12. Materials & Scattering

Dr. Hamish Carr & Dr. Rafael Kuffner dos Anjos

Agenda

- Scattering Function
- Surface Frame
- BSDFs and their properties
- Approximations
- Different models

Material Models

- We need to model large numbers of photons
  - So we want statistical behaviour, capturing:
  - Specular reflection
  - Glossy reflection
  - Lambertian (matte) reflection
  - Subsurface scattering
  - Transmission
- All captured in a scattering function

Scattering Functions

Textures

Bi-directional reflectance dist. func. (BRDF)

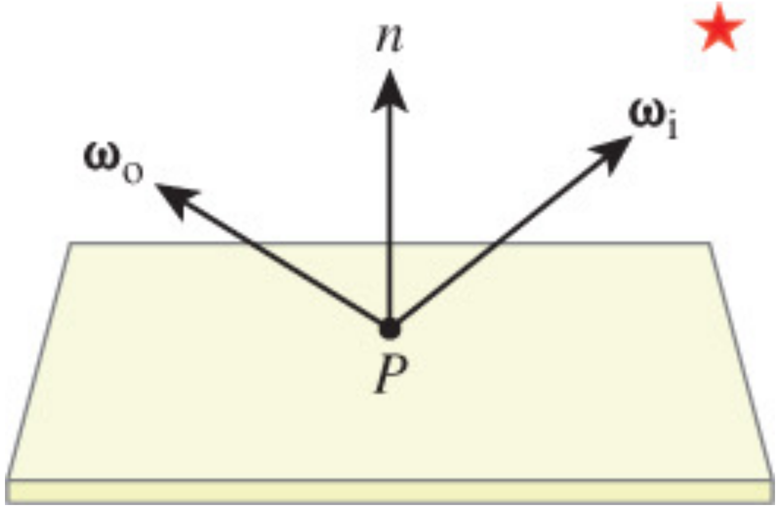
Bi-directional transmission dist. func. (BTDF)

Bi-directional scattering dist. func. (BSDF)

Bi-dir scattering & subsurface d.f. (BSSDF)

All of which share common features

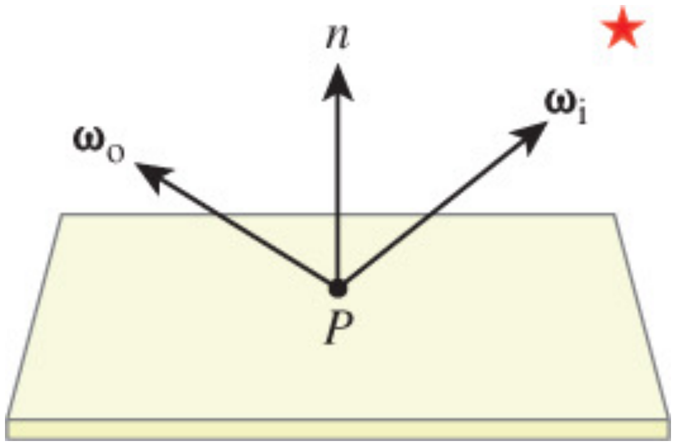
Scattering Functions



- Photon arrives at point P (u,v coords)
  - From direction ω_i
  - Could go any direction, so parametrise by:
  - ω_o- direction light goes out
  - And get a *probability* of going that way out

UNIVERSITY OF LEEDS

Mathematically



$$f_s(P, \omega_i, \omega_o): T^2 \times S^2 \times S^2 \rightarrow \mathbb{R}^+$$

T^2: texture coordinates

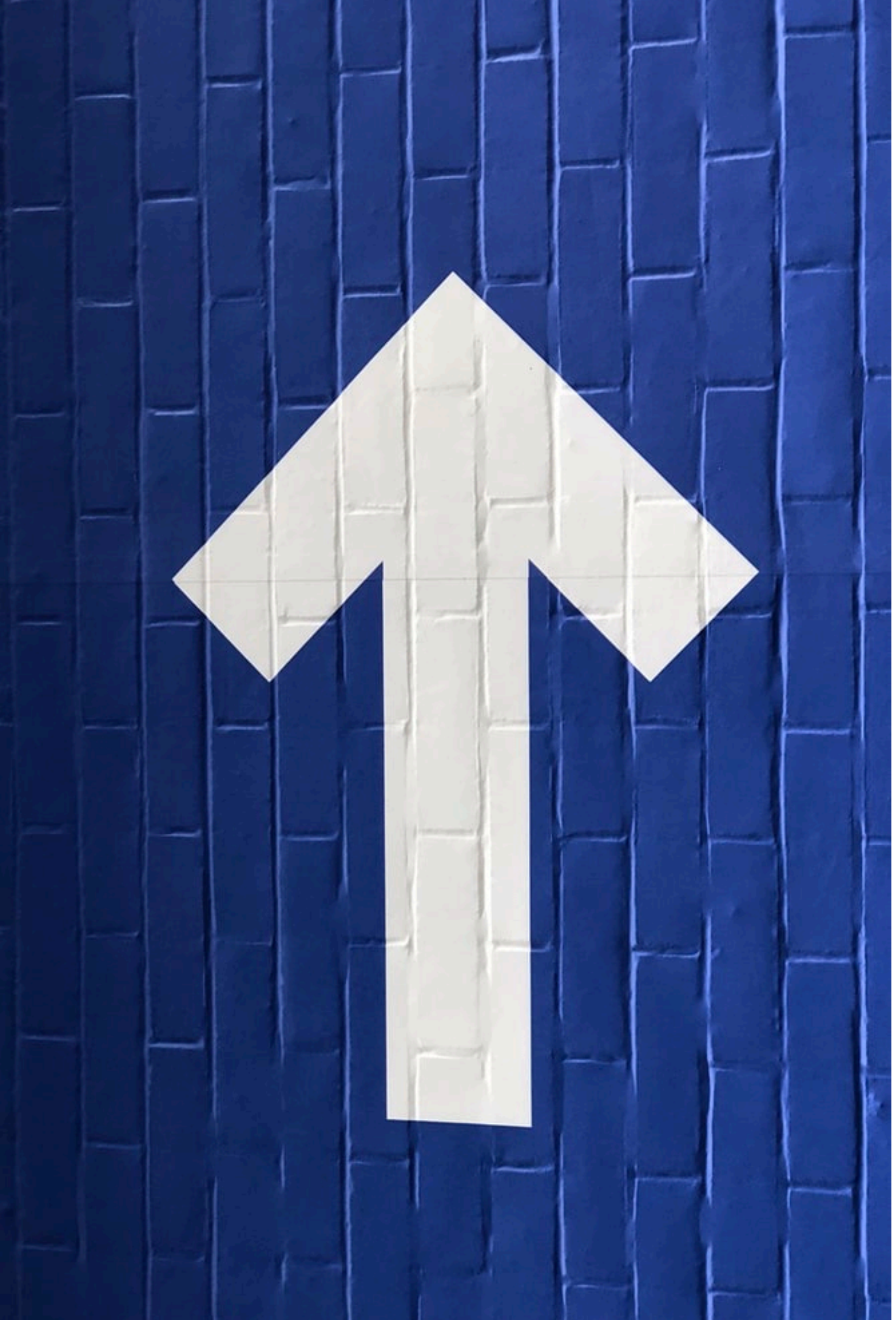
S^2: parametrised sphere
(used for directions)

- Note we have a *6-D* function
  - 2D – texture coords for point
  - 2x2D – spherical coordinates for directions

UNIVERSITY OF LEEDS

How do we use this function?

- We are using 2 coordinates (u,v) to locate our point
- This does not give us the orientation we need for the scattering function



Directions

- Directions are assumed to be unit vectors
  - And can be parametrised to a sphere
- Texture coordinates give:
  - Surface tangents
  - Surface normal
  - i.e. a local coordinate frame at P
- So we can express it in these terms

Finding the Surface Frame

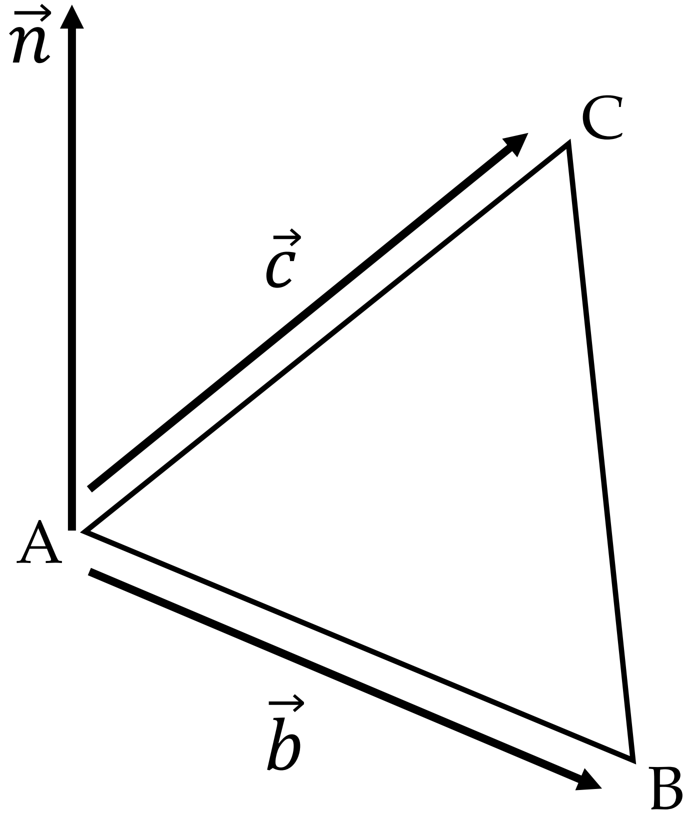
- Consider dS/du and dS/dv: partial derivatives of the surface
- These form a basis for the tangent plane
  - Not necessarily orthonormal
  - But still useful
- In texture coordinates,
  - $\frac{\partial S}{\partial u} = (1,0)_{u,v}$
  - $\frac{\partial S}{\partial v} = (0,1)_{u,v}$

Computation

- We assume that the normal vector is known
- And take two edges $\vec{b}, \vec{c}$ of a triangle ΔABC
- We will compute them twice:
  - In spatial coordinates
  - In texture coordinates
- Then use a matrix inverse to convert them

Spatial Frame

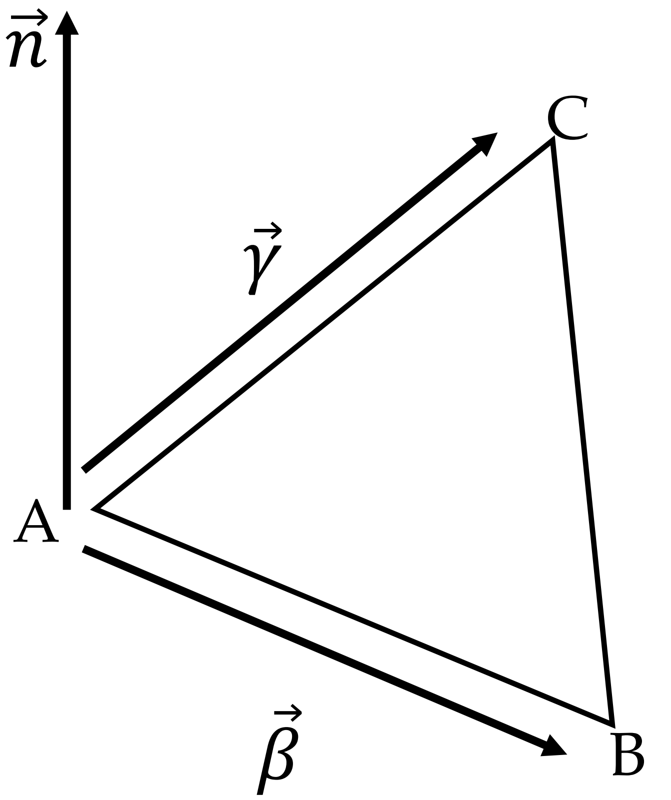
- Take the vectors
  - $\vec{b} = B - A$
  - $\vec{c} = C - A$
  - $\vec{n} = \vec{b} \times \vec{c}$
- These form a frame
- Now ignore $\vec{n}$
  - Since we work in its plane



UNIVERSITY OF LEEDS

Texture Vectors

- Compute b,c in texture space
- Texture Edge Vectors:
  - $\vec{\beta} = (u(B) - u(A), v(B) - v(A))$
  - $\vec{\gamma} = \big(u(C) - u(A), v(C) - v(A)\big)$



Setting Up Equations

- We can express $\vec{\beta}, \vec{\gamma}$ in texture space:
  - $\vec{\beta} = \beta_u(1,0)_{u,v} + \beta_v(0,1)_{u,v}$
  - $\vec{\gamma} = \gamma_u(1,0)_{u,v} + \gamma_v(0,1)_{u,v}$
- And convert back to spatial coordinates:
  - $\vec{b} = b_u \frac{\partial s}{\partial u} + b_v \frac{\partial s}{\partial v}$
  - $\vec{c} = c_u \frac{\partial s}{\partial u} + c_v \frac{\partial s}{\partial v}$

Matrix Form

$$\begin{bmatrix} \vec{b} \\ \vec{c} \end{bmatrix} = \begin{bmatrix} b_u & b_v \\ c_u & c_v \end{bmatrix} \begin{bmatrix} \frac{\partial S}{\partial u} \\ \frac{\partial S}{\partial v} \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial S}{\partial u} \\ \frac{\partial S}{\partial v} \end{bmatrix} = \begin{bmatrix} b_u & b_v \\ c_u & c_v \end{bmatrix}^{-1} \begin{bmatrix} \vec{b} \\ \vec{c} \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial S}{\partial u} \\ \frac{\partial S}{\partial v} \end{bmatrix} = \frac{1}{b_u c_v - c_u b_v} \begin{bmatrix} c_v & -b_v \\ -c_u & b_u \end{bmatrix} \begin{bmatrix} \vec{b} \\ \vec{c} \end{bmatrix}$$

Problem

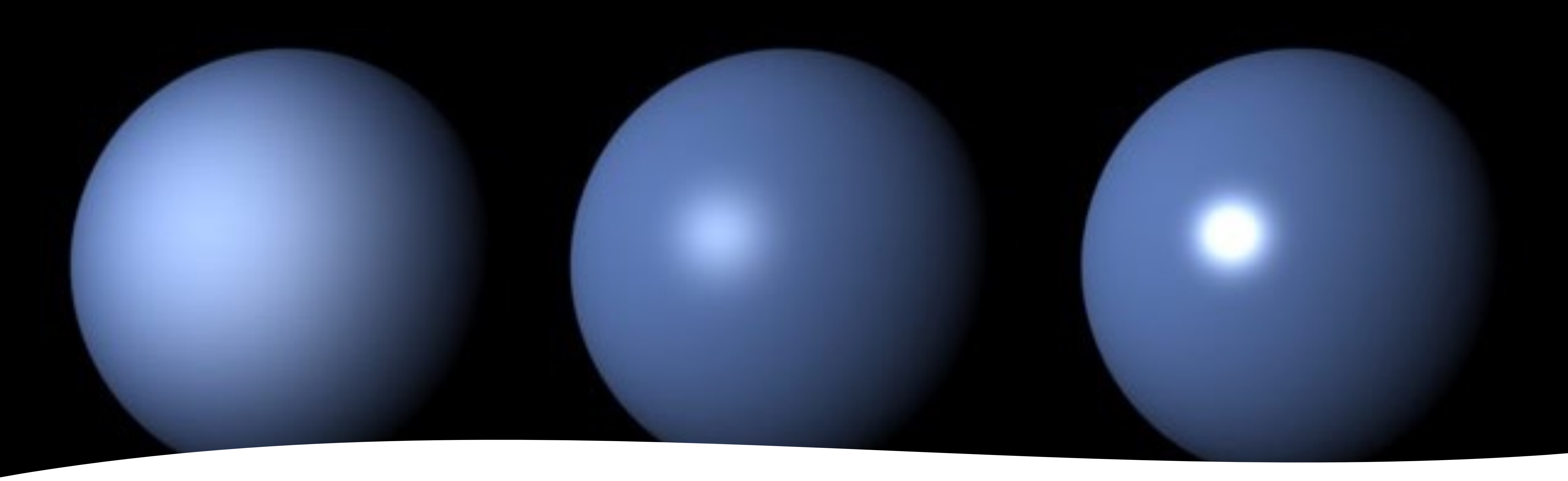
- How do we use our function ?
  $$f_s(P, \omega_i, \omega_o): T^2 \times S^2 \times S^2 \rightarrow \mathbb{R}^+$$
- Storing 6D functions is impractical
- 32 samples / parameter gives $2^{30} \sim 1$ GB data

Solution?

- So in practice, we never actually use them
- Instead we use Euclidean products of:
  - BSDF variation in one texture
  - Local basis in a second texture
  - Another texture if needed for orientation

Sources of BSDFs

- Measured (expensive, noisy):
  - Build a measuring device, store GBs of data
- Analytic (clean, expressive):
  - Build a mathematical model (how?)
- Artistic (cheap(er), acceptable):
  - Pay an artist to paint it
  - Requires training your artist in BSDFs



Blinn-Phong Shading

- Just *one* way to do a BSDF:
  - No specular (mirrored reflections)
  - Gloss (called specular)
  - Diffuse (Lambertian)
  - No subsurface scattering
  - Ambient (substitutes for light transport)
  - Emissive

Translucency

- Translucency is transmitted light
- May be in addition to reflected light
- Green glass transmits green, reflects darker
- High-quality rendering does refraction
- Other rendering does *blending (compositing)*
  - Combines light from behind with material
  - And causes problems if not ray-tracing

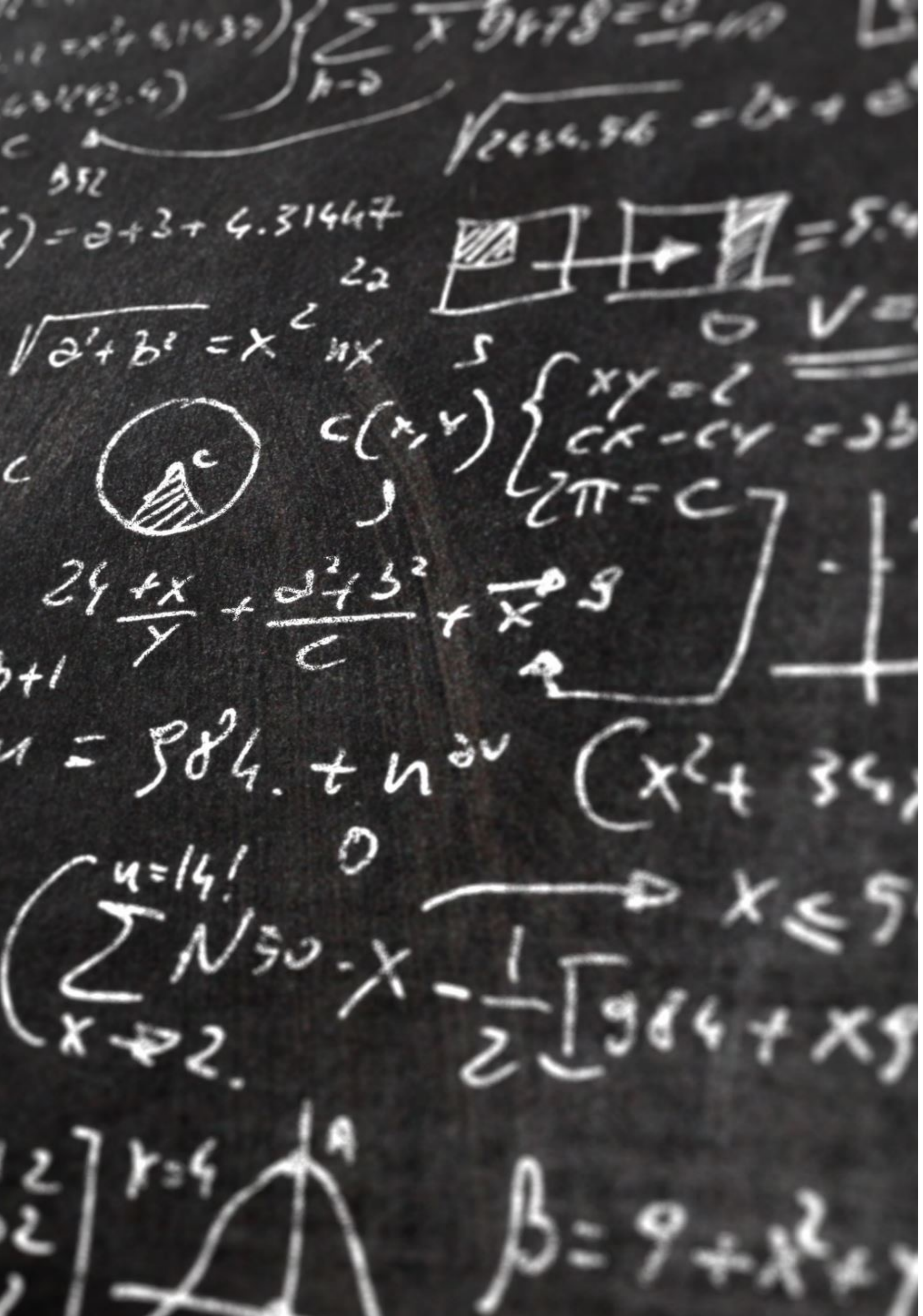
Alpha Opacity

- Translucency is modelled by *alpha*
  - The percentage of (R,G,B) transmitted
- Stored as an extra channel in a buffer/image
  - Often stored premultiplied
- Can be stored in a texture
  - E.g. stained glass window

UNIVERSITY OF LEEDS

Emission

- Surfaces may emit light
- In ray-tracing, these are luminaires
- The rest of the time, they are an extra term
  - And are crucial for inter-object lighting
    - Lights in-scene, torches, gun blast
  - Can be handled with extra textures
    - Light maps instead of lighting



Radiance Function

- Also known as the plenoptic function
- Measures incoming light everywhere:
  - $L(X, \omega)$: $\mathbb{R}^3 \times S^2 \rightarrow \mathbb{R}$
  - Units are radiance ($W/m^2 sr$)
- Consists of direct *and* indirect light

Ground Truth

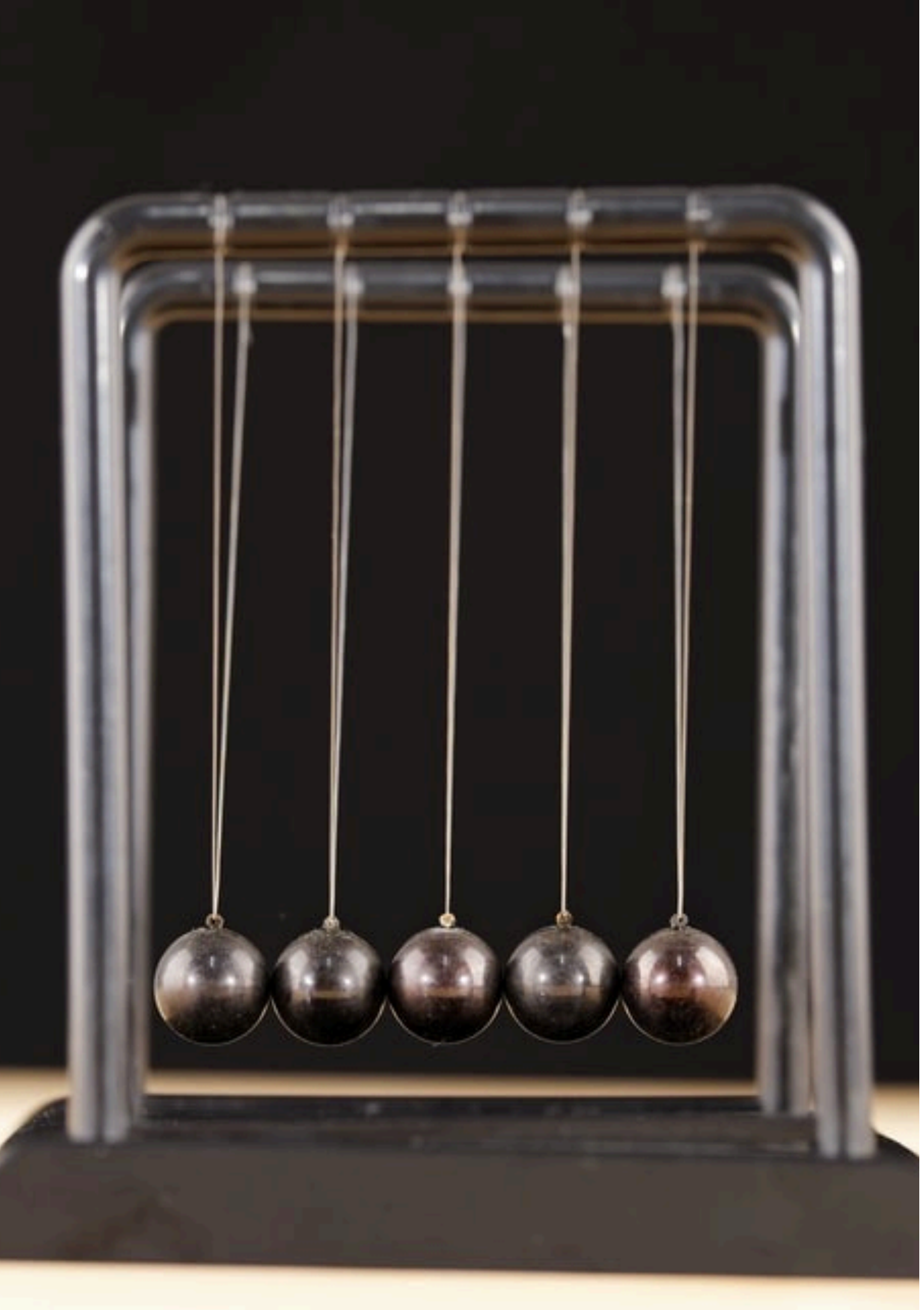
- We don’t have an ideal solution
- We have lots of approximations
- So this is *very* much a matter of choice
  - Which effects you want to represent
  - How realistic they look (the eyeball test)
- And this is not likely to change soon

Approaches

Empirical / phenomenological	Constructed to "look right"
Measured	Physically expensive to measure Huge storage requirements Limited creative control
Physically based	Mathematical analysis of physics

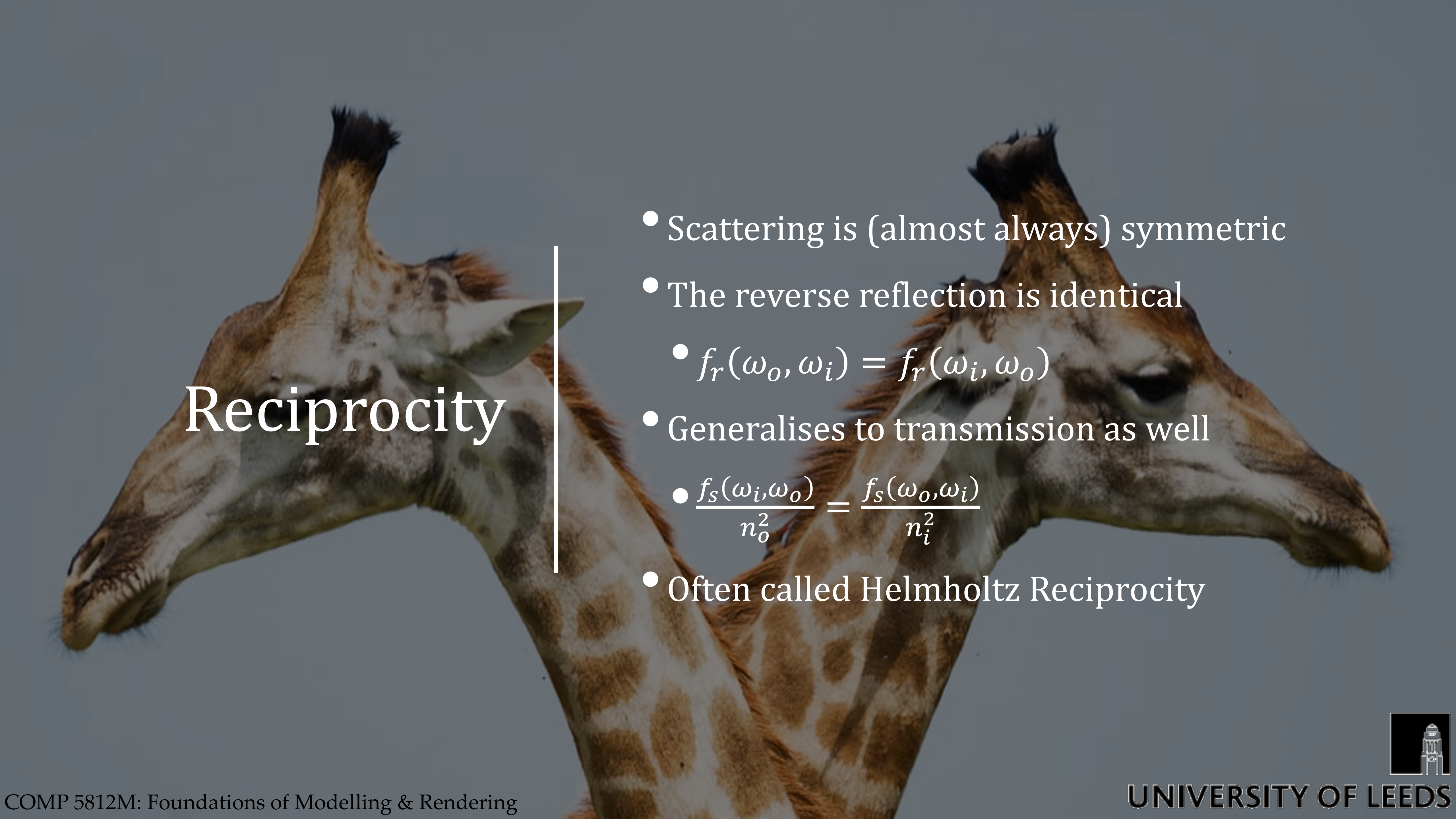
Physical Constraints

- Energy Conservation
- Reciprocity
- Surface vs. Subsurface
- Volumetric Representations
- Material Scale
- Specular Representation



Energy Conservation

- You can't *add* light once it's created
- Unless the object is a luminaire
  - in which case emission is dominant term
- So total light output < total light input
- Expressed in integral form:
  - $\int_{\omega_o,\omega_i} f_r(\omega_o, \omega_i)\, d\omega_o d\omega_i \leq 1$



Reciprocity

- Scattering is (almost always) symmetric
- The reverse reflection is identical
  - $f_r(\omega_o, \omega_i) = f_r(\omega_i, \omega_o)$
- Generalises to transmission as well
  - $\frac{f_s(\omega_i,\omega_o)}{n_o^2} = \frac{f_s(\omega_o,\omega_i)}{n_i^2}$
- Often called Helmholtz Reciprocity

Surface vs. Subsurface

- Subsurface scattering may *travel*
  - i.e. light emerges somewhere else
- This complicates matters
  - but for now we'll skip that

Volumetric Representation

- This all still assumes that we have *surfaces*
- We might have volumetric effects
  - Complex interactions in a volume of space
- These are an entirely different question
  - Heavily used in science & medicine
- Typically a different code path
  - We'll come back to this

Material Scale

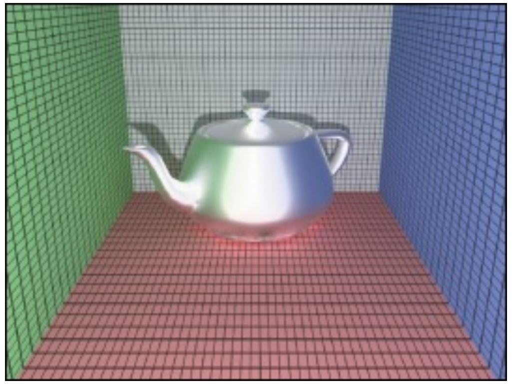
- Visible light has wavelength of about $1\mu m$
- Human hair is about $15\mu m$ across
  - And we can feel it with our fingers
  - Let alone see it visually
- Scratches in a surface can cause interference
- But in practice, we hand-wave this
  - For now . . .

Specular Reflection

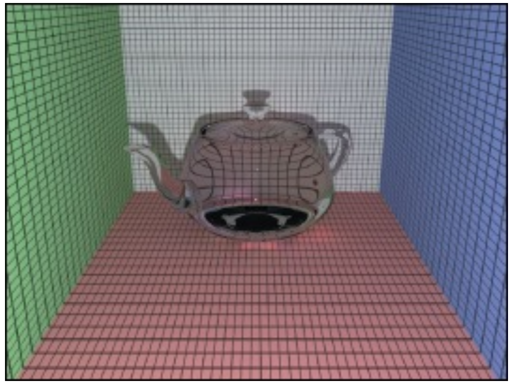
- There are (nearly) perfect mirrors
- All light reflects in a single direction
  - Usually (but not always) $\theta_r = \theta_i$
- Which means a range of 10^{10} in albedo
- This causes numerical problems in codepath
- So commonly handled as a separate case
- Referred to as impulse reflection

UNIVERSITY OF LEEDS

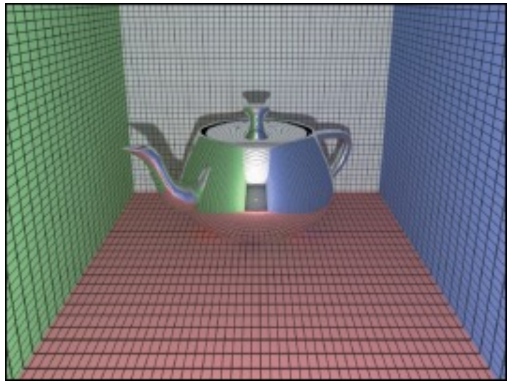
Scattering Phenomena



Reflective (all in S_+^2)



Transmissive (all in S_-^2)



Mirror (Impulse)

Glossy (Blurred Mirror)

Lambertian (ω_o indep. of ω_i)

Retroreflective

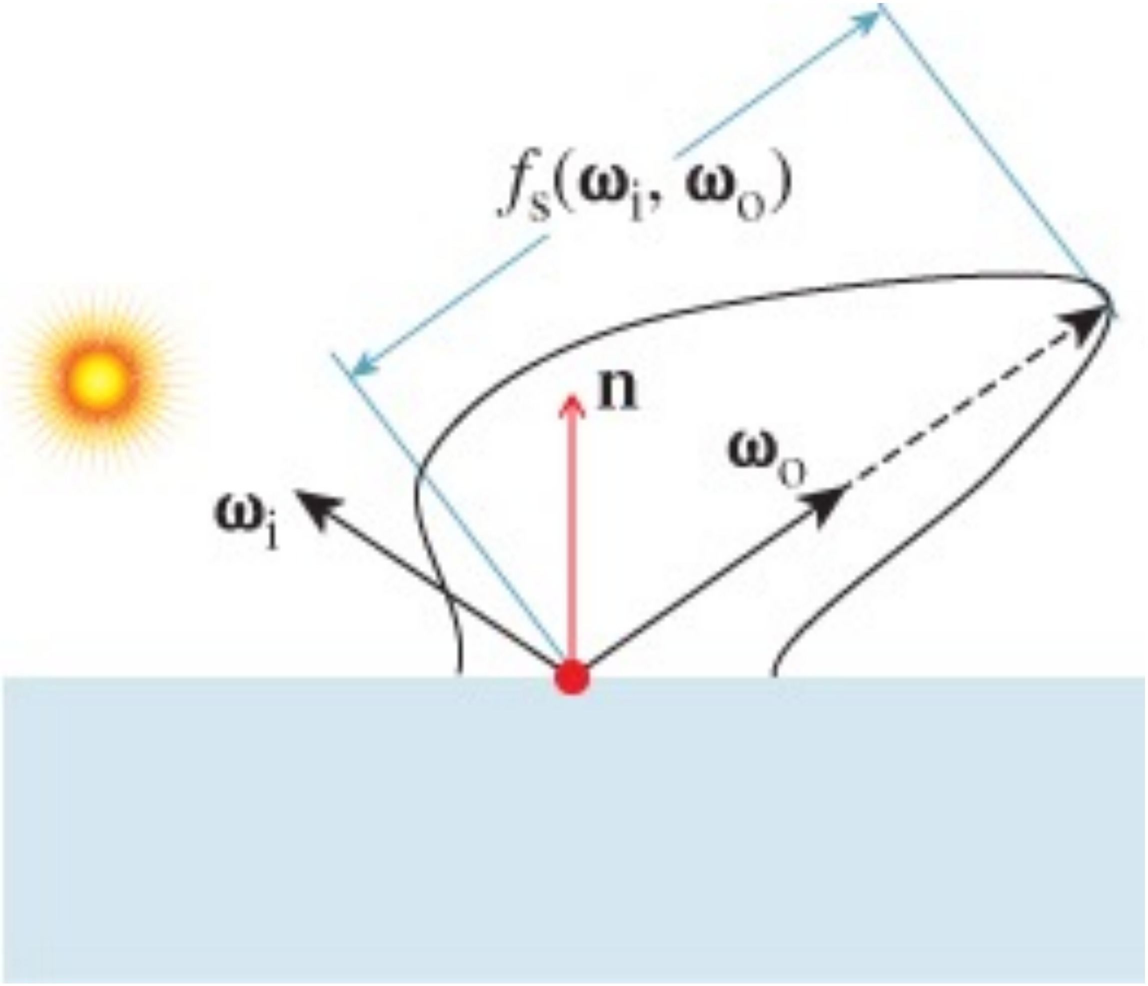
Refractive

&c.

UNIVERSITY OF LEEDS

Plotting the BSDF

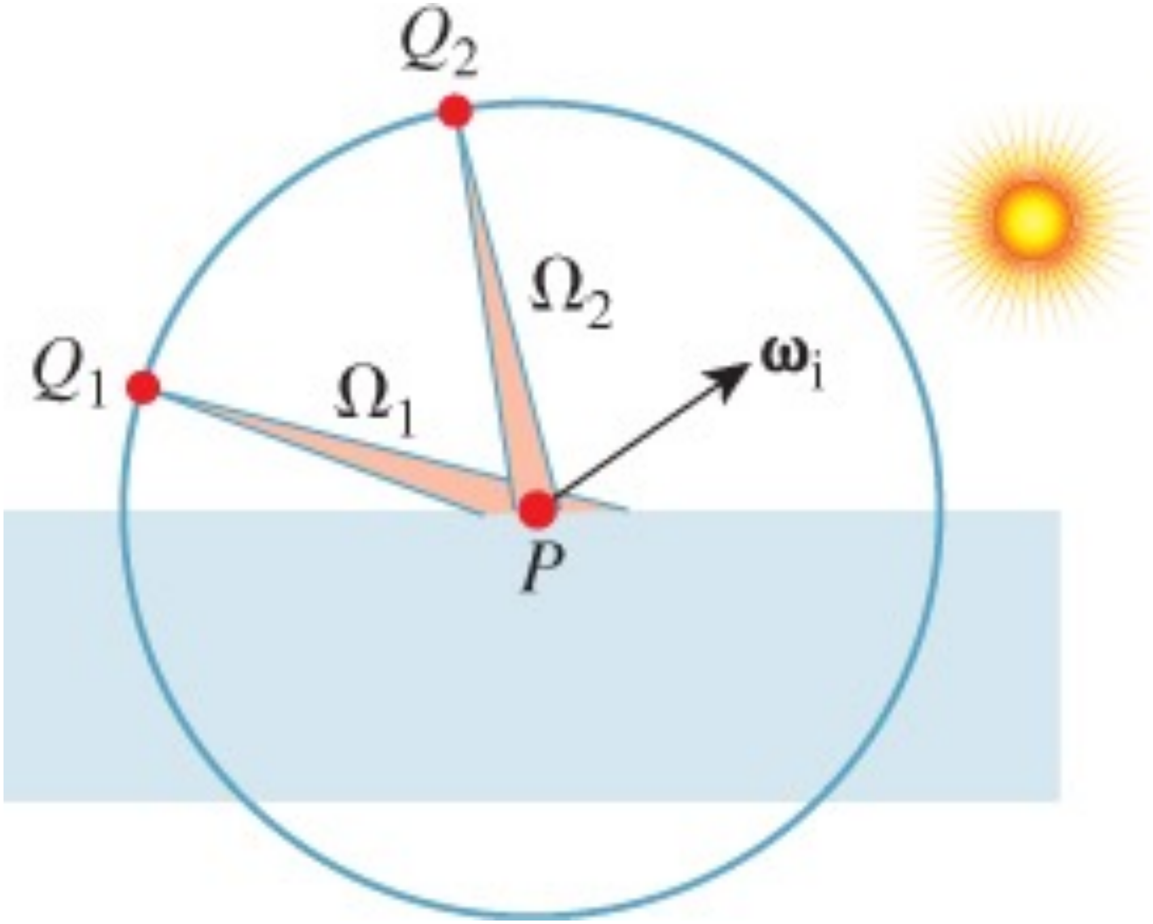
- Consider light with respect to surface frame
- We can plot the distribution of reflection
- Radiance or probability density function (pdf)



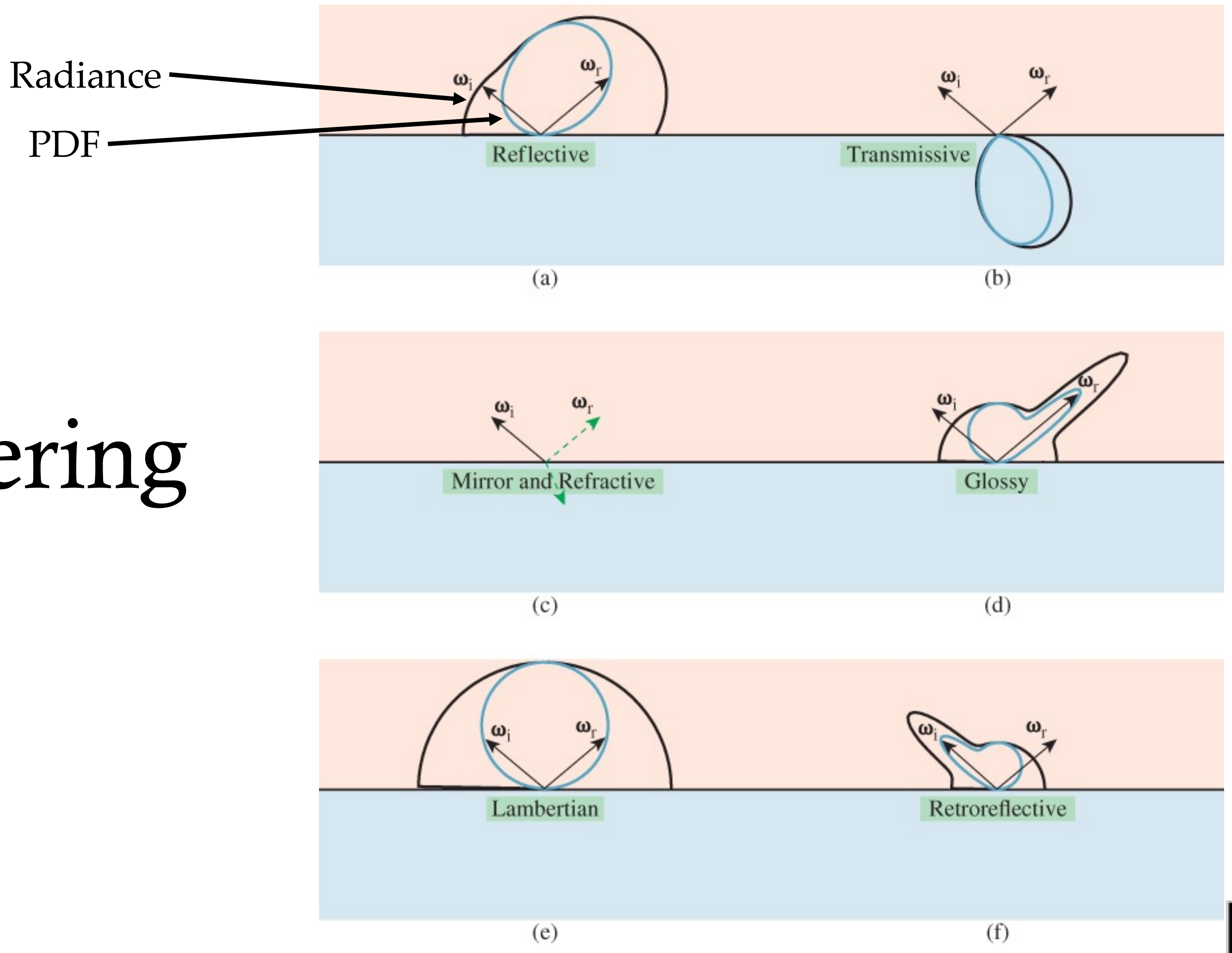
BSDF

Radiance vs. PDF

- Radiance is a physical quantity
- pdf is integrated
  - Over a small patch
  - Uses cosines like "diffuse" reflection
- Carries forward through our integration
  - Because samples are actually integrals



Examples of Scattering



Mirror "Scattering"

- Mirrors do absorb light
- But reflect most of it
- Insulators preserve spectral distribution
- Conductors have "preferred" frequencies
- So typically stored as RGB albedo
- This ignores polarisation
- We'll see geometric hacks next term

Lambertian vs. Phong

- Lambertian ~ Phong diffuse
- Effectively captured in ρ term
- But cosine term needed for pdf
  - Based on dot product

$$p(\omega_o) = \frac{1}{\pi}(\omega_o \cdot \mathbf{n}). \qquad (27.16)$$

- Which explains success of Phong model
  - it's *close* to the truth

UNIVERSITY OF LEEDS

Lafortune Model

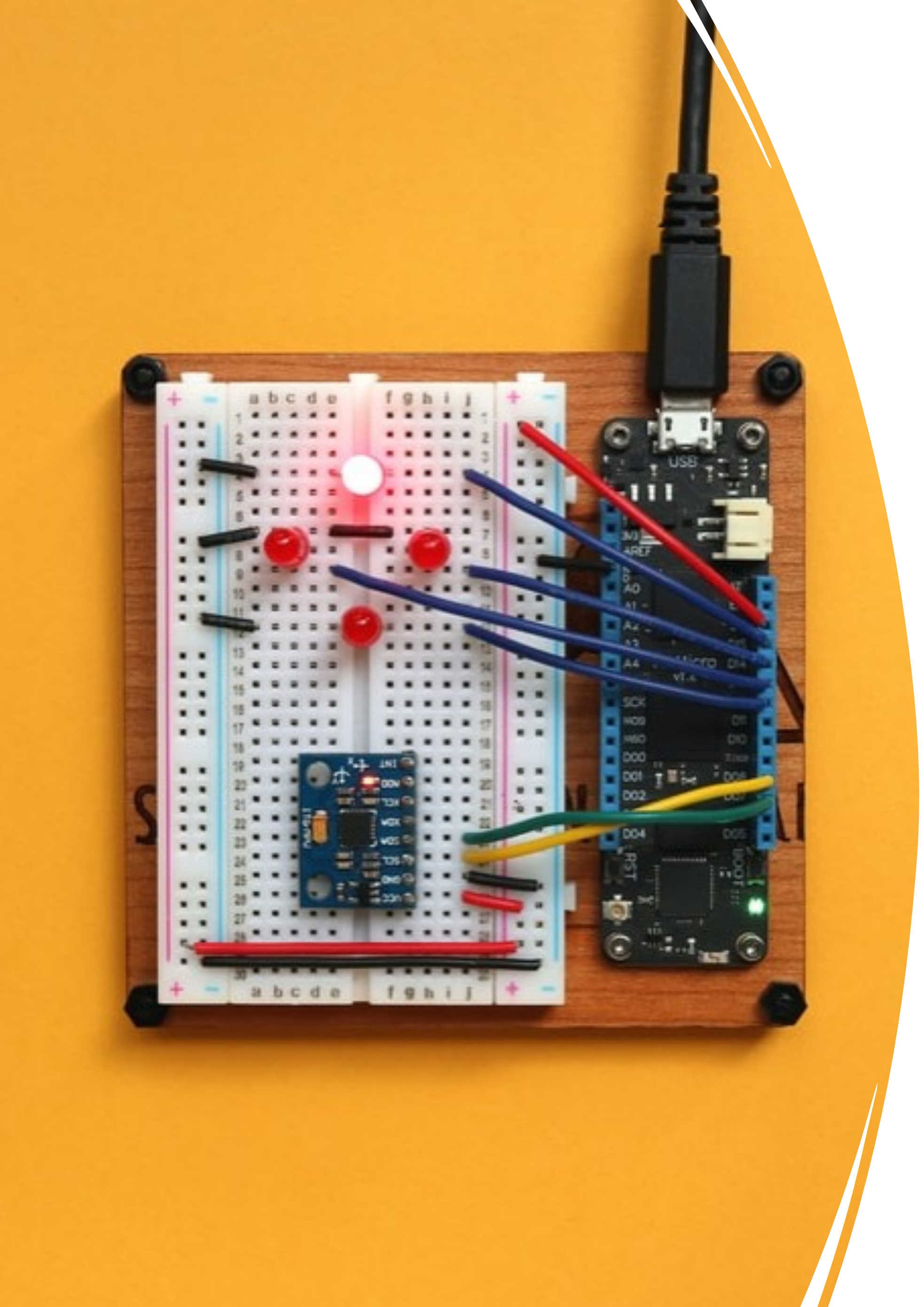
Measured BRDFs are *not* centred

i.e. the angle of reflection is off

So they generalised by adding more terms

Multiple specular lobes

With different properties

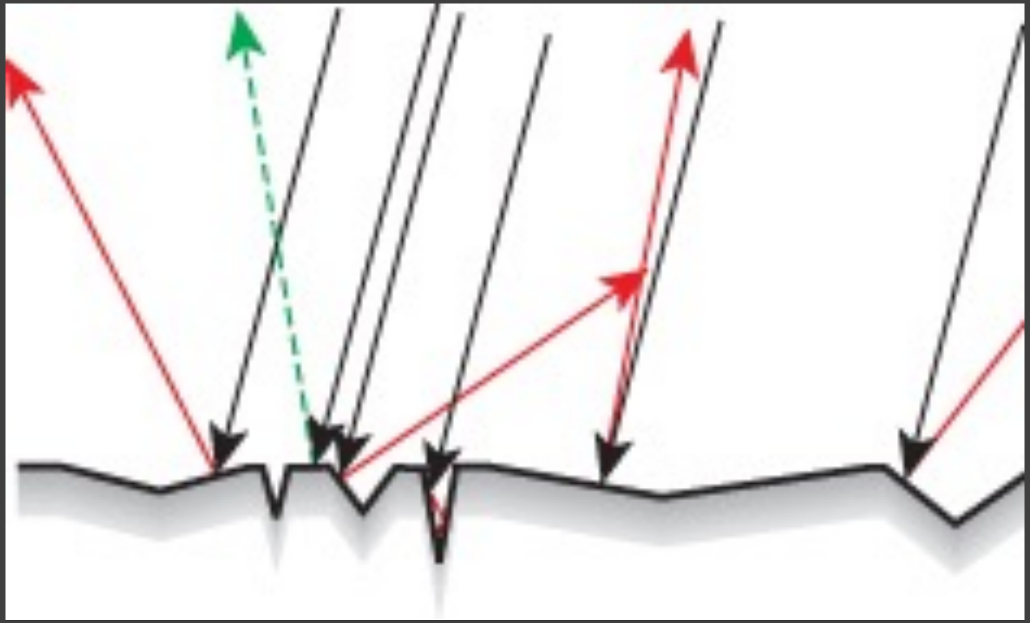


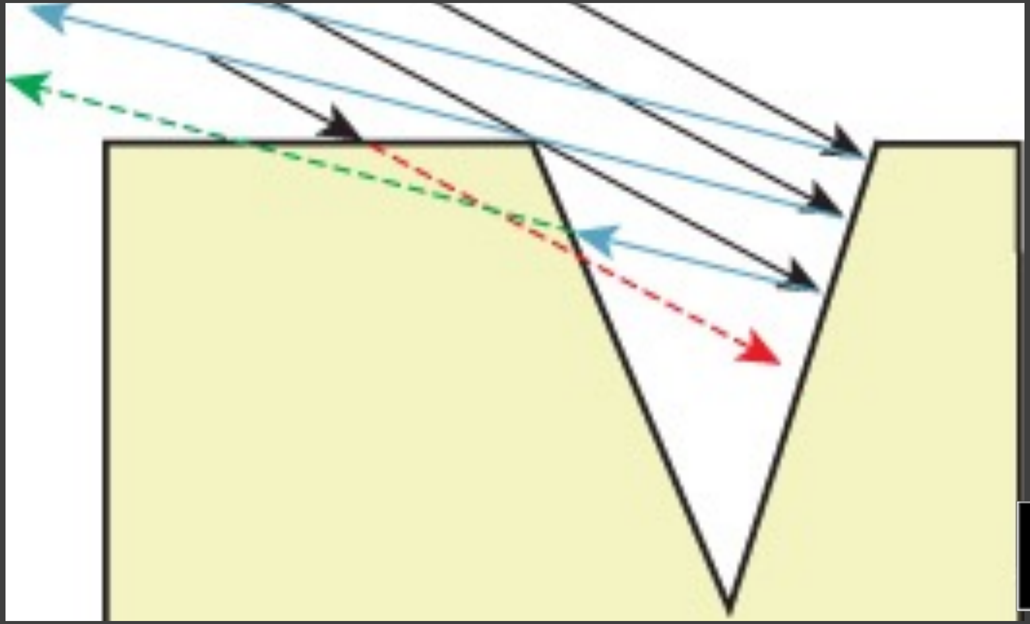
Measured BRDFs

- Still too expensive in practice
- Massive complex measuring equipment
- And 6 parameters
  - So $100^6 = 10^{12}$ storage requirements
- Effectively a texturing problem anyway
- *And* you end up interpolating in them

Torrance-Sparrow Model

- Light actually reflects from microfacets
- These occlude each other
- So the BRDF isn't quite what you expect





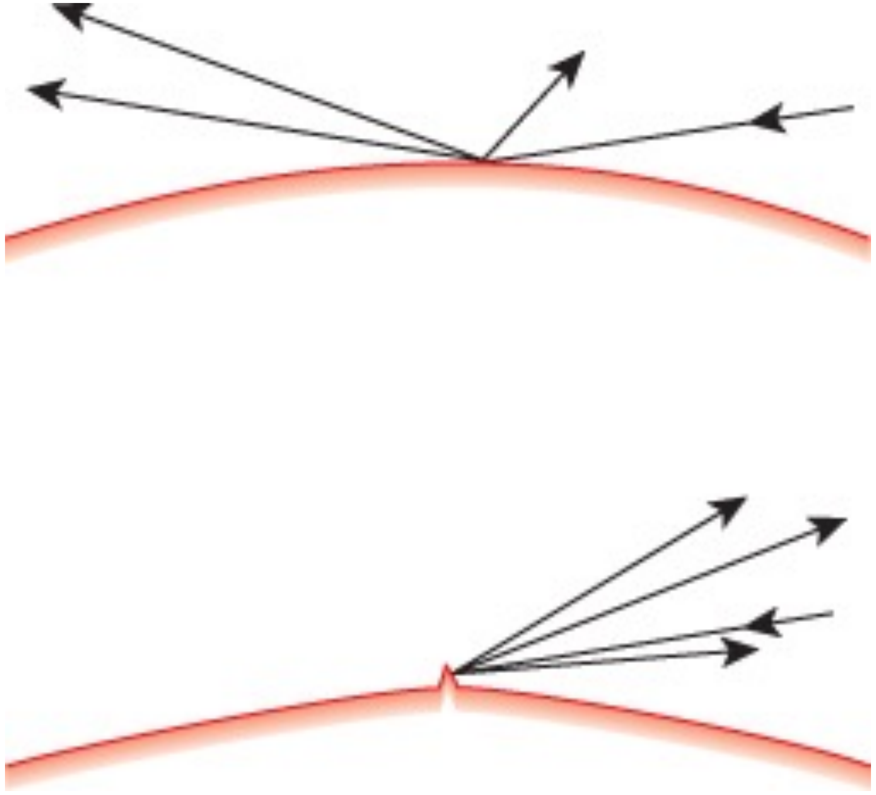
Cook-Torrance Model

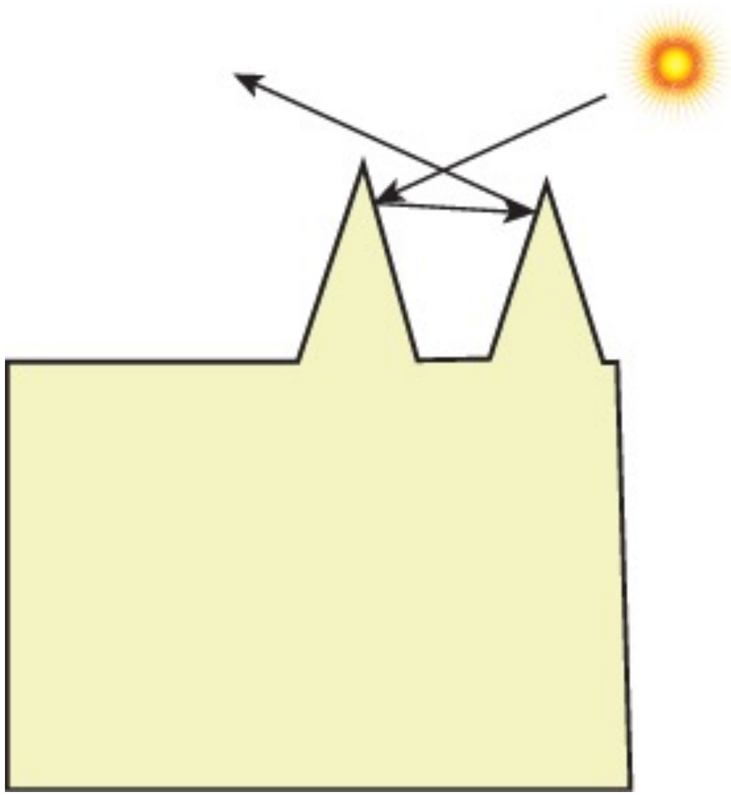
- Diffuse reflection is subsurface
- Specular reflection is surface
- Extra terms in the equation

UNIVERSITY OF LEEDS

Oren-Nayar Model

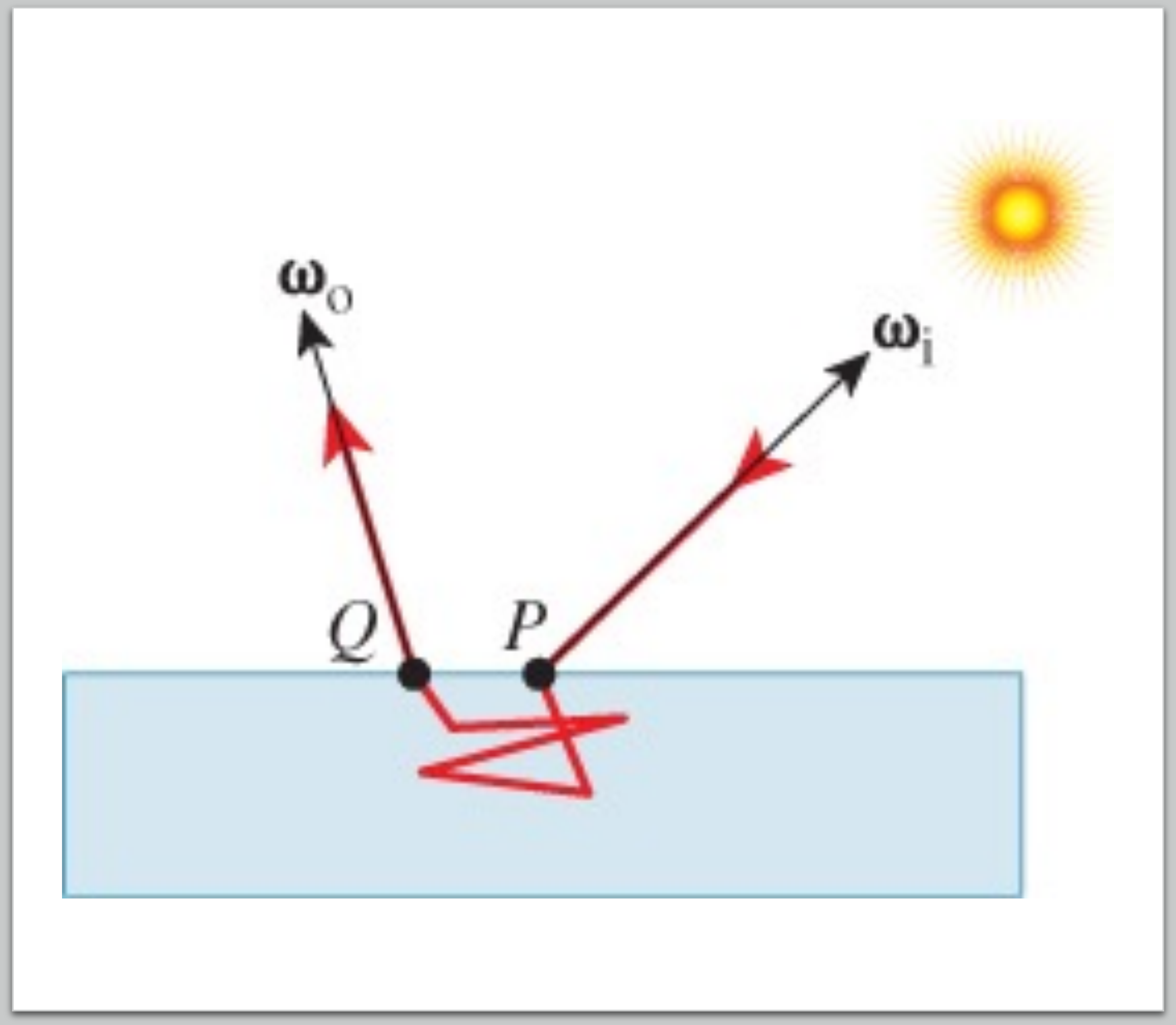
- Micro-roughness near the silhouette
- Causes extra reflection towards viewer
- Extra lightness near boundaries





Subsurface Scattering

- Light goes in at P, emerges at Q
- After complex interactions & bounces
- Even the textbooks say this is too complex
    - There are ways to approximate it (next term)



The Practical Reality

- People still use Phong lighting for real-time
- And they add various tricks to improve it
  - Per-pixel shading
  - Texture-based material storage
  - Geometric hacks (next term)
- Where it falls down is global illumination
  - The ”ambient” term

Images by

- S3 Vinicius “amnx” Amano
- S7 Nick Fewings
- S19 Matthias Gotzke
- S20 Anh Tuan To
- S21 Engin Akyurt
- S22 Vincent Van Zalinge
- S30 Julianne Liebermann
- S31 Vince Fleming
- S39 Jorge Ramirez