1}{\pi\,\alpha_x\,\alpha_y\,\cos^4\theta_\mathrm{m}\left(1+\tan^2\theta_\mathrm{m}\left(\frac{\cos^2\phi_\mathrm{m}}{\alpha_x^2}+\frac{\sin^2\phi_\mathrm{m}}{\alpha_y^2}\right)\right)^2}$$

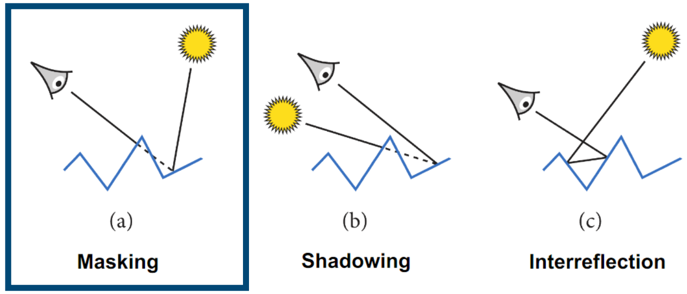
- θ_m and ϕ_m: spherical coordinates of microfacet normal ω_m

Microfacet effects

- Microfacets occlude each other
  - Not the entire surface is visible to the camera ➔ Masking
  - Not the entire surface is lit by the light ➔ Shadowing
- Light scatters between microfacets



The masking function

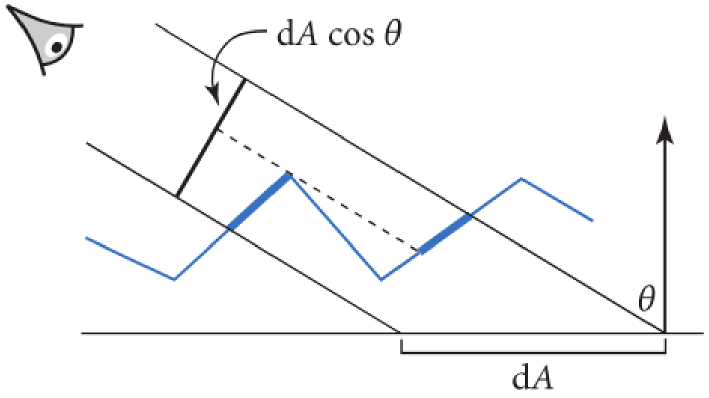


- What fraction of microfacets with normal ω_m is visible from ω?
  - $0 \leq G(\omega, \omega_m) \leq 1$
- For Trowbridge-Reitz, this is

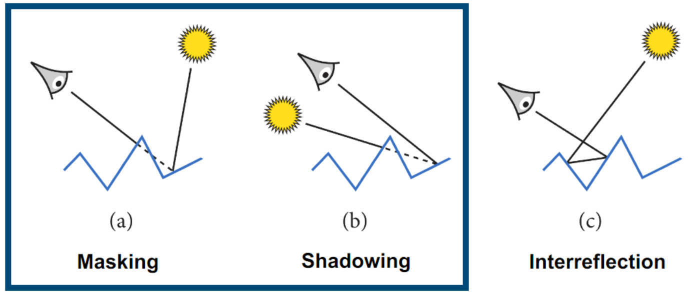
$$G_1(\omega) = \frac{1}{1 + \Lambda(\omega)}$$

$$\Lambda(\omega) = \frac{\sqrt{1 + \alpha^2 \tan^2 \theta} - 1}{2}$$

$$\alpha = \sqrt{\alpha_x^2 \cos^2 \phi + \alpha_y^2 \sin^2 \phi}$$



The masking-shadowing function



- Bidirectional form of G
- Models the masking and shadowing between a pair of directions

$$G(\omega_o, \omega_i) = \frac{1}{1 + \Lambda(\omega_o) + \Lambda(\omega_i)}$$

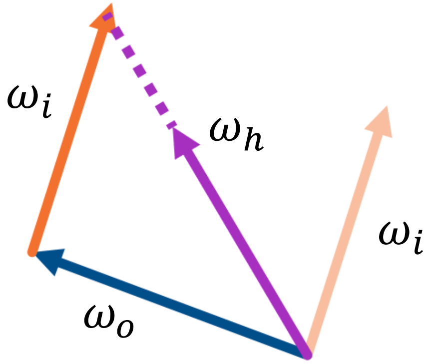
$$\Lambda(\omega) = \frac{\sqrt{1 + \alpha^2 \tan^2 \theta} - 1}{2}$$

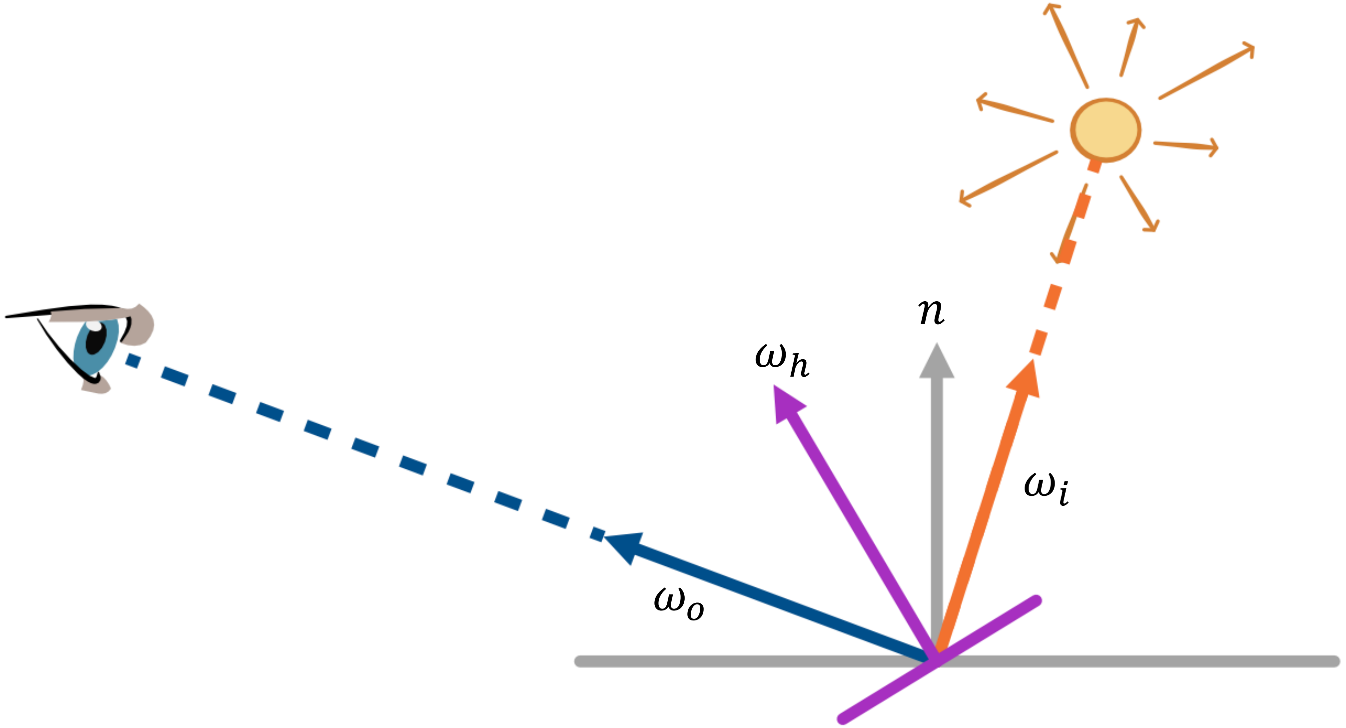
$$\alpha = \sqrt{\alpha_x^2 \cos^2 \phi + \alpha_y^2 \sin^2 \phi}$$

A Microfacet BRDF assuming mirror-like microfacets

- Given a pair of directions ω_i and ω_o
- Compute the "half vector" ω_h: the normal of a mirror-like microfacet that will reflect ω_i to ω_o

$$\omega_h = \frac{\omega_i + \omega_o}{\|\omega_i + \omega_o\|}$$





A Microfacet BRDF assuming mirror-like microfacets

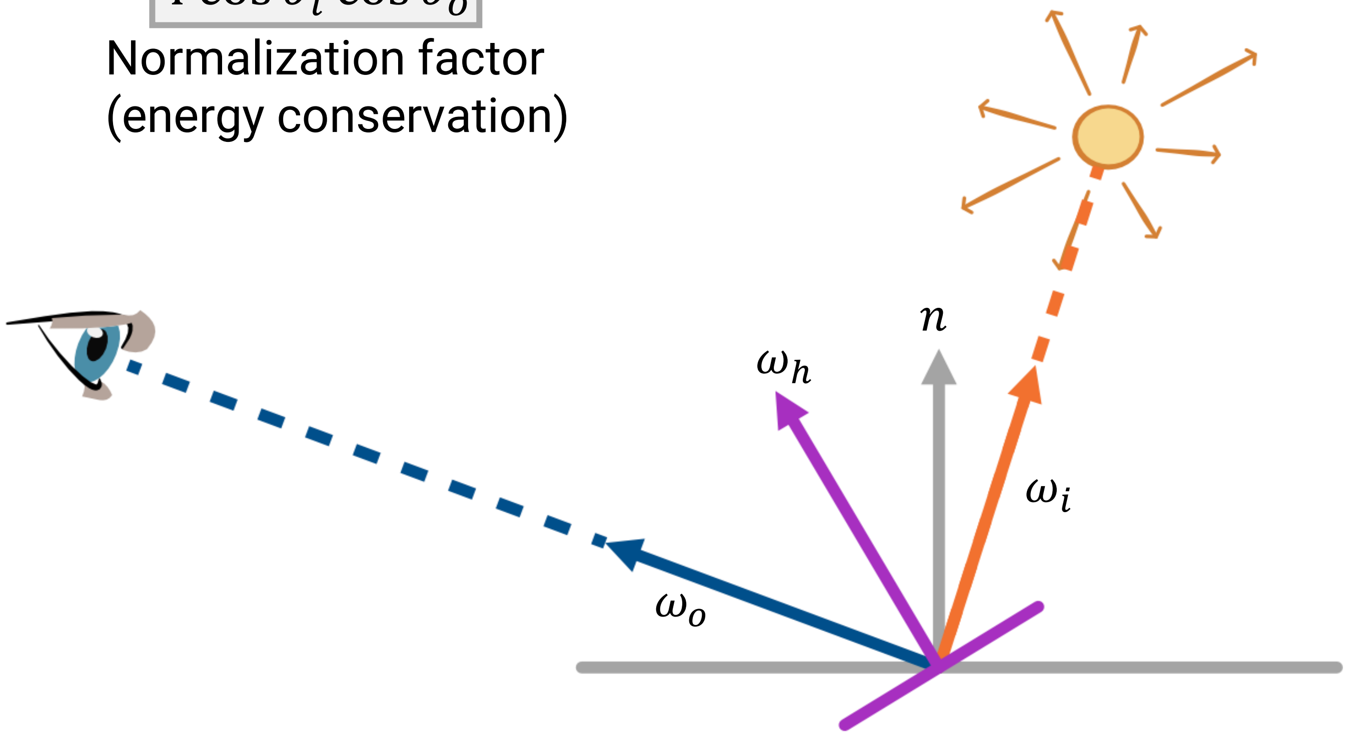
Probability that a microfacet has normal ω_h

Geometry masking

$$f_r(x, \omega_i, \omega_o) = \frac{\boxed{D(\omega_h)}\boxed{G(\omega_i, \omega_o)}\boxed{F_r(\omega_o, \omega_h)}}{\boxed{4 \cos\theta_i \cos\theta_o}}$$

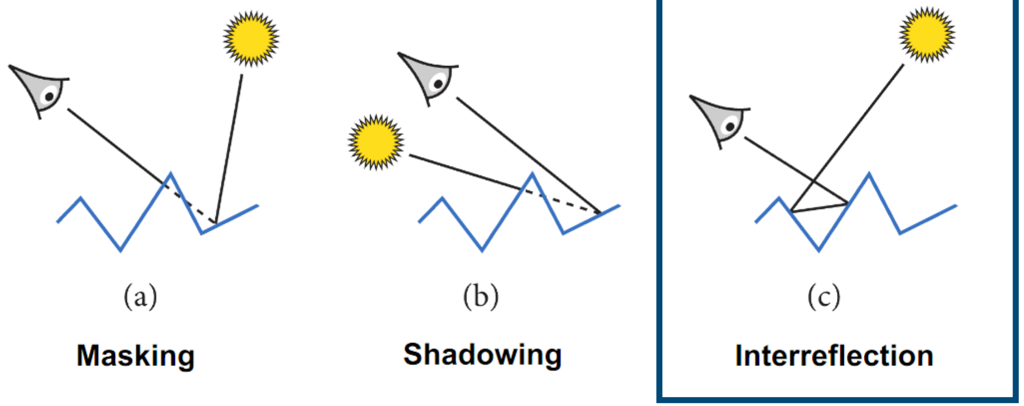
Fresnel term on the mirror-like microfacet

Normalization factor (energy conservation)

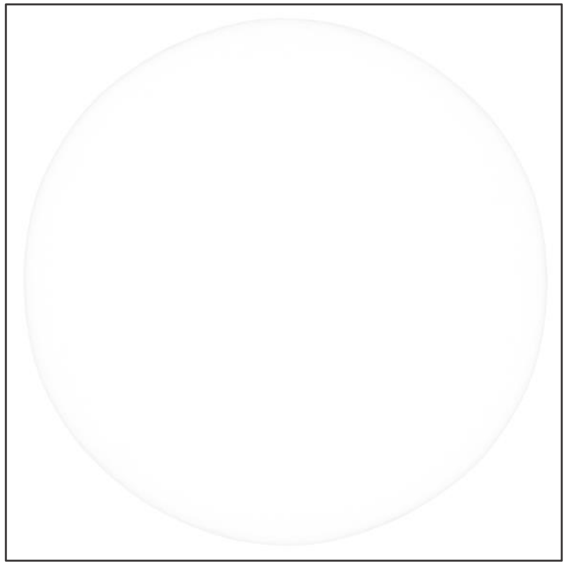


Microfacet BRDFs and energy conservation

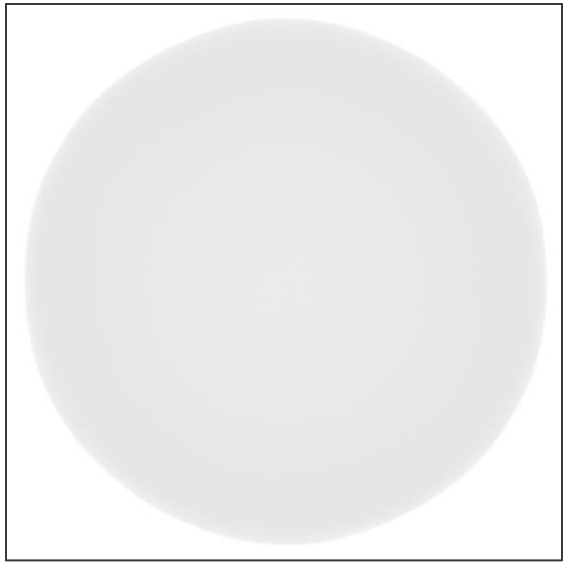
- “White furnace test”
  - $L_i = 1$ from all directions, image should be white
- Darkening = energy loss



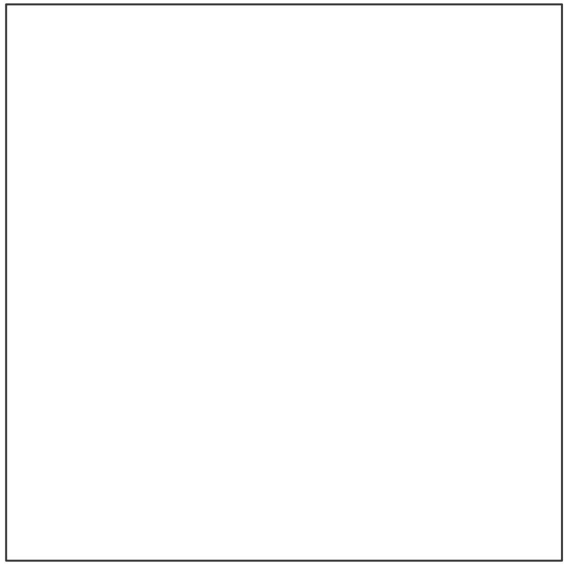
Often unaccounted for



$\alpha = 0.2$



$\alpha = 0.6$

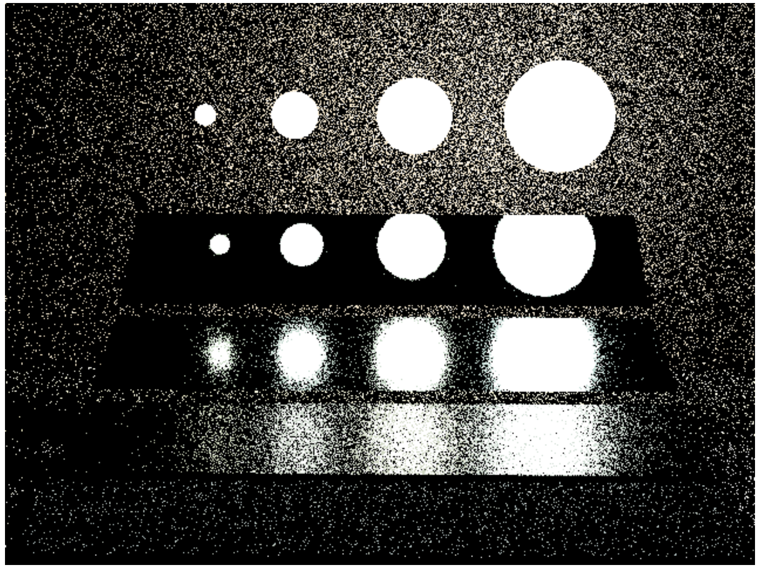


$\alpha = 0.6$ with multi-scatter

BSDF sampling is helpful on glossy surfaces

- Generate directions proportional to the BRDF
  - $p(\omega_i) \propto f_r(x, \omega_i, \omega_o)$

- E.g., for a microfacet BRDF:
  1. Sample a microfacet normal (“half vector”) proportional to the NDF $D(\omega_h)$
  2. Reflect ω_o about this normal to get ω_i



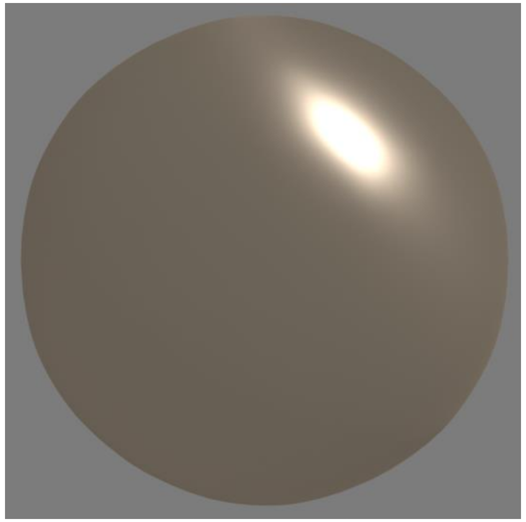
Combined BSDFs

Example: a metallic-roughness model

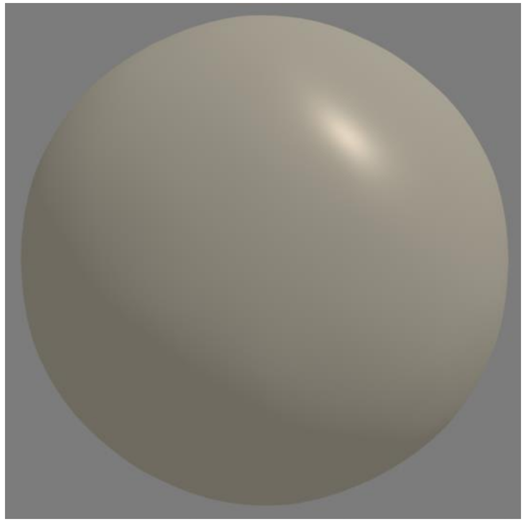
- Two components:
  - Microfacet BRDF f_m with a conductor Fresnel term
  - Diffuse BRDF f_d
- Additional parameter: “metallic” m
  - Interpolates between the two

$$f_r = m f_m + (1 - m) f_d$$

- To sample this with a probability $p(\omega) \propto m f_m + (1 - m) f_d$:
  - Decide between diffuse and microfacet
    - with probability m for the former, $1 - m$ for the latter
  - Sample the selected component



$m = 0.9$



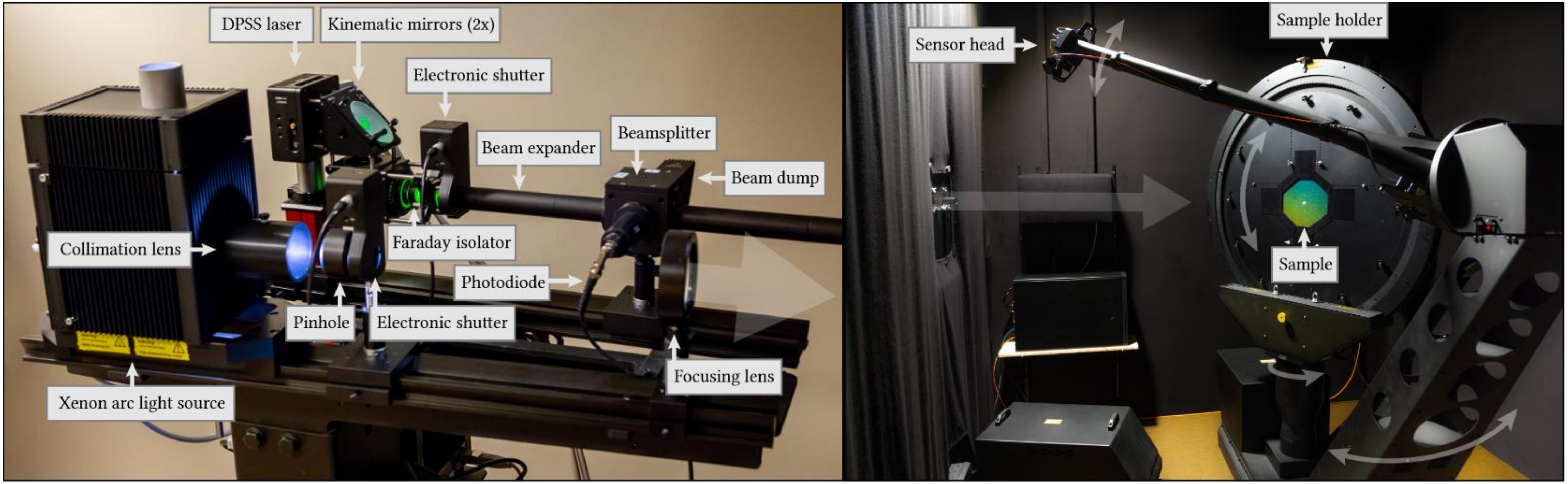
$m = 0.1$

Layer models

Example from: Jakob et al. 2014. “A Comprehensive Framework for Rendering Layered Materials”

What else is possible with materials?

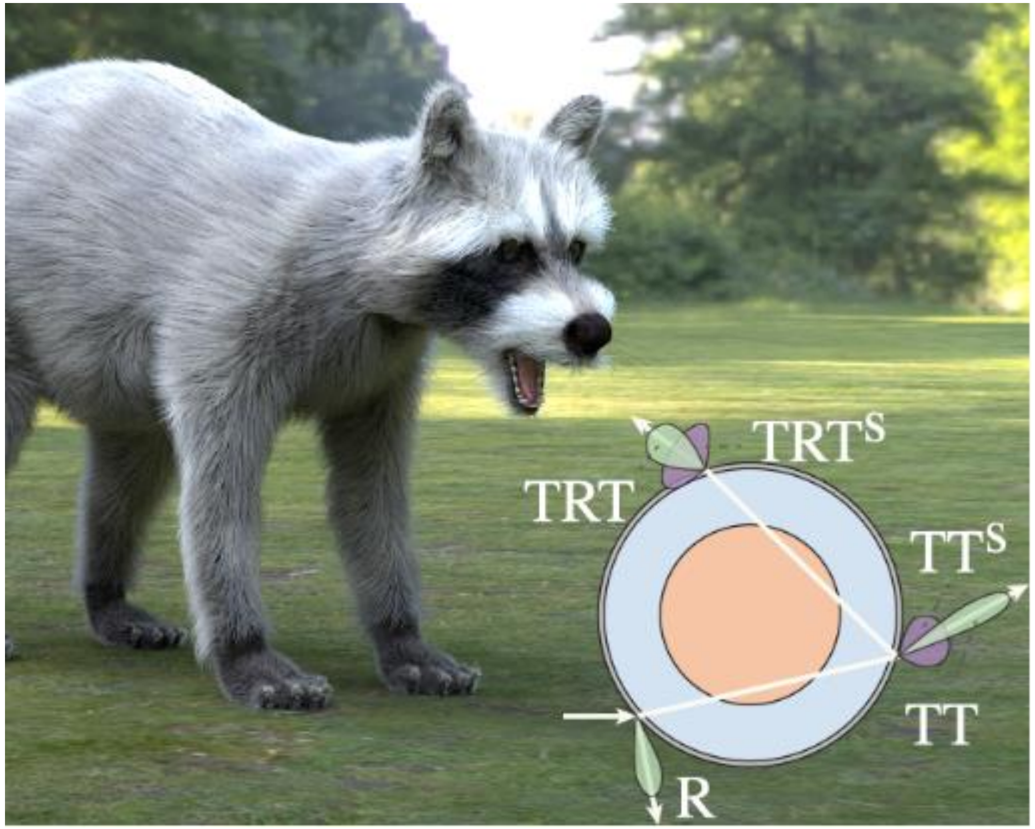
Capturing BSDF data



Example from: Dupuy and Jakob 2018. “An Adaptive Parameterization for Efficient Material Acquisition and Rendering”

Dataset and interactive viewer: https://rgl.epfl.ch/materials

Fabrics, hair, and fur



Example from Yan et al. 2017. “An Efficient and Practical Near and Far Field Fur Reflectance Model”

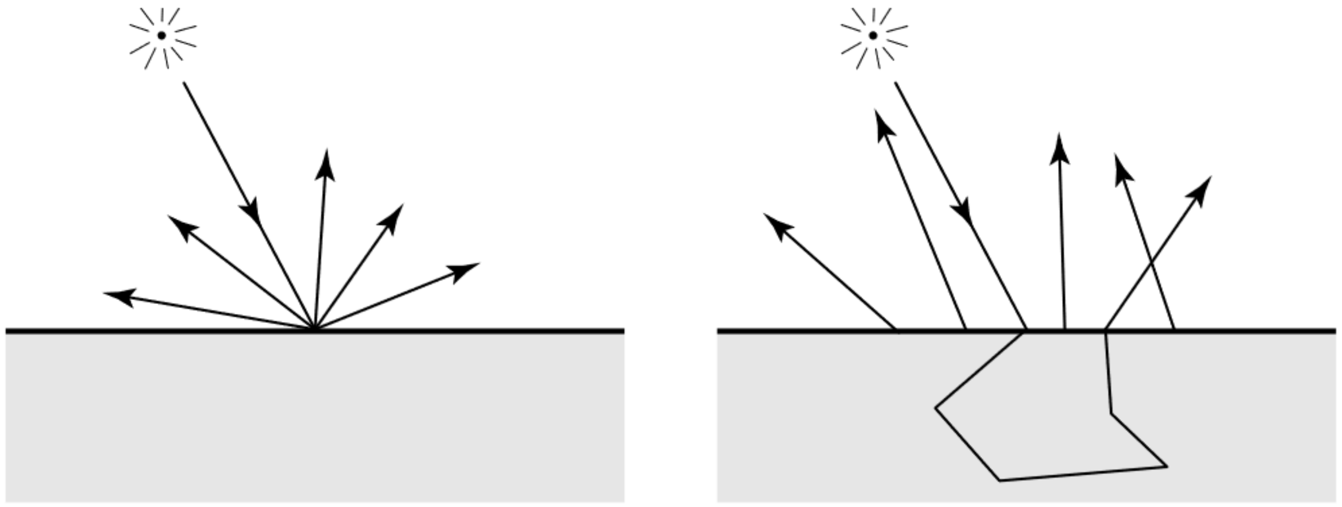
Pearlescence

- Wave-optical interference causes colorful effects
- Can be modelled as part of the BSDF

Example from: Guillén et al. 2020. A General Framework for Pearlescent Materials

Subsurface scattering approximations with the BSSRDF

- Light enters at one point, leaves at another
- Extends the BSDF with another parameter: $f(x, y, \omega_i, \omega_o)$



(a) (b)

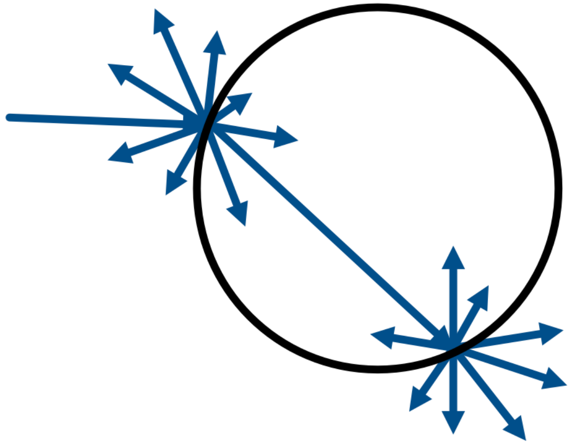
And many, many, many more

Reading materials

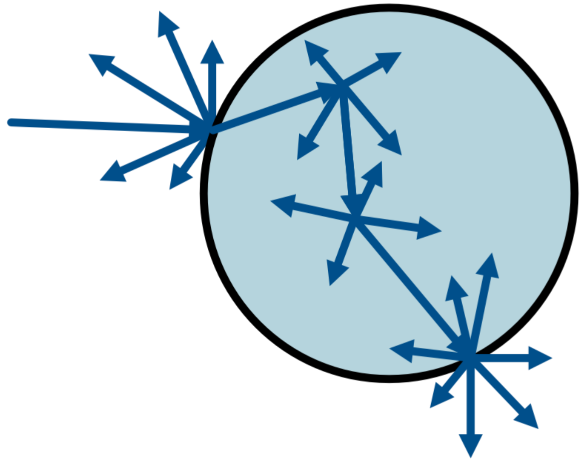
- https://www.pbr-book.org/4ed/Reflection_Models
- Trowbridge & Reitz. “Average irregularity representation of a rough surface for ray reflection“ 1974
  - https://pharr.org/matt/blog/images/average-irregularity-representation-of-a-rough-surface-for-ray-reflection.pdf
- Walter et al. “Microfacet Models for Refraction through Rough Surfaces” 2007
- Eric Heitz. “Sampling the GGX Distribution of Visible Normals” 2018.
  - https://jcgt.org/published/0007/04/01/paper.pdf

Volumes

Real-world objects are not thin shells of emptiness



The thin-shell surface model we assumed so far



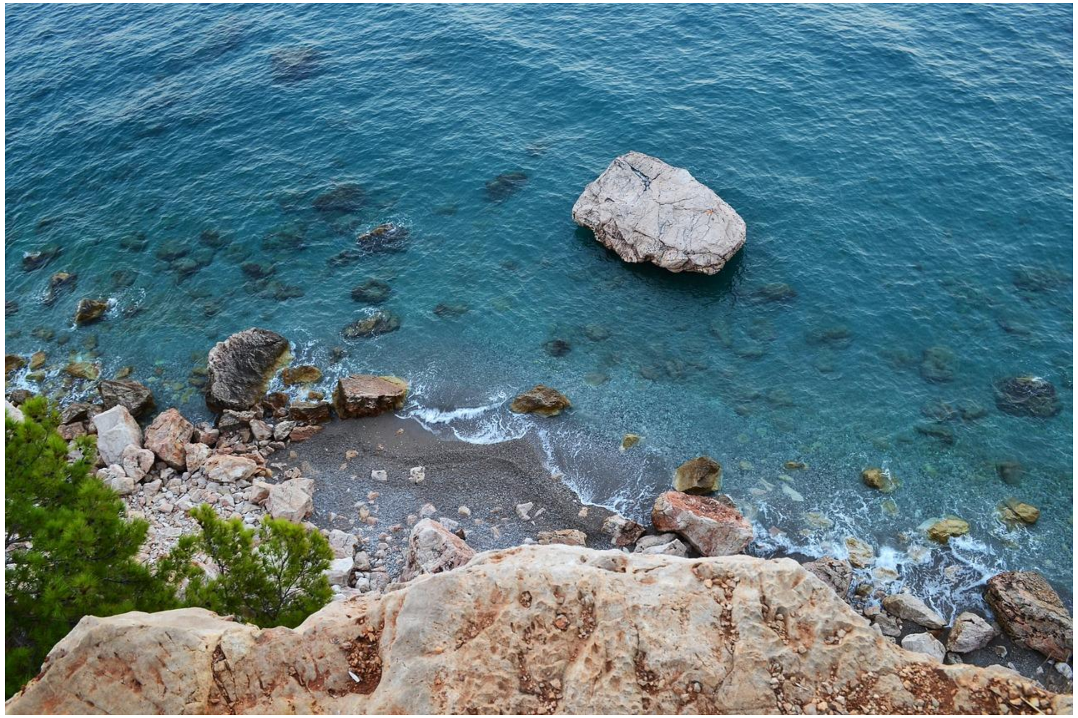
Volumetric scattering underneath surfaces

Responsible for subsurface scattering

(c) Lernert & Sander

Light enters at one point, scatters underneath, leaves at another

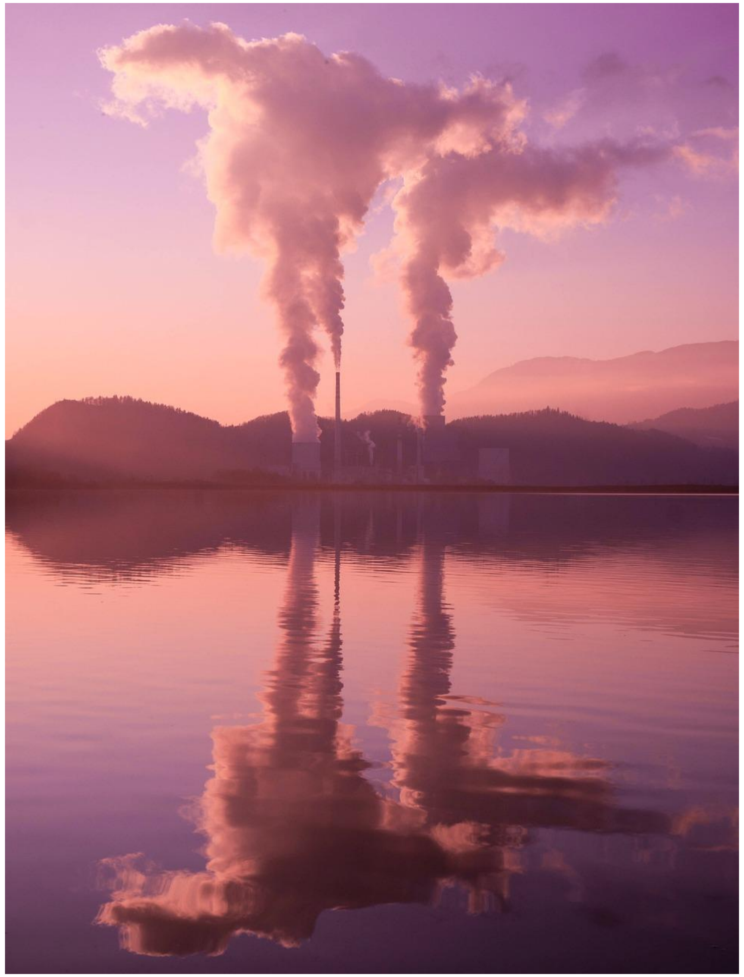
Responsible for the color of liquids



Light is absorbed as it travels through the volume

Some wavelengths (colors) more than others

Not all volumes have a surrounding surface



Like smoke, fog, or gases

Volumes scatter, emit, or absorb light

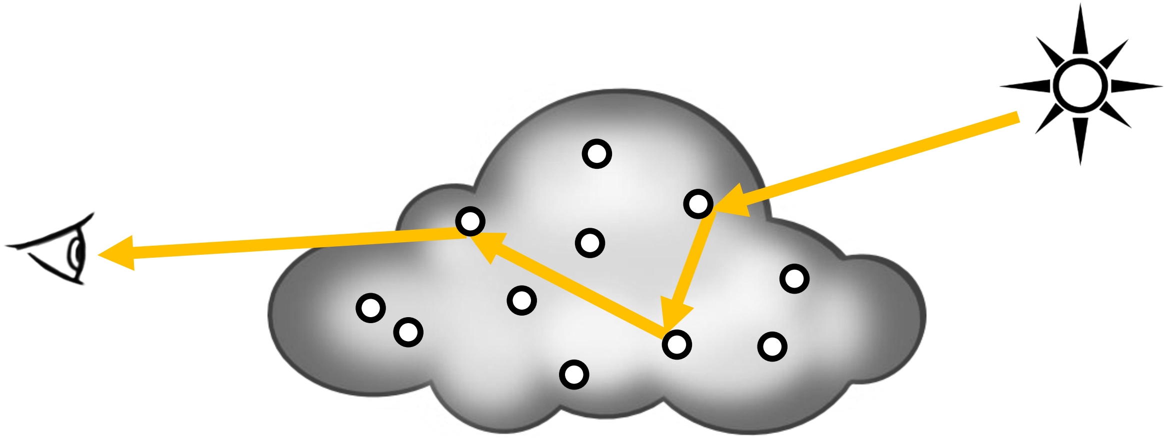
http://coclouds.com

http://wikipedia.org

http://commons.wikimedia.org

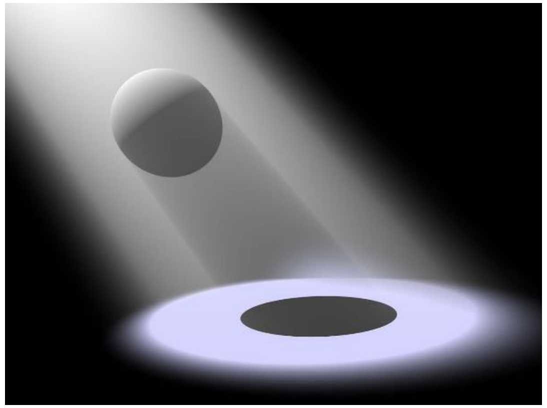
Light interacts with particles in the volume

- Modeling individual particles is not practical
- Instead: aggregate into statistics
  - Same idea as microfacet BSDFs
- Parameters:
  - Absorption coefficient μ_a
    - Fraction of radiance absorbed per unit distance
  - Scattering coefficient μ_s
    - Fraction scattered (in or out) per unit distance
  - Emission L_e
  - Phase function f_p
    - Analog of the BSDF on surfaces



Volume Representation

- Homogeneous:
  - Volume parameters are the same everywhere

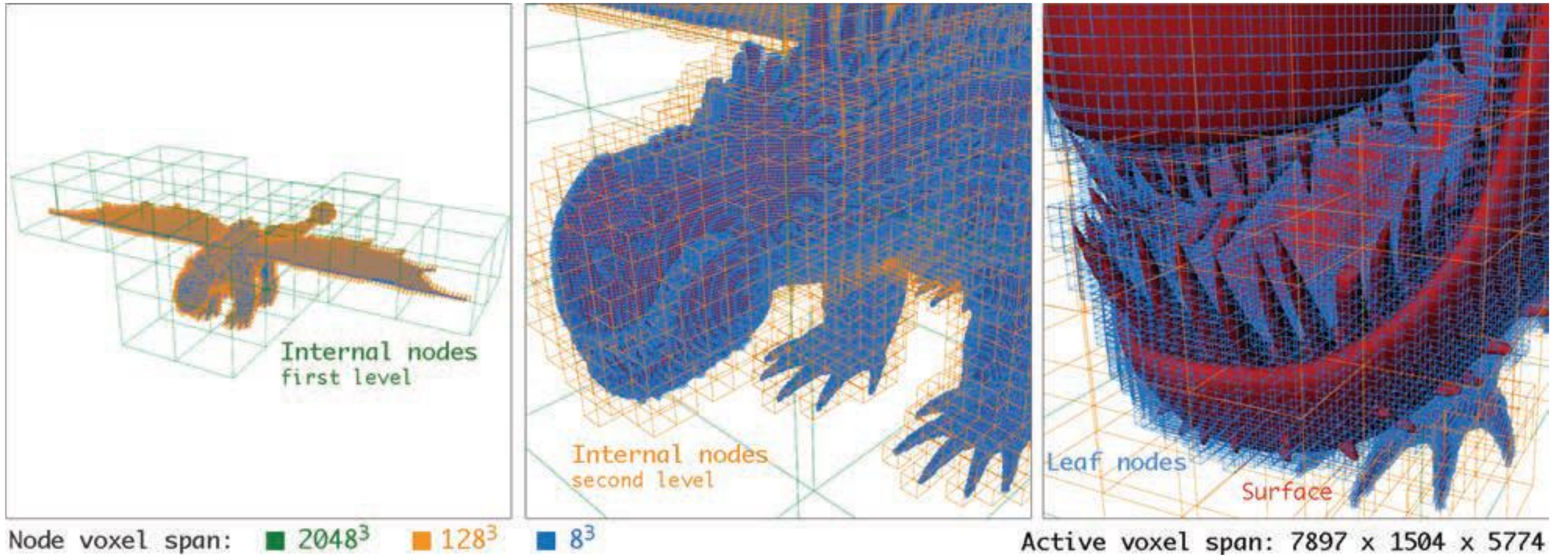


- Heterogeneous:
  - Parameters vary across the volume
  - Can be represented using **3D textures**

http://wikipedia.org

Representation Example: OpenVDB

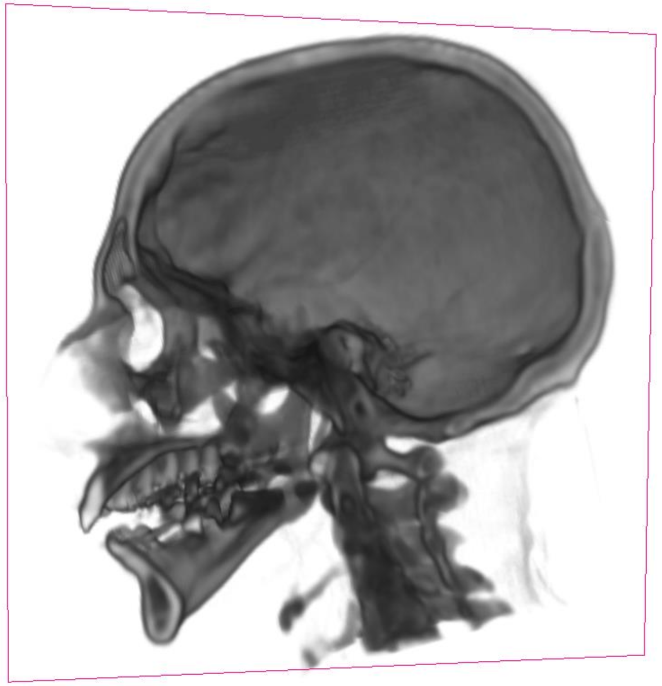
- https://www.openvdb.org/
- Hierarchical voxel representation



Data Acquisition Examples

- Real-world measurements via tomography
- Simulation, e.g.,
  - Fluids,
  - Fire and smoke,
  - Fog

https://docs.blender.org

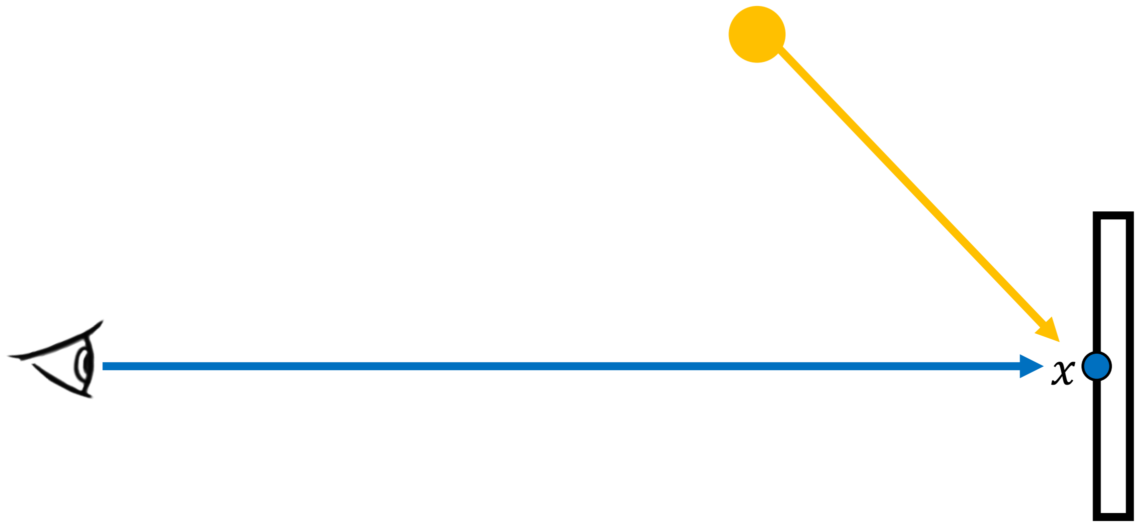


Rendering Volumes

Mathematical Formulation of Volumetric Light Transport

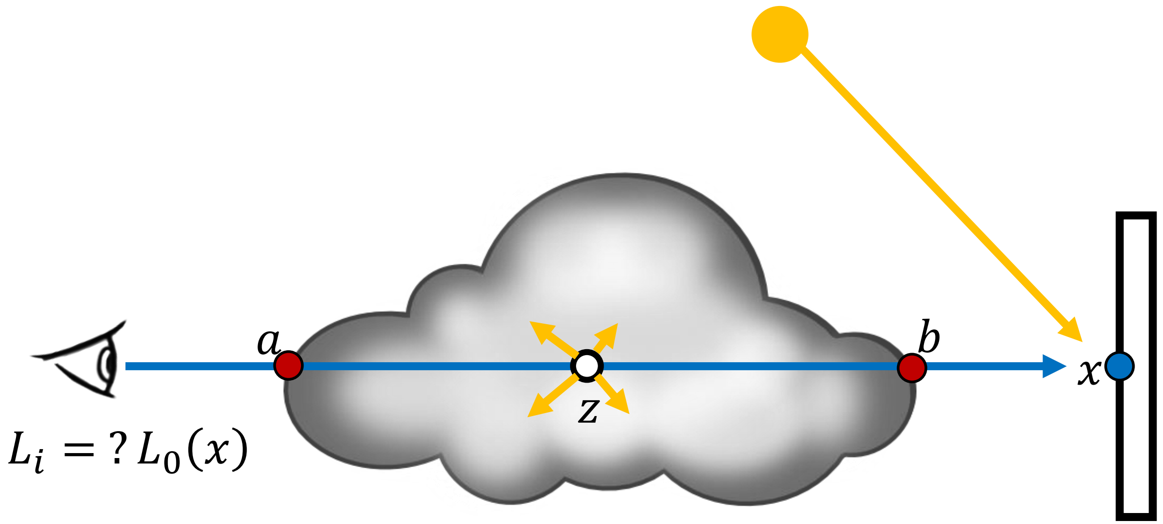
So far: Assume Vacuum

- Compute $L_o(x, \omega_o)$ using the rendering equation



Attenuation = Absorption + Out-Scattering

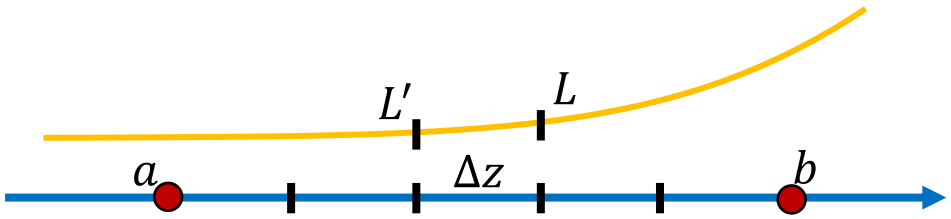
- Every point in the volume might absorb light or scatter it in other directions
- Modeled by absorption and scattering densities: $\mu_a(z)$ and $\mu_s(z)$



http://commons.wikimedia.org

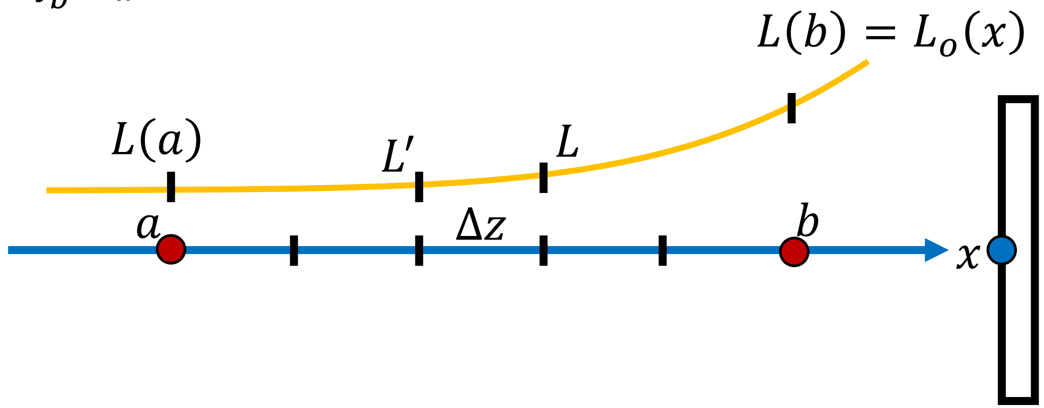
Computing Absorption – Intuition

- Consider a small segment Δz
- Along that segment, radiance is reduced from L to L'
- Let μ_a be the fraction of radiance absorbed per unit distance
  - $L' - L = -\mu_a \, \Delta z \, L$



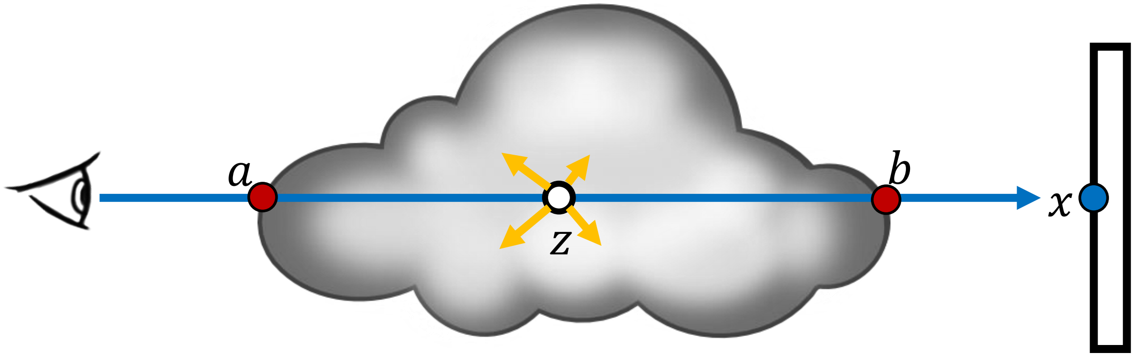
Computing Absorption – Exponential Decay

- $\Delta L = -\mu_a \, L \, \Delta z$
- For infinitely small Δz
  - $dL = -\mu_a \, L \, dz$
  - $\frac{dL}{dz} = -\mu_a L$
- A differential equation that models exponential decay
- Solution: $L(a) = L_o(x) \, e^{-\int_b^a \mu_a(t) \, dt}$



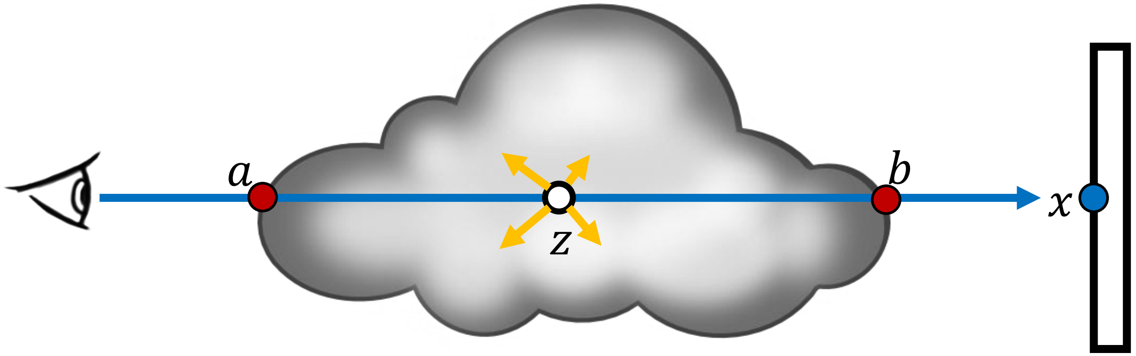
Computing Out-Scattering

- Same as absorption, only different factor!
- $\mu_s(z)$: fraction of light scattered at point z
- $L(a) = L_o(x)\, e^{-\int_b^a \mu_s(t)\, dt}$



Computing Attenuation

- Fraction of light that is neither absorbed nor out-scattered
- $\mu_t = \mu_a + \mu_s$
- $L(a) = L_o(x)\, T(a,b)$
- Attenuation: $T(a,b) = e^{-\int_b^a \mu_t(t)\, dt}$



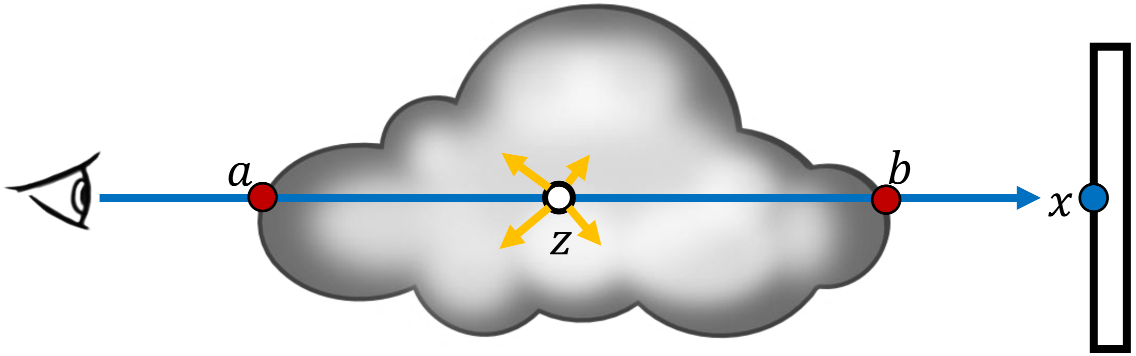
Computing Attenuation – Homogeneous

- Simple case: constant density / attenuation
  - $\mu_t(z) = \mu_t \quad \forall z$

Heterogeneous attenuation is harder
→ We'll cover it in the summer term lecture

- $T(a,b) = e^{-\int_b^a \mu_t(t)\,dt} = e^{-\boxed{(a-b)}\mu_t}$

distance travelled in the volume



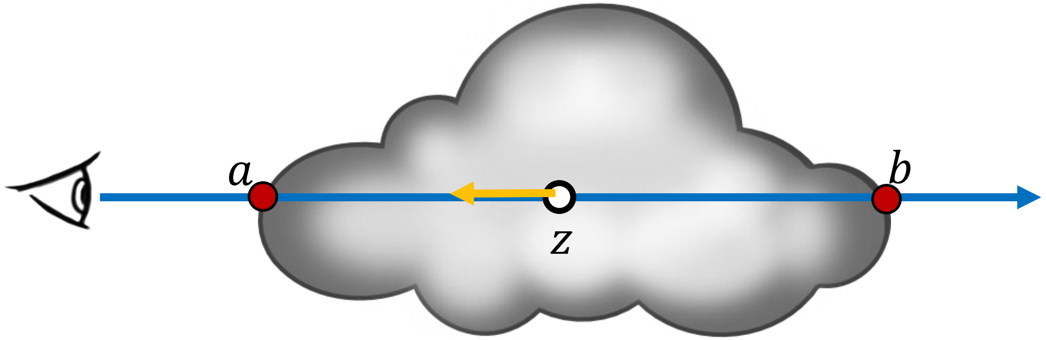
Every Point Might Emit Light

http://wikipedia.org

- Assume z emits $L_e(z)$ towards a
- Some of that light might be absorbed or out-scattered: It is attenuated
- $L(a) = L_e(z)\, T(z,a)$

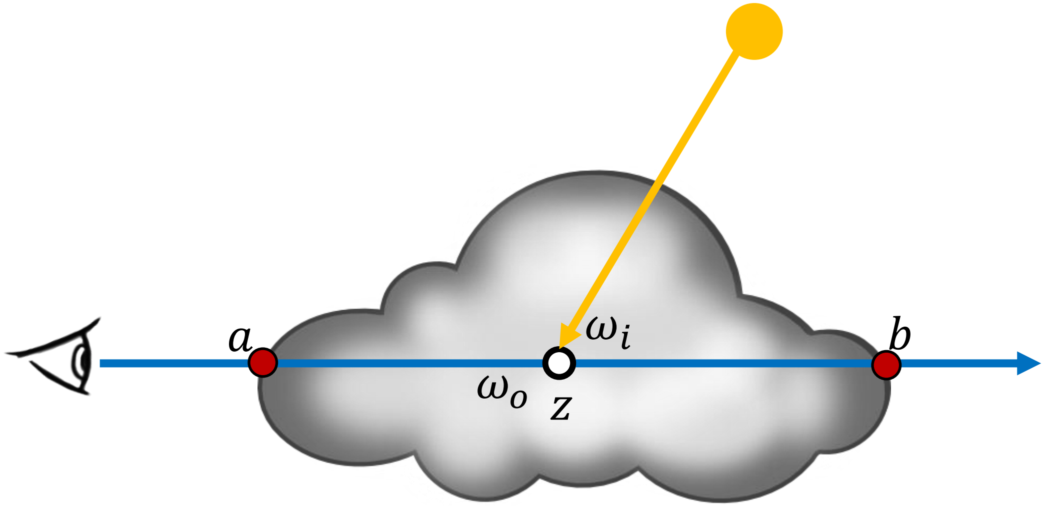
- Happens at every point along the ray!
- $L(a) = \int_a^b L_e(z)\, T(z,a)\, dz$

- Can be estimated via MC:
  - Sample random distance $z \in [a, b]$



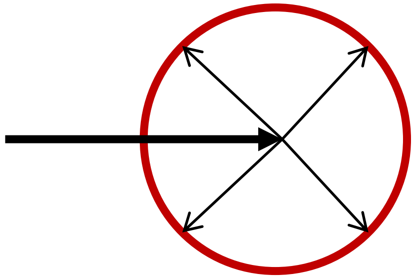
Volumetric Direct Illumination

- Account for the (attenuated) direct illumination at every point z
- Similar to the rendering equation:
  - $L_o(z, \omega_o) = \int_\Omega L_i(z, \omega_i)\, f_p(\omega_i, \omega_o)\, d\omega_i$
  - Integration over the whole sphere Ω
- The **phase function** f_p takes on the role of the BSDF



Phase Functions

- $L_o(z, \omega_o) = \int_\Omega L_i(x, \omega_i)\, \boldsymbol{f_p(\omega_i, \omega_o)}\, d\omega_i$
- Describe what fraction of light is reflected from ω_i to ω_o
- Similar to BSDF for surface scattering
- Simplest example: isotropic phase function
  - $f_p(\omega_i, \omega_o) = \frac{1}{4\pi}$
  - (energy conservation: $\int_\Omega \frac{1}{4\pi} d\omega = 1$)



Example: Henyey-Greenstein phase function

- $f_p(\omega_i, \omega_o) = \frac{1}{4\pi} \frac{1-g^2}{(1+g^2+2g\cos(\omega_i,\omega_o))^{\frac{3}{2}}}$

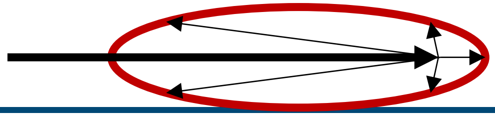
- g: asymmetry (scalar)
- $\cos(\omega_i, \omega_o)$: cosine of the angle formed by ω_i and ω_o

Henyey-Greenstein: Asymmetry Parameter

- $g = 0$: isotropic
- Negative g: back scattering
- Positive g: forward scattering

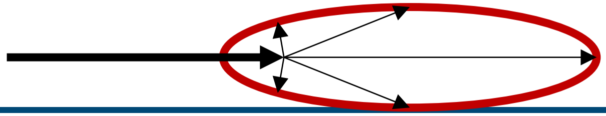
Back Scattering

http://commons.wikimedia.org



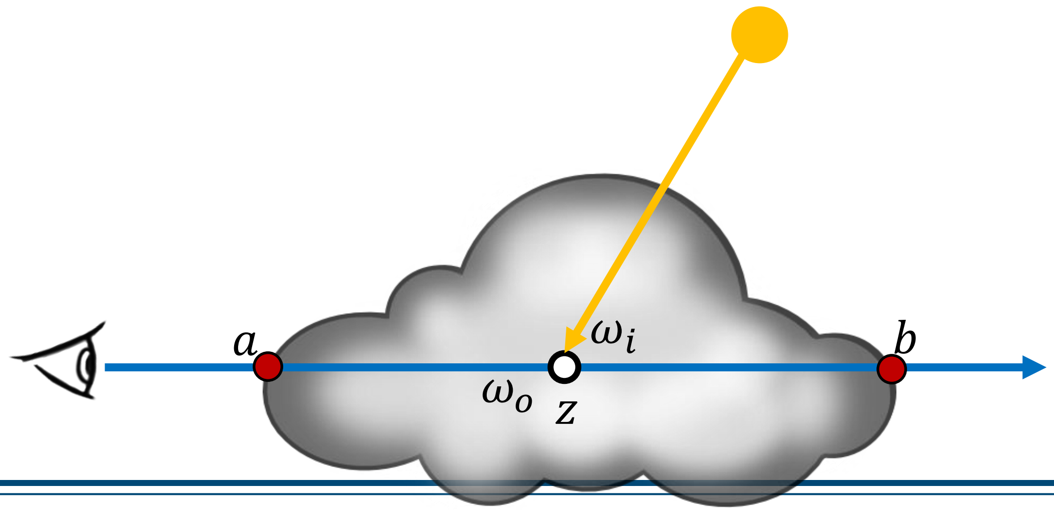
Forward Scattering

http://coclouds.com



How to Estimate Volume Direct Illumination

- Reflected radiance at a point z:
  - $L_o(z, \omega_o) = \int_\Omega L_i(x, \omega_i)\, f_p(\omega_i, \omega_o)\, d\omega_i$
- In our framework:
  - Sum over all point lights (as for surfaces)
  - Trace shadow ray (as for surfaces)
  - Estimate attenuation along the shadow ray (as for surfaces)

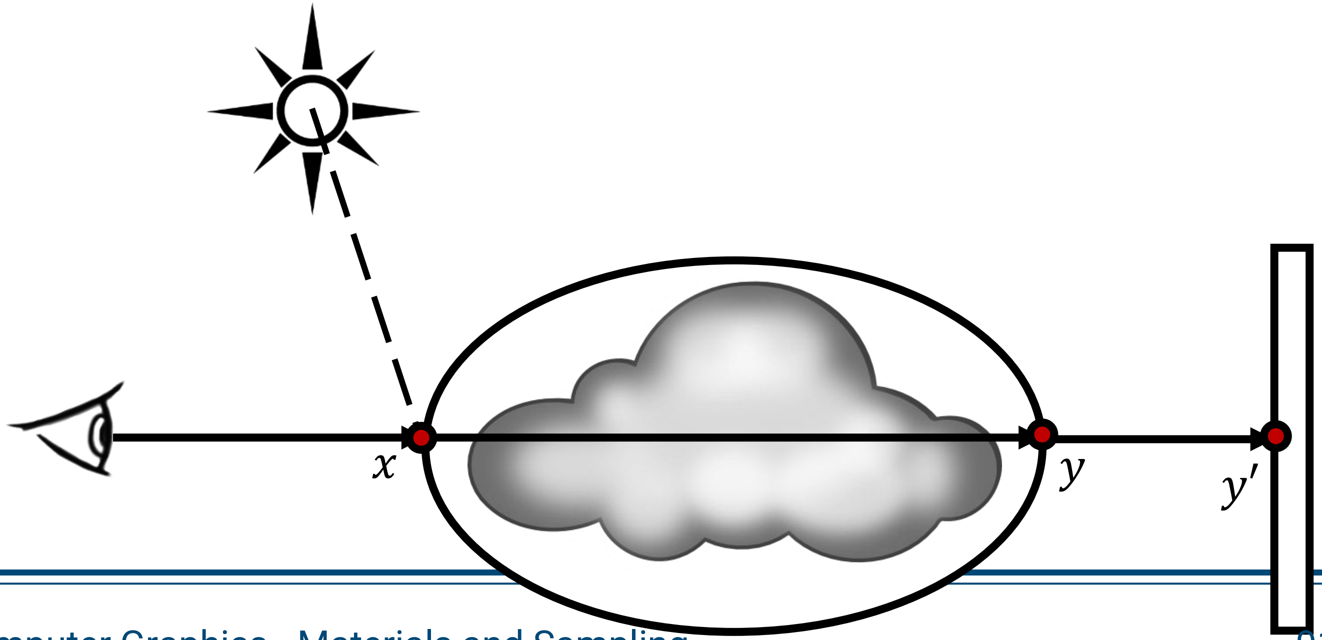


Putting it all Together

A Simple Volume Integrator

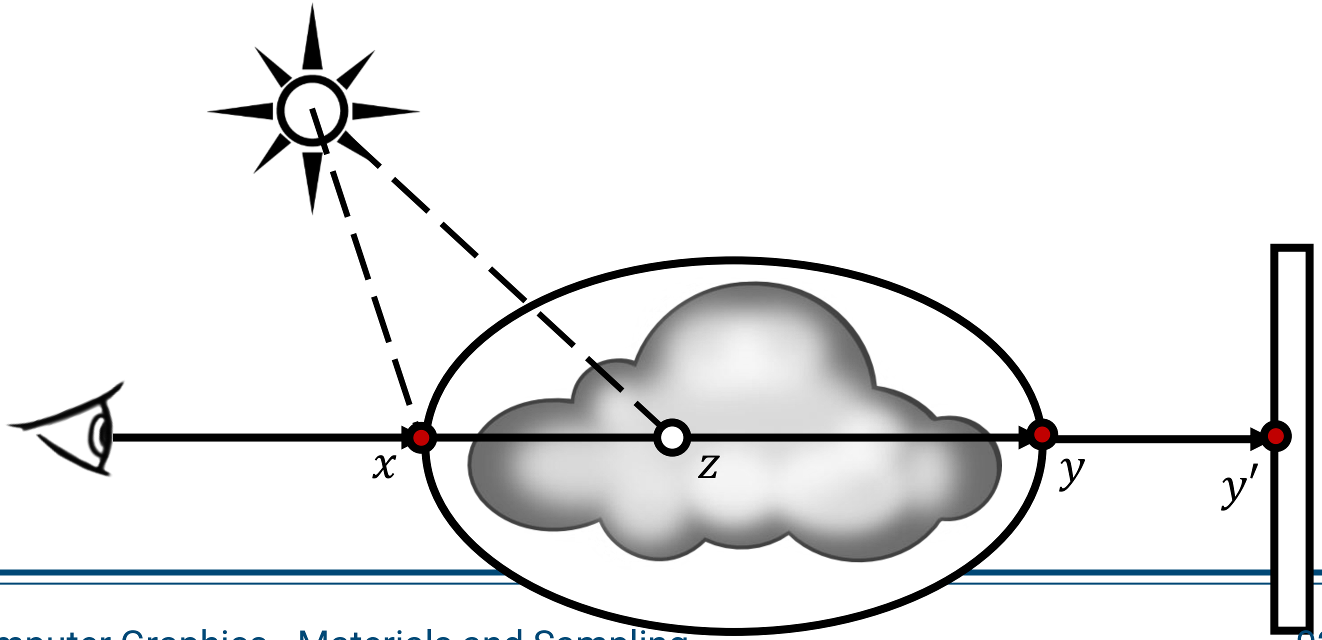
A Simple Volume Integrator

- Estimate direct illumination at x
- If volume: continue straight ahead until no volume (yields intersections y, z)



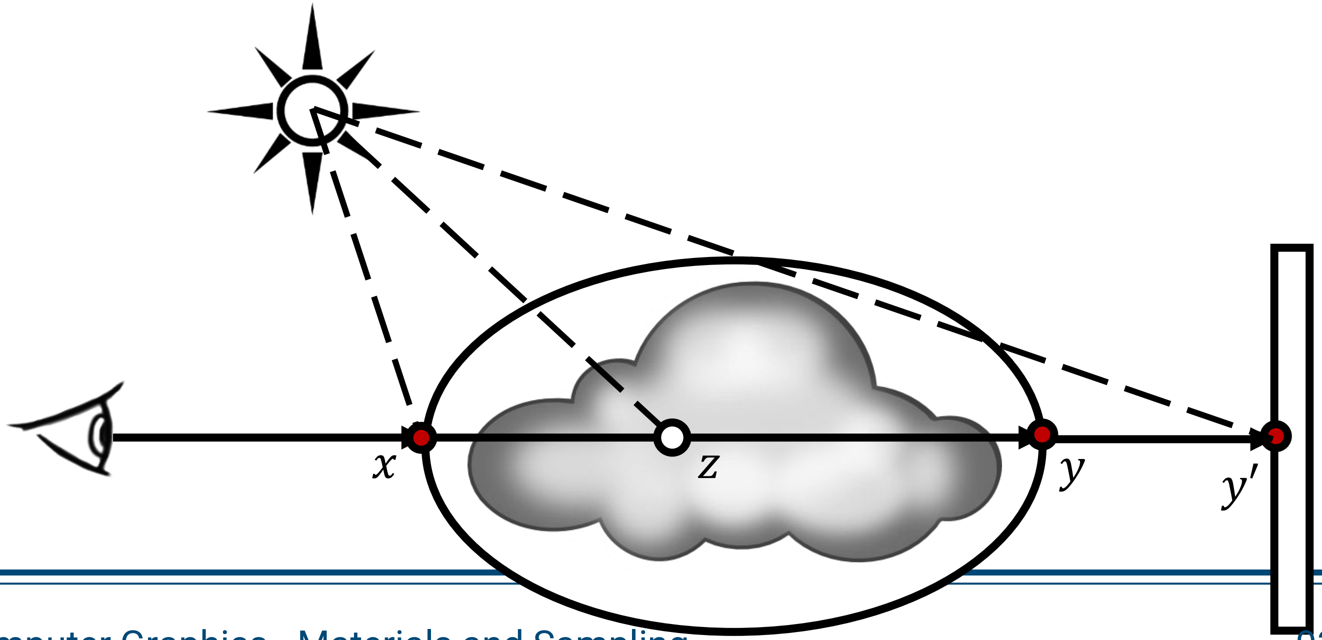
A Simple Volume Integrator

- Estimate direct illumination at x (as before)
- If volume: continue straight ahead until no volume (yields intersections y, z)
- Sample scatter distance z
    - Add attenuated emission from z to x
    - Add attenuated in-scattered direct light



A Simple Volume Integrator

- Estimate direct illumination at x (as before)
- If volume: continue straight ahead until no volume (yields intersections y, z)
- Sample scatter distance z
  - Add attenuated emission from z to x
  - Add attenuated in-scattered direct light
- Compute direct light at y and y'
  - Attenuate shadow rays through volume



Reading materials

- https://www.pbr-book.org/4ed/Volume_Scattering

Summary

Now we know how to...

- use Monte Carlo integration to compute soft shadows and global illumination
- model non-trivial surface appearances with BSDFs
- model (simple) volumetric scattering effects