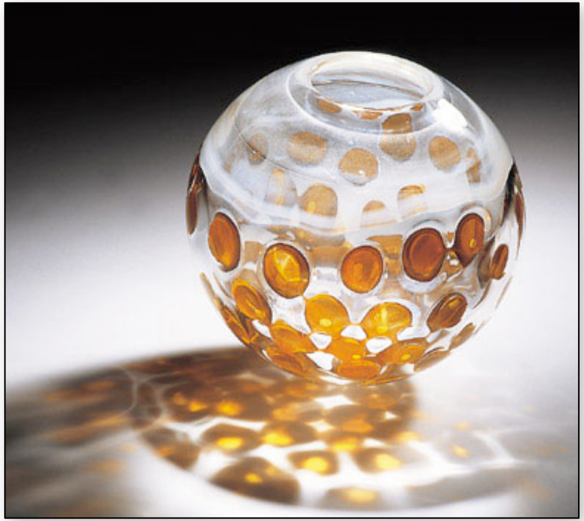
The Rendering Equation

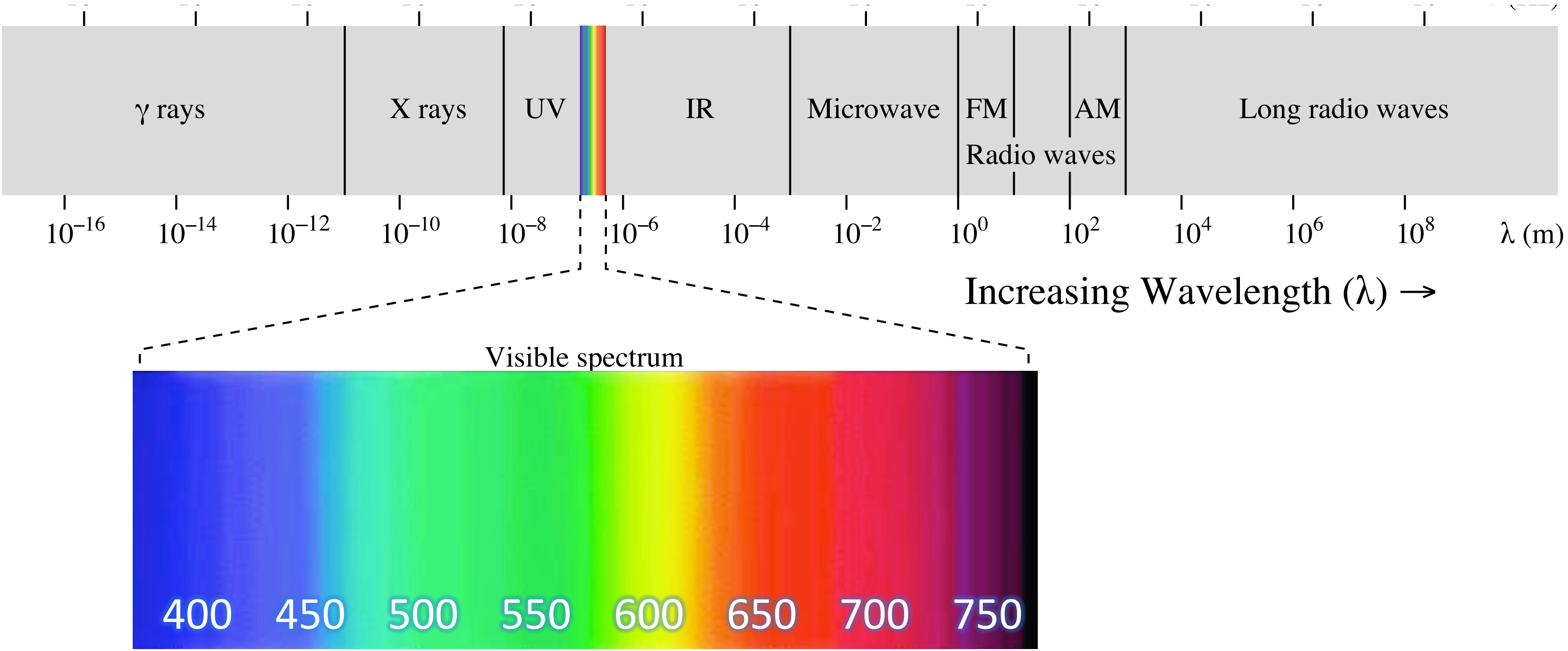
Cengiz Öztireli

UNIVERSITY OF CAMBRIDGE

Rendering – Simulating Light



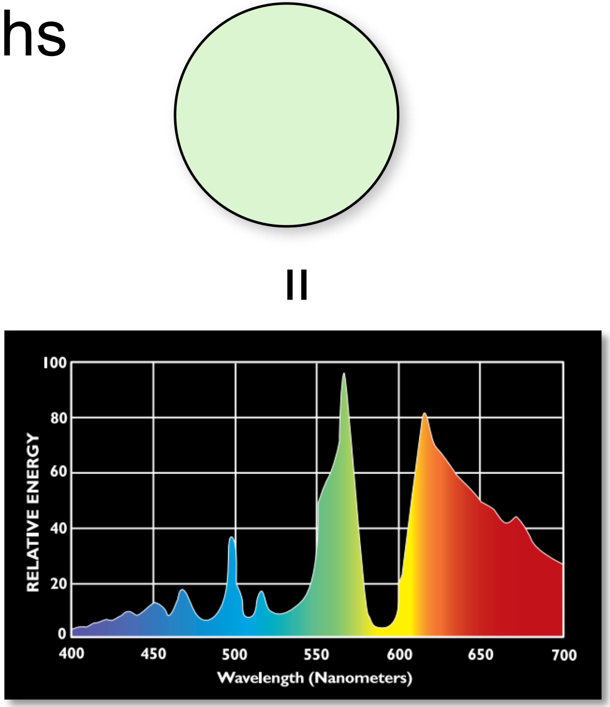
Light and Colors



UNIVERSITY OF CAMBRIDGE

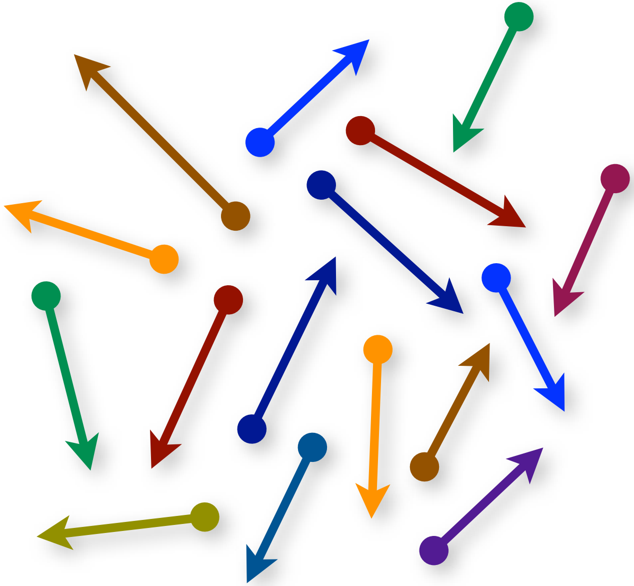
Light and Colors

- Light can be a mixture of many wavelengths
- Spectral power distribution (SPD)
  - P(λ) = intensity at wavelength λ
  - intensity as a function of wavelength
- We perceive these distributions as colors



Measuring Light

- How do we measure light



Measuring = Counting photons

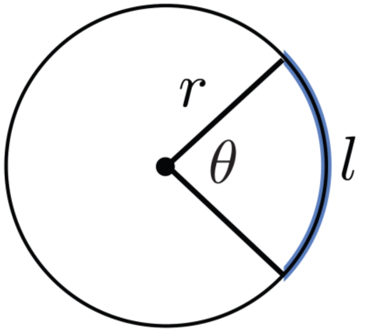
Measuring Light

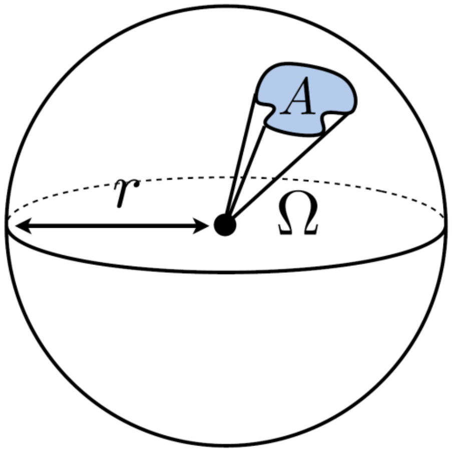
- Radiometry
  - Studies the measurement of electromagnetic radiation, including visible light

Basic Definitions

Solid angle is defined as the ratio of the area covered on a sphere by an object to the area given by the square of the radius of the sphere.

- Angle: $\theta = \dfrac{l}{r}$
  - circle: 2π radians
- Solid angle: $\Omega = \dfrac{A}{r^2}$
  - sphere: 4π steradians



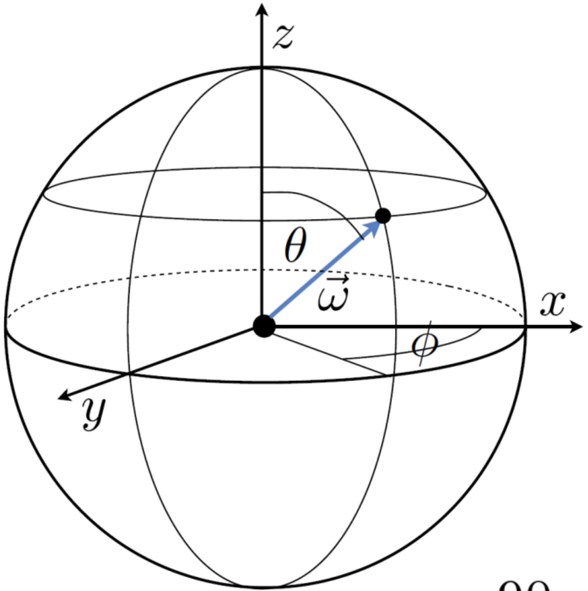


Basic Definitions

- Direction
  - point on the unit sphere
  - parameterized by two angles

$$\vec{\omega} = (\theta, \phi)$$

zenith (θ) azimuth (ϕ)



$$\vec{\omega}_x = \sin\theta\cos\phi$$

$$\vec{\omega}_y = \sin\theta\sin\phi$$

$$\vec{\omega}_z = \cos\theta$$

$$latitude = \frac{90}{\pi}(\pi - \theta) \qquad longitude = \frac{90}{\pi}\phi$$

UNIVERSITY OF CAMBRIDGE

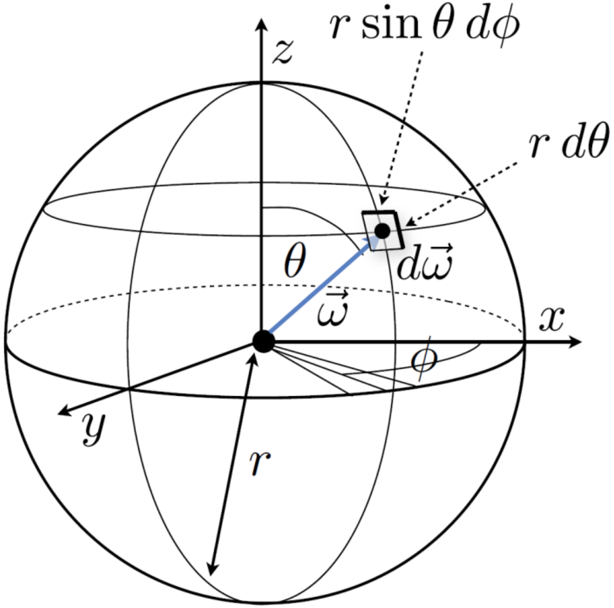
Basic Definitions
Differential version of an area on a sphere is defined by considering a very small square on the sphere.
The differential solid angle can then be defined by dividing the differential area by squared radius of the sphere.
Note that we denote the differential solid angle with the same symbol as direction.
Integrating the differential solid angle gives the area of the unit sphere as expected.

- Differential Solid Angle

$$dA = (rd\theta)(r\sin\theta d\phi)$$

$$d\vec{\omega} = \frac{dA}{r^2} = \sin\theta d\theta d\phi$$

$$\Omega = \int_{S^2} d\vec{\omega} = \int_0^{2\pi}\int_0^{\pi} \sin\theta d\theta d\phi = 4\pi$$



Basic Definitions

- Assume light consists of photons with
  - $\mathbf{x}$: Position
  - $\vec{\omega}$: Direction of motion
  - λ : Wavelength

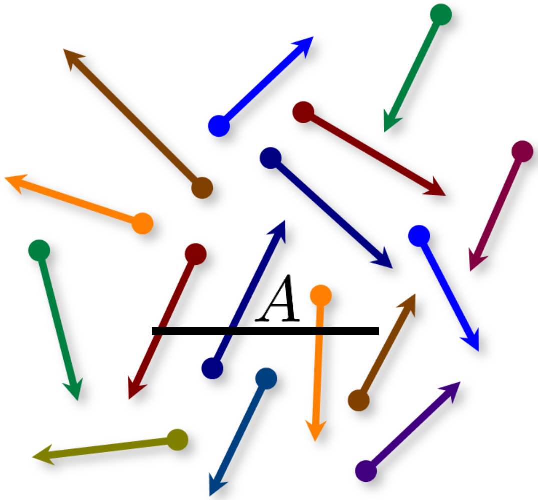
- Each photon has an energy of: $\dfrac{hc}{\lambda}$
  - $h \approx 6.63 \cdot 10^{-34}\, m^2 \cdot kg/s$: Planck's constant
  - $c = 299{,}792{,}458\, m/s$: speed of light in vacuum
  - Unit of energy, Joule : $[J = kg \cdot m^2/s^2]$

Radiometry

- Flux (radiant flux, power)
  - total amount of energy passing through surface or space per unit time

$$\Phi(A) \qquad \left[\frac{J}{s} = W\right]$$

  - examples:
    - number of photons hitting a wall per second
    - number of photons leaving a lightbulb per second

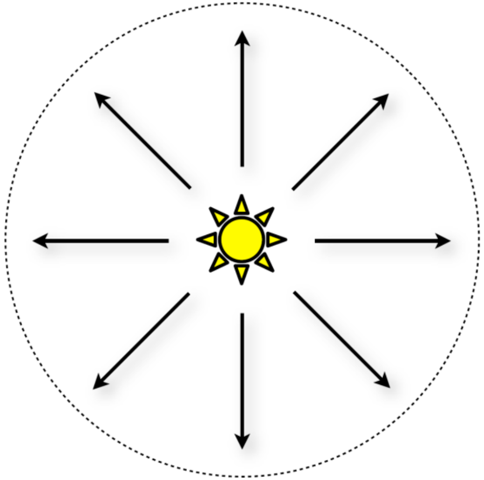


Radiometry
Radiant intensity is defined with differential flux and solid angle.
Intuitively, it is a very local measure of energy per unit time per solid angle.

- Radiant intensity
  - Power (flux) per solid angle = directional density of flux

$$I(\vec{\omega}) = \frac{d\Phi}{d\vec{\omega}} \qquad \left[\frac{W}{sr}\right] \qquad \Phi = \int_{S^2} I(\vec{\omega})d\vec{\omega}$$

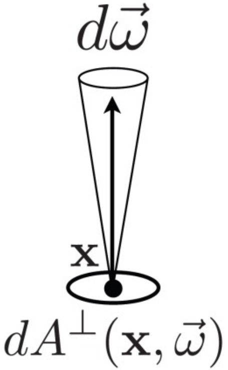
  - example:
    - power per unit solid angle emanating from a point source



Radiometry

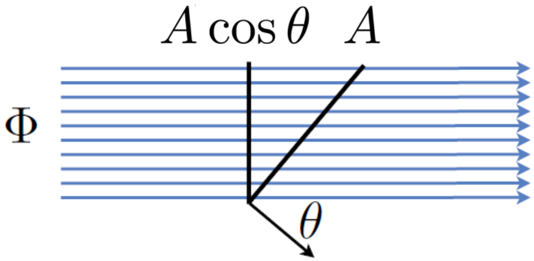
Radiance is the most important quantity. It is energy per unit time per solid angle per unit **perpendicular** area. Intuitively, due to the cosine term, it is independent of the angle the actual area makes with the light direction. This makes sense because a large area that is tilted with respect to the light direction does not get more flux. Radiance remains constant along a ray. We will utilize it in the rendering equation.

- Radiance
  - Radiant intensity per perpendicular unit area



$$L(\mathbf{x}, \vec{\omega}) = \frac{d^2\Phi(A)}{d\vec{\omega}\, dA^{\perp}(\mathbf{x}, \vec{\omega})}$$

$$= \frac{d^2\Phi(A)}{d\vec{\omega}\, dA(\mathbf{x}) \cos\theta} \quad \left[\frac{W}{m^2 sr}\right]$$

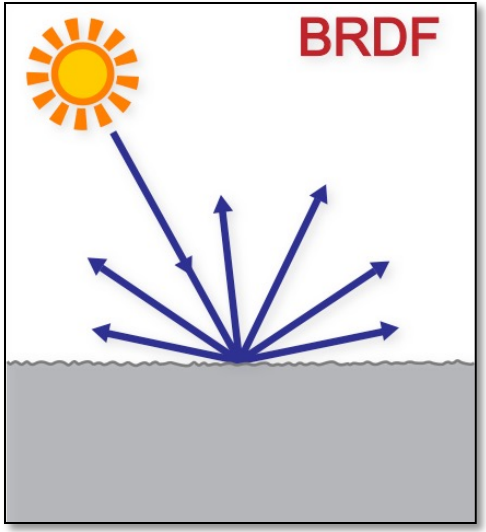


  - remains constant along a ray

Reflection Models

Now that we have studied how to measure light, we can slowly go into how to use the theory in practice. BRDF and its variants are the most commonly used functions to represent and render light in a digital scene. With BRDF, we only consider reflection, i.e. no light passes through any surface in a scene. Surfaces only reflect.

- **B**idirectional **R**eflectance **D**istribution **F**unction (BRDF)



BRDF

The BRDF function relates infinitesimal reflected radiance to incoming radiance.
Intuitively, it measures how much light is reflected in the direction w_r for light coming in the direction w_i.
This can be defined for each point on a surface.

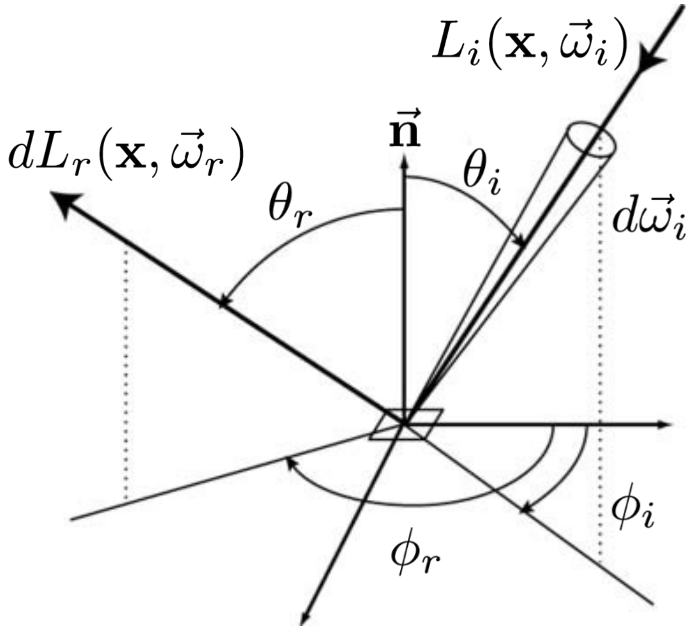
- **B**idirectional **R**eflectance **D**istribution **F**unction

$$f_r(\mathbf{x}, \vec{\omega}_i, \vec{\omega}_r) = \frac{dL_r(\mathbf{x}, \vec{\omega}_r)}{L_i(\mathbf{x}, \vec{\omega}_i)\cos\theta_i d\vec{\omega}_i} \qquad [1/sr]$$

BRDF

infinitesimal reflected radiance

infinitesimal solid angle



Reflection Equation

We can finally define the reflection equation, which is simply the integral form of the previous expression. This tells us that the total reflected light for a given direction is the integral of light coming in all directions on the hemisphere H^2 with weighting terms given by the BRDF and the cosine of the incoming light angle. Note that we do not integrate over the whole sphere as we assume light is only reflected off a surface.

- The BRDF provides a relation between incident radiance and differential reflected radiance
- From this we can derive the **Reflection Equation**

$$f_r(\mathbf{x}, \vec{\omega}_i, \vec{\omega}_r) = \frac{dL_r(\mathbf{x}, \vec{\omega}_r)}{L_i(\mathbf{x}, \vec{\omega}_i) \cos\theta_i \, d\vec{\omega}_i}$$

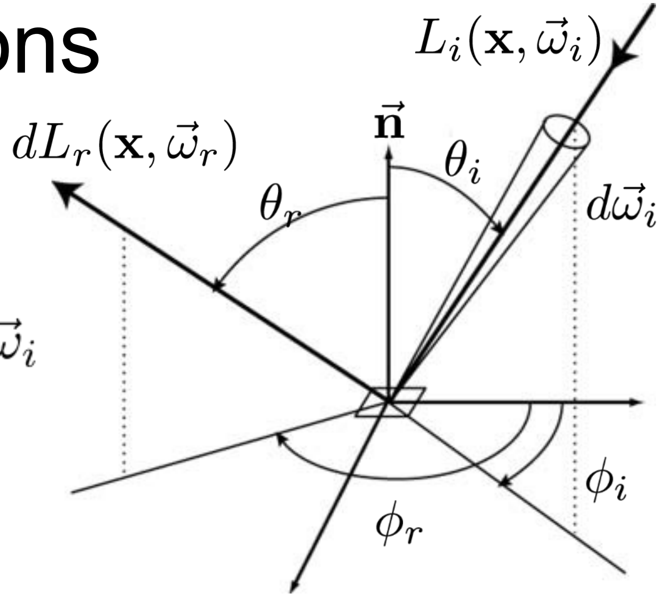
$$f_r(\mathbf{x}, \vec{\omega}_i, \vec{\omega}_r) L_i(\mathbf{x}, \vec{\omega}_i) \cos\theta_i = \frac{dL_r(\mathbf{x}, \vec{\omega}_r)}{d\vec{\omega}_i}$$

$$\int_{H^2} f_r(\mathbf{x}, \vec{\omega}_i, \vec{\omega}_r) L_i(\mathbf{x}, \vec{\omega}_i) \cos\theta_i \, d\vec{\omega}_i = L_r(\mathbf{x}, \vec{\omega}_r)$$

Reflection Equation

- The reflected radiance due to incident illumination from all directions

$$L_r(\mathbf{x}, \vec{\omega}_r) = \int_{H^2} f_r(\mathbf{x}, \vec{\omega}_i, \vec{\omega}_r) L_i(\mathbf{x}, \vec{\omega}_i) \cos\theta_i \, d\vec{\omega}_i$$



The Rendering Equation

The step from reflectance to rendering equation is adding a term that accounts for potentially emitted light. This makes sure that energy is conserved everywhere in the scene.

- The outgoing light is the sum of emitted and incoming

$$L_o(\mathbf{x}, \vec{\omega}_o) = L_e(\mathbf{x}, \vec{\omega}_o) + L_r(\mathbf{x}, \vec{\omega}_o)$$

$$\boxed{\underbrace{L_o(\mathbf{x}, \vec{\omega}_o)}_{\text{outgoing light}} = \underbrace{L_e(\mathbf{x}, \vec{\omega}_o)}_{\text{emitted light}} + \underbrace{\int_{H^2} f_r(\mathbf{x}, \vec{\omega}_i, \vec{\omega}_o) L_i(\mathbf{x}, \vec{\omega}_i) \cos\theta_i \, d\vec{\omega}_i}_{\text{reflected light}}}$$

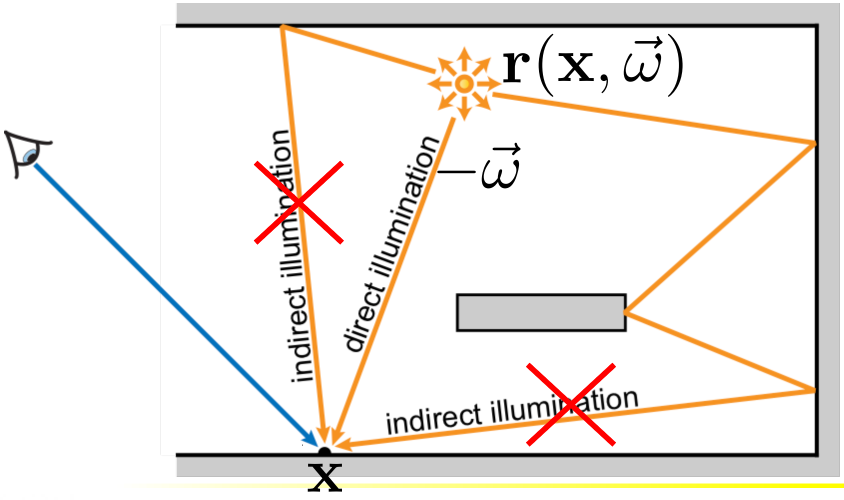
Energy is conserved!

Direct Illumination

How do we use the rendering equation in practice? One way is local or direct illumination.
In this model, light can only come from light sources, i.e. we ignore light reflected from a surface and landing on another surface.
This is useful for fast renderings and is what rasterizers such as the ones in OpenGL typically assume.

- All light comes directly from emitters, i.e. light sources

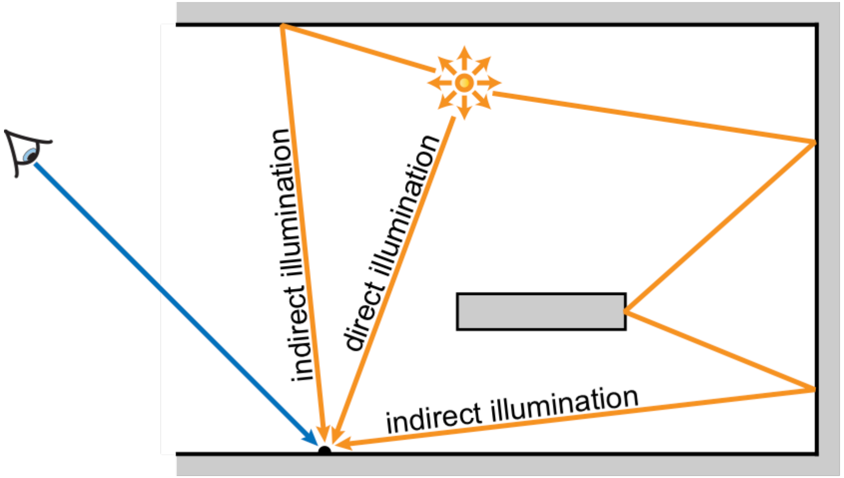
$$L_o(\mathbf{x}, \vec{\omega}_o) = L_e(\mathbf{x}, \vec{\omega}_o) + \int_{H^2} f_r(\mathbf{x}, \vec{\omega}_i, \vec{\omega}_o) L_i(\mathbf{x}, \vec{\omega}_i) \cos\theta_i \, d\vec{\omega}_i$$

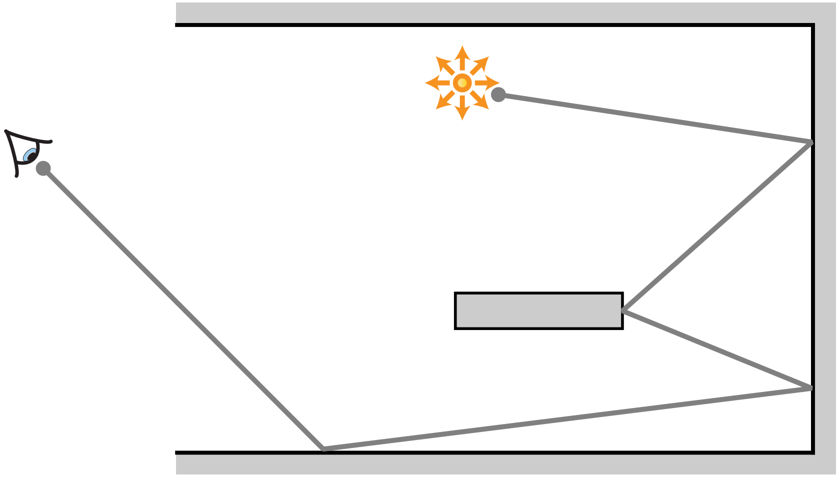


$$L_i(\mathbf{x}, \vec{\omega}) = L_e(\mathbf{r}(\mathbf{x}, \vec{\omega}), -\vec{\omega})$$

Global Illumination
In general, we should take reflection of light from surfaces into account. Those are called indirect illumination. This can get quite complicated as light can reflect multiple times in a scene. That forms a light path.

- Consider all light – including bounces





x

light path

Global Illumination

- Connects a **light source** to a **sensor**
- Constructed by tracing from:
  - light source… *light tracing*
  - from sensor… *path tracing*
  - or from both… *bidirectional path tracing*
- Length of light path:
  - 2 segments… *direct illumination, direct lighting*
  - >2 segments… *indirect illumination, indirect lighting*

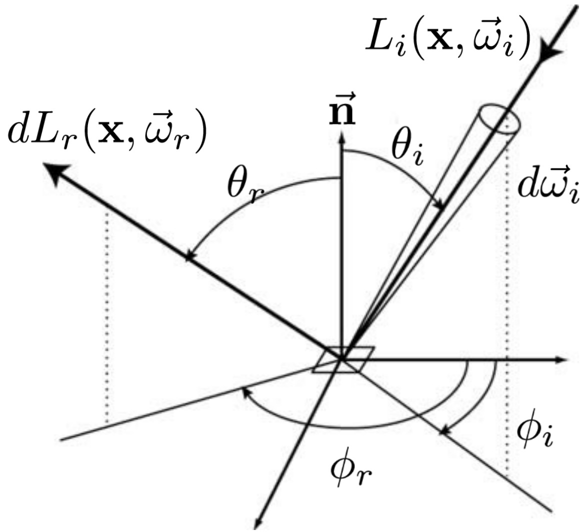
Global Illumination

Direct illumination | **Indirect illumination** | **Direct + Indirect**

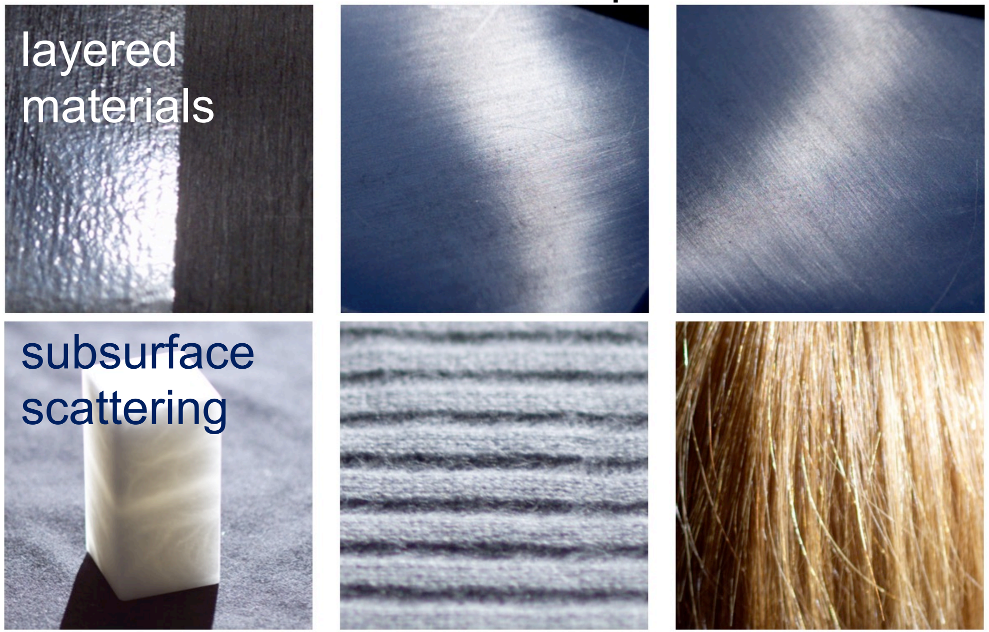
BRDFs – Complex Reflections

The rendering equation can be used to render any real-world material with complex reflection characteristics.
The BRDF defines e.g. if the reflection is anisotropic.
If there is non-reflections, e.g. some light passes through, we can define related functions.

$$\int_{H^2} \boxed{f_r(\mathbf{x}, \vec{\omega}_i, \vec{\omega}_r)} L_i(\mathbf{x}, \vec{\omega}_i) \cos\theta_i \, d\vec{\omega}_i$$

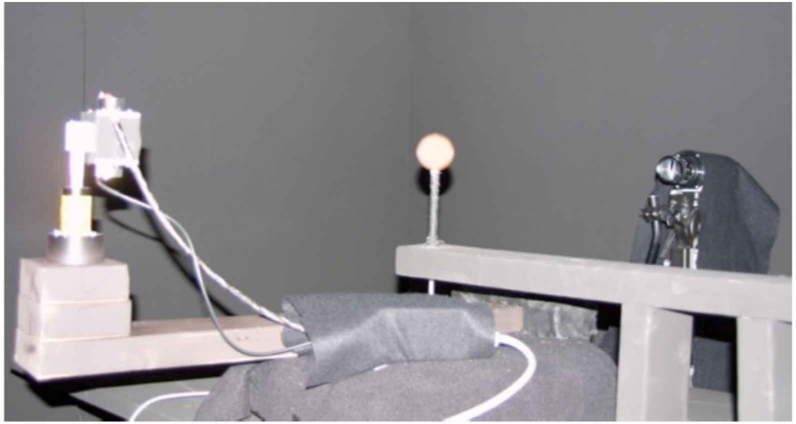


anisotropic reflections

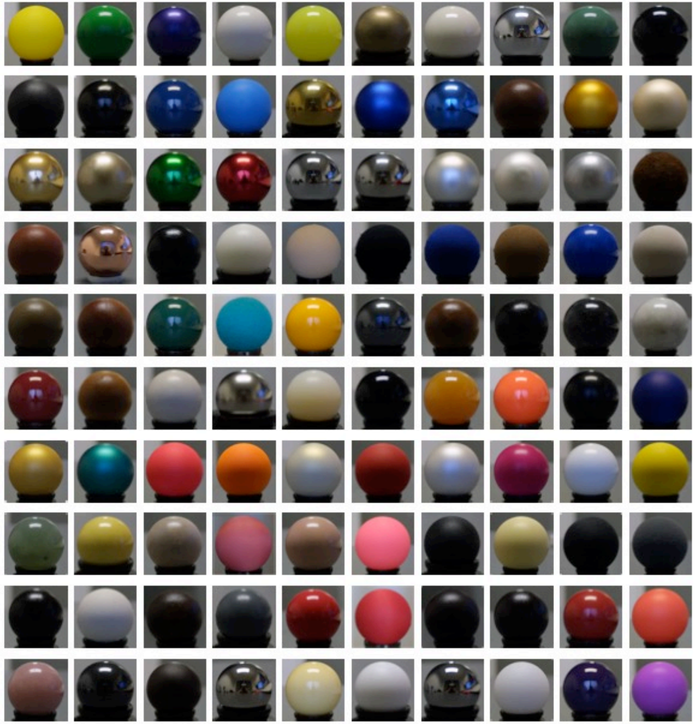


volumetric structures

Measuring BRDFs

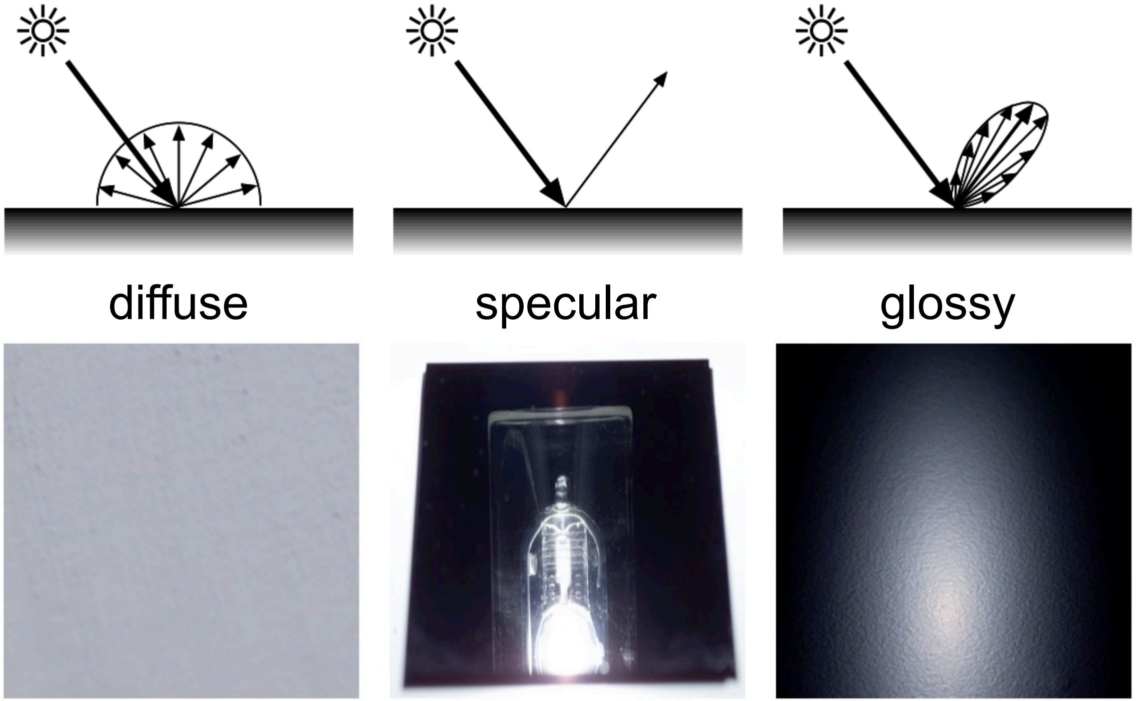


Matusik et al.: Efficient Isotropic BRDF Measurement, Eurographics Symposium on Rendering 2003



Simpler Reflections

Simple reflection models are useful for fast rendering and can be combined for a wide range of effects. They result from assuming a special form for the BRDF.



Hendrik Lensch, Efficient Image-Based Appearance Acquisition of Real-World Objects, Ph.D. thesis, 2004

Diffuse Reflection

Diffuse reflection means the BRDF is constant and hence can be taken out of the integral.
We have a constant times the integral of the incoming light in all directions weighted by the cosine term.
This also means the reflected light is independent of the outgoing light direction, hence no view-dependent effects.

- For diffuse reflection, the BRDF is a constant:

$$L_r(\mathbf{x}, \vec{\omega}_r) = \int_{H^2} f_r(\mathbf{x}, \vec{\omega}_i, \vec{\omega}_r) L_i(\mathbf{x}, \vec{\omega}_i) \cos\theta_i \, d\vec{\omega}_i$$

$$L_r(\mathbf{x}) = f_r \int_{H^2} L_i(\mathbf{x}, \vec{\omega}_i) \cos\theta_i \, d\vec{\omega}_i$$

$$L_r(\mathbf{x}) = f_r \, E_i(\mathbf{x})$$

Ticking

Ticking session: Thursday, October 27, 2022, 13 - 14 PM

- In case you cannot make it, let us know asap.
- Please check Moodle after the submission for your slot.
- The slots are approximate, prepare to be ready 13:00 - 14:00.

Ticking format

- 5 minutes via Zoom or similar
- Link to the virtual room on Moodle
- Please be prepared to explain & run the code