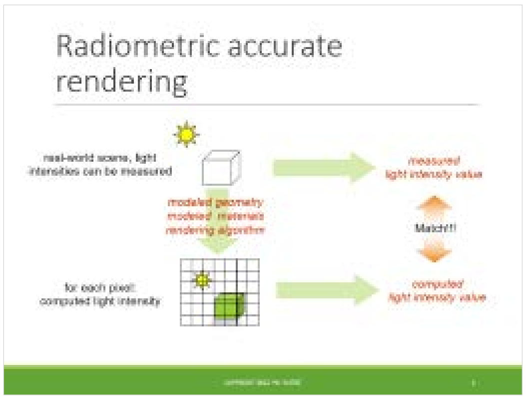
The Rendering Equation

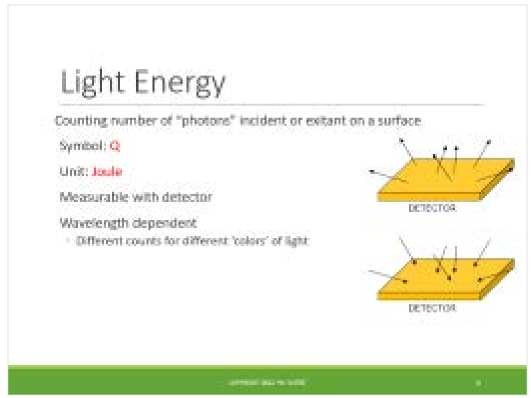
$$L_o = L_e + \int_{\Omega} f_r \, L_i \cos\theta \, d\omega$$

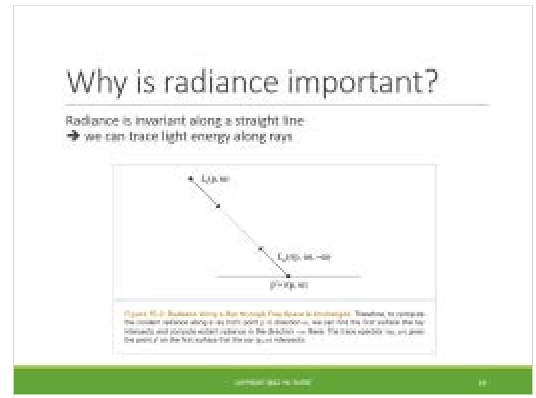
FUNDAMENTALS OF COMPUTER GRAPHICS

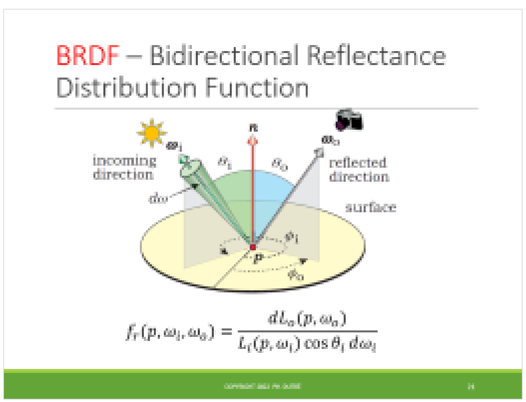
PHILIP DUTRÉ
DEPARTMENT OF COMPUTER SCIENCE

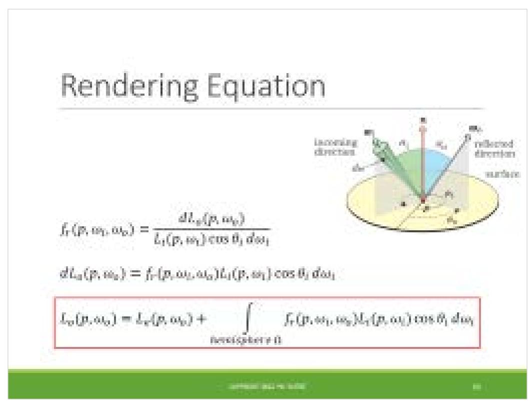
Overview Lecture

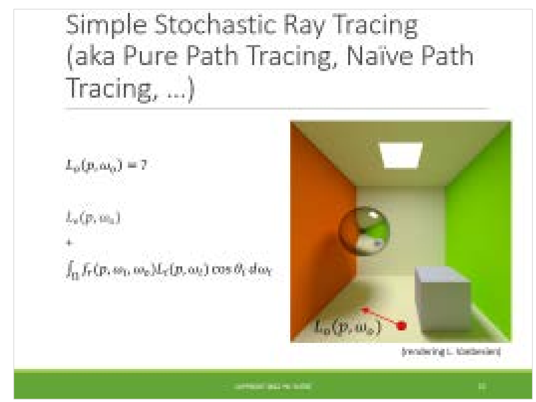






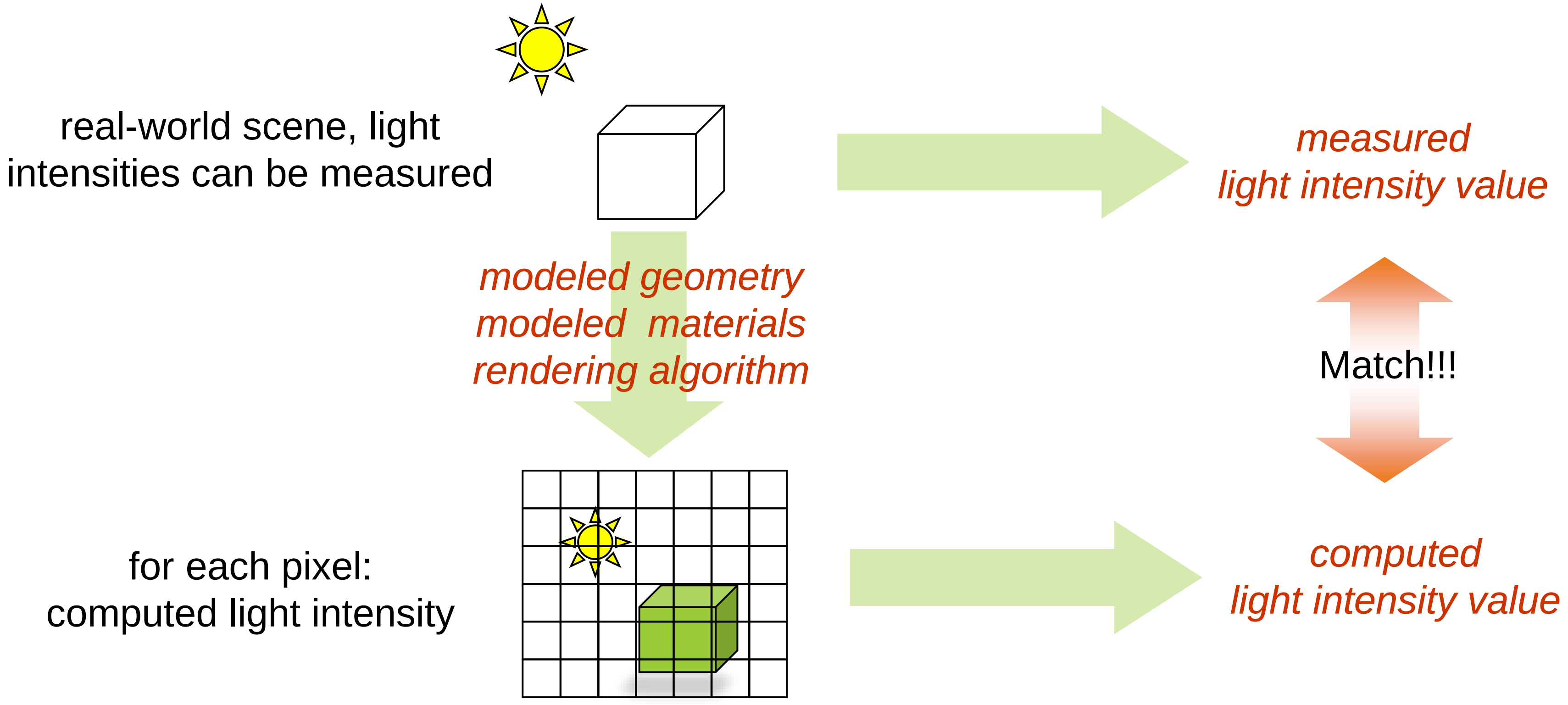




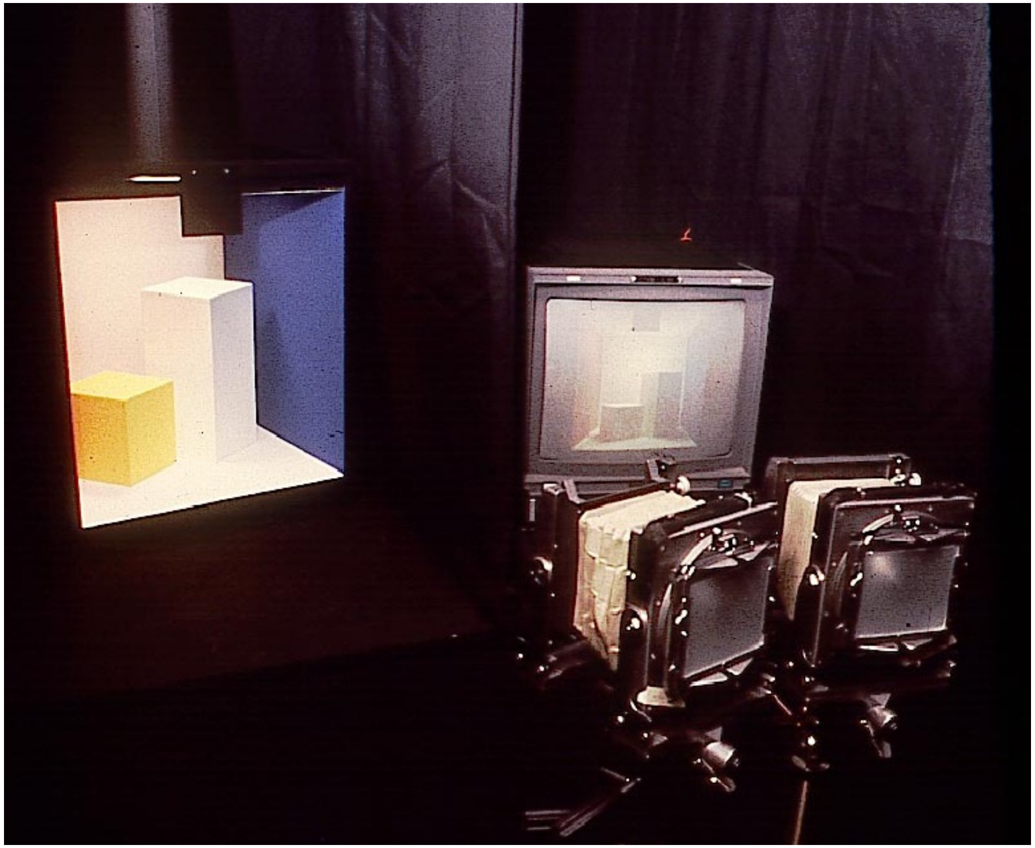


Relevant sections in book: Chapter 13
(Illustrations from *Ray Tracing From The Ground Up*, *Physically-Based Rendering*, *Fundamentals of Computer Graphics*)
(Page numbering might skip some slides due to 'hidden' slides in my presentation.)

Radiometric accurate rendering



Visual comparison



Radiometric / Perceptual match (Meyer1986)

Visual comparison

Light Energy

Counting number of “photons” incident or exitant on a surface

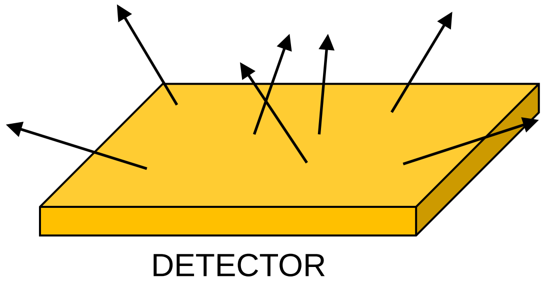
Symbol: Q

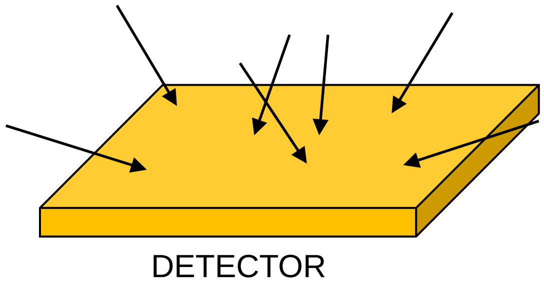
Unit: Joule

Measurable with detector

Wavelength dependent

- Different counts for different ‘colors’ of light





Power

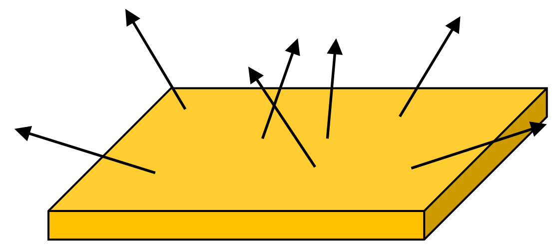
Energy per unit time (dQ/dt)

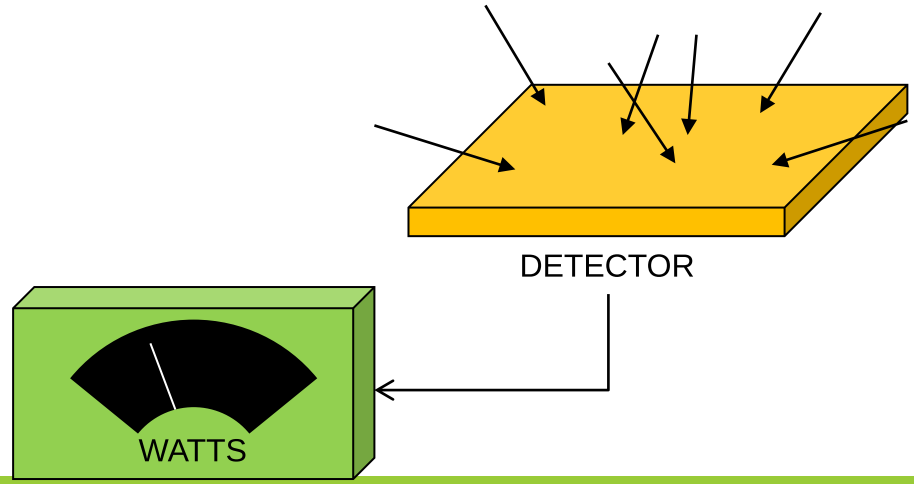
- "radiant flux"

Symbol: P

Unit: Watt (Joule/sec)

- "photons" per second
- measurable as current from a detector





Irradiance

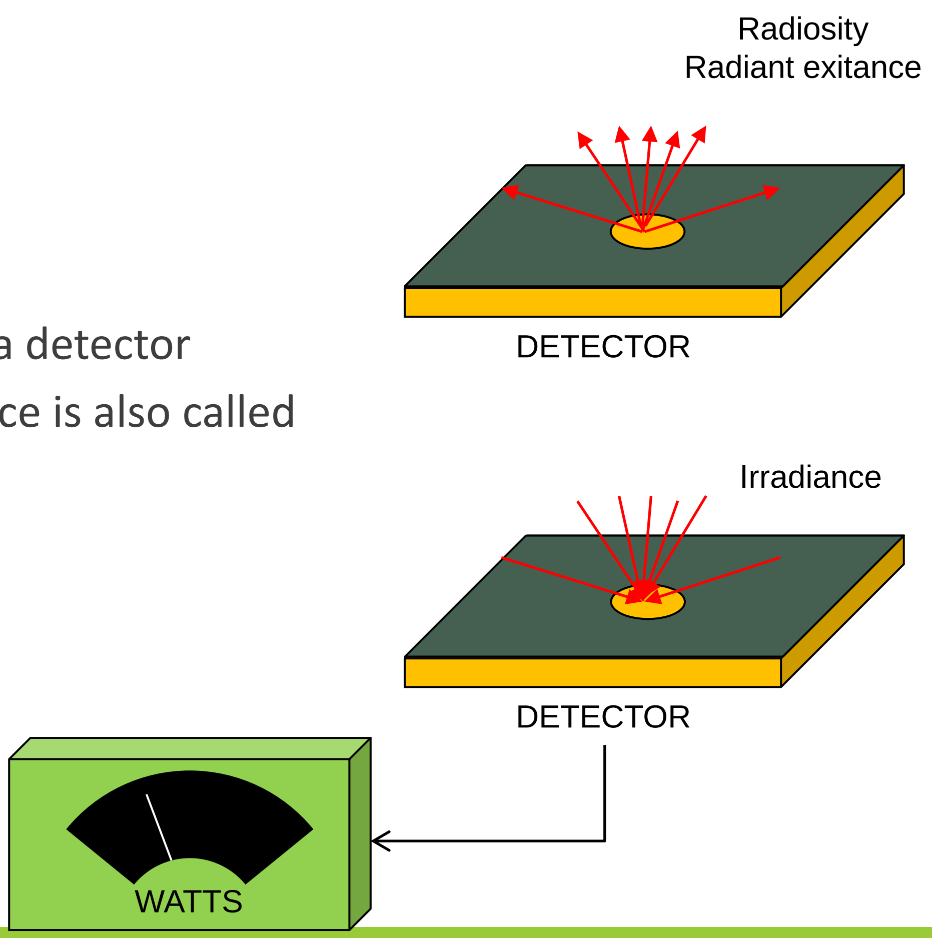
Power per unit area (dP/dA)

- Defined with respect to a surface

Symbol E

Unit Watt/m^2

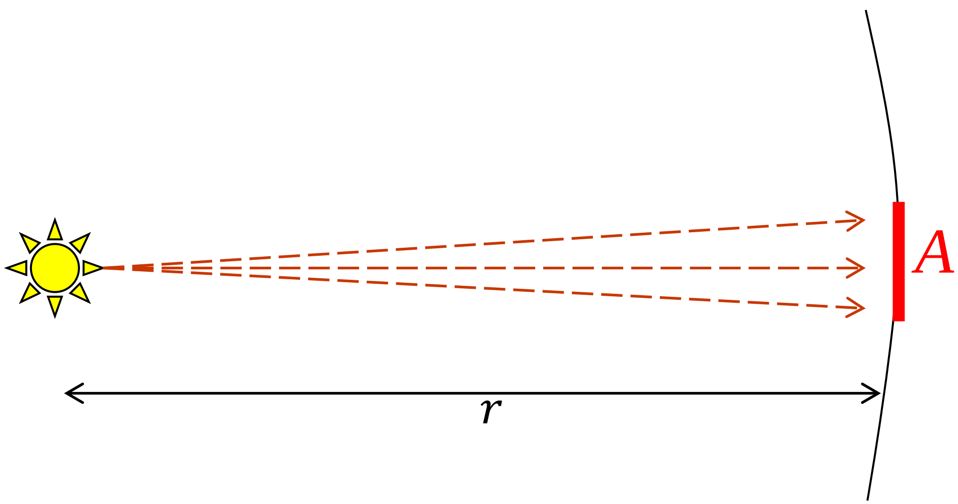
- measurable as power on a small-area detector
- area density of power ***exiting*** a surface is also called radiant exitance (M) or radiosity (B)



Irradiance example

Uniform (point) light source with power P illuminates small surface A at distance r

- Think of A as a piece of the area of a sphere around the light
- P is uniformly spread over the area of this sphere



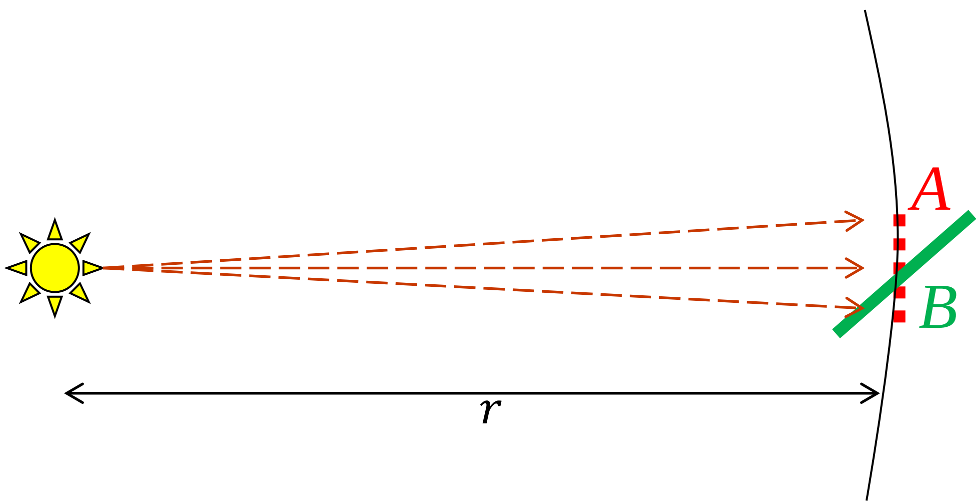
$$P_A = P\frac{A}{4\pi r^2}$$

$$E = \frac{P_A}{A} = \frac{P}{4\pi r^2}$$

Irradiance example

Uniform (point) light source with power P illuminates small surface A at distance r

- Think of A as a piece of the area of a sphere around the light
- P is uniformly spread over the area of this sphere



$$P_B = P_A = P\frac{A}{4\pi r^2}$$

$$E = \frac{P_B}{B} = \frac{P_B\cos\theta}{A} = \frac{P\cos\theta}{4\pi r^2}$$

Solid angle

Angle $\theta = s/r$

- subtended arc of circle
- s/r is unitless and called radians
- whole circle: 2π radians

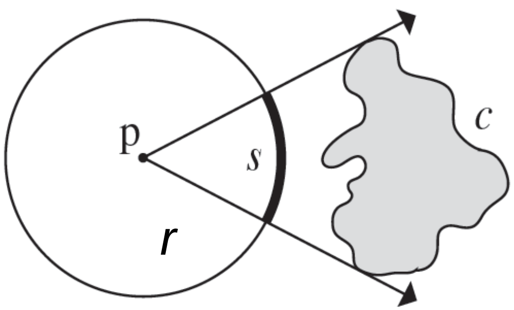


Figure 5.8: Planar Angle. The planar angle of an object c as seen from a point p is equal to the angle it subtends as seen from p, or equivalently as the length of the arc s on the unit sphere.

Solid angle

Solid angle

- subtended area of sphere
- s/r^2 is unitless and called steradians
- whole sphere: 4π steradians

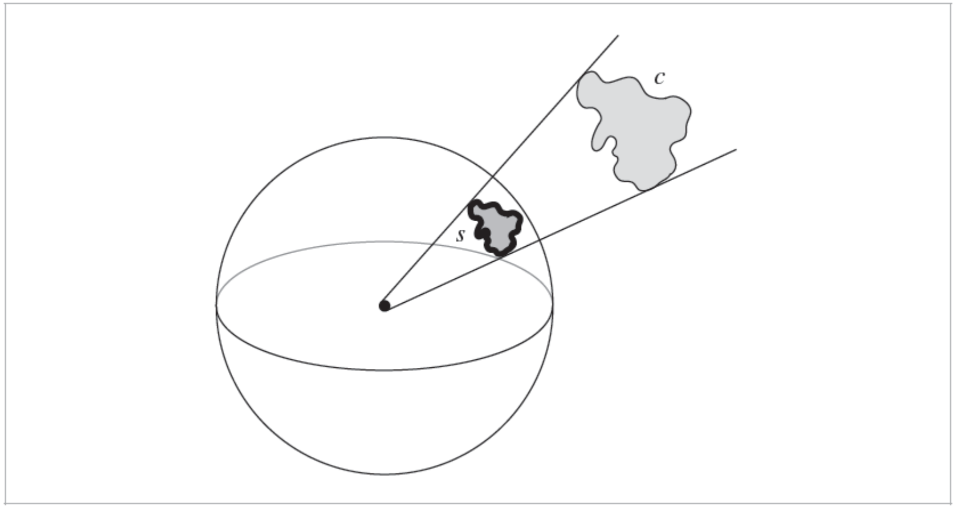


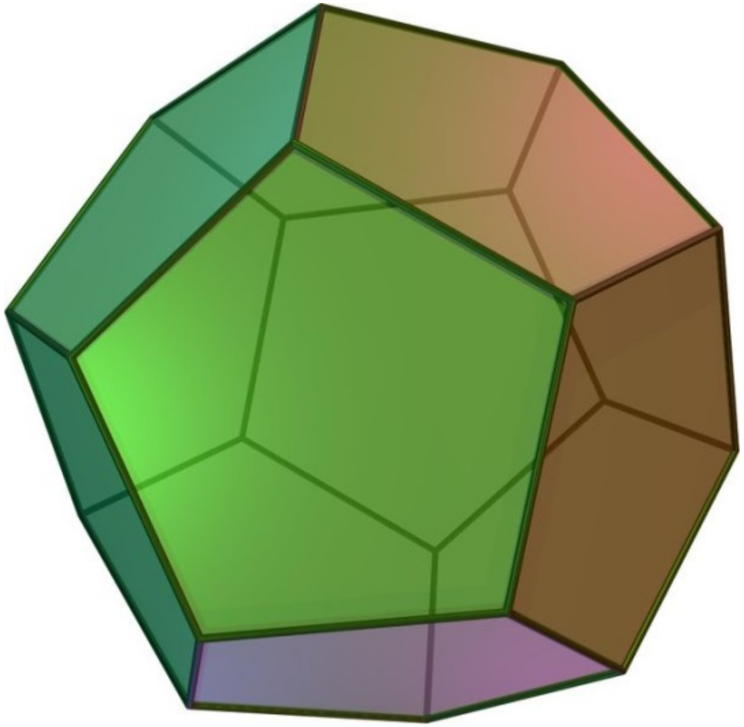
Figure 5.9: Solid Angle. The solid angle s subtended by an object c in three dimensions is computed by projecting c onto the unit sphere and measuring the area of its projection.

Solid angle and dodecahedron

Full sphere = 4π steradian = 12.566 sr

Dodecahedron

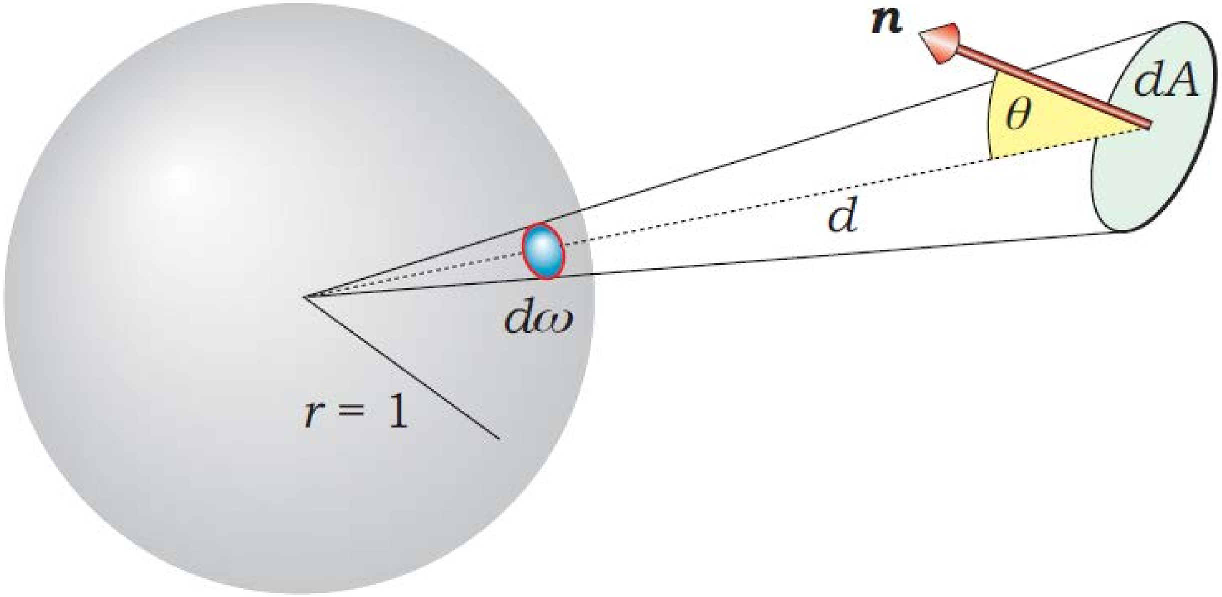
- 1 face = 1 steradian



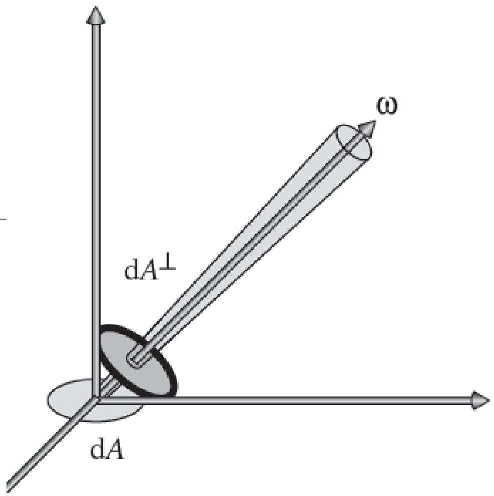
Solid angle

Differential solid angle

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



Radiance



Power per unit solid angle, per unit area

- Irradiance per unit solid angle
- Measures *light flow* near a point x and direction ω
- "Counts photons" that
  - (a) go through a little area around x perpendicular to ω and
  - (b) are traveling in directions that fall in a little solid angle $d\omega$ around ω

$$L(x \to \omega) = \frac{d^2 P}{dA^{\perp} d\omega} = \frac{d^2 P}{dA \cos\theta \, d\omega}$$

Symbol: L

Unit: Watt/m^2 sr

Radiance

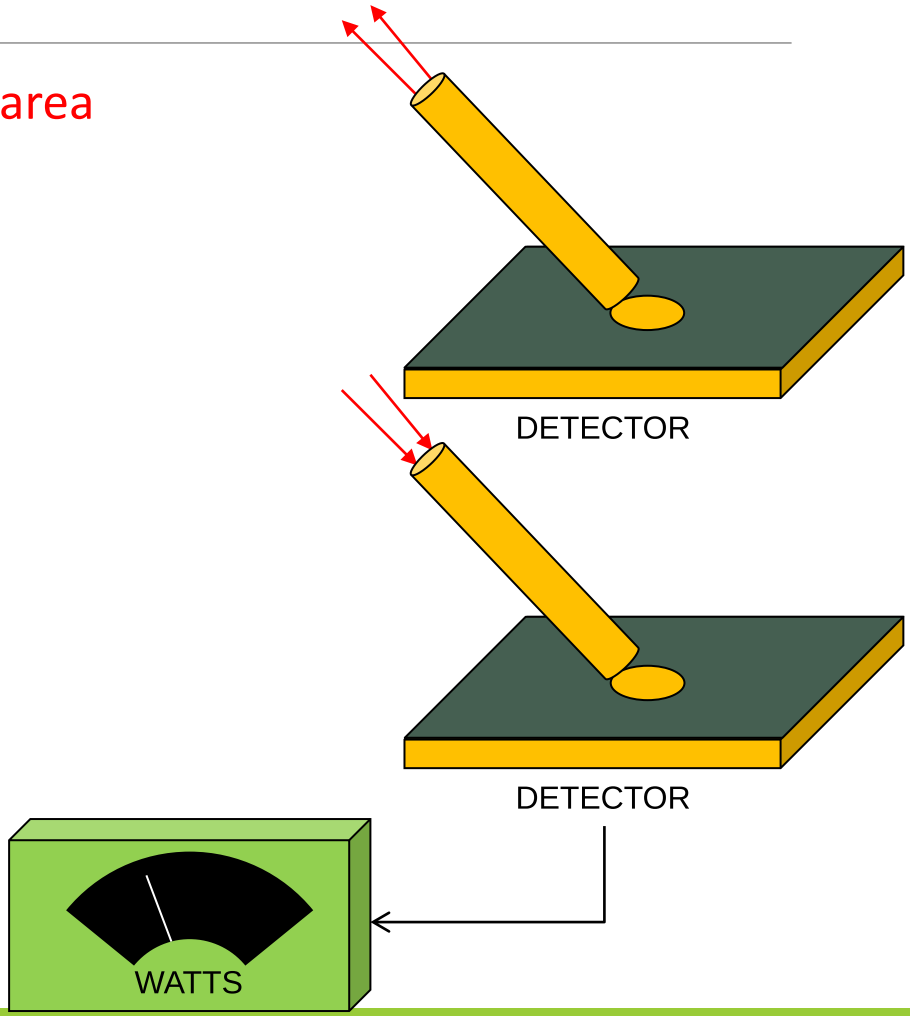
Power per unit solid angle, per unit area

- $L(x \to \omega)$
- x is position
- ω is direction

5D function

Defined for each wavelength

- (in practice r-g-b)



16

Some remarks w.r.t. notation

Outgoing radiance:

- $L_o(p, \omega)$ or $L(p \to \omega_o)$

Incoming radiance:

- $L_i(p, \omega)$ or $L(p \leftarrow \omega_o)$

All radiance is also wavelength dependent

- $L(p, \omega) \Rightarrow L(p, \omega, \lambda)$
- Often implicitly assumed as rgb-triplets

Integral over hemisphere of directions:

- 2D integral over hemispherical domain (see appendix)

Why is radiance important?

Radiance is invariant along a straight line
➔ we can trace light energy along rays

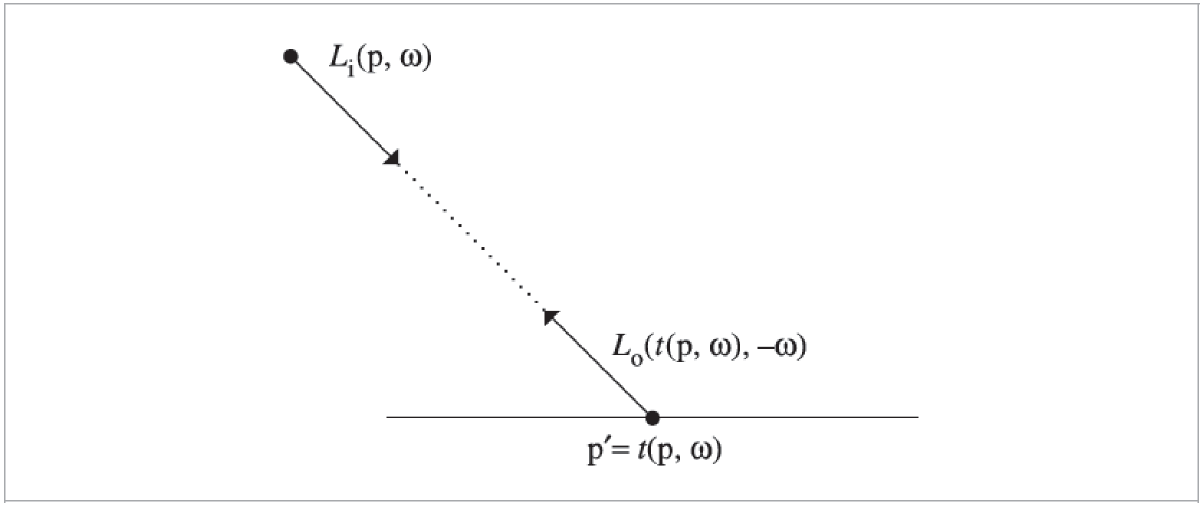
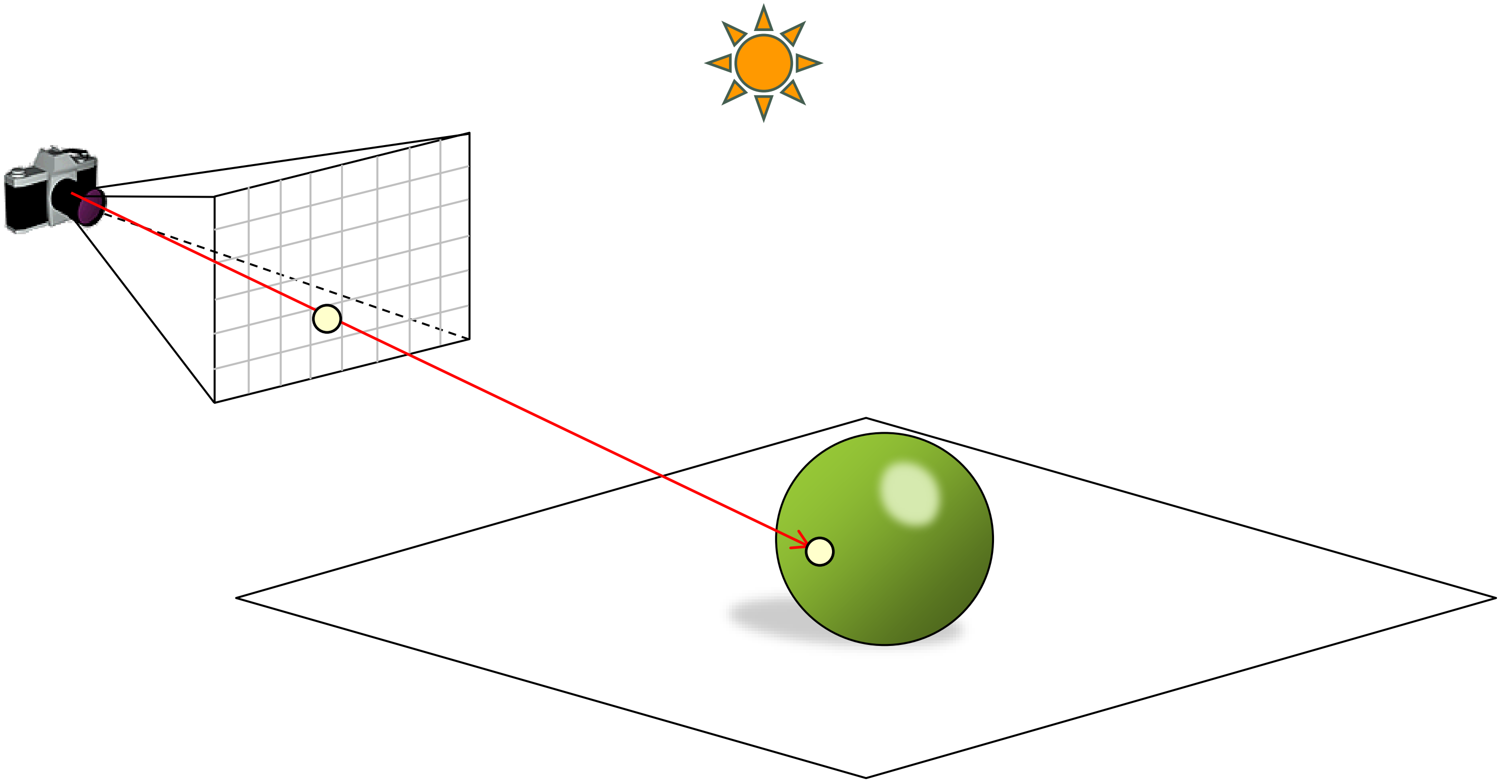
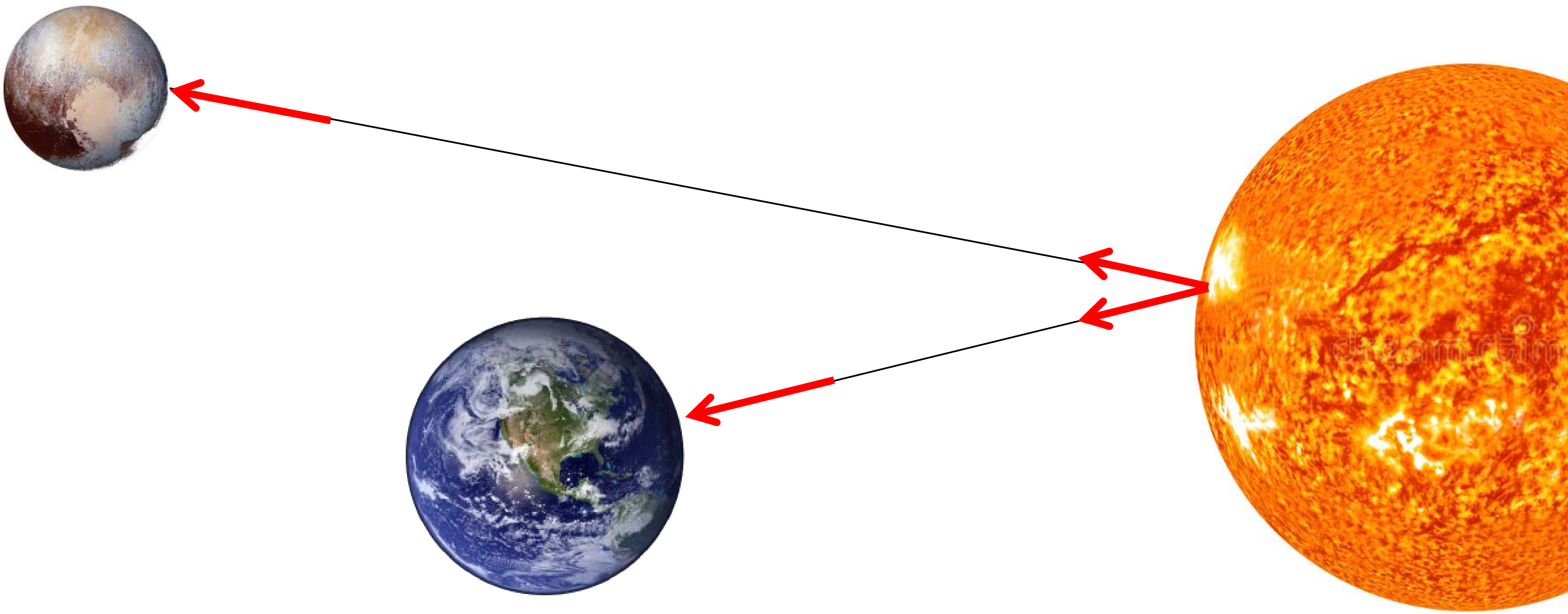


Figure 15.3: Radiance along a Ray through Free Space Is Unchanged. Therefore, to compute the incident radiance along a ray from point p in direction ω, we can find the first surface the ray intersects and compute exitant radiance in the direction $-\omega$ there. The trace operator $t(p, \omega)$ gives the point p' on the first surface that the ray (p, ω) intersects.

Why is radiance important?

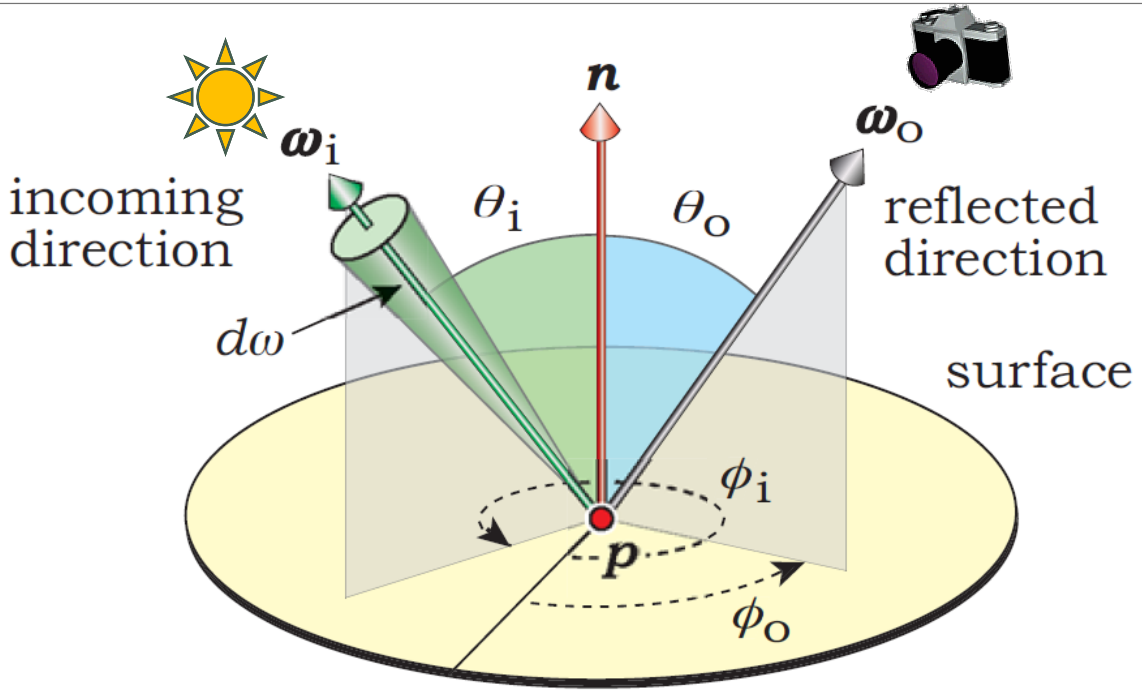


Why is radiance important?



Same radiance received from the Sun on the surface of Earth and Pluto?

BRDF – Bidirectional Reflectance Distribution Function



$$f_r(p, \omega_i, \omega_o) = \frac{dL_o(p, \omega_o)}{L_i(p, \omega_i) \cos\theta_i \, d\omega_i}$$

Properties of the BRDF

Reciprocity: $f_r(p, \omega_i, \omega_o) = f_r(p, \omega_o, \omega_i)$

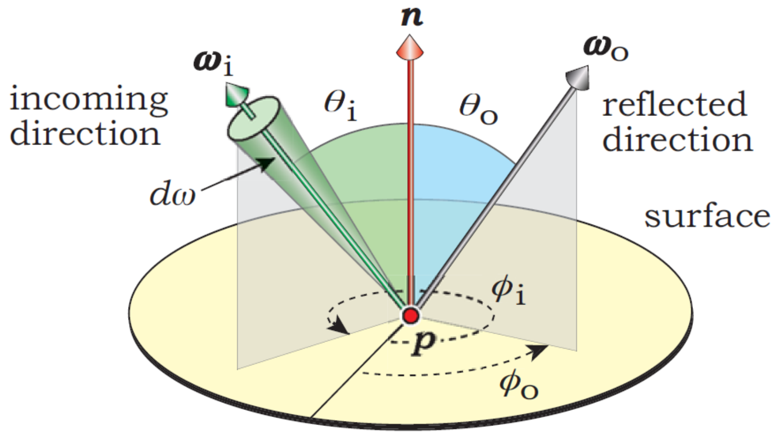
BRDF usually does not dependent on the intensity of light

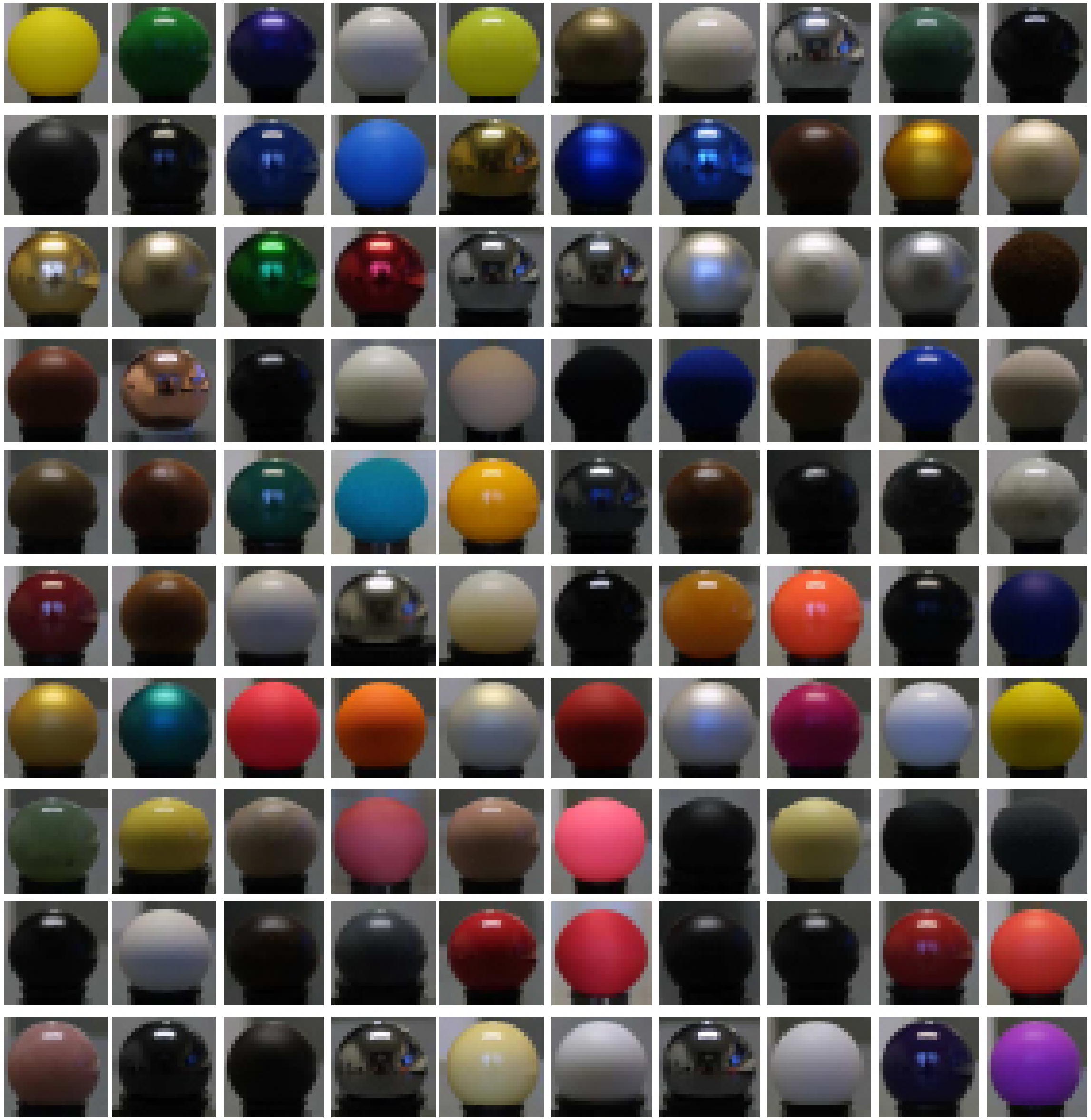
BRDF is dependent on wavelength

- $f_r(p, \omega_i, \omega_o) \Rightarrow f_r(p, \omega_i, \omega_o, \lambda)$

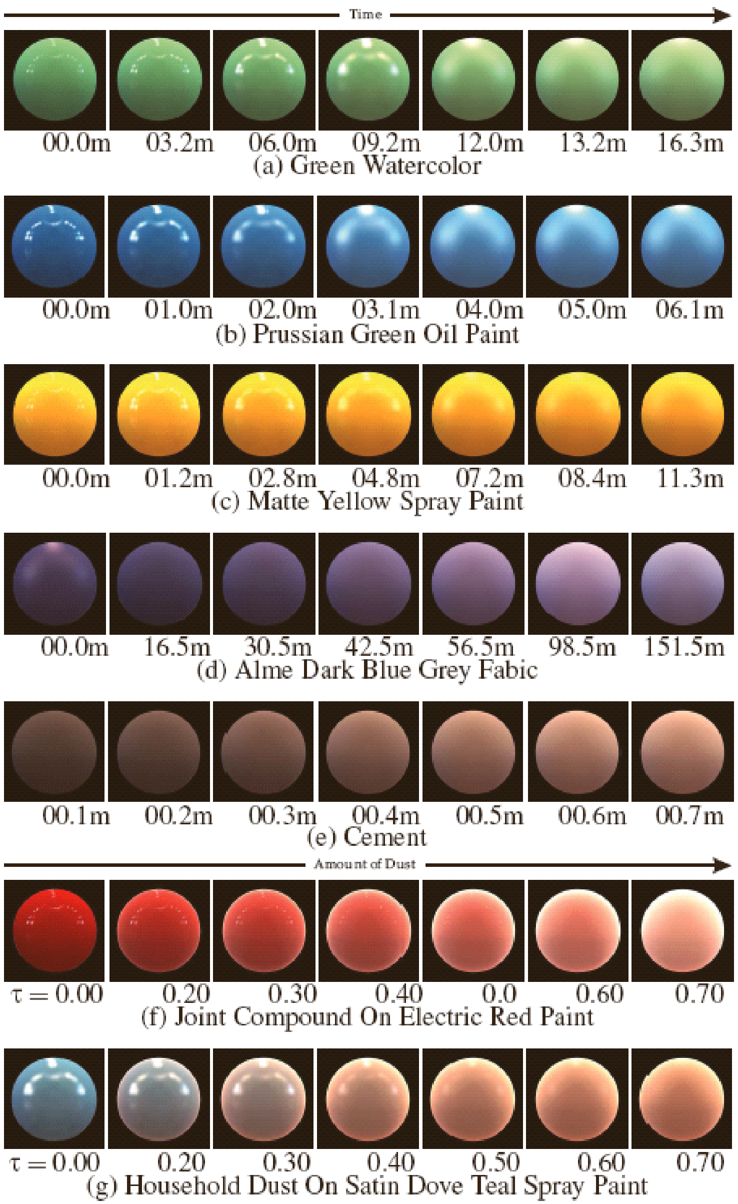
Conservation of energy

- All reflected energy < incoming energy
- $\forall \omega_o : \int_\Omega f_r(p, \omega_o, \omega') \cos\theta' d\omega' \leq 1$



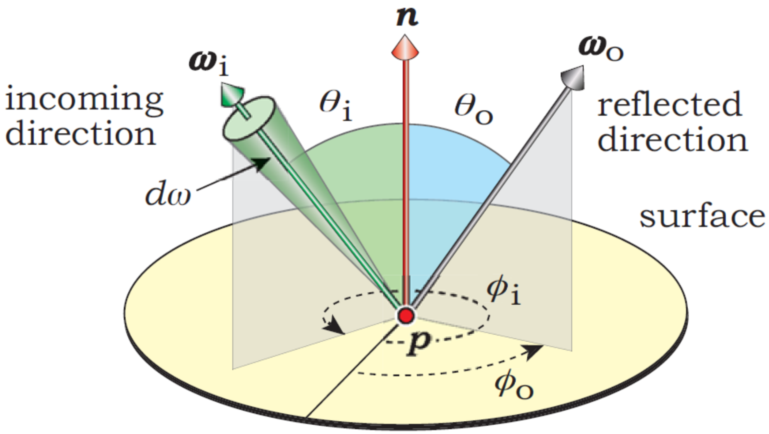


https://cdfg.csail.mit.edu/wojciech/brdfdatabase



https://www1.cs.columbia.edu/CAVE/databases/tvbrdf/about.php

Rendering Equation



$$f_r(p, \omega_i, \omega_o) = \frac{dL_o(p, \omega_o)}{L_i(p, \omega_i) \cos\theta_i \, d\omega_i}$$

$$dL_o(p, \omega_o) = f_r(p, \omega_i, \omega_o) L_i(p, \omega_i) \cos\theta_i \, d\omega_i$$

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

Rendering Equation

Dallas, August 18-22 Volume 20, Number 4, 1986

THE RENDERING EQUATION

James T. Kajiya
California Institute of Technology
Pasadena, Ca. 91125

ABSTRACT. We present an integral equation which generalizes a variety of known rendering algorithms. In the course of discussing a monte carlo solution we also present a new form of variance reduction, called Hierarchical sampling and give a number of elaborations shows that it may be an efficient new technique for a wide variety of monte carlo procedures. The resulting rendering algorithm extends the range of optical phenomena which can be effectively simulated.

The equation is very much in the spirit of the radiosity equation, simply balancing the energy flows from one point of a surface to another. The equation states that the transport intensity of light from one surface point to another is simply the sum of the emitted light and the total light intensity which is scattered toward x from all other surface points. Equation (1) differs from the radiosity equation of course because, unlike the latter, no assumptions are made about reflectance characteristics of the surfaces involved.

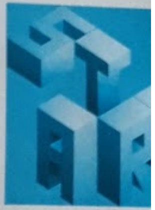
Each of the quantities in the equation are new quantities which we call

Figure 6. A sample image. All objects are neutral grey. Color on the objects is due to caustics from the green glass balls and color bleeding from the base polygon.

EQUATION

Illuminated Mathematics

QUANTIFYING LIGHT IS THE KEY TO GREAT 3-D ANIMATION.



Light is one of the most challenging aspects of computer-generated imagery. It bounces around, reflects off surfaces, and changes color; if you don't get all those elements right, your animation will look no more real than Bugs Bunny—not necessarily bad, but not what the folks at places like Pixar are going for. In the mid-'80s, mathematicians quantified this process: If you know the intensity and color of the light arriving at a particular spot, you can figure out the same information for the light that bounces off. Here's the equation at the heart of every 3-D rendering package. —JULIE REHMEYER

$L_0(\lambda, x)$ The intensity in watts of a particular color (wavelength λ) of light from spot x to the viewer's eyeball.

L_e Light emitted from x. Unless some object—a lamp, the sun, a firefly, phosphorescent algae—is emitting a glow at x, this will be zero.

Ω A hemisphere around x. Each point on Ω represents one of the possible directions from which light could hit x.

$\int$ Strain your brain back to calculus and you may remember that this is a fancy summation—adding up infinitely many infinitely tiny things. Here, you take Ω—every possible angle light may travel to x—compute how much each unit of that light bounces toward your eye, and then add 'em all up.

$$L_0(\lambda,x) = L_e + \int_\Omega f \, L_i \cos\theta \, d\omega$$

f This describes how much of the light that hits a material will reflect in a specific direction. A red surface, for example, will absorb every other color of light but red. A matte surface will spread light in all directions, reflecting only a bit toward your eye, while a mirrored surface will bounce all of the light in a single direction.

L_i The amount of incoming light of wavelength λ that strikes x from a particular direction.

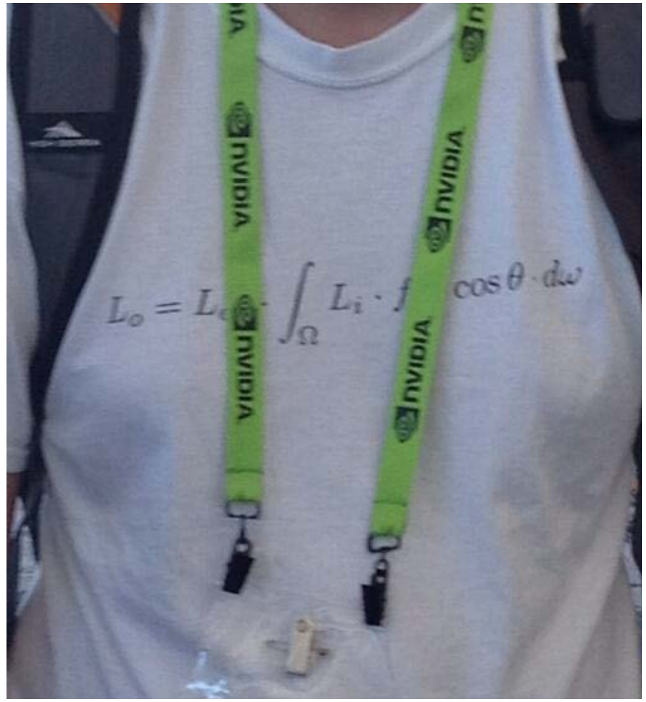
$\cos\theta$ An adjustment for the angle of the light. Just as the low winter sun offers less heat than the high summer sun, less of the light coming from a low angle will bounce off x toward the eye than that arriving from a high angle.

$d\omega$ An infinitely small section of the hemisphere Ω around x—in other words, just one of the directions from which light could strike spot x.

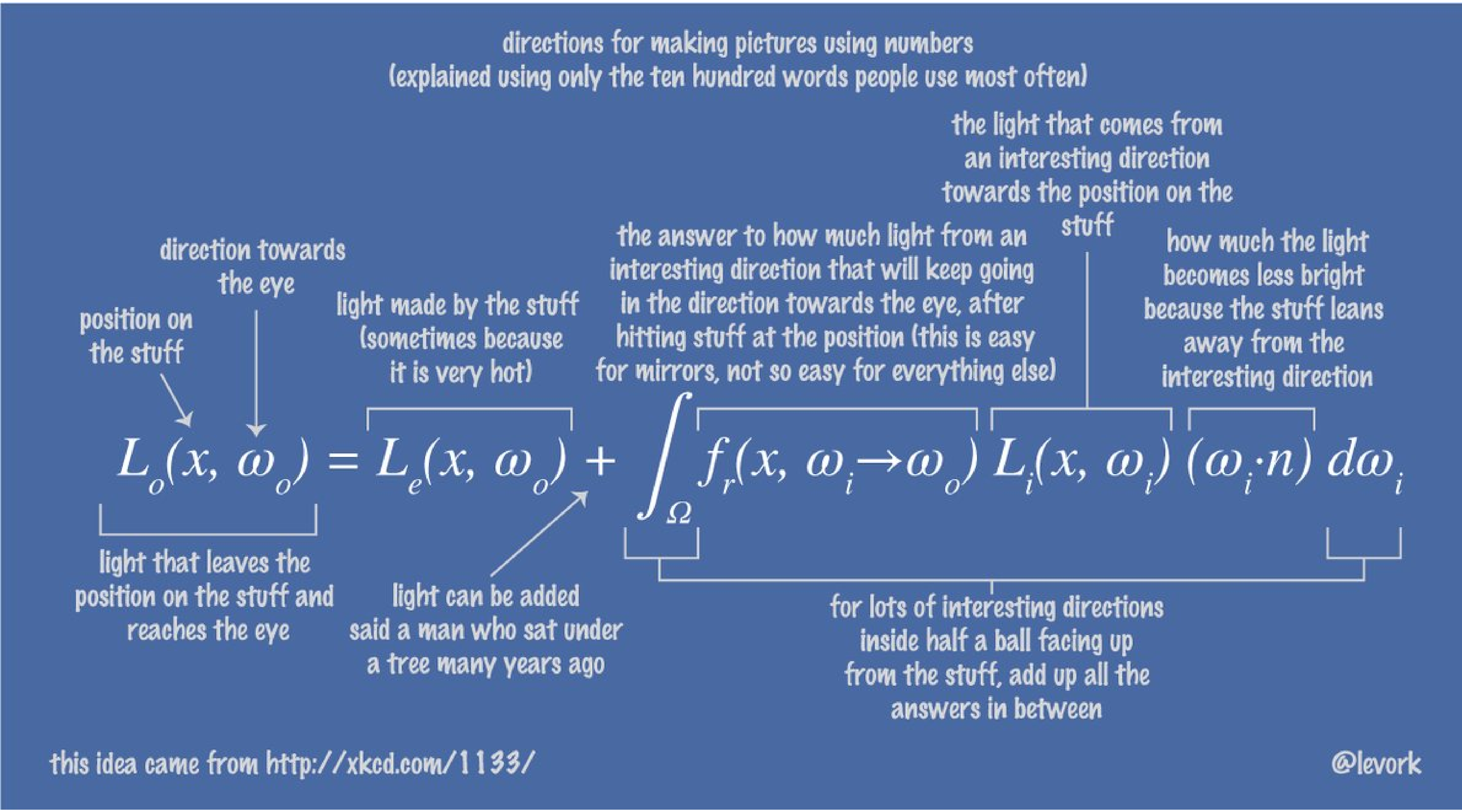
Software for making animated films like *Despicable Me* relies on the equation above to model how light interacts with objects.

(WIRED magazine, July 2010)

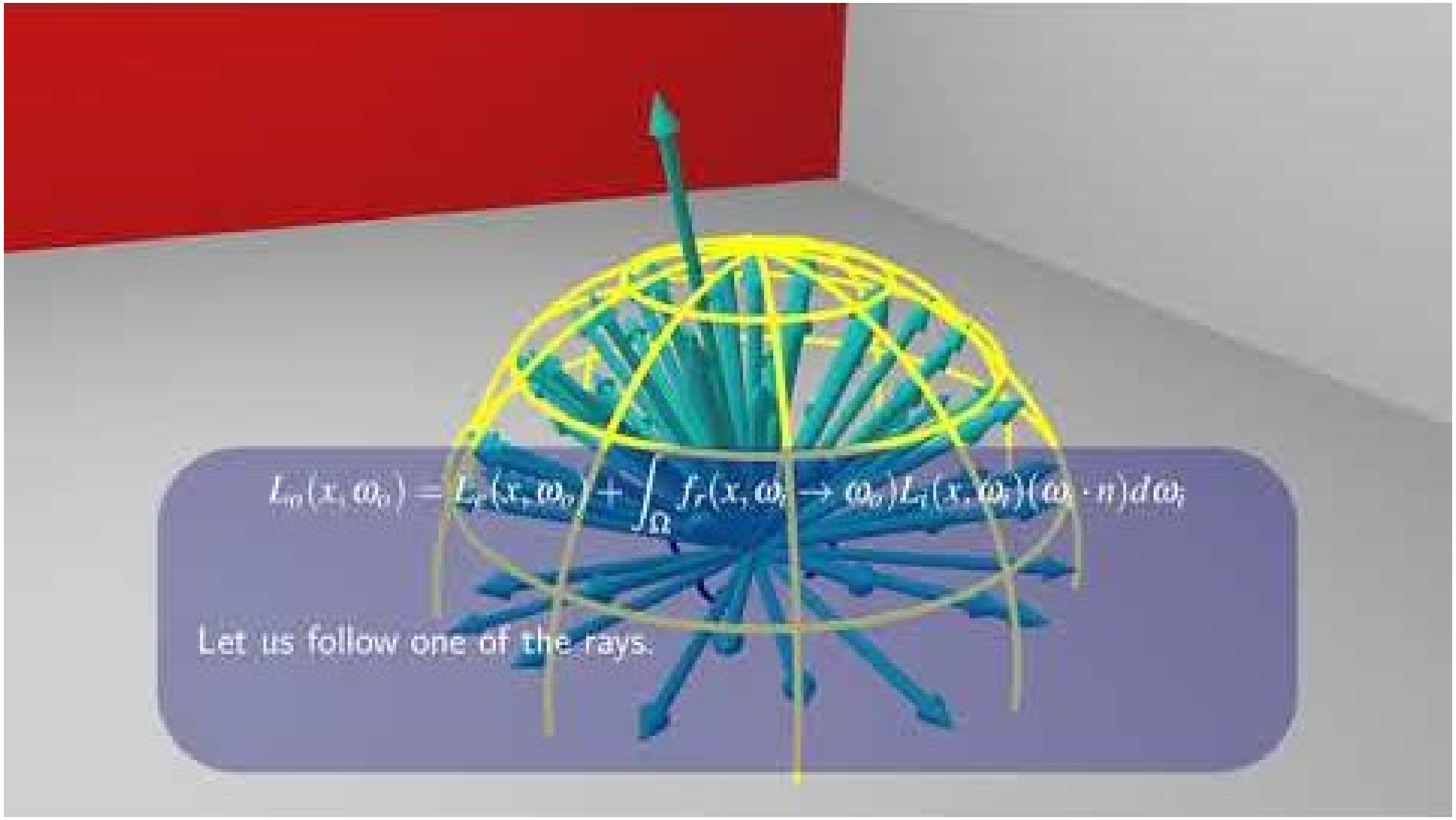

Kajiya said ...
... and there was global illumination.


NVIDIA


$L_o = L_e + \int_\Omega L_i \cdot f_r \cdot \cos\theta \cdot d$
TOKYO


directions for making pictures using numbers
(explained using only the ten hundred words people use most often)
the light that comes from an interesting direction towards the position on the stuff
direction towards the eye
position on the stuff
light made by the stuff (sometimes because it is very hot)
the answer to how much light from an interesting direction that will keep going in the direction towards the eye, after hitting stuff at the position (this is easy for mirrors, not so easy for everything else)
how much the light becomes less bright because the stuff leans away from the interesting direction
$L_o(x, \omega_o) = L_e(x, \omega_o) + \int_\Omega f_r(x, \omega_i \rightarrow \omega_o)\, L_i(x, \omega_i)\, (\omega_i \cdot n)\, d\omega_i$
light that leaves the position on the stuff and reaches the eye
light can be added said a man who sat under a tree many years ago
for lots of interesting directions inside half a ball facing up from the stuff, add up all the answers in between
this idea came from http://xkcd.com/1133/
@levork

Evaluating the Rendering Equation



https://www.youtube.com/watch?v=eo_MTI-d28s

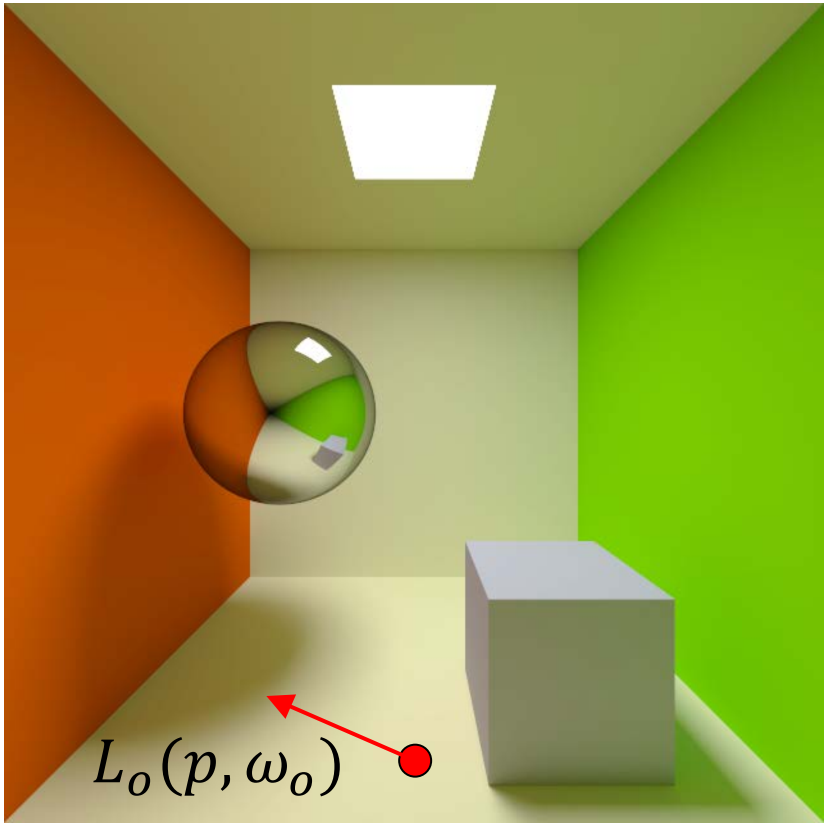
Simple Stochastic Ray Tracing (aka Pure Path Tracing, Naïve Path Tracing, …)

$L_o(p, \omega_o) = ?$

$L_e(p, \omega_o)$

$+$

$\int_\Omega f_r(p, \omega_i, \omega_o) L_i(p, \omega_i) \cos\theta_i \, d\omega_i$



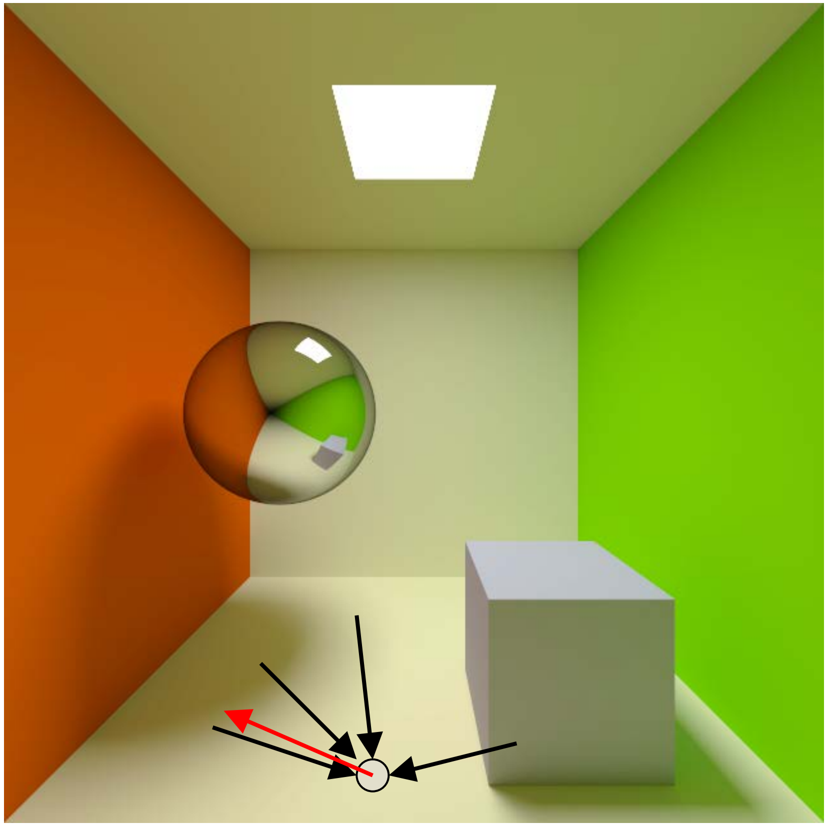
(rendering L. Vanbesien)

Simple Stochastic Ray Tracing

Generate N random directions ω_j on hemisphere Ω around p, using probability density $p(\omega_j)$

$$\int_\Omega f_r(p,\omega_i,\omega_o)L_i(p,\omega_i)\cos\theta_i \, d\omega_i$$

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



(rendering L. Vanbesien)

Simple Stochastic Ray Tracing

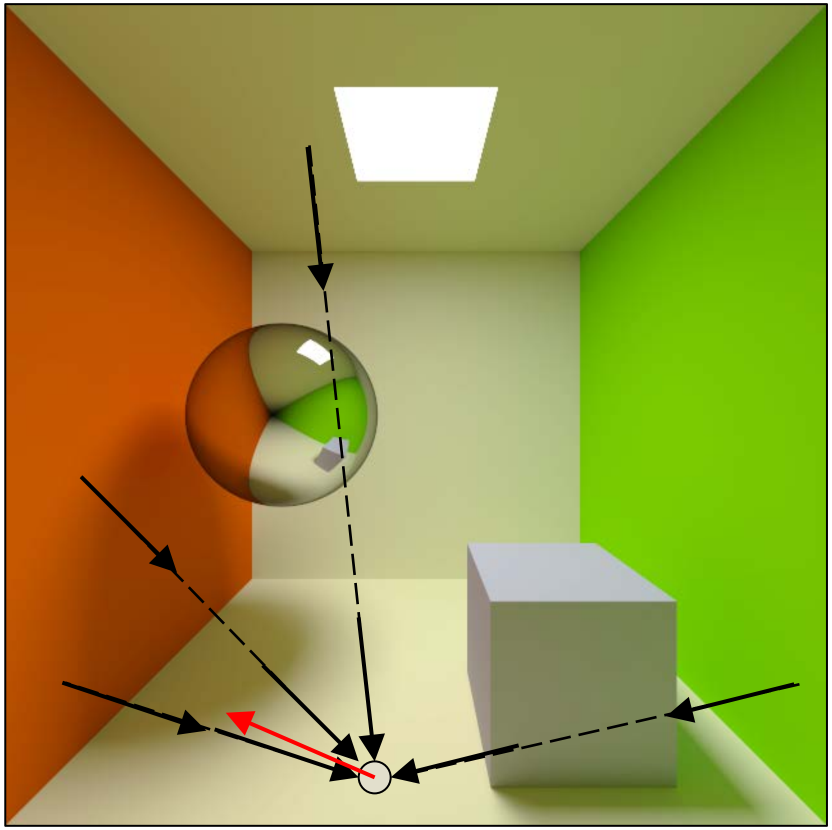
Generate N random directions ω_j on hemisphere Ω around p, using probability density $p(\omega_j)$

$$\int_\Omega f_r(p, \omega_i, \omega_o) L_i(p, \omega_i) \cos\theta_i \, d\omega_i$$

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

Evaluate $L_i(p, \omega_j)$?

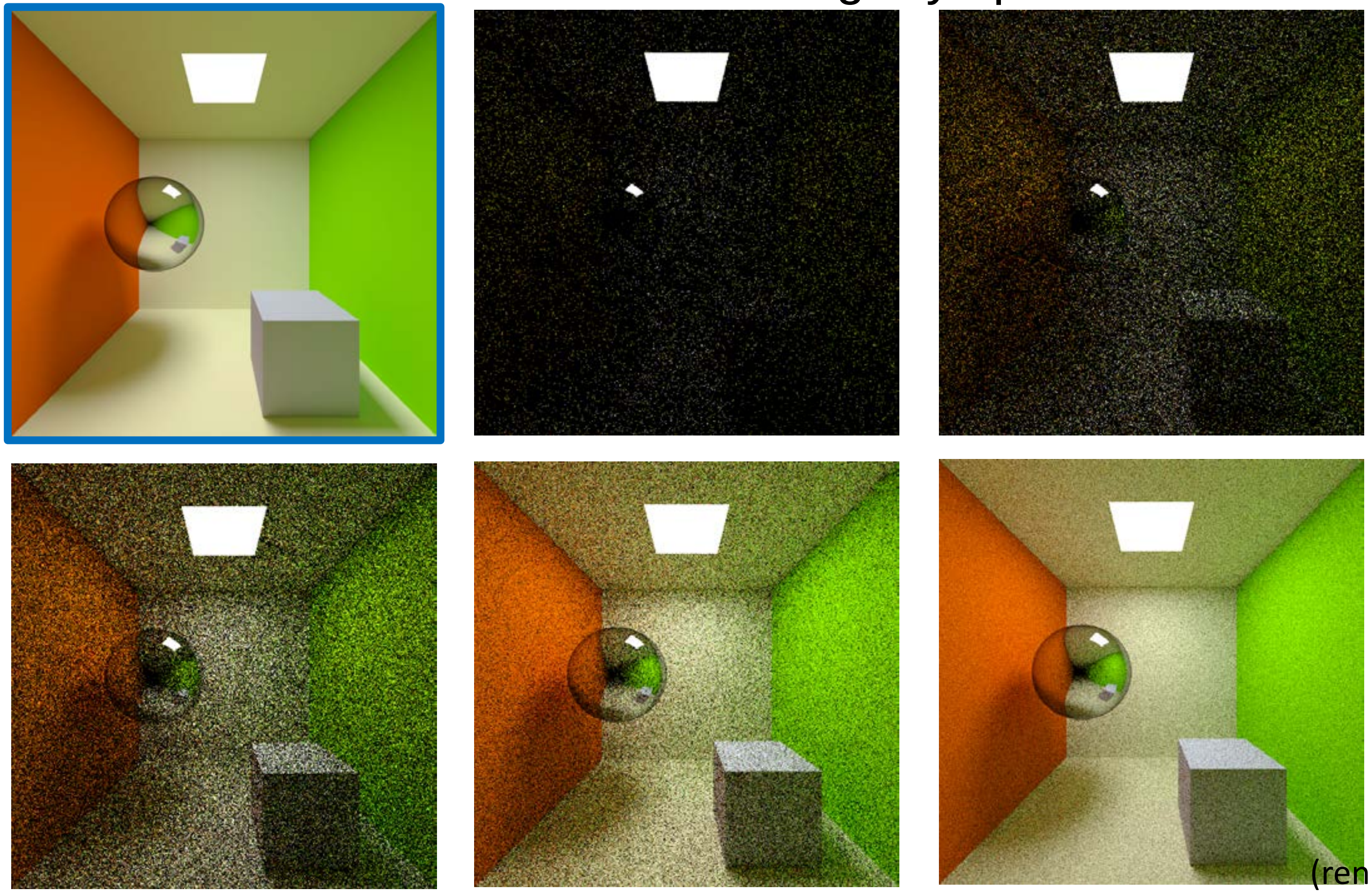
- Radiance is invariant along straight lines
- Trace ray to find closest intersection
- When path hits light source, we have a contribution
- Recursion



(rendering L. Vanbesien)

Simple Stochastic Ray Tracing

Reference - 1 - 4 - 16 - 64 - 256 viewing rays/pixel. No branching.



(rendering L. Vanbesien)

Simple Stochastic Ray Tracing

Choices?

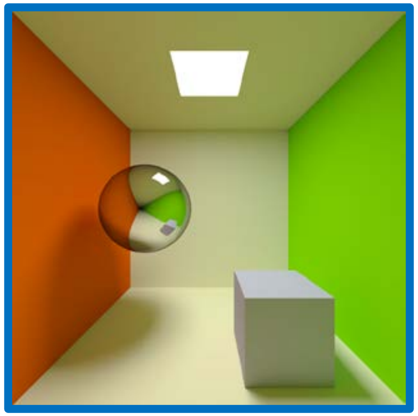
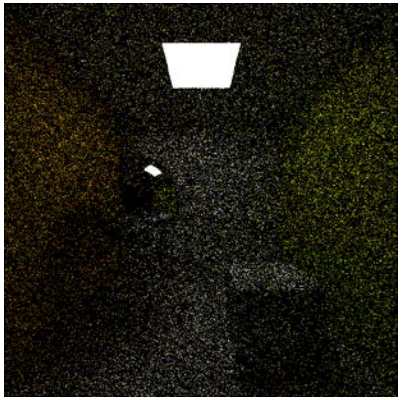
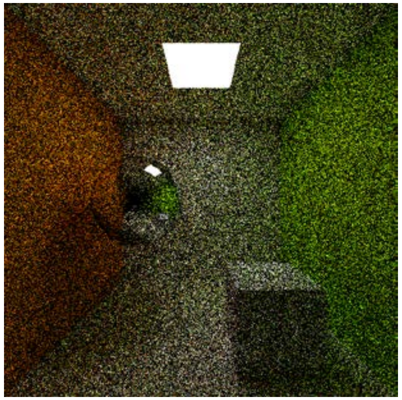
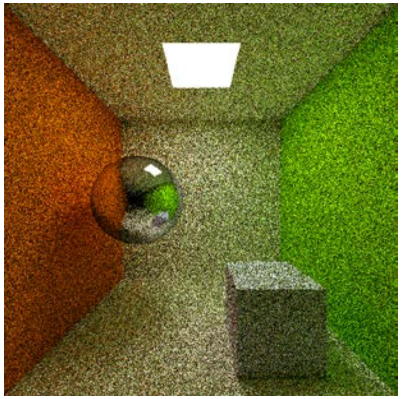
- # viewing rays per pixel (anti-aliasing)
- # random rays over the hemisphere for each surface point: "branching factor"

Algorithm known as "path tracing":

- Branching factor = 1
- Multiple viewing rays per pixel, but each ray branches only once at each reflection
- 'path that is reflected through the scene'

Simple Stochastic Ray Tracing Branching factor

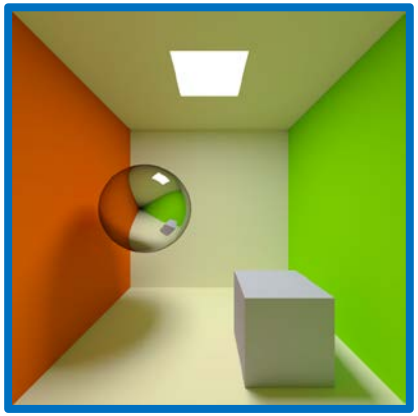
Reference 4 viewing rays/pixel
Branching factor 1 - 2 - 4

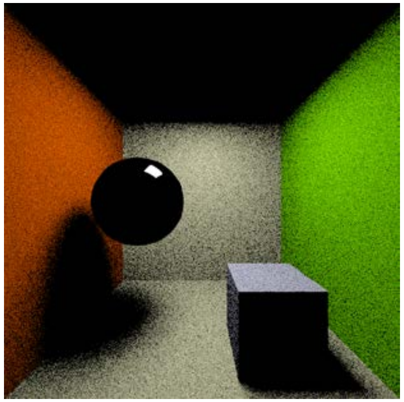
(rendering L. Vanbesien)

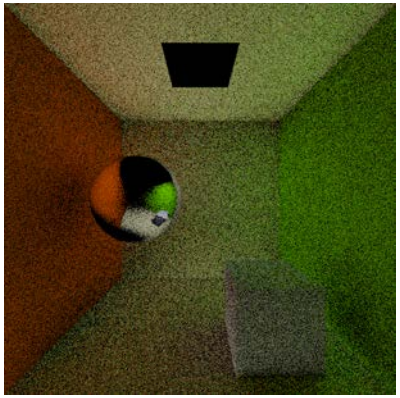
Simple Stochastic Ray Tracing
Depth of Recursion

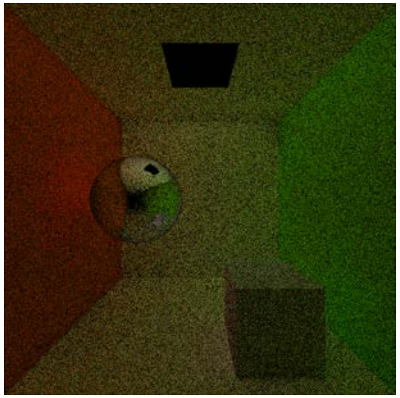
Reference



Recursion depth 1 - 2 - 3 is shown (same brightness)







Russian Roulette

Choose ‘absorption probability’ α

- probability α that ray will *not be reflected* ➔ in this case, we stop the recursion
- probability $1-\alpha$ that ray will be reflected ➔ recursive evaluation is multiplied by $1/(1-\alpha)$

E.g. $\alpha = 0.9$ $(1-\alpha = 0.1)$

- 10% probability that ray is reflected
- If the ray is reflected, radiance along reflected ray ‘counts’ for 1 / 0.1 = 10 times as much

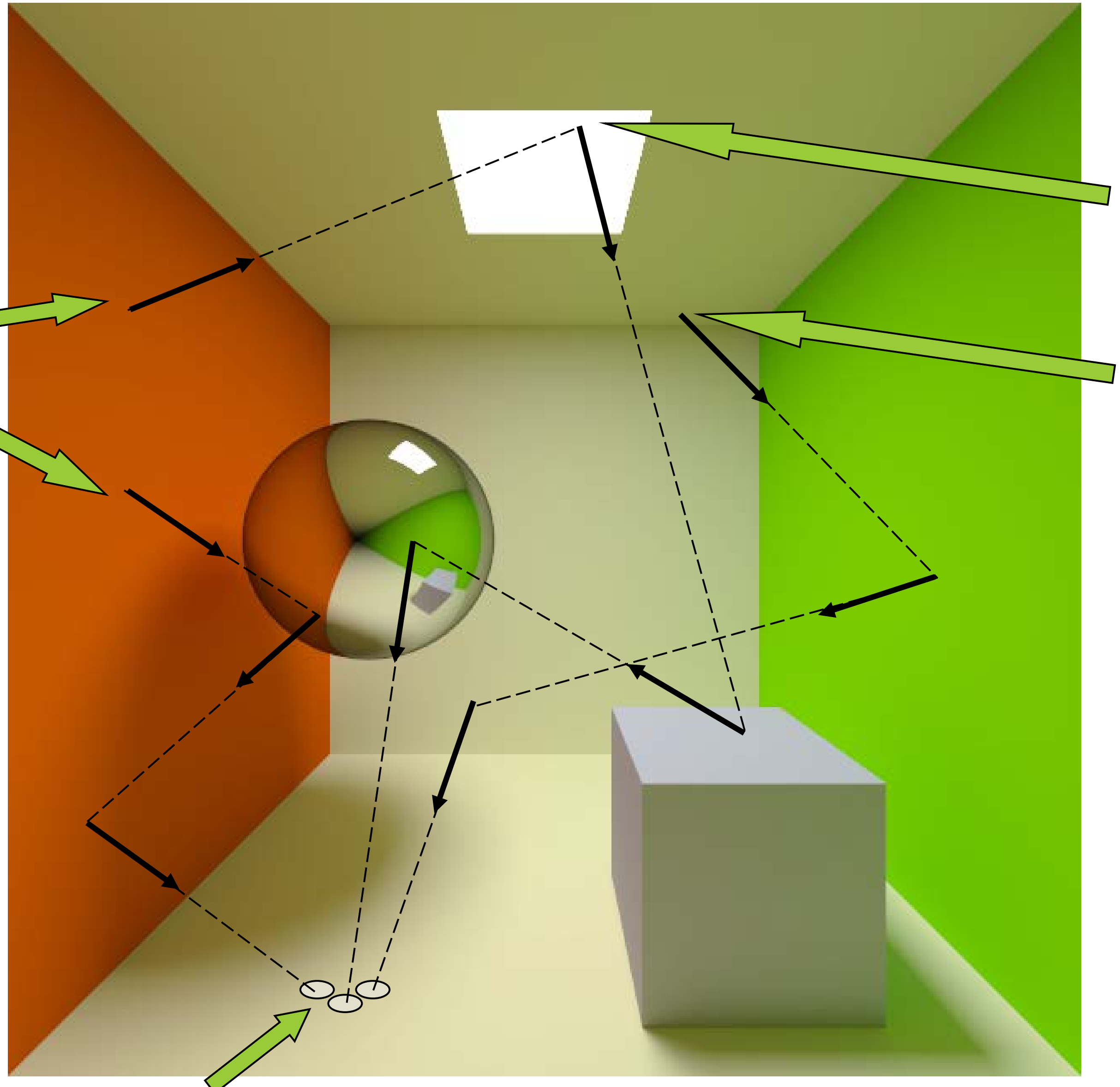
E.g. $\alpha = 0.7$ (or $1-\alpha = 0.3$)

- 30% probability that ray is reflected
- If the ray is reflected, radiance along reflected ray ‘counts’ for 1 / 0.3 = 3.3333 times as much

Trivial stop-condition if $f_r = 0$ (set $\alpha = 1$)

- E.g. light sources are often modeled with $f_r = 0$

Russian Roulette ends recursion



Light source contribution

Russian Roulette ends recursion

multiple viewing rays result in multiple visible points through pixel