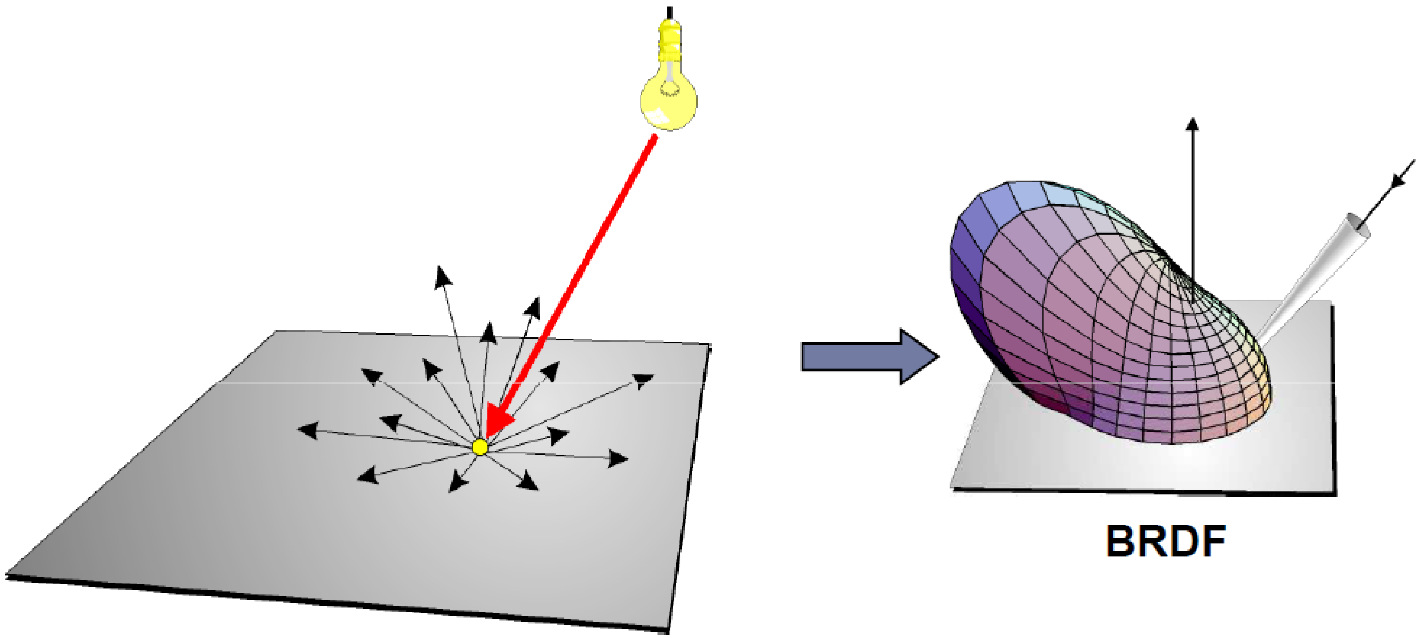
Local Illumination Models

CS7GV3 – Real-time Rendering

Overview

- General definition of Illumination Models
- Local Illumination Models on the Shader
- How do we simulate different light reflection behaviour say between shiny plastic and metal that is the same colour

Scattering



Energy is scattered from a surface in a distribution that depends on the surface's microscopic geometry. This distribution can be described as a function that records the reflected energy per direction = *BRDF*

BRDF

- General light reflected from a point on a surface is categorised by the Bi-directional Reflectance Distribution Function
  - Function of some direction (in local-illumination we are interested in the viewer) and incoming light direction
  - Can be written as

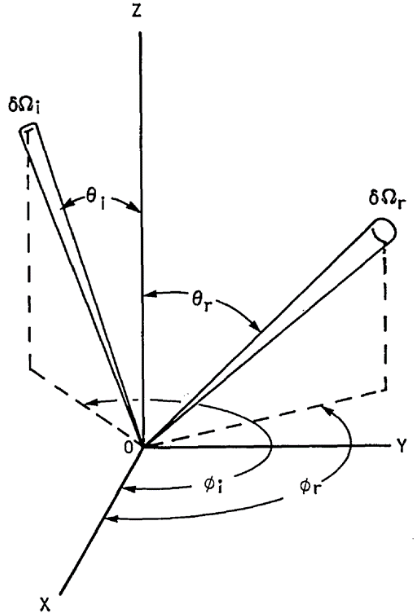


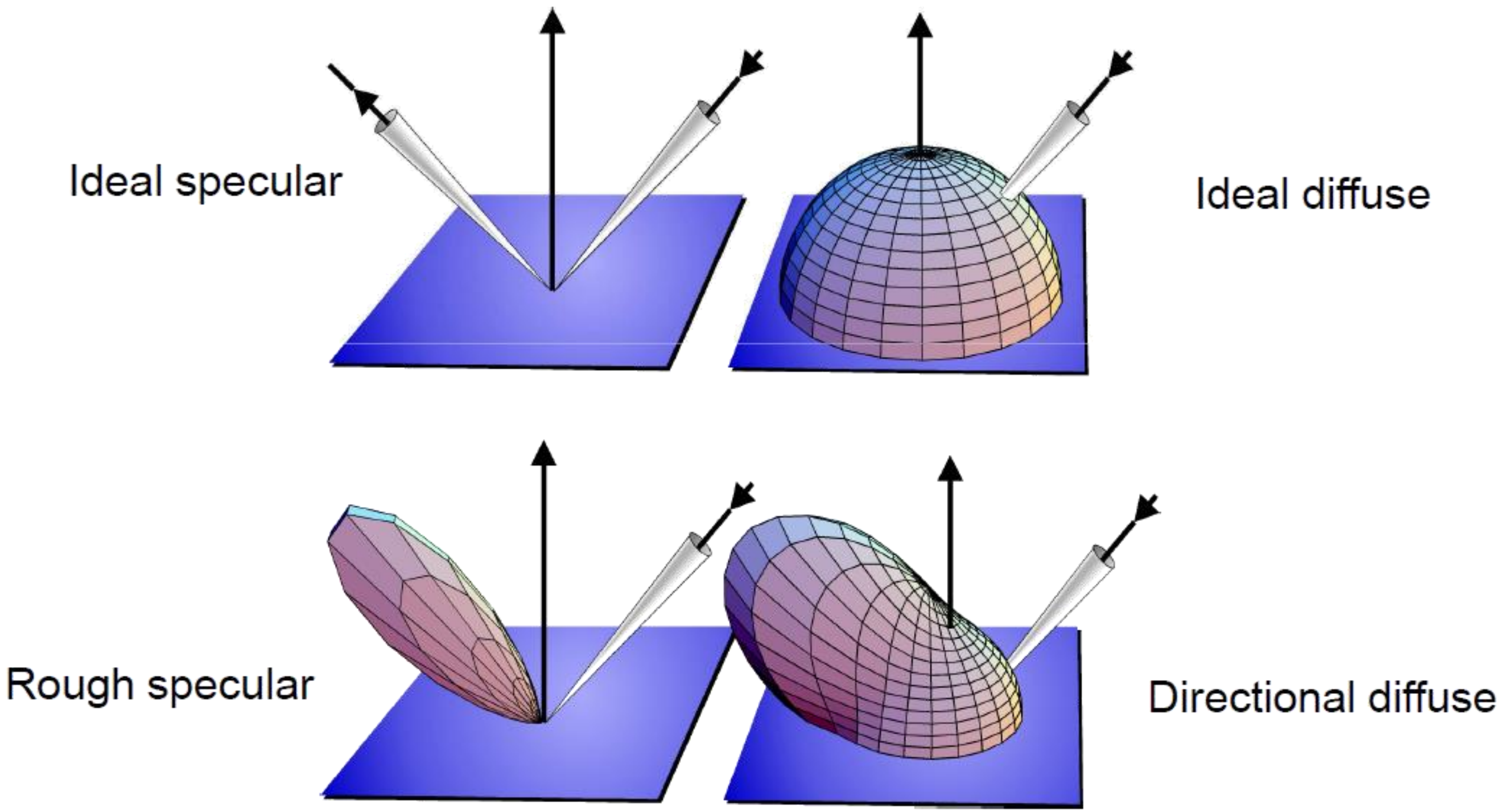
Fig. 1

$$BRDF = f(\boldsymbol{\theta}_{in}, \boldsymbol{\varphi}_{in}, \boldsymbol{\theta}_{ref}, \boldsymbol{\varphi}_{ref}) = f(\mathbf{L}, \mathbf{V})$$

Citation

Fred E. Nicodemus, "Directional Reflectance and Emissivity of an Opaque Surface," Appl. Opt. **4,** 767-775 (1965); https://www.osapublishing.org/ao/abstract.cfm?uri=ao-4-7-767

BRDF Approximations



Reflectance Equation

- The *reflectance equation relates reflected* radiance to incoming radiance that is scattered according to the surface's BRDF

$$\underbrace{L_r(x,\omega_r)}_{\text{Reflected radiance}} = \int_\Omega \underbrace{f_r(x,\omega_i,\omega_r)}_{\text{BRDF}} \underbrace{L_i(x,\omega_i)}_{\substack{\text{Incident radiance} \\ \Rightarrow \textbf{recursive}}} \underbrace{\cos\theta_i d\omega_i}_{\substack{\text{Projected} \\ \text{Solid Angle}}}$$

Ω = domain of integration

Hemisphere if surface is opaque, otherwise the sphere

Kajiya, J. T. 1986. The rendering equation. *SIGGRAPH Comput. Graph.* 20, 4 (Aug. 1986), 143-150.

* A simplified variant on the rendering equation we saw previously N.B. Emission and geometry terms are removed

Reflectance Equation

- Actually more complex than this:
  - Integrate over entire sphere (for transparency)
  - Need to integrate over time (for exposure, motion blur)
  - Integrate over surface of lens (for aperture effects)
  - Integrate over the visible spectrum (for dispersion etc.)
  - And more...
- More accurate equation:

$$L_r(x, \lambda, \omega_r) = \int_T \int_\Lambda \int_\Omega f_r(x, t, \lambda, \omega_i, \omega_r) L_i(x, t, \lambda, \omega_i) \cos\theta_i \, d\omega_i \, d\lambda \, dt$$

  - ...and this doesn't include the camera...
- We'll need to simplify for real-time

Rendering

- Rendering: determine the *most appropriate colour to* assign to a pixel associated with an object in a scene.
- The visible colour of an object at a point depends on:
  - geometry of the object at that point (*normal direction)*
  - position, geometry and colour of the light sources (*luminaires)*
  - position and visual response of the viewer / camera
  - *surface reflectance properties of the object at that point*
  - scattering by any *participating media (e.g. smoke, rising hot air)*
- Units:
  - Physically based: radiance, irradiance, power, energy etc.
  - Many talk about intensity which is non-physically based
- Most pipelines ignore physics units: everything is a tweakable from 0.0 to 1.0

Local Illumination Requirements

- Illumination Model (or Shading Model)
  - Approximation of a BRDF

$$L_r(x,\omega_r) = \int_\Omega \boxed{f_r}(x,\boxed{\omega_i,\omega_r})L_i(x,\omega_i)\cos\theta_i d\omega_i$$

- Surface Model
  - Defines variation in BRDF over surface (e.g. texture)

$$L_r(x,\omega_r) = \int_\Omega \boxed{f_r(x,}\omega_i,\omega_r)L_i(x,\omega_i)\cos\theta_i d\omega_i$$

- Lighting Source Model
  - Defines the source of incoming illumination

$$L_r(x,\omega_r) = \int_\Omega f_r(x,\omega_i,\omega_r)\boxed{L_i(x,\omega_i)}\cos\theta_i d\omega_i$$

Light Sources

- In Real-time rendering we normally employ *isotropic point light sources defined by position, colour and power*
  - Isotropic ⇒ radiates energy equally in all directions
  - Point ⇒ has no area!

- The power per unit area (irradiance E) at a point *x* at distance *r from* the point light source may now be determined:
  - power is radiated isotropically, through the sphere centered at the light position
  - at a distance *r* from the source, the surface area of this sphere is *4πr2*
  - therefore the power per unit area at *x is:*

$$E = \frac{\Phi}{4\pi r^2}$$

  - note: this assumes the surface at *x is perpendicular to the direction to the* light source.

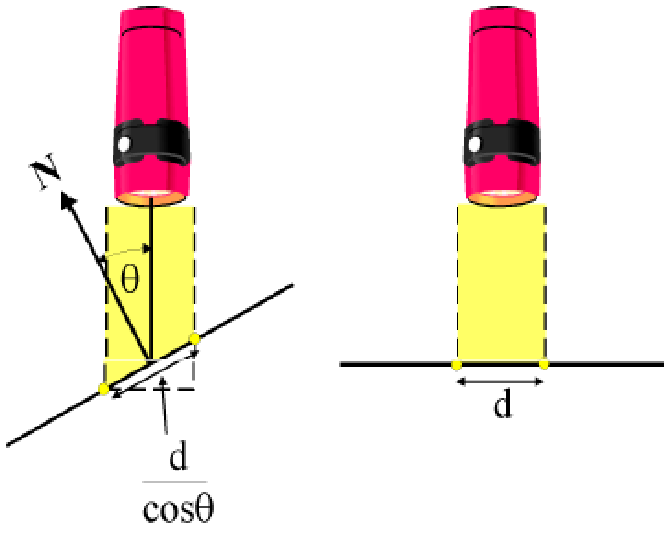
The Cosine Rule

- A surface oriented perpendicular to a light source will receive more energy (and thus appear brighter) than a surface oriented at an angle to the light source.
- The irradiance *E* is inversely proportional to area:

$$E \propto \frac{1}{A}$$

- As the area increases the irradiance decreases therefore:

$$E = \frac{\Phi \cos\theta}{4\pi r^2}$$

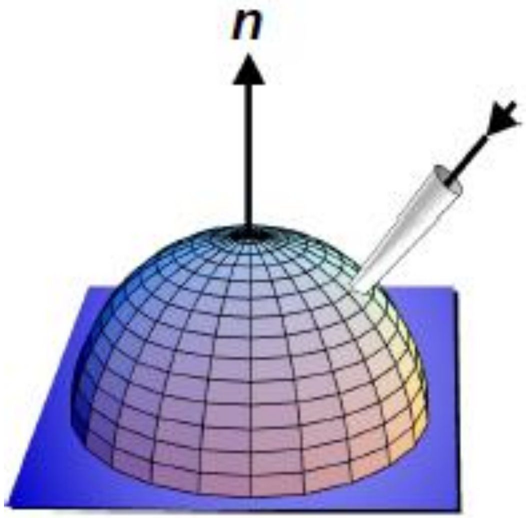


- As θ increases, the irradiance and thus the brightness of a surface decreases by cosθ

Lambertian Illumination

- We can use the cosine rule to implement shading of *Lambertian or diffuse surfaces.*
- Diffuse surfaces reflect light in all directions equally:
    - BRDF is a constant with respect to reflected direction
    - surface may be characterized by a reflectance ρ_d rather than a BRDF:

$$\rho_d(x) = \frac{\Phi_i(x)}{\Phi_r(x)}$$

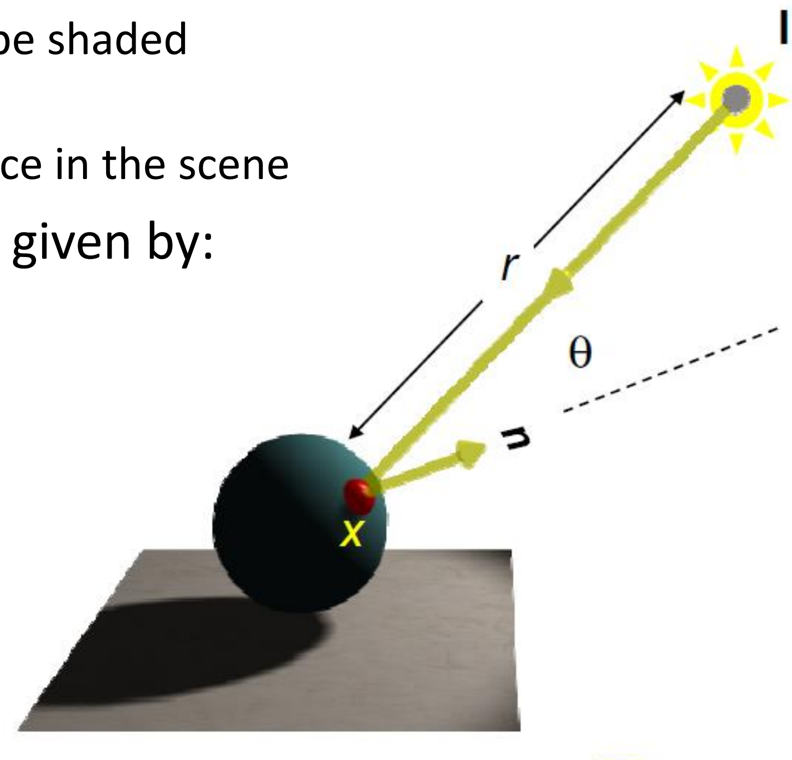


- the reflectance gives the ratio of the total reflected power to the total incident power

$$f_r(x) = \frac{\rho_d(x)}{\pi}$$

Lambertian Illumination

- To shade a diffuse surface we need to know:
  - normal to the surface at the point to be shaded
  - diffuse reflectance of the surface
  - positions and powers of the light source in the scene
- Contribution from a single source is given by:



$$L_{r,d}(\mathbf{x}, \cdot) = \frac{\rho_d}{\pi} \cos\theta \frac{\Phi_L}{4\pi r^2}$$

$$= \frac{\rho_d}{\pi} (\mathbf{n} \cdot \mathbf{l}) E$$

No dependence on directions

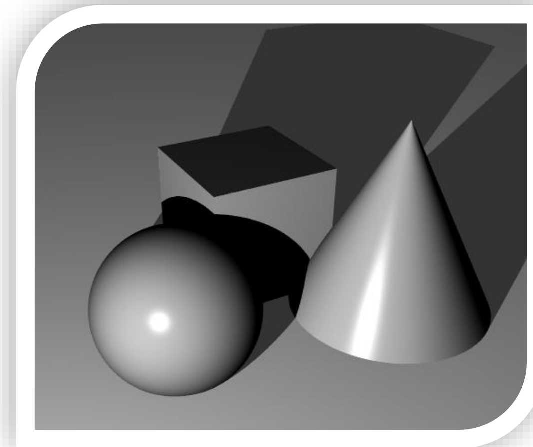
In practice repalced by single co-efficient Kd

- This is *local illumination*

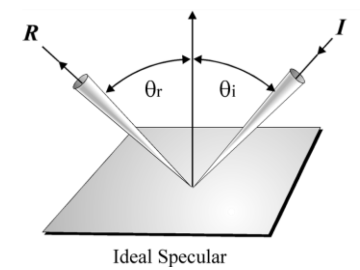
In practice:replaced by the light intensity L (attenuation accounted for elsewhere)

$$I_d = k_d (n.1) L_d$$

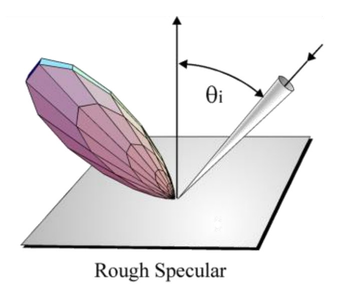
Specular Highlights



- Specular surfaces exhibit a high degree of coherence in their reflectance, i.e. the reflected radiance depends very heavily on the outgoing direction.

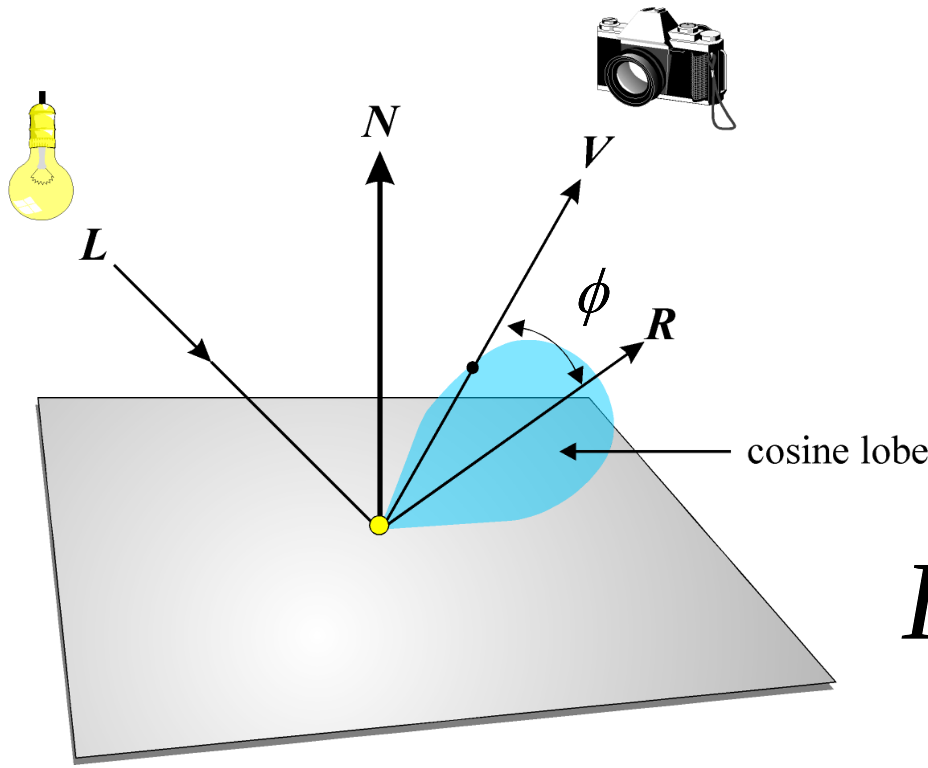


- An ideal specular surface is optically smooth (smooth even at resolutions comparable to the wavelength of light).



- Most specular surfaces (rough specular) reflect energy in a tight distribution (or lobe) centered on the optical reflection direction:

Phong Model of Specular Reflection



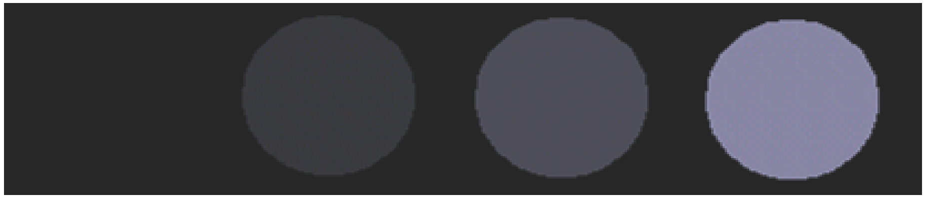
- Intensity reflected light depends on:
  - Viewer direction
  - Incoming light direction
  - Light colour + brightness (L)
  - (α) Phong exponent/shininess
  - (k_s) specular reflectivity coefficient

$$I_S = k_S L_S \cos^{\alpha} \phi$$

Local model: only consider reflections of light sources. Assume that the BRDF of shiny surfaces may be approximated by a spherical cosine function raised to a power (known as the Phong exponent, indicative of the shininess).

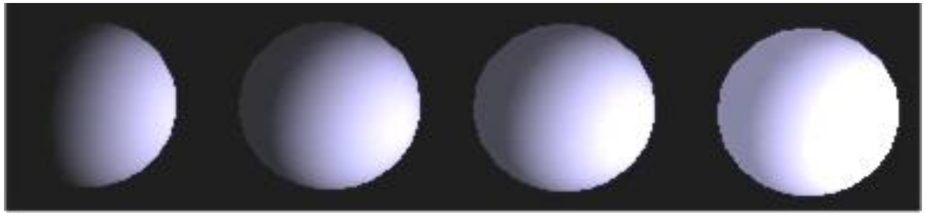
Ambient Light

- Objects lit by ambient light are lit evenly on all surfaces in all directions
- Certain lights e.g. tube lights in class-rooms or kitchens try to achieve this by using large diffusers
- Ambient illumination is characterised by an intensity L_a identical at every point in the scene



$$I = L_a k_a$$

where k_a is proportion of ambient light reflected

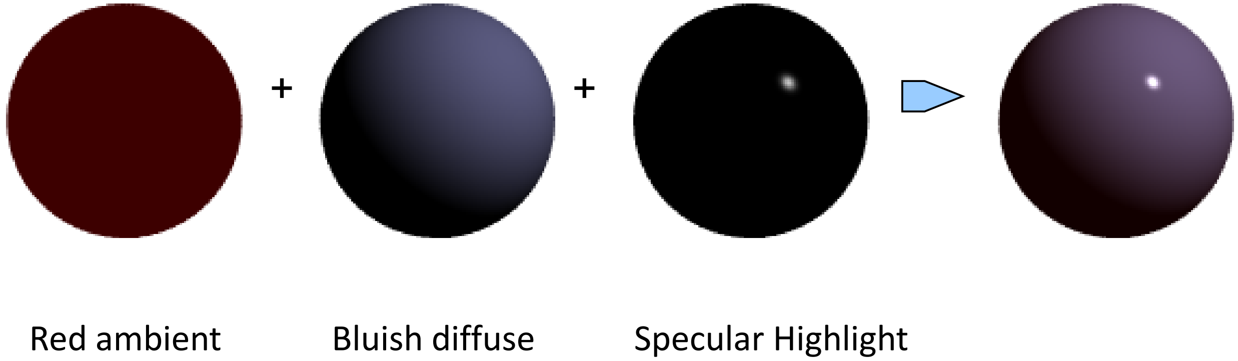


$$I = L_a k_a + I_d$$

N.B. Usually we add only very small amounts of ambient light

Putting it all together

- The intensity of light from one point is a sum of the diffuse, specular and ambient components:



Phong Illumination Model

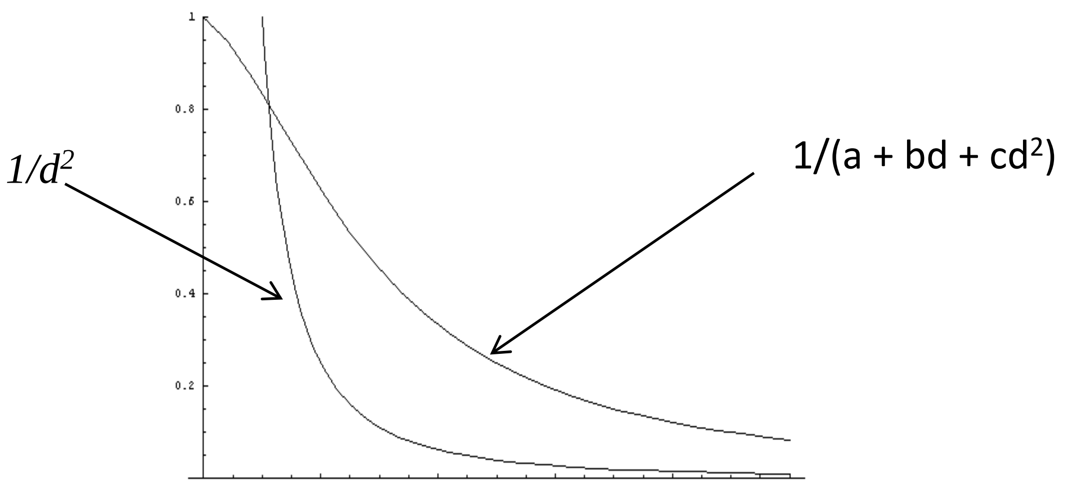
```
//.vs
varying vec3 lightDir, normal, pos;
void main()
{
lightDir = normalize(vec3(gl_LightSource[0].position - gl_ModelViewMatrix * gl_Vertex));
pos = vec3(gl_ModelViewMatrix * gl_Vertex);
normal = normalize(gl_NormalMatrix * gl_Normal);
gl_Position = gl_ModelViewProjectionMatrix * gl_Vertex;
}
```

Phong Illumination Model

```
//.fs
varying vec3 lightDir, normal, pos;
void main()
{
float i_diffuse, i_specular, i_ambient;
vec3 n;
vec3 viewDir = normalize(-pos);
vec3 refDir = normalize(-reflect(lightDir, normal));
vec4 color;
n = normalize(normal);
i_diffuse = max(dot(lightDir,n), 0.0);
i_specular = max( pow( dot(refDir, viewDir), 100), 0.0);
i_ambient = 1;
color = vec4(0.5, 0.5, 1.0, 1) * i_diffuse + vec4(1, 1, 1, 1)* i_specular + i_ambient*vec4(.2, 0, 0, 1);
gl_FragColor = color;
}
```

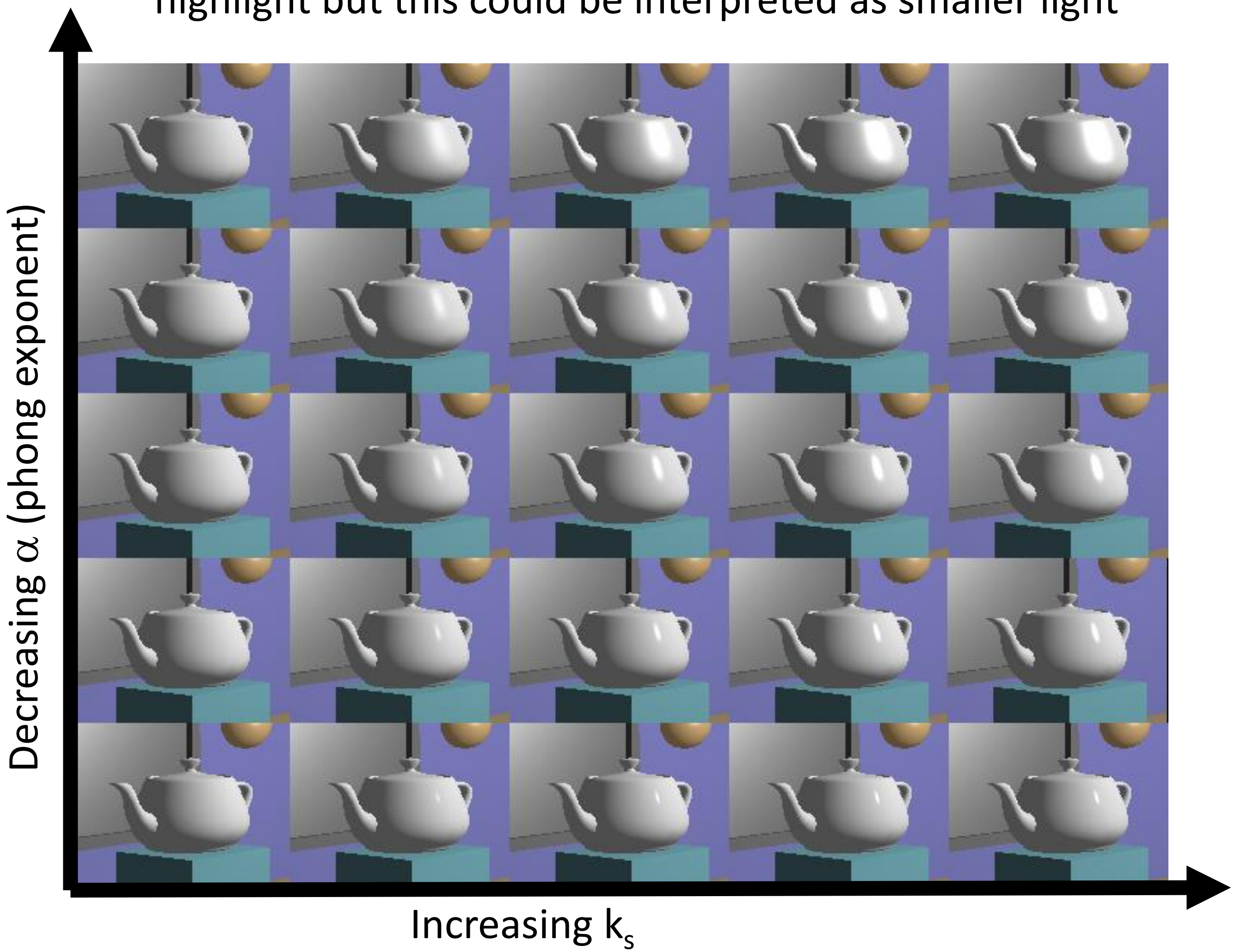
Light Attenuation

- Physically accurate attenuation
  - Attenuation factor = $1/d^2$
  - d is distance from light to surface
  - In practice this is very harsh – falls off too quickly
- Quadratic Attenuation with softer fall off
  - Attenuation factor = 1/(a + bd + cd2)



Phong Shading Limitations

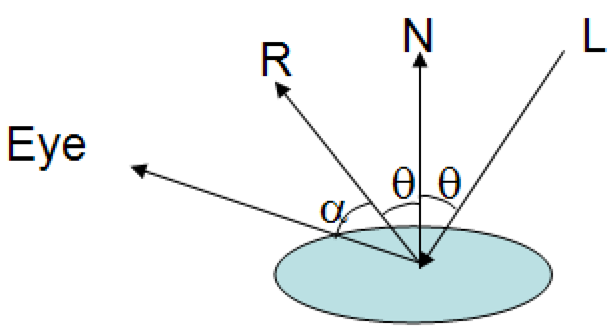
For a point light source: increasing shininess leads to smaller highlight but this could be interpreted as smaller light



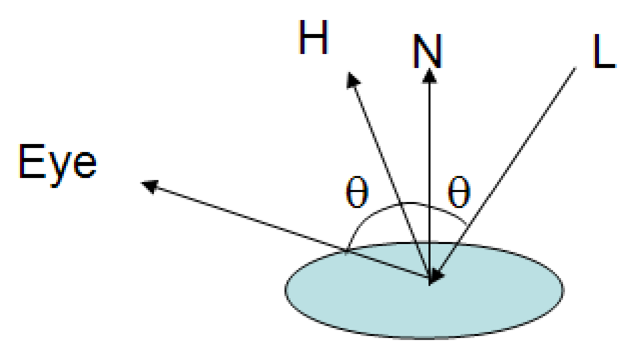
Phong & Blinn-phong models (1/2)

[1975]

- The formula for specular is the Phong model
  - not physically correct
  - looks nice in practice and very simple to evaluate
- Blinn simplification
  - use angle with half-vector
  - also standard in Computer Graphics



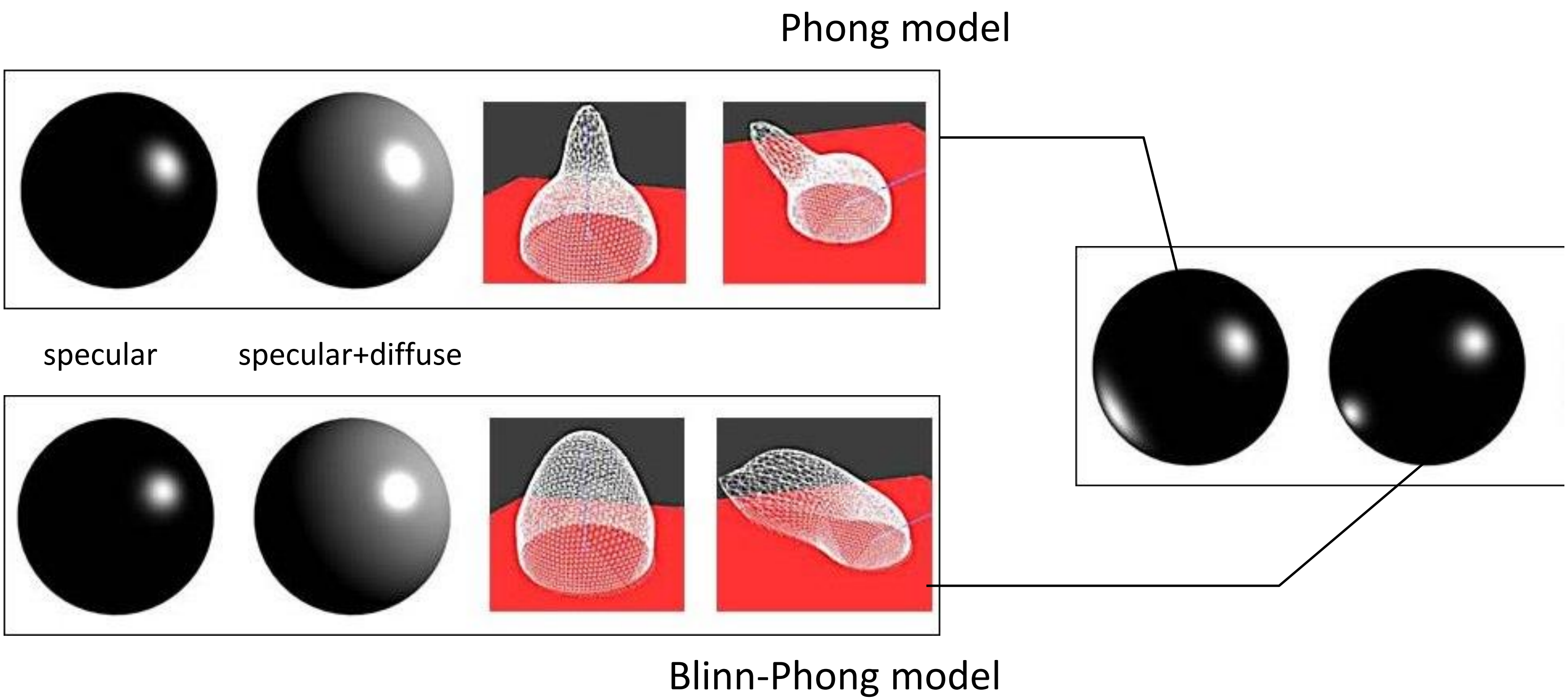
Phong



Blinn-Phong

Phong & Blinn-phong models (2/2)

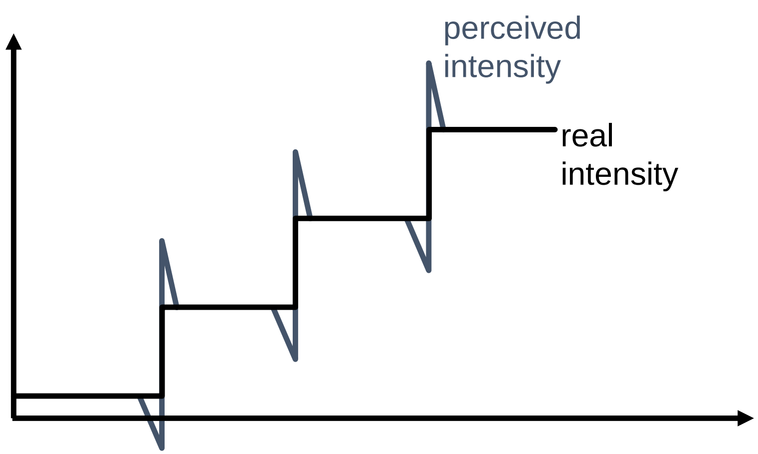
- Difference is a matter of taste!
- Blinn-phong tends to be more predictable

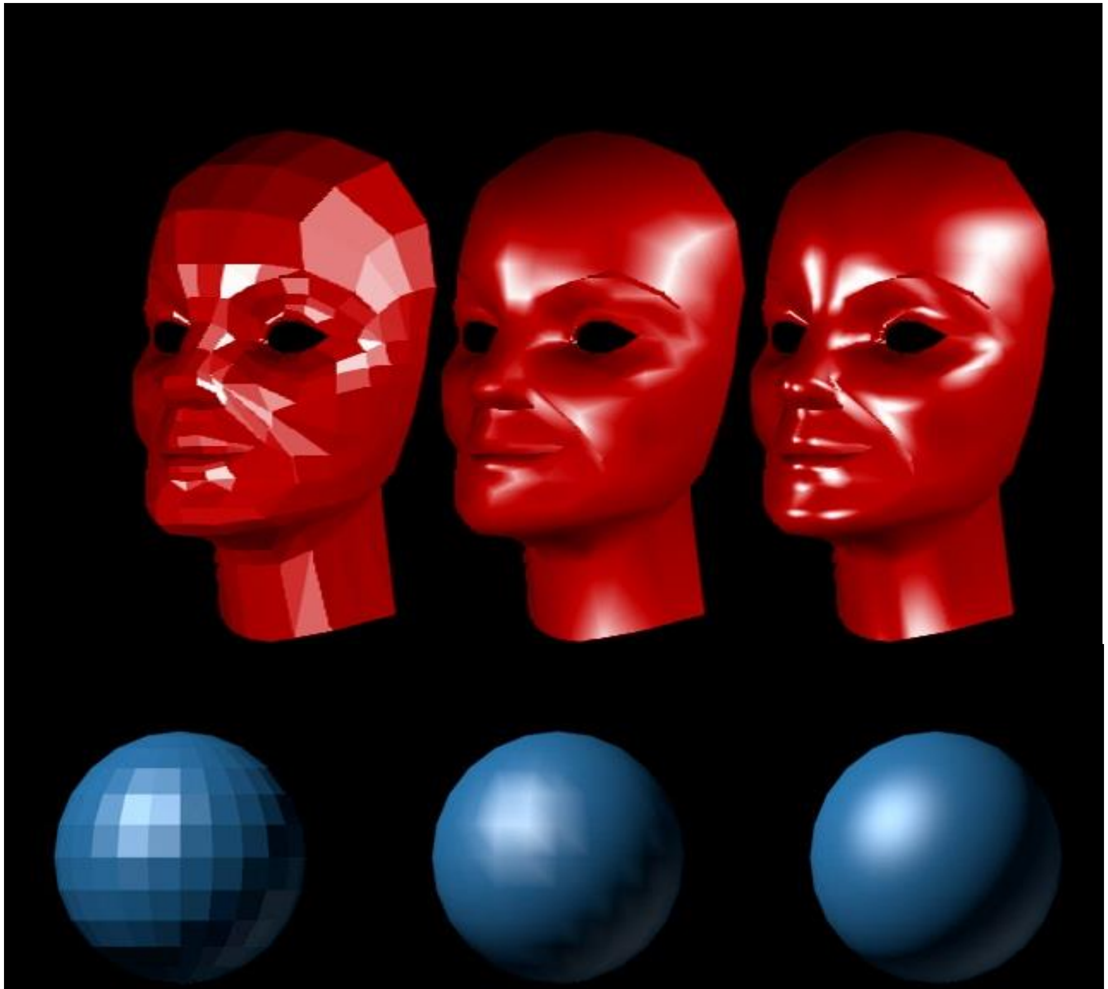


Shading In-Brief

Flat vs. Gouraud shading

- Flat shading creates “facetted” objects
  - requires highly tesselated surfaces





Flat, Gouraud, Phong Shading

Gouraud vs. Phong shading

- Means per-pixel vs. per vertex shading
- Per pixel is much nicer
  - And more "correct"

A historical note on graphics terms...

- Lambertian Illumination Model
  - Uses diffuse only reflectance
- Phong Illumination Model
  - Simulates diffuse, specular and ambient reflectance terms
- Gouraud Shading
  - Evaluate lighting model at vertices only, interpolate colour for each pixel
- Phong Shading
  - Evaluate Phong Model at each pixel
- These distinctions are less relevant with PS and VS shaders

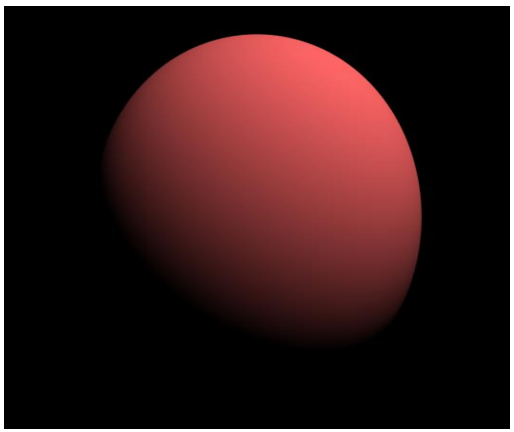
Advanced Illumination

Phong Illumination

- Local illumination is mainly combination of
  - Diffuse
    - Perfect Diffuse when light is scattered equally in all directions
    - E.g. from perfect matte surface
  - Specular
    - Perfect specular when light strikes mirror-like surface
    - Thin beam of light in reflected direction $\theta_i = \theta_r$
- Combining elements of both approximates some complexities of real-behaviour

  ... However Phong itself misses a few things due to its specific approximations

Real-World BRDFs



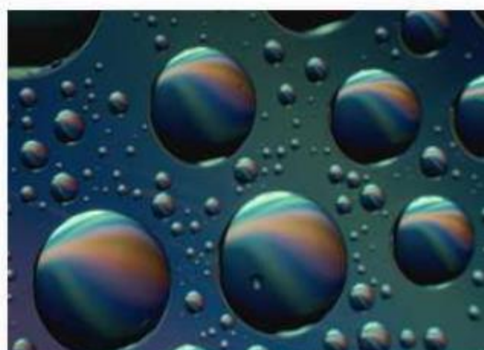
Complex Surface Scattering Examples


Multi-layered Surfaces


Coloured Glass


Human Skin


Stone


Thin Film


Tarnished Metal

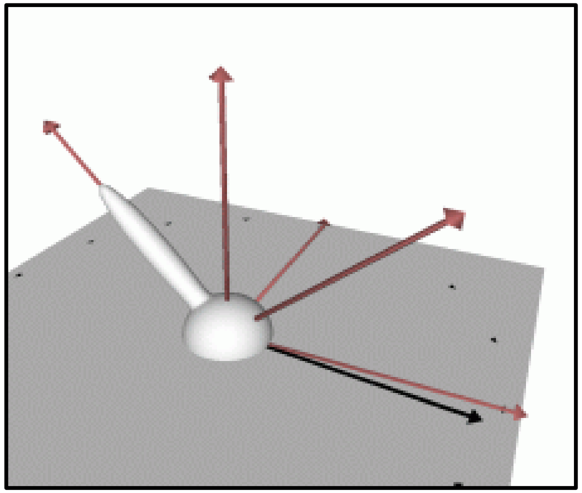
Lighting Models: Beyond Phong

- Lighting models
  - Torrance-Sparrow [1967]
  - Blinn [1977]
  - Cook-Torrance [1982]
  - Ward Anisotropic [1992]
  - Oren-Nayar [1994]
  - Schlick [1994]
  - He-Torrance-Sillion-Greenberg [1991]
  - Lafortune et al. [1997]
  - Ashikhmin-Shirley [2000]
  - Hapke/Lommel-Seeliger [1963]
  - Strauss [1990]
  - Poulin-Fournier [1990]
  - Minnaert [1941]

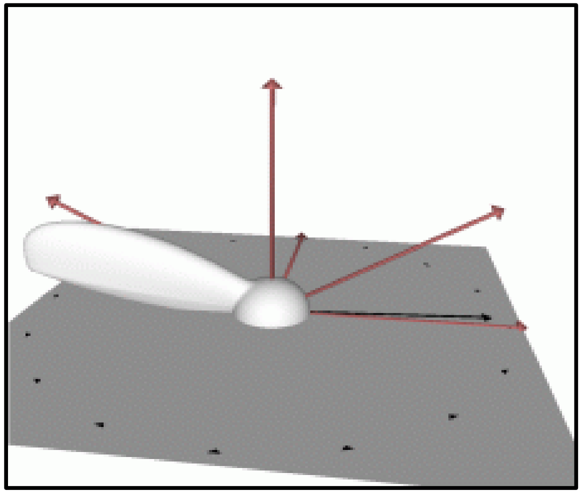
Minnaert - An empirical model with applications in Fabric modelling (velvet, satin)

BRDF

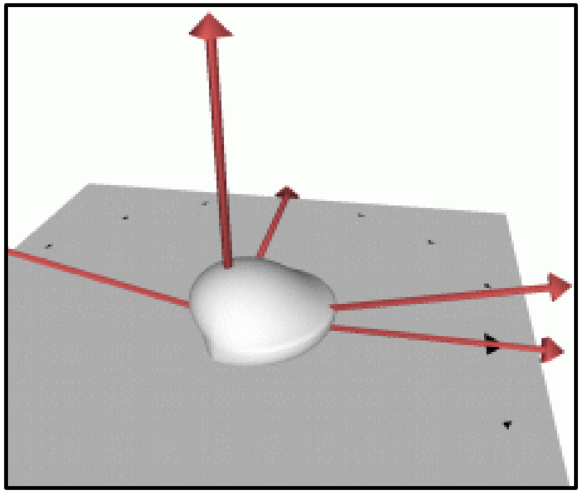
- The bidirectional reflectance distribution function is surface



Phong BRDF



Cook-Torrance BRDF



Oren-Nayar BRDF

- Given incoming light direction, v_i, and outgoing light direction, v_o, how much light is reflected?
- Can depend on wavelength, λ.

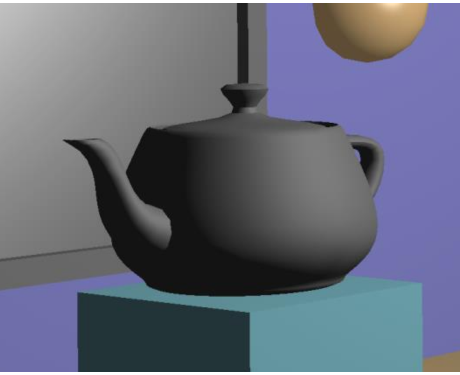
Measuring BRDFs

- Gonioreflectometer: measure reflected light for various light and reflection directions of the hemisphere.

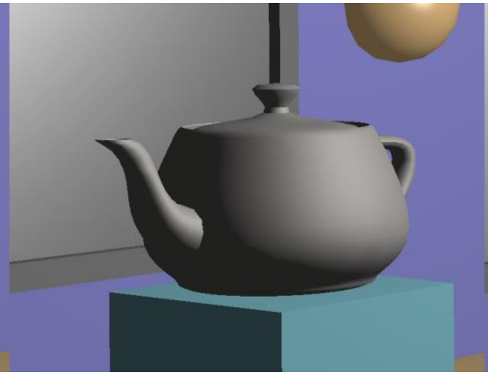
Real BRDF

Selection of Models described in Watt Policarpo – the Computer Image

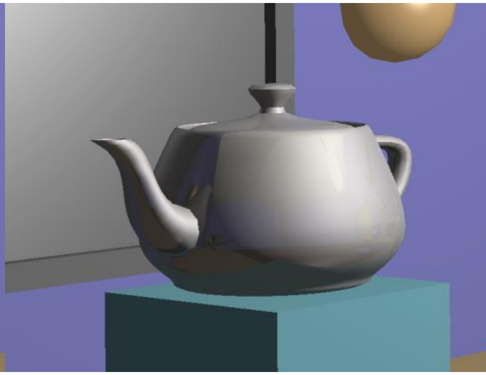
Iron



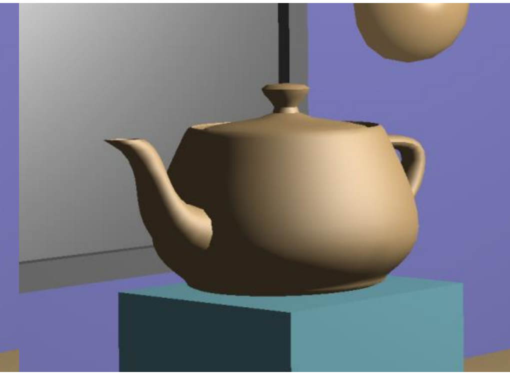
Steel



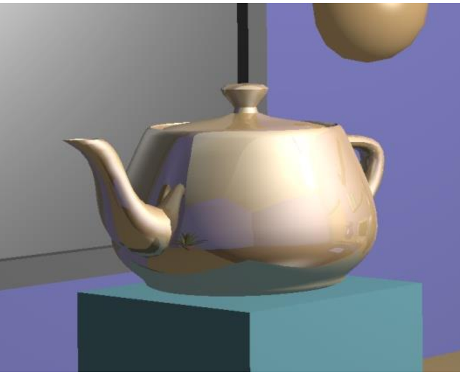
Stainless Steel



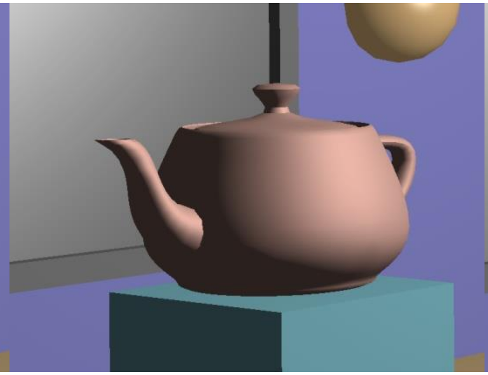
Antique Brass



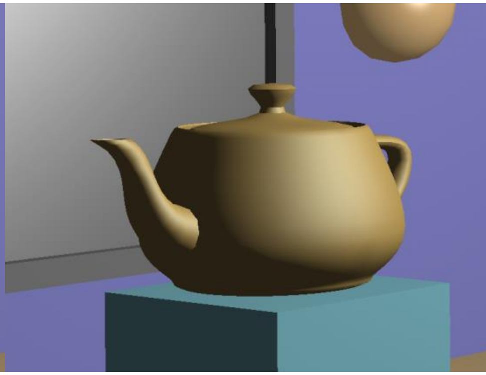
Polished Brass



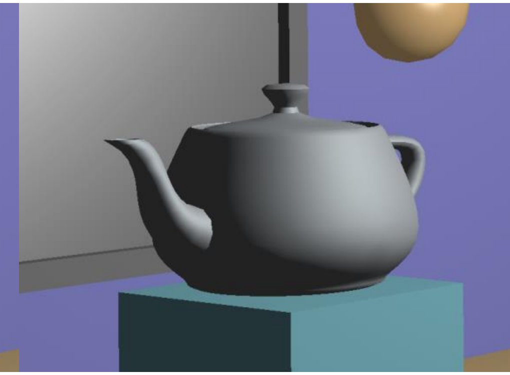
Copper



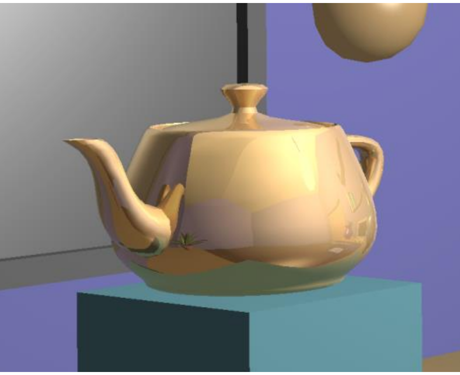
Bronze



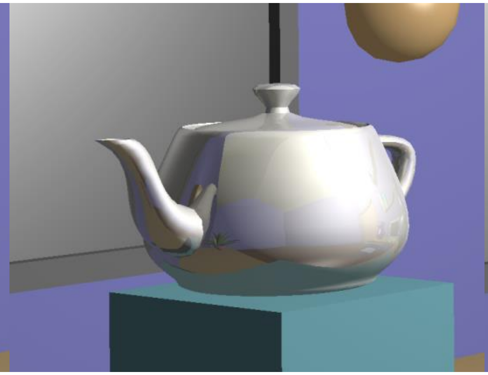
Lead



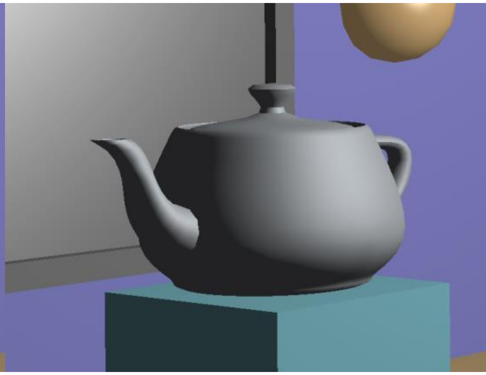
Polished Gold



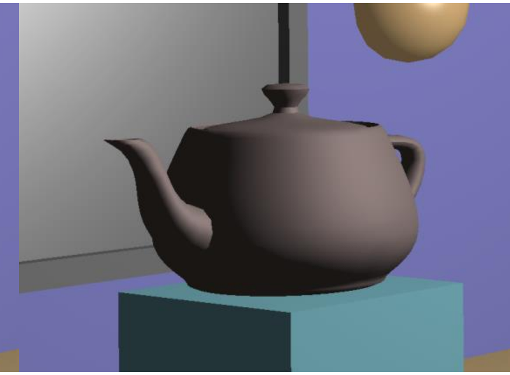
Silver Plate



Chromium Plate



Graphite



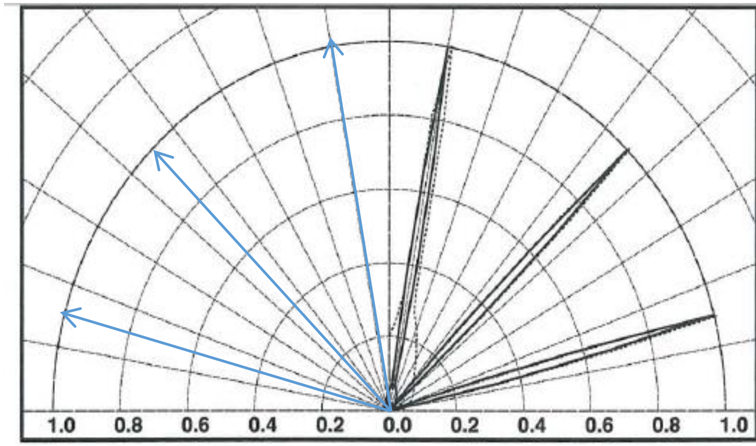
Real-BRDF

- In reality
  - function is not monochromatic: different behaviours for different wavelengths (separate BRDF required)
  - Scattering of light by atmosphere
  - Sub surface scattering
- Formally: BRDF is a four-dimensional function that defines how light is reflected at an opaque surface.

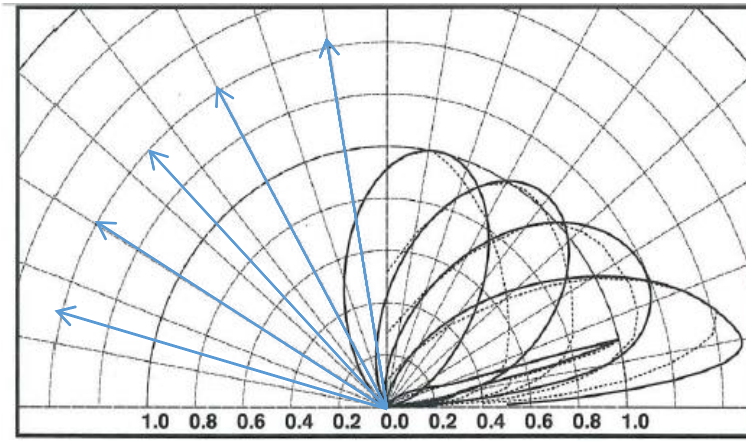
[Nicodemus 1965]

  - Based on (with respect to the surface normal n)
    - incoming light direction, ωi,
    - outgoing direction, ωo,
  - return the ratio of reflected radiance exiting along ωo to the irradiance incident on the surface from direction ωi.
  - has units sr^{-1}, with steradians (sr) being a unit of solid angle.

Real BRDF – variant with λ



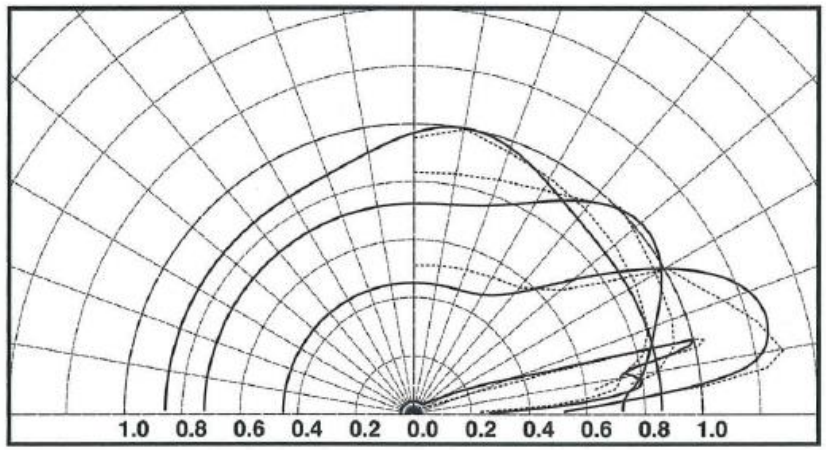
- Normalized BRDF's of roughened aluminum at
  - incidence angles of θ_i=10°, 45°, and 75°.
  - wavelength λ = *2.0μm.*
- Strong specular reflection at this wavelength



- Normalized BRDF's of roughened aluminum at
  - incidence angles of θ_i = 10°, 30°, 45°, 60°, and 75°.
  - λ = *0.5/μm.*
- Strong directional diffuse and emerging specular reflection at this wavelength.

- Images From: Xiao D. He, Kenneth E. Torrance, Francois X. Sillion, and Donald P. Greenberg. 1991. **"A comprehensive physical model for light reflection."** In *Proceedings of the* SIGGRAPH '91.

Real BRDF – variant with θ



- Normalized BRDF's of roughened magnesium oxide ceramic
  - incidence angles of θ_i = 10°,45°,60° and 75
  - λ= *0.5μm.*
- Strong uniform diffuse and emerging specular reflection

- Image From: Xiao D. He, Kenneth E. Torrance, Francois X. Sillion, and Donald P. Greenberg. 1991. **“A comprehensive physical model for light reflection.”** In *Proceedings of the* SIGGRAPH '91.

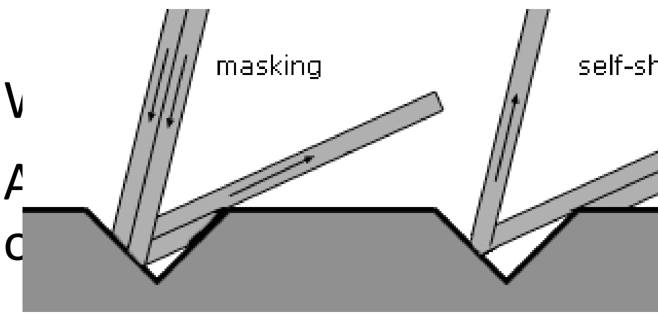
Perfect Reflection Models

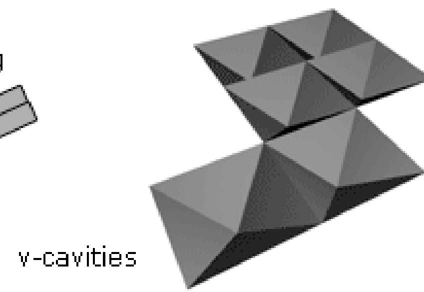
- Most imitations of real BRDF obtained through small changes to specular component – usually very subtle effects
- We will look at:
  - Emprically spread perfect diffuse. Phong (1977) : already discussed this
  - Physically based specular. Blinn (1977), Cook-Torrance (1982)

Torrance-Sparrow (1967)

- Gaussian distribution of microfacets:
- Analysis of the geometrical attenuation in symmetrical V-shaped grooves
  - Masking, shadowing

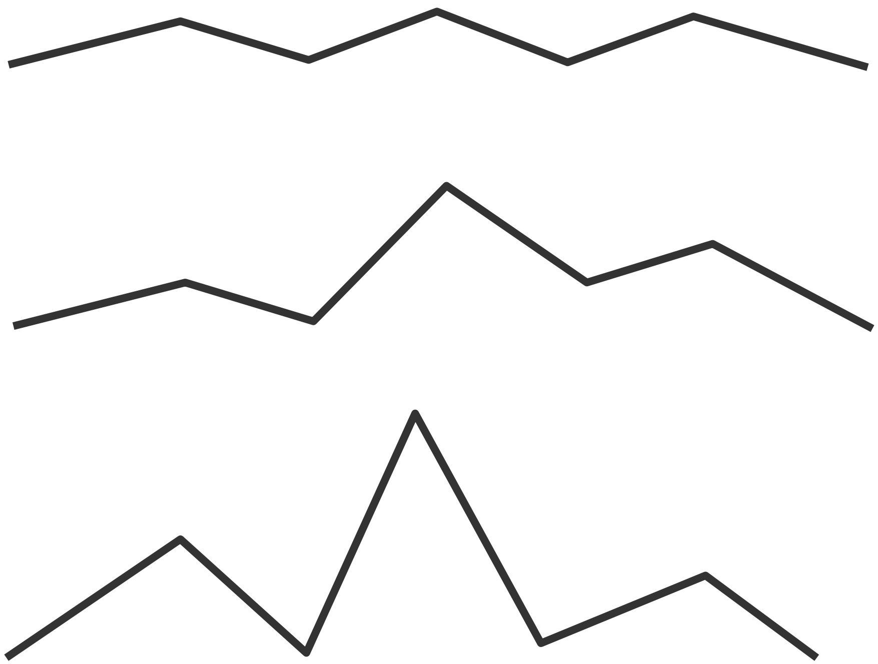
- V
- A d their lighting models
  c

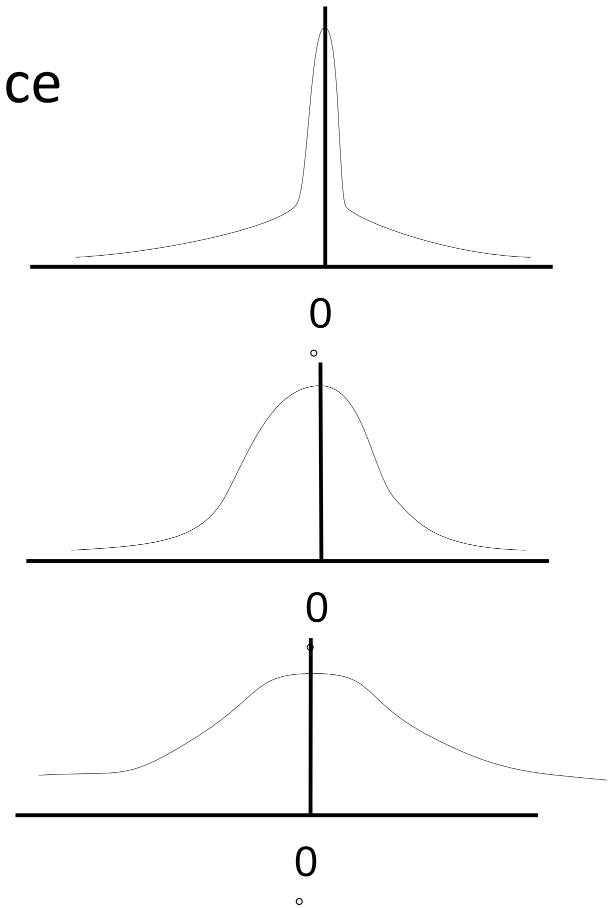




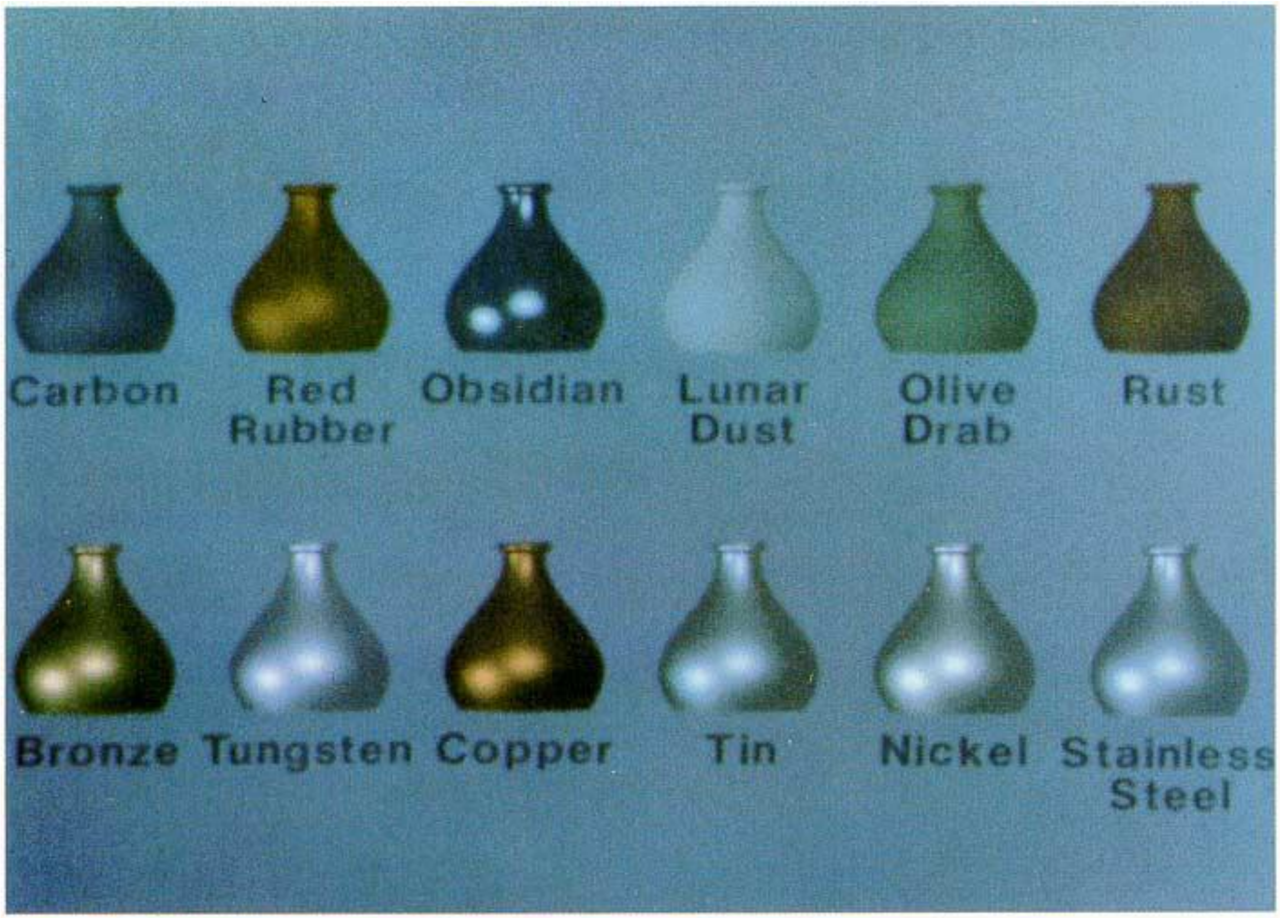
Distribution of microfacets

- Phong-specular assumes uniform surface
  - In reality : not necessarily uniform





Cook-Torrance results (1982)

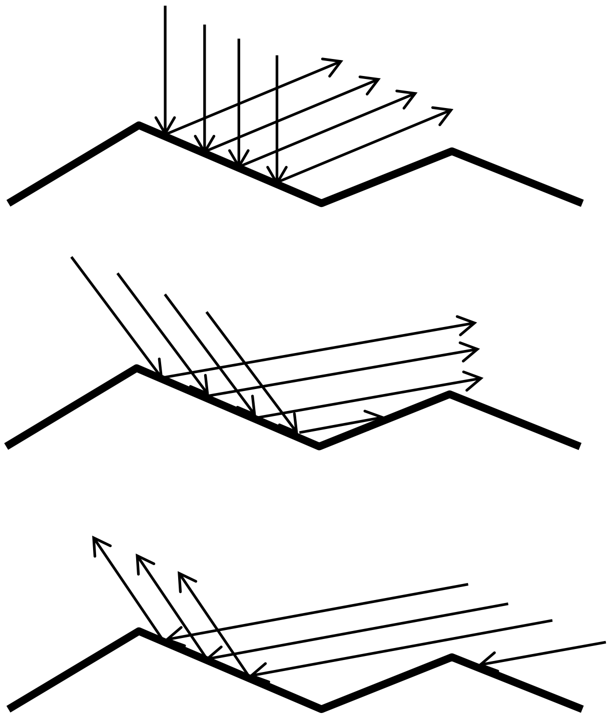


Cook-Torrance

- Metals
  - Copper, Tin, Nickel, Stainless Steel
- Nonmetals
  - Carbon, Rubber, Obsidian, Matte finish

Cook-Torrance (1982)

- G: geometrical attenuation



- No attenuation: G = 1
- Masking : G =
- Shadowing: G =

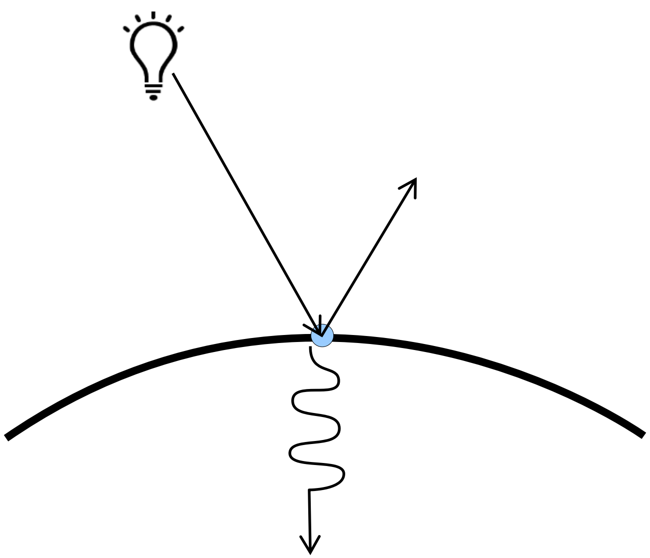
Consider all 3 states by taking the minimum over all possibilities:

Cook-Torrance

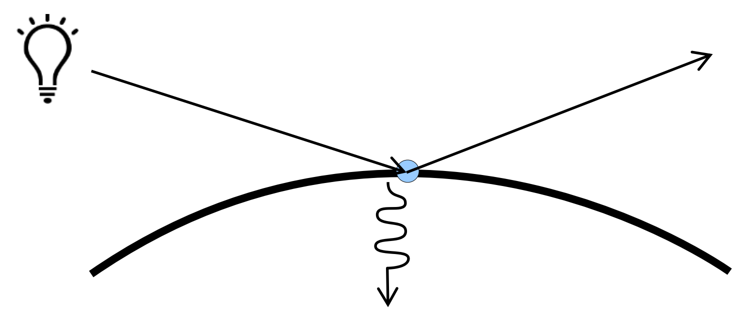
- Blinn (1977) not to be confused with Blinn-Phong shading
  - Microfacets are ellipsoids
    - c is the eccentricity of the ellipsoids
      - c=0: very shiny
      - c=1: very diffuse
- Cook Torrance extended Blinn to acount for spectral composition of highlights with dependency on
  - material type
  - angle of incidence
- Subtle effect on size and colour of the highlight compared to Phong – only changes the specular component
- For shiny metalic surfaces

Fresnel Effect : Imperfect reflection

- Degree of absorption depends on angle of incidence



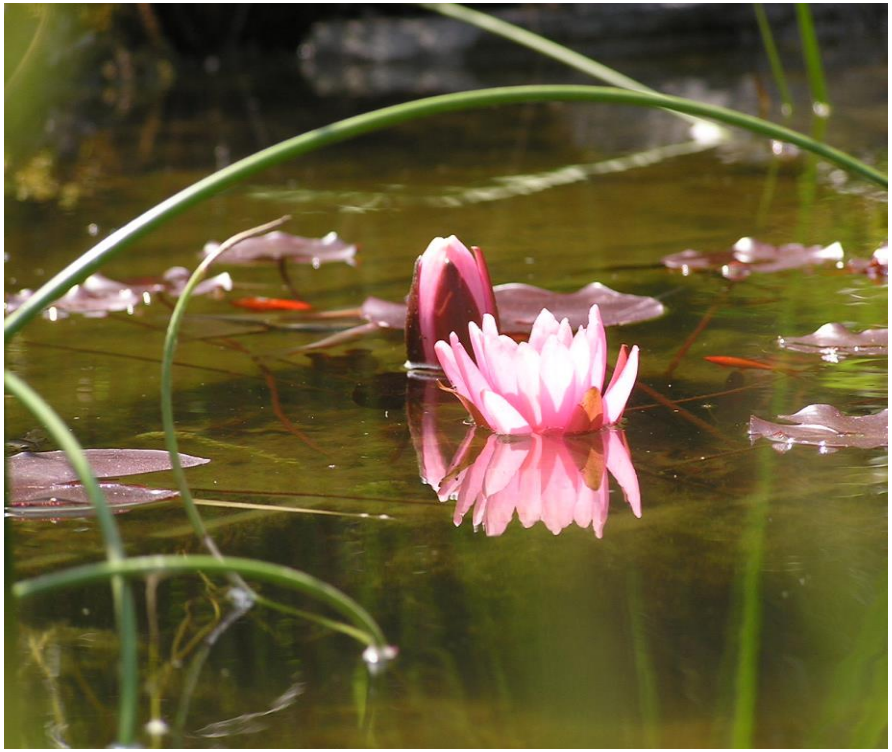
More absorption



Less absorption

Fresnel Effect

- Has major impact on water rendering
- Degree of **refraction** depends on angle of incidence



Ward

- Physically based
  - Measurement on real material
    - Gonio-réflectometer
  - Data sets
- Approximation by gaussians
- Anisotropic model