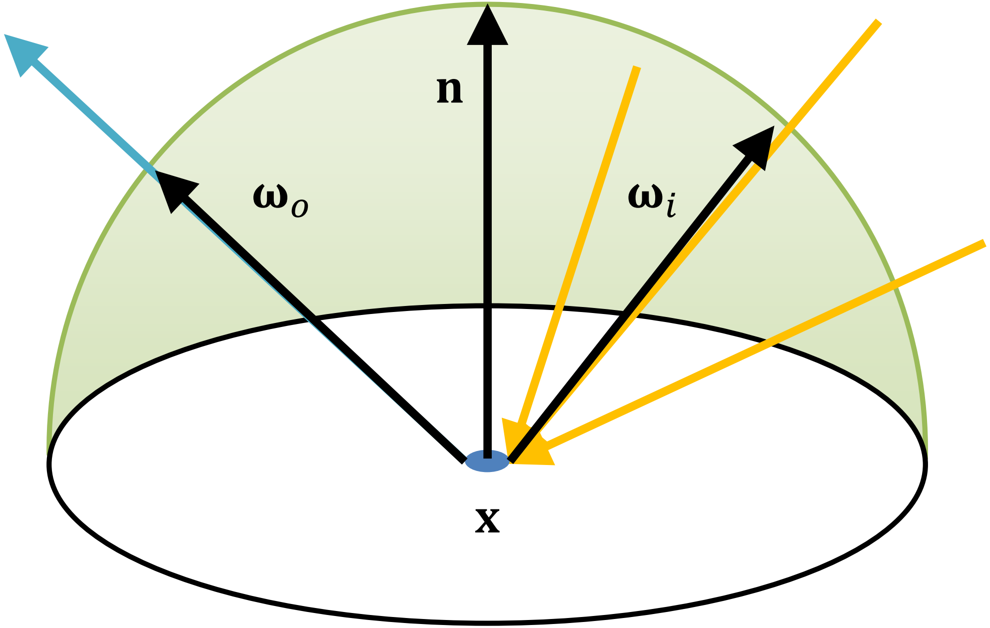
Image courtesy of Epic Games

Shading Models

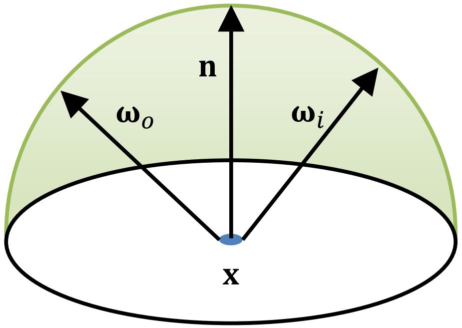
Shading!?

- Now we know which Pixels are occupied by which object.
  - e. g. from rasterization
- So what color should they be?
  - How is light reflected by an object?

Recap: Rendering Equation



Recap: Rendering Equation



indirect illumination

BRDF

$$L(\mathbf{x},\boldsymbol{\omega}_o) = \underbrace{L_e(\mathbf{x},\boldsymbol{\omega}_o)}_{\text{emitted light}} + \underbrace{\int_{\Omega} L(\mathbf{x},\boldsymbol{\omega}_i) f_r(\mathbf{x},\boldsymbol{\omega}_i,\boldsymbol{\omega}_o)\, d\boldsymbol{\omega}_i}_{\text{reflected light}}$$

BRDF

- The **B**idirectional **R**eflectance **D**istribution **F**unction
- describes how a surface reflects light.
  - at location $\mathbf{x}$
  - from incoming direction $\boldsymbol{\omega}_i$
  - into outgoing direction $\boldsymbol{\omega}_o$

$$f_r(\mathbf{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o)$$

BRDF

- reciprocity

$$f_r(\mathbf{x}, \boldsymbol{\omega}_1, \boldsymbol{\omega}_2) = f_r(\mathbf{x}, \boldsymbol{\omega}_2, \boldsymbol{\omega}_1) \;\; \forall \boldsymbol{\omega}_1, \boldsymbol{\omega}_2$$

- energy conservation

$$\int_{\Omega} f_r(\mathbf{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o)\, d\boldsymbol{\omega}_i \leq 1 \;\; \forall \boldsymbol{\omega}_o$$

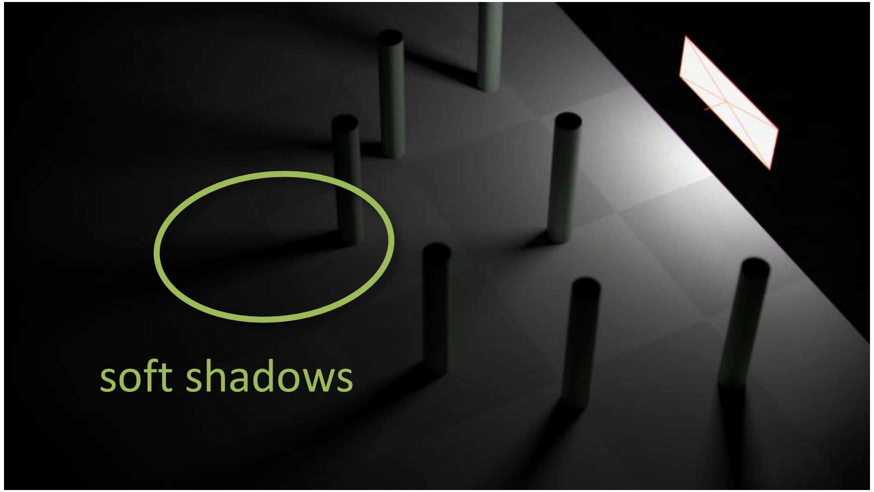
- positivity

$$f_r(\mathbf{x}, \boldsymbol{\omega}_1, \boldsymbol{\omega}_2) \geq 0$$

Light Sources

- most general light source: area light
- problem: at every location, light is coming from a range of directions
  - integration needed
  - analytic solution only for extremely trivial cases

  → simplification needed



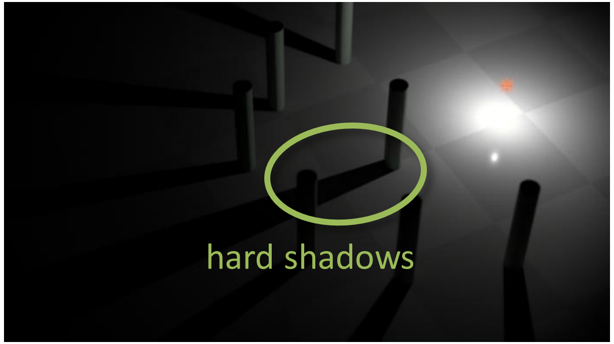
Light Sources

- To achieve real-time:
  - local shading
    - direct illumination only
    - $O(crazy)$ reduces to $O(n)$
  - consider analytical light sources only:
    - point light (infinitesimally small)
    - directional light (infinitely far away)

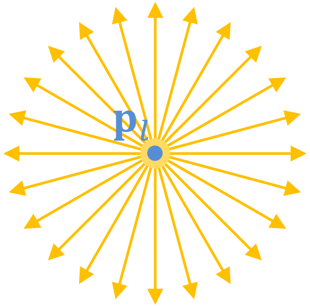
    → light just from a single direction $\mathbf{l}$

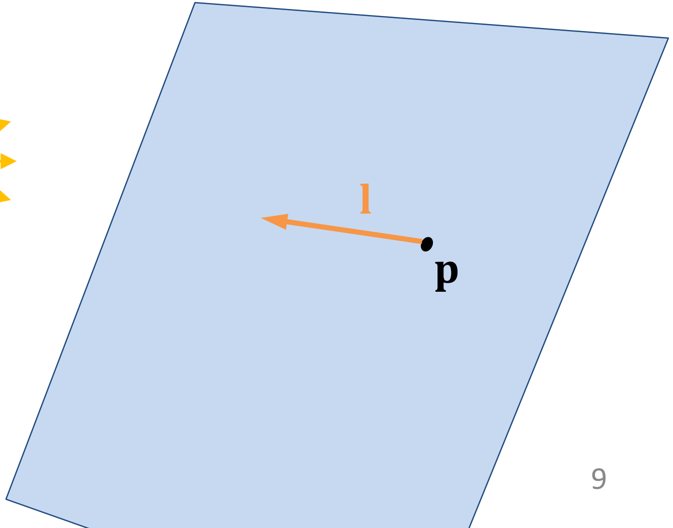
    → integral goes away

Point Light

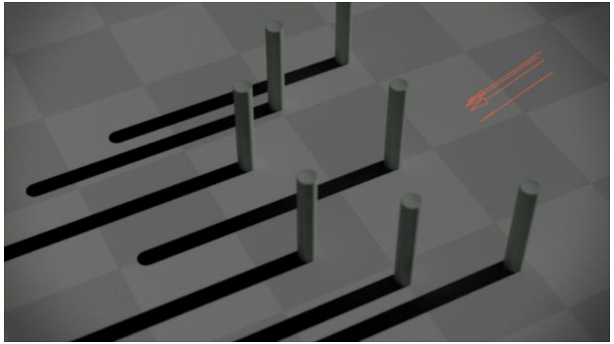


- Simplification: light source infinitesimally small
  - At every location $\mathbf{p}$, light incoming from a single direction only
- Parameters:
  - $\mathbf{p}_l$ light position

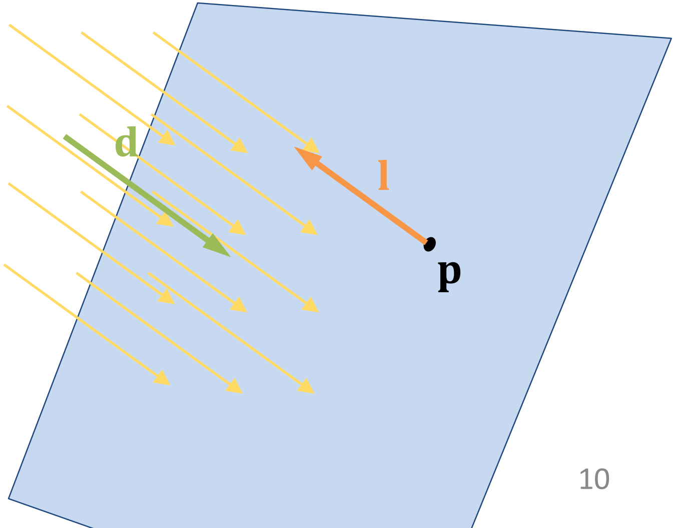




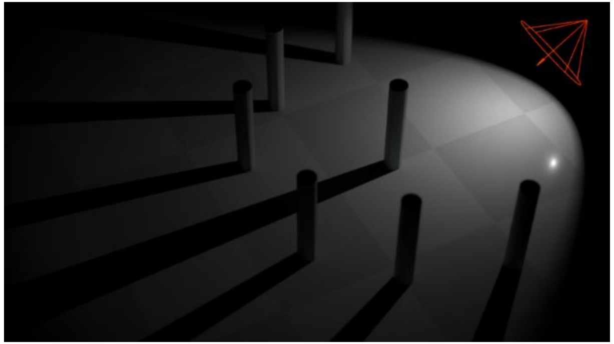
Directional Light



- Simplification: light source infinitely far away
  - At every location $\mathbf{p}$, light from the same direction only
- Parameters:
  - $\mathbf{d}$ light direction



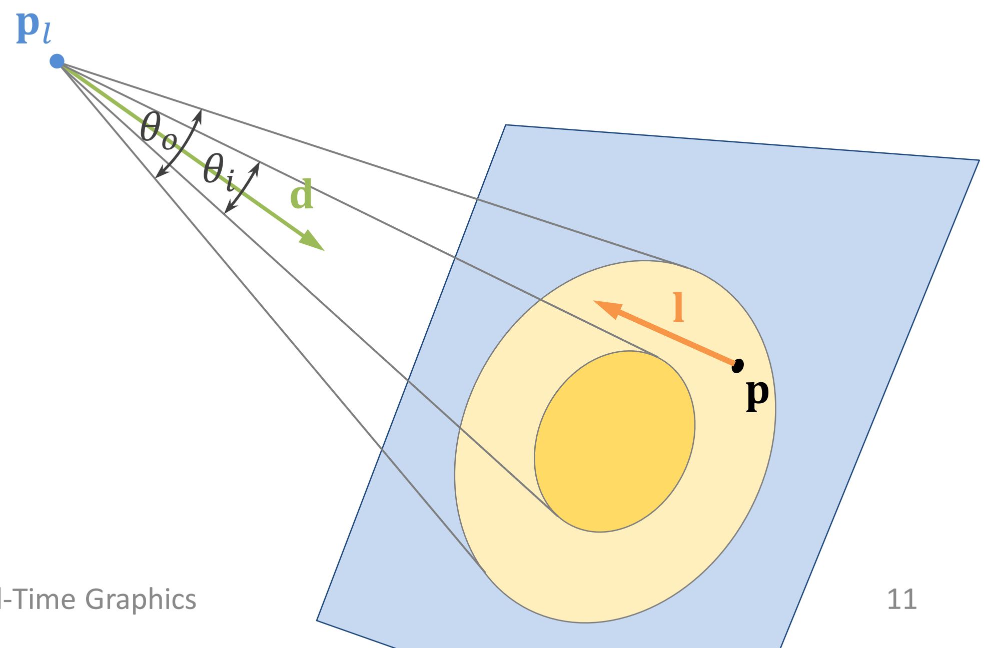
Spot Light



- like point light

\+ volume constrained to a cone

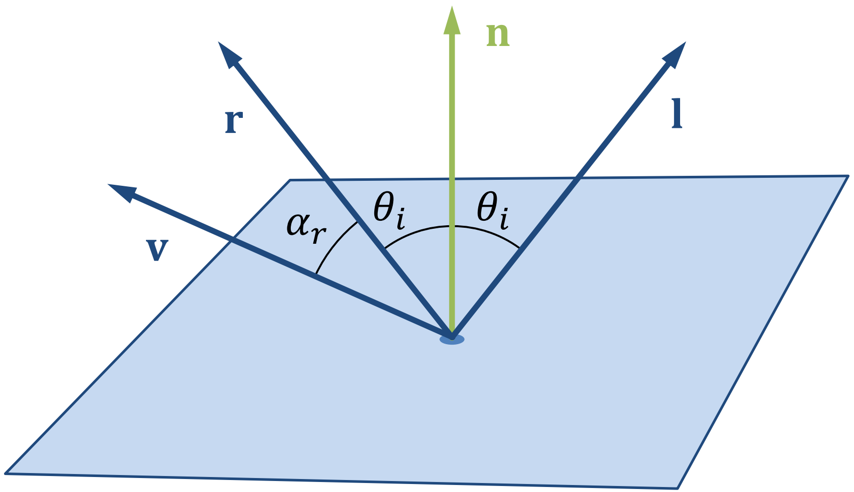
- Parameters:
  - $\mathbf{p}_l$ light position
  - $\mathbf{d}$ cone direction
  - θ_i inner cone angle
  - θ_o outer cone angle



Recap: Phong Shading

- $\mathbf{L}_o = \mathbf{c}_e + (\mathbf{c}_d \circ \cos\theta_i + \mathbf{c}_s \circ (\cos\alpha_r)^m) \circ \mathbf{B}_L$

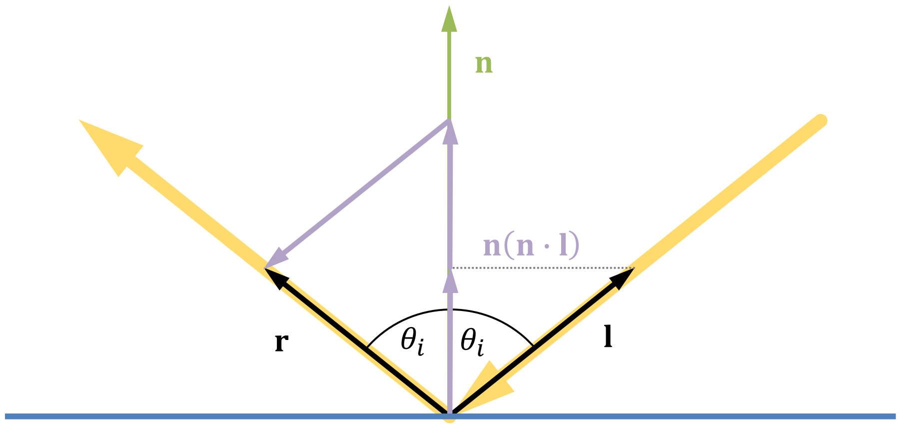
  - $\mathbf{c}_e$ emissive color
  - $\mathbf{c}_d$ diffuse color
  - $\mathbf{c}_s$ specular color
  - m specular power
  - $\mathbf{B}_L$ light color



$$\mathbf{r} = 2\mathbf{n}(\mathbf{n} \cdot \mathbf{l}) - \mathbf{l}$$

*all vectors assumed to be normalized

Reflection



$$\mathbf{r} = 2\mathbf{n}(\mathbf{n} \cdot \mathbf{l}) - \mathbf{l}$$

(given that $\mathbf{n}$ and $\mathbf{l}$ are normalized)

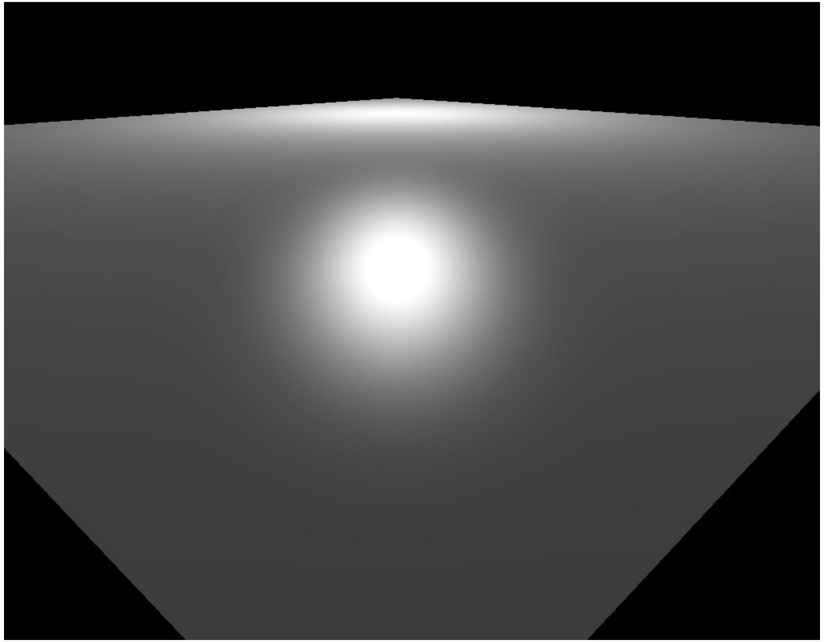
Phong Shading

- phenomenological
  - not a physically meaningful BRDF
- highlights always circular
- energy conservation ignored
  - large m should lead to smaller, but also stronger highlight

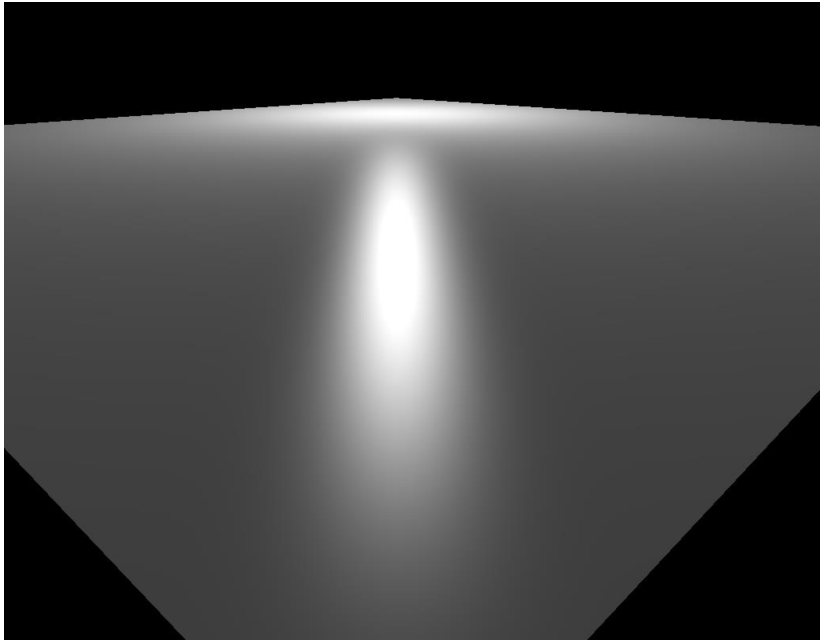
  $\rightarrow$ normalization needed

Highlight Shape

Phong



Blinn-Phong

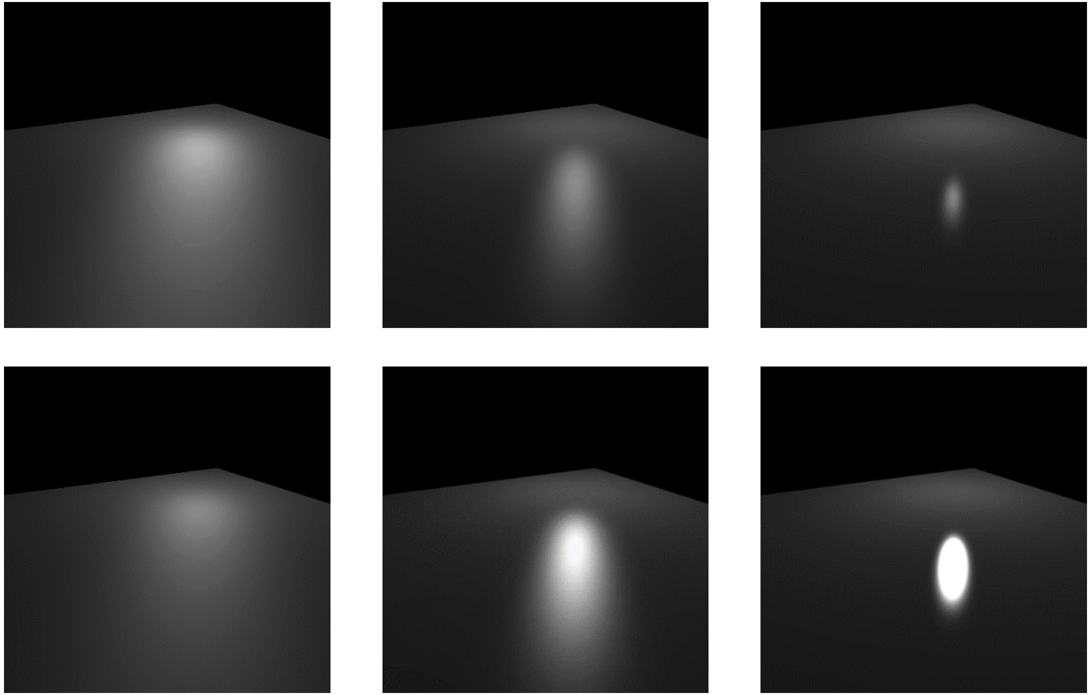


Normalization

$m = 8$ $m = 48$ $m = 512$

unnormalized

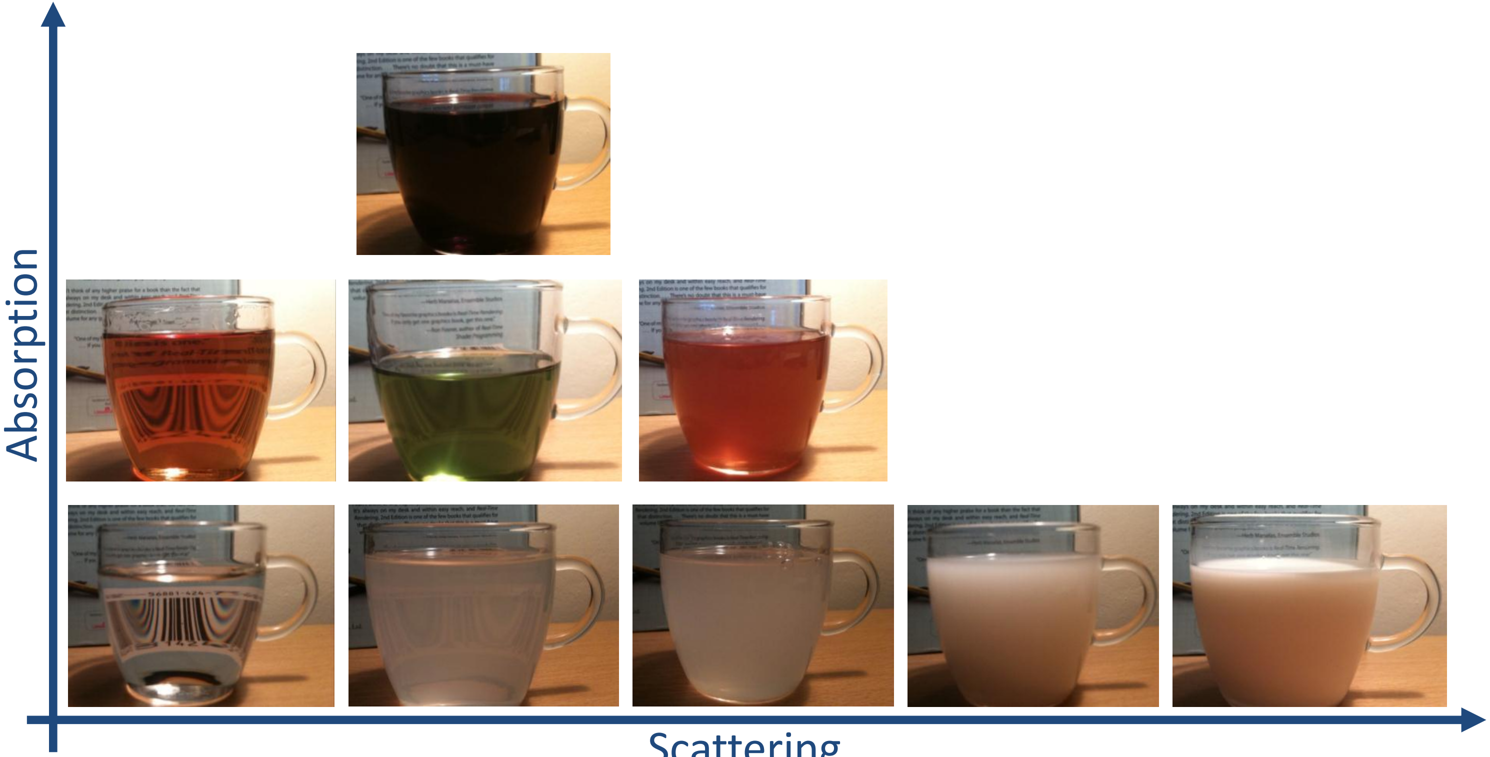
normalized



Some Physics

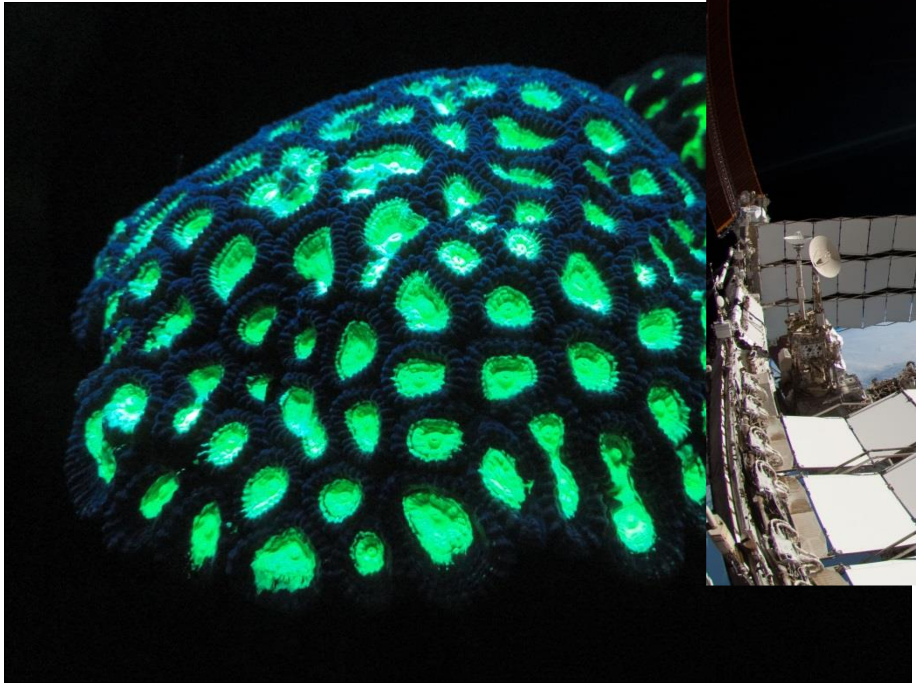
- All light-matter interaction boils down to:
  - scattering
  - absorption
  - emission

Absorption vs. Scattering

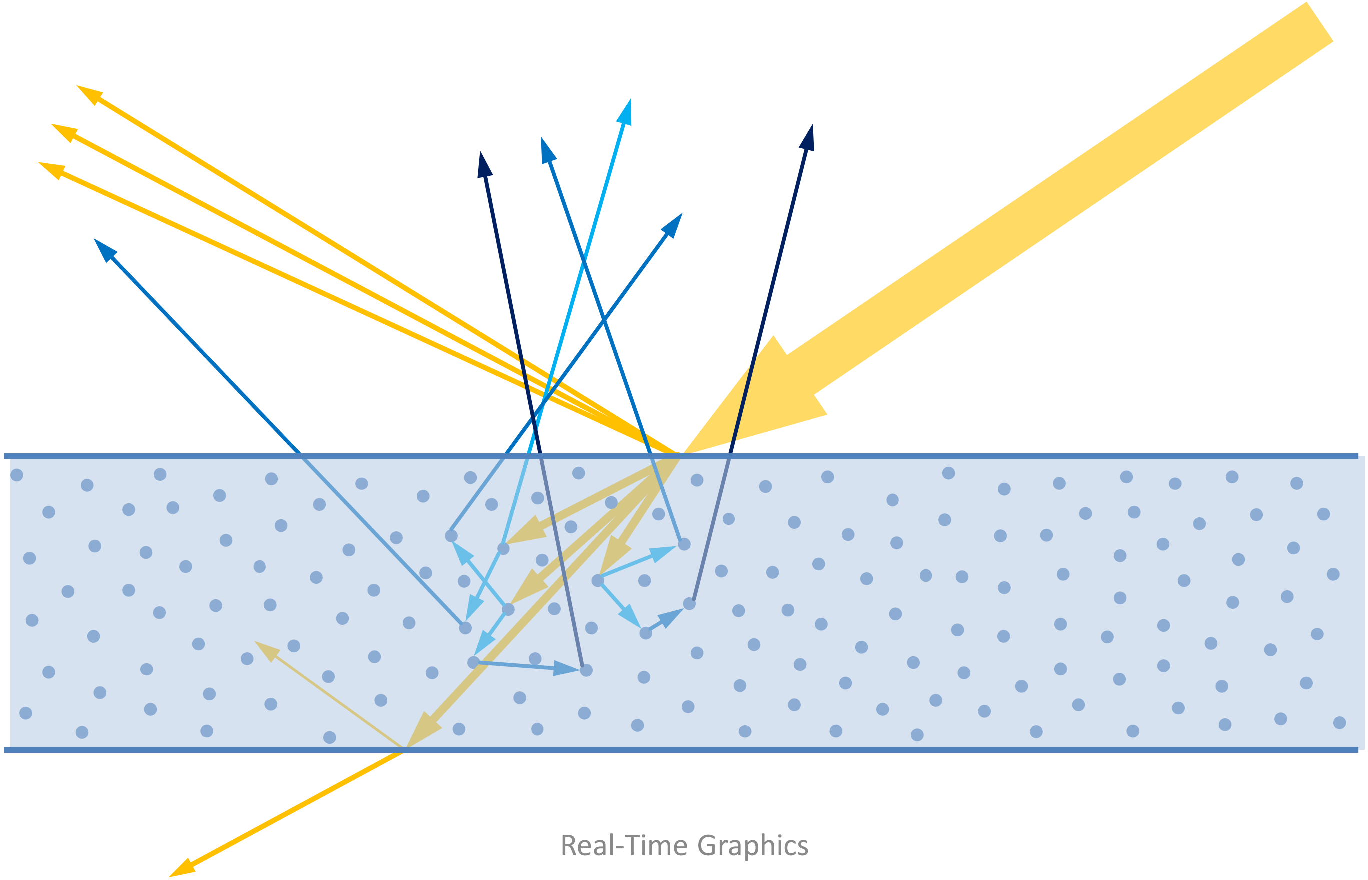


Images by Naty Hoffman

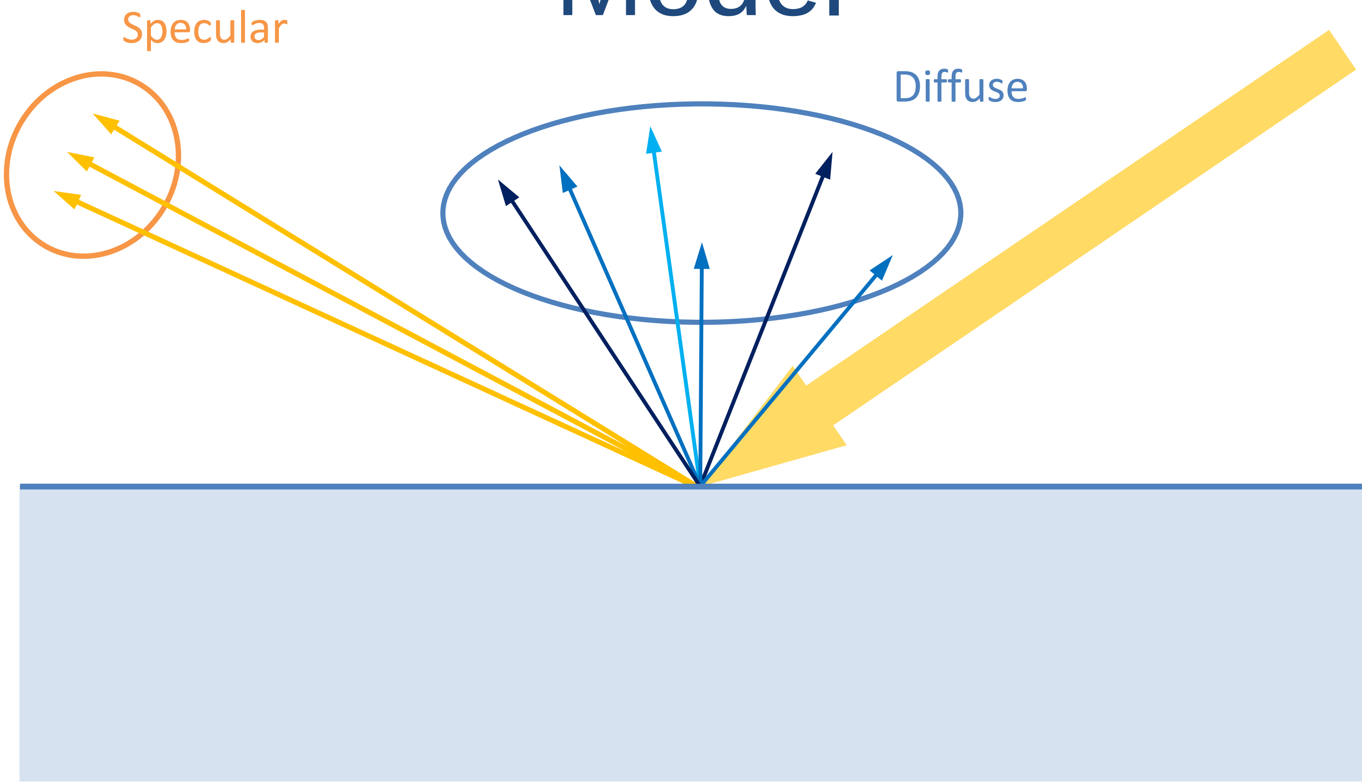
Emission

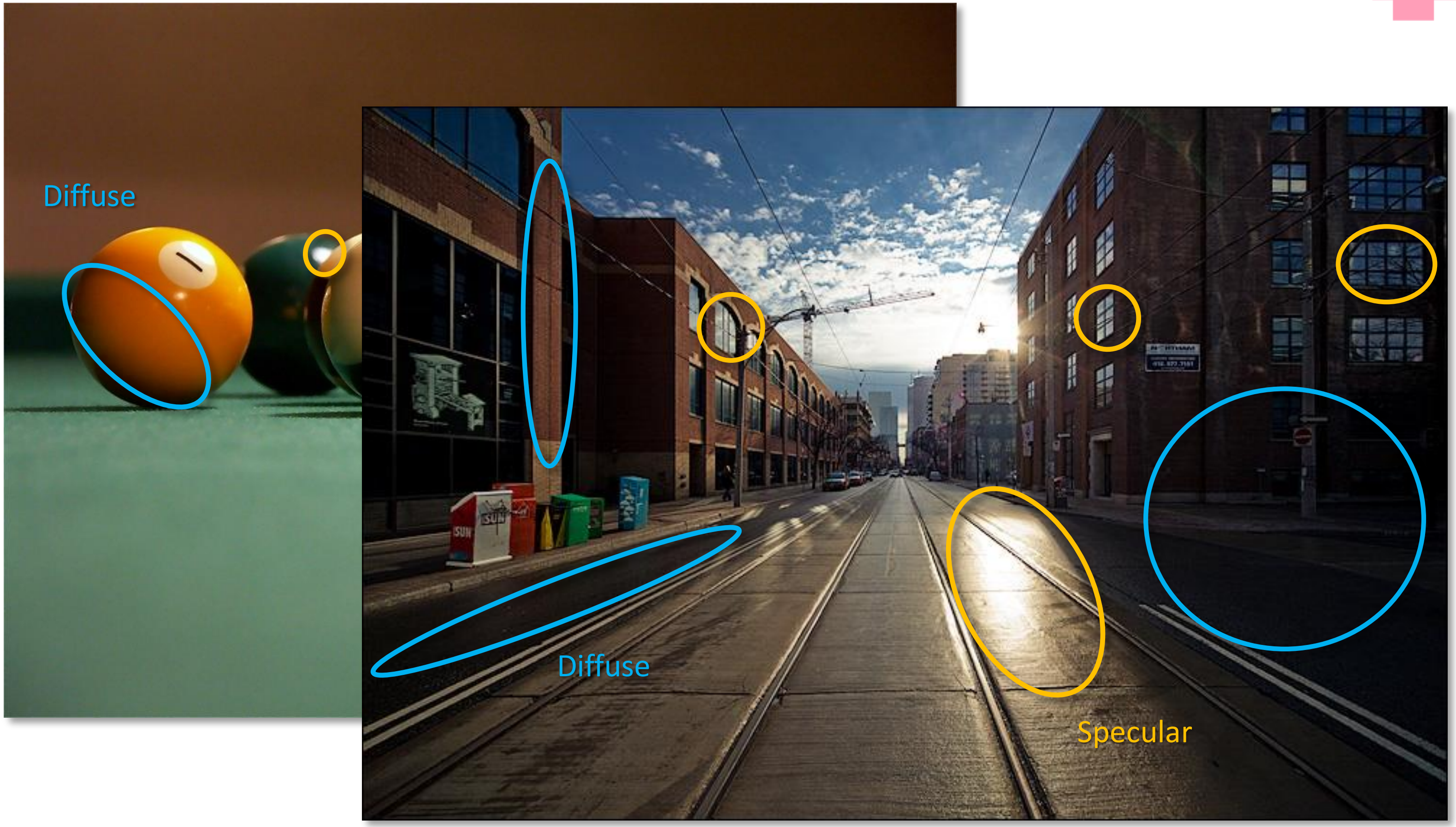


Nature



Model



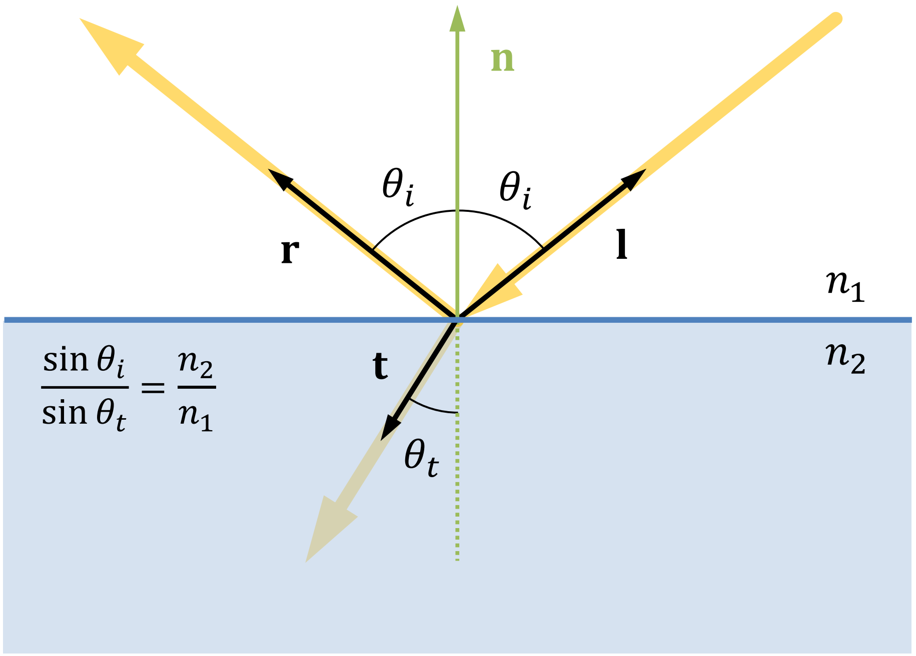

Diffuse
Diffuse
Specular

Fresnel Equations

- At the boundary between two media,
  - phase speed of light different in each medium
- part of the light wave is reflected, and
- part of the light wave is transmitted.

- So what is the exact ratio?
  - Augustin-Jean Fresnel knows.

Fresnel Equations



$$R_s = \left| \frac{n_1 \cos\theta_i - n_2 \sqrt{1 - \left(\frac{n_1}{n_2}\sin\theta_i\right)^2}}{n_1 \cos\theta_i + n_2 \sqrt{1 - \left(\frac{n_1}{n_2}\sin\theta_i\right)^2}} \right|^2$$

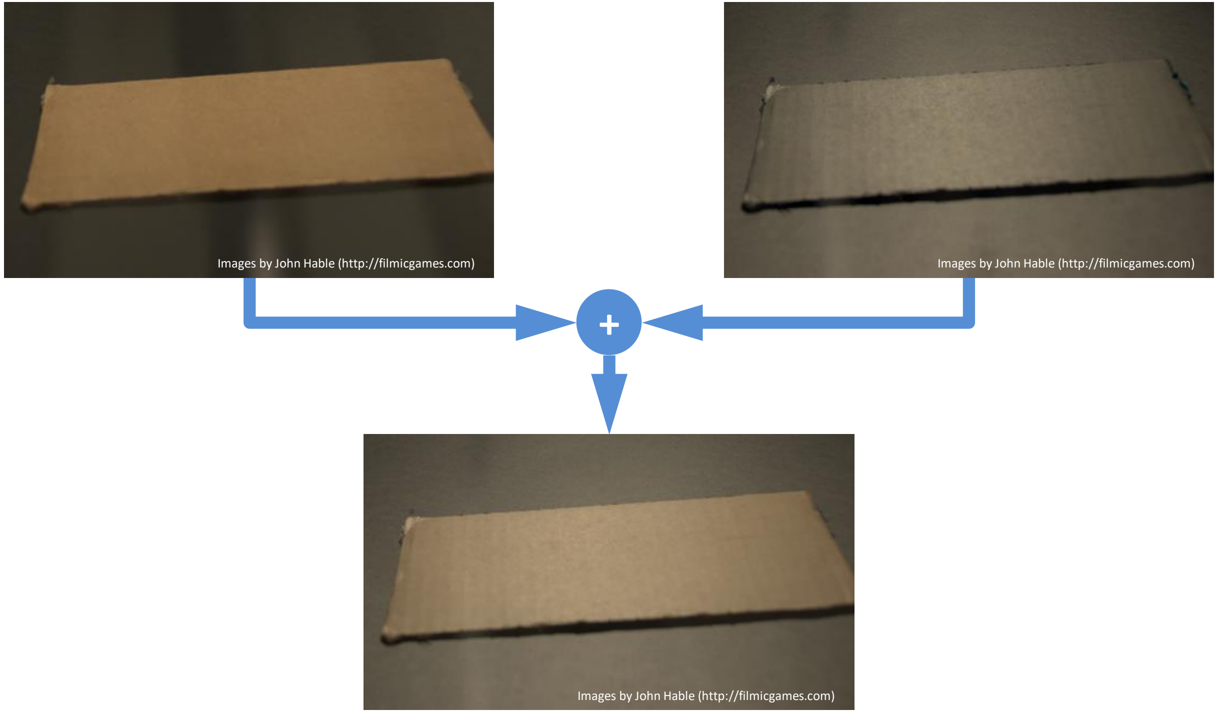
$$R_p = \left| \frac{n_1 \sqrt{1 - \left(\frac{n_1}{n_2}\sin\theta_i\right)^2} - n_2 \cos\theta_i}{n_1 \sqrt{1 - \left(\frac{n_1}{n_2}\sin\theta_i\right)^2} + n_2 \cos\theta_i} \right|^2$$

$$R = \frac{R_s + R_p}{2} \qquad T = 1 - R$$

Examples

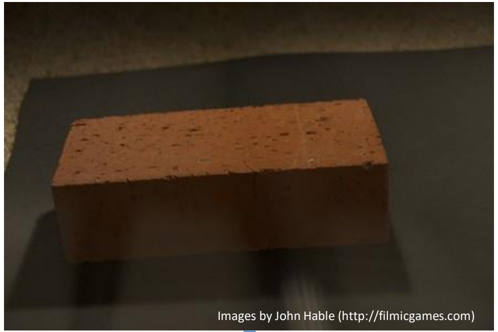
Diffuse

Specular



Examples

Diffuse



Specular

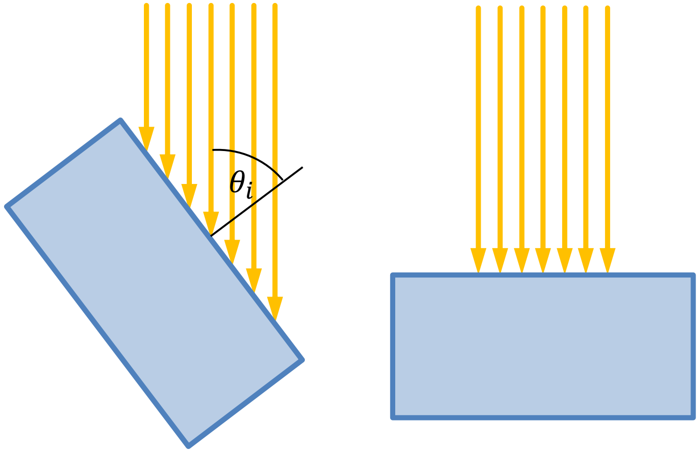
Lambert Shading

- Assumption: perfect diffuse reflector
  - Scatters light evenly in all directions

    → view independent

    - depends only on orientation of surface towards light
  - surface irradiance from light:

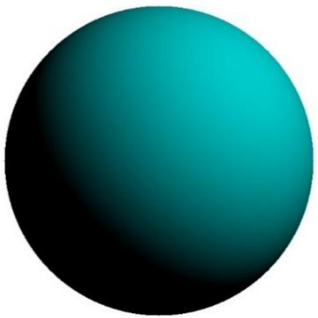
$$E_L \propto \cos\theta_i$$

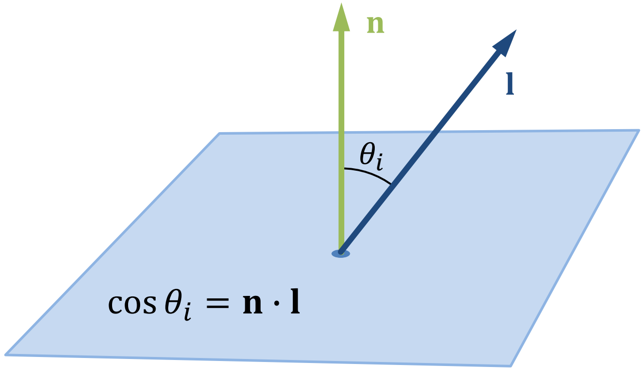


Lambert Shading

$$L_o = c_d \circ \max(\mathbf{n} \cdot \mathbf{l}, 0) \circ I_L$$

- c_d diffuse reflectance („Albedo“)
- I_L irradiance from light





Schlick's Approximation [Schlick 1994]

$$R = R_0 + (1 - R_0)(1 - \cos\theta_i)^5$$

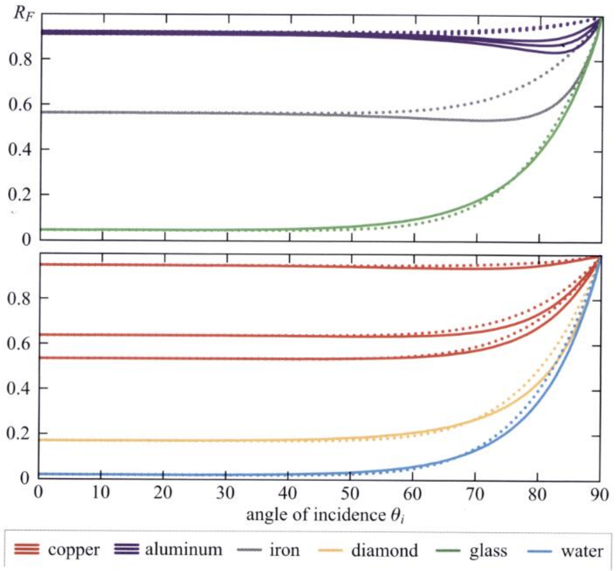
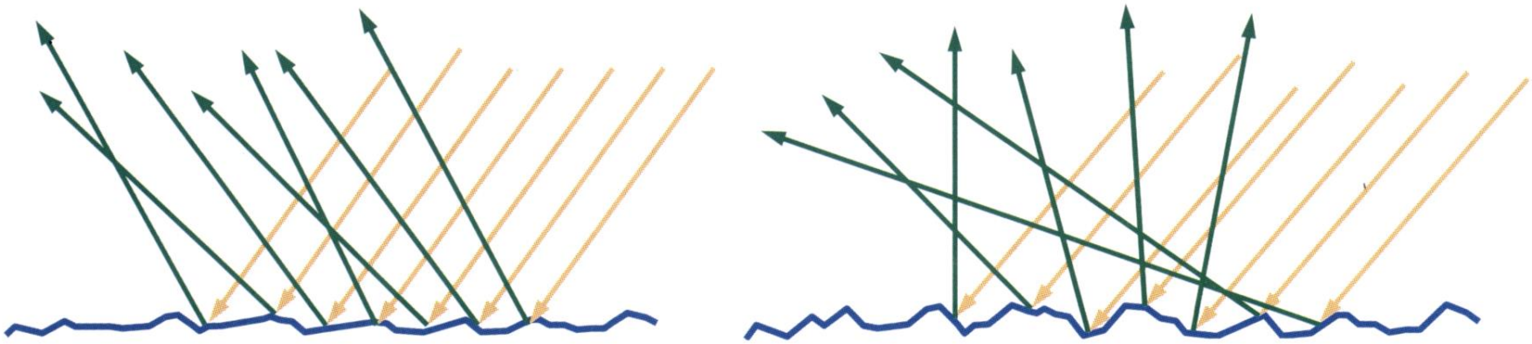
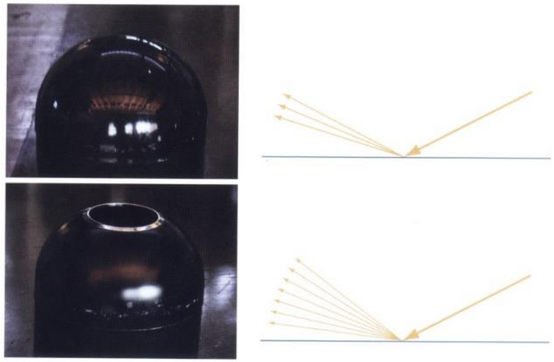


Image from [Akenine-Möller et al. 2008]

Microfacet Models

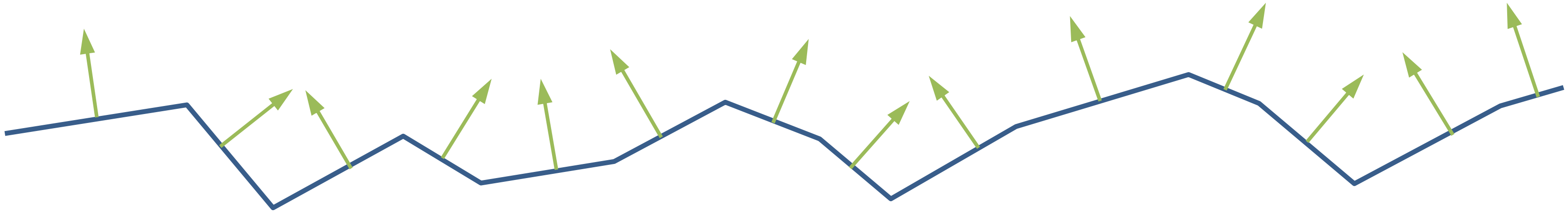




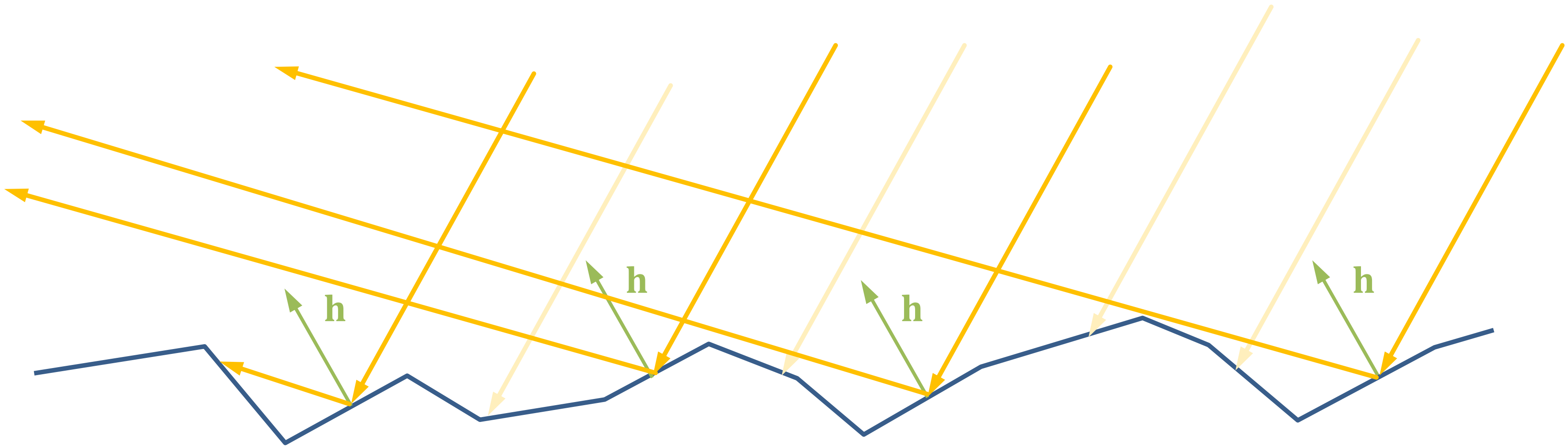
Images from [Akenine-Möller et al. 2008]

Microfacet Models

- Think of surface as made up of tiny mirrors
  - lots of tiny mirrors
- Model distribution of mirrors
- Only mirrors oriented halfway between light and view direction reflect light into camera

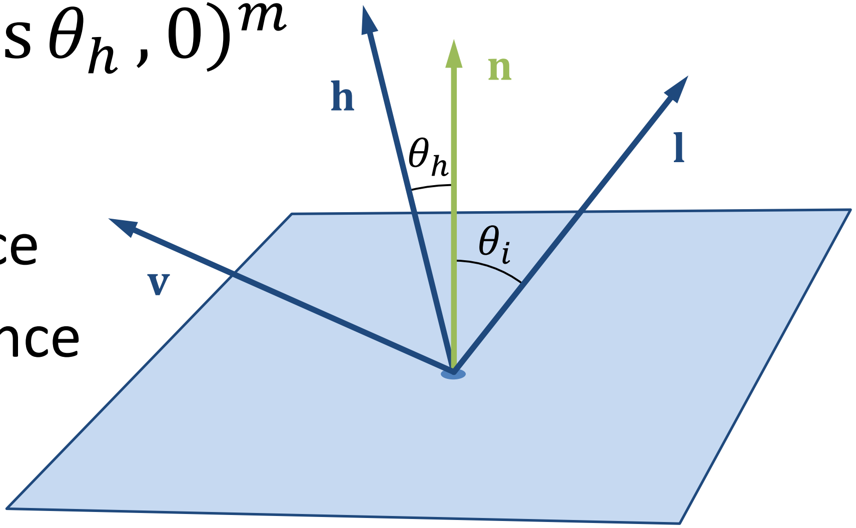


Microfacet Models



Blinn-Phong Model [Blinn 1977]

- $f_r = \frac{c_d}{\pi} + \frac{m+8}{8\pi} c_s \max(\cos\theta_h, 0)^m$
  - c_d diffuse reflectance
  - c_s specular reflectance
  - m specular power

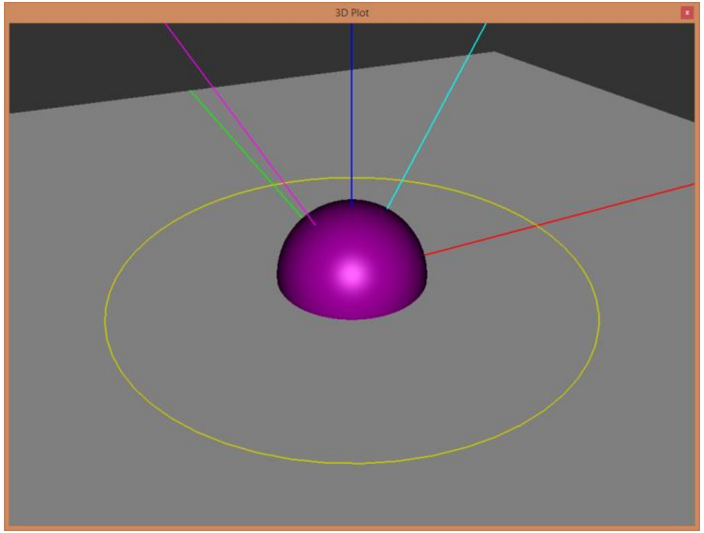


$$\mathbf{h} = \frac{\mathbf{v} + \mathbf{l}}{\|\mathbf{v} + \mathbf{l}\|}$$

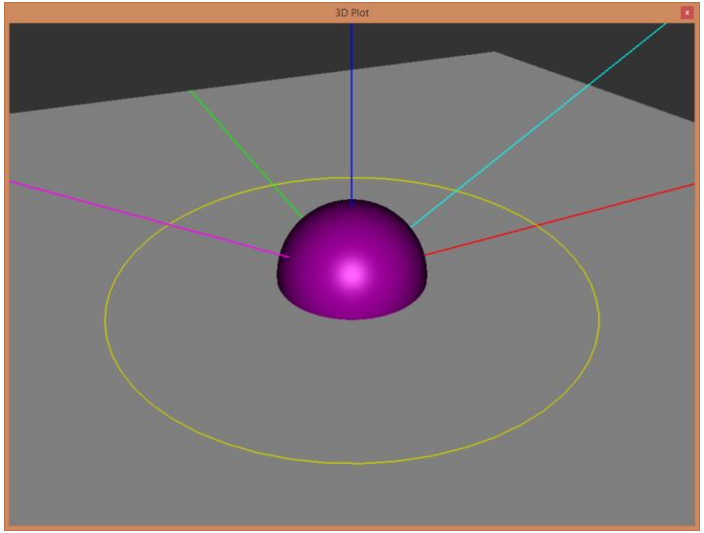
*all vectors assumed to be normalized

Lambert BRDF

$\theta_i = 30°$

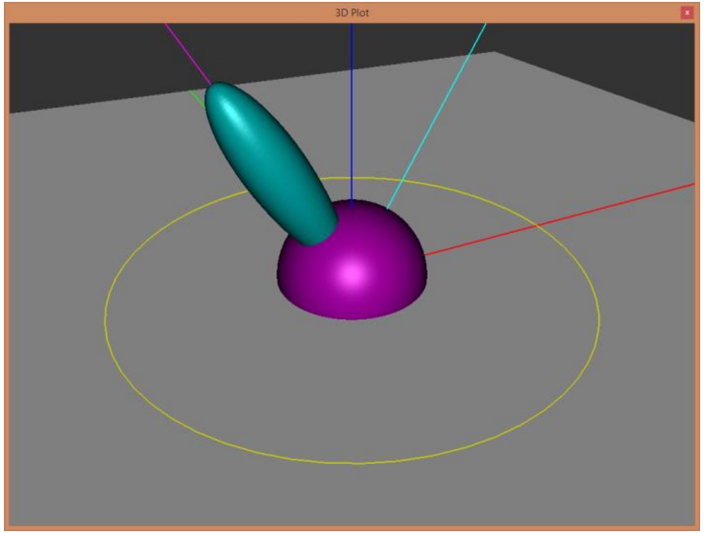


$\theta_i = 60°$

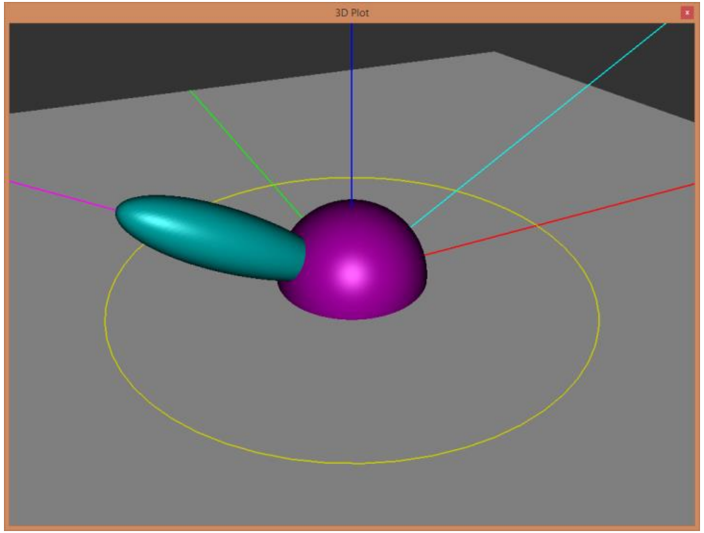


Phong BRDF

$\theta_i = 30°$

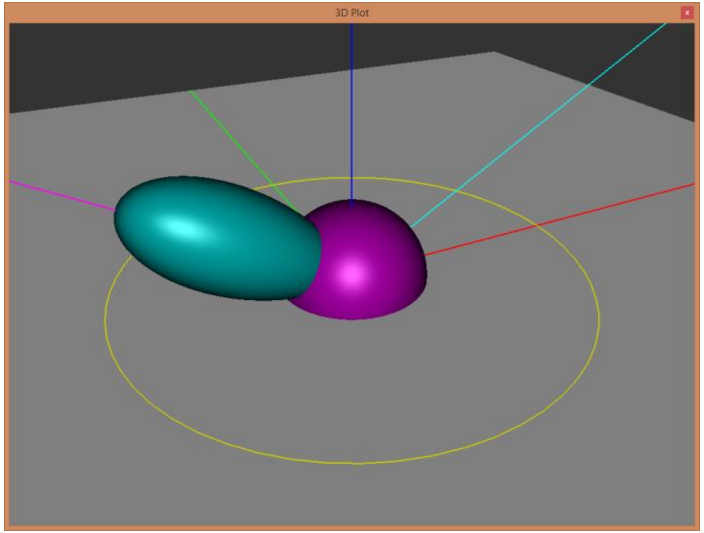


$\theta_i = 60°$

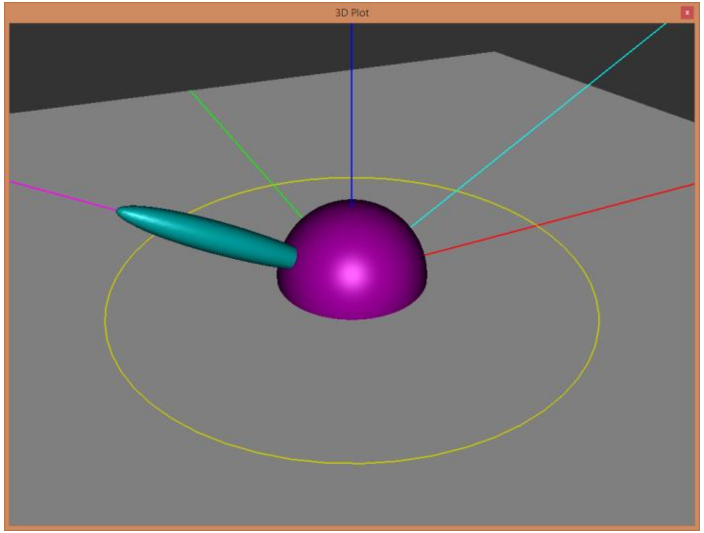


Phong BRDF

$m = 8$

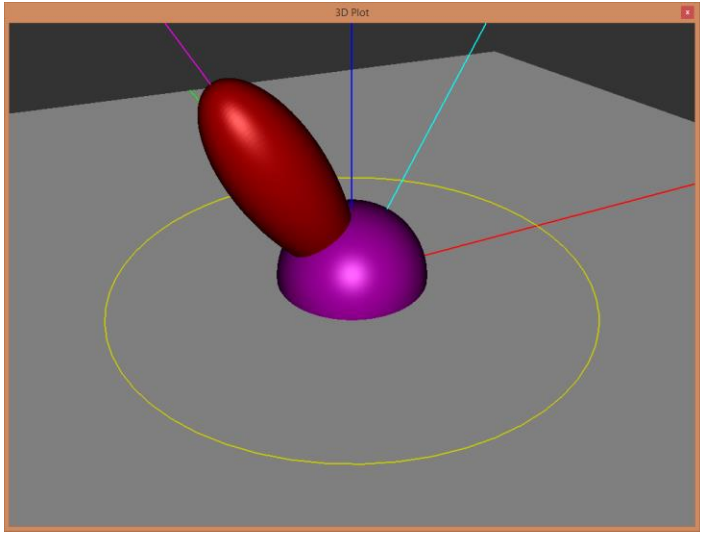


$m = 100$

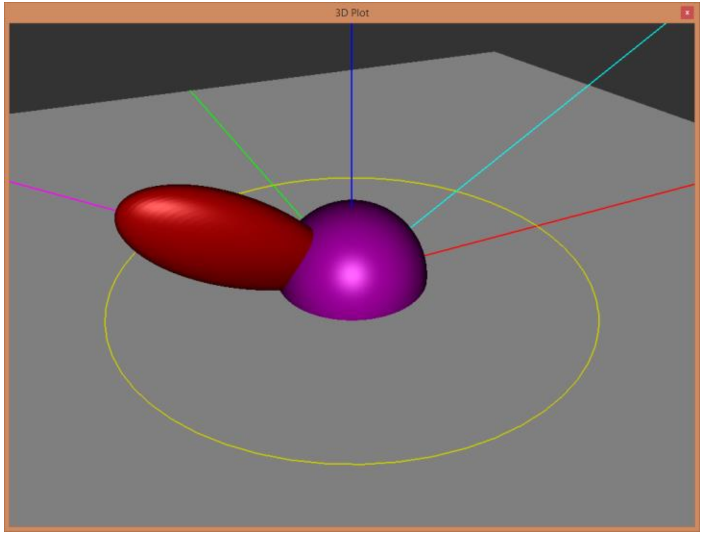


Blinn-Phong BRDF

$\theta_i = 30°$

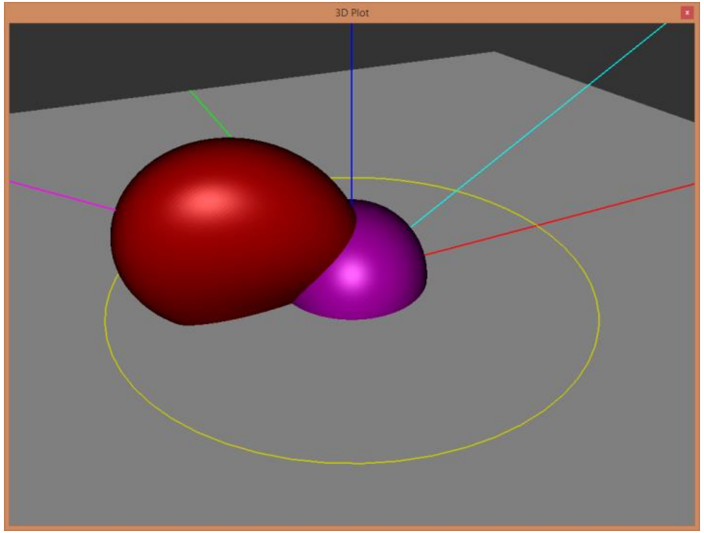


$\theta_i = 60°$

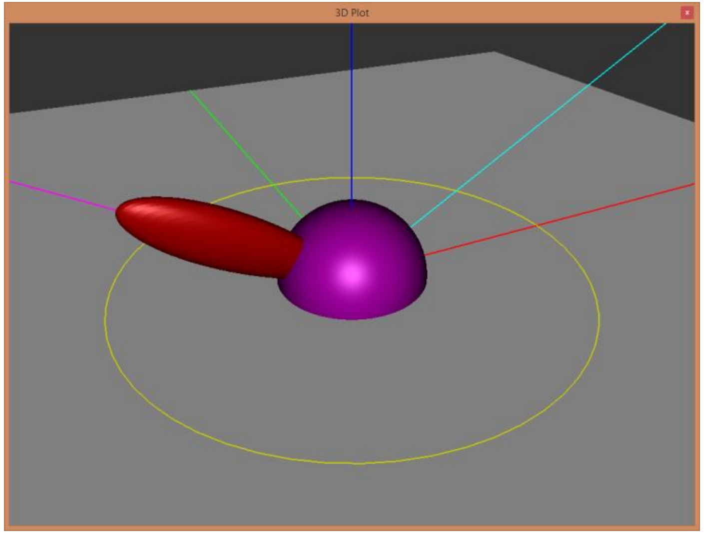


Blinn-Phong BRDF

$m = 8$



$m = 100$



Microfacet Models

microfacet distribution

fresnel reflectance

geometric attenuation

$$f_r(\mathbf{l}, \mathbf{v}) = \frac{D(\mathbf{h})F(\mathbf{l}, \mathbf{v})G(\mathbf{l}, \mathbf{v})}{4(\mathbf{n} \cdot \mathbf{v})(\mathbf{n} \cdot \mathbf{l})}$$

Microfacet Distribution

- fraction of microfacets oriented in direction $\mathbf{h}$
- e. g. Beckmann Distribution [Beckmann, Spizzichino 1963]:

$$D(\mathbf{h}) = \frac{1}{4m^2(\mathbf{n}\cdot\mathbf{h})^4}\exp\left(\frac{(\mathbf{n}\cdot\mathbf{h})^2 - 1}{m^2(\mathbf{n}\cdot\mathbf{h})^2}\right)$$

m root mean squared slope of microfacets (corresponds to roughness)

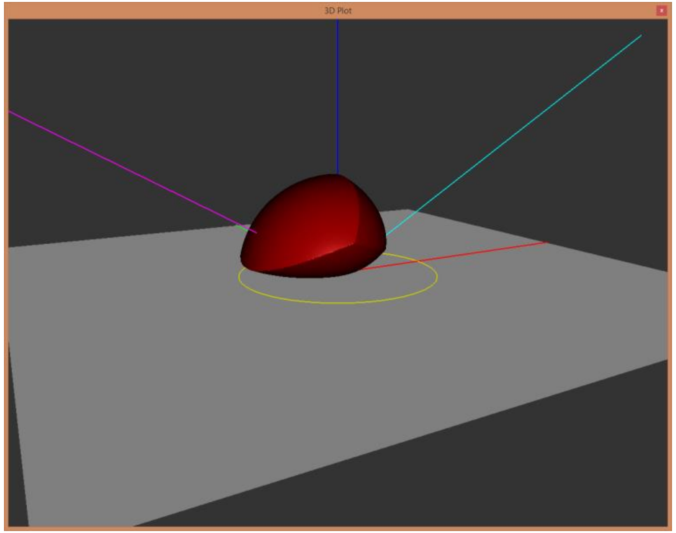
Geometric Attenuation

- accounts for
  - shadowing
    - light blocked from reaching microfacet by other microfacets
  - masking
    - reflected light blocked from reaching camera by other microfacets
- e. g. Torrance-Sparrow model [Torrance, Sparrow 1967]:

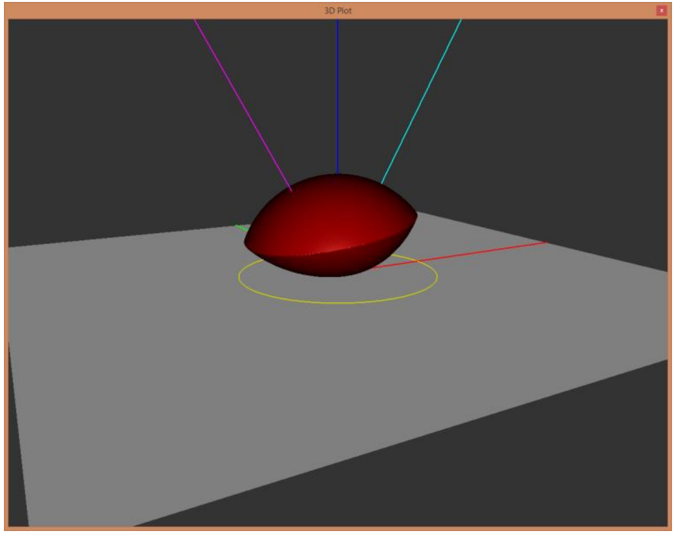
$$G(\mathbf{l}, \mathbf{v}) = \min\left(1, \frac{2\cos\theta_h \cos\theta_o}{\cos\alpha_h}, \frac{2\cos\theta_h \cos\theta_i}{\cos\alpha_h}\right)$$

Geometric Attenuation

$\theta_i = 60°$

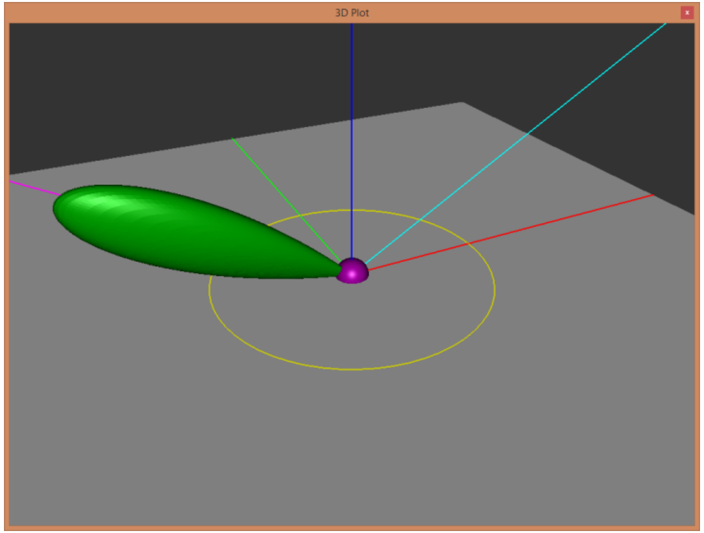


$\theta_i = 30°$

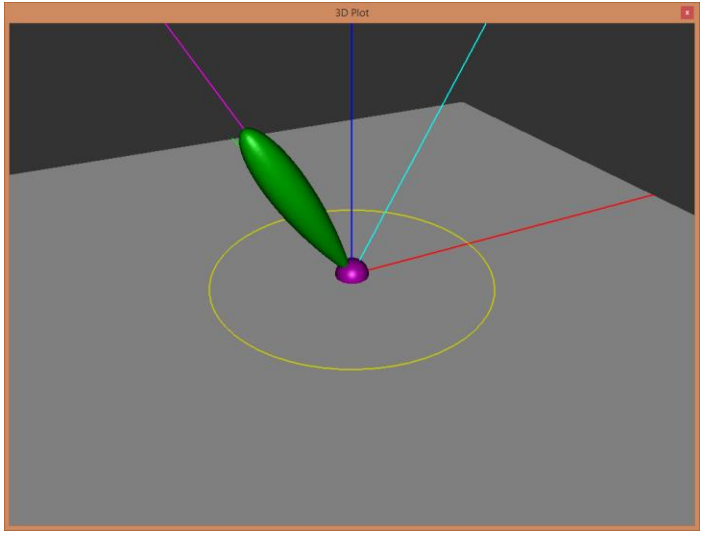


Cook-Torrance BRDF

$\theta_i = 60°,\ m = 0.069$

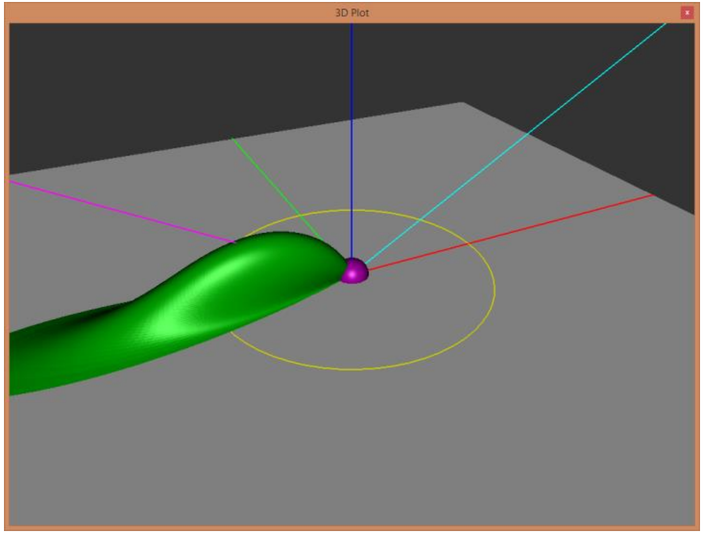


$\theta_i = 30°,\ m = 0.069$

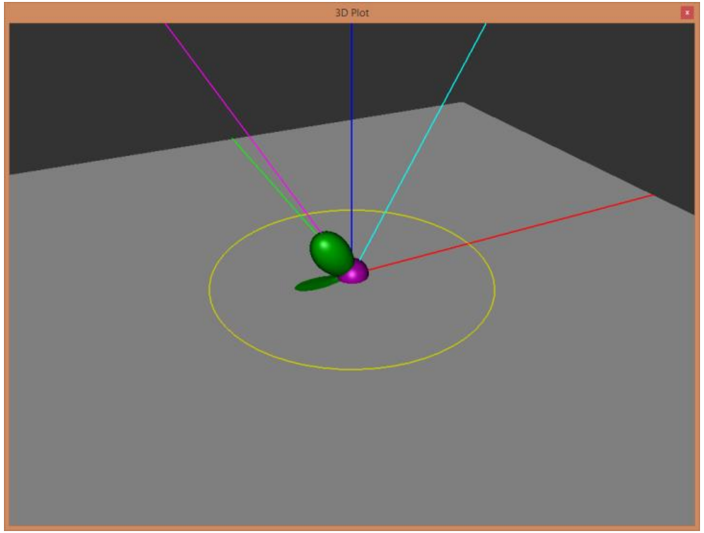


Cook-Torrance BRDF

$\theta_i = 60°,\ m = 0.24$

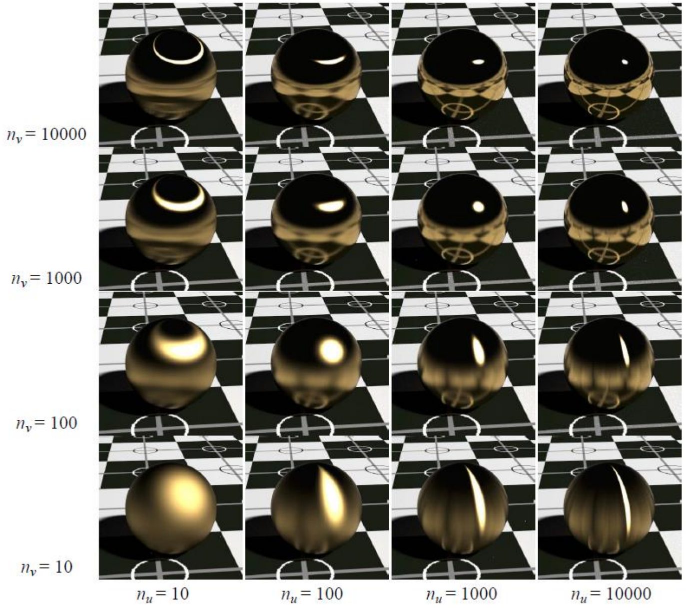


$\theta_i = 30°,\ m = 0.24$



[Ashikhmin, Shirley 2000]

- anisotropic model

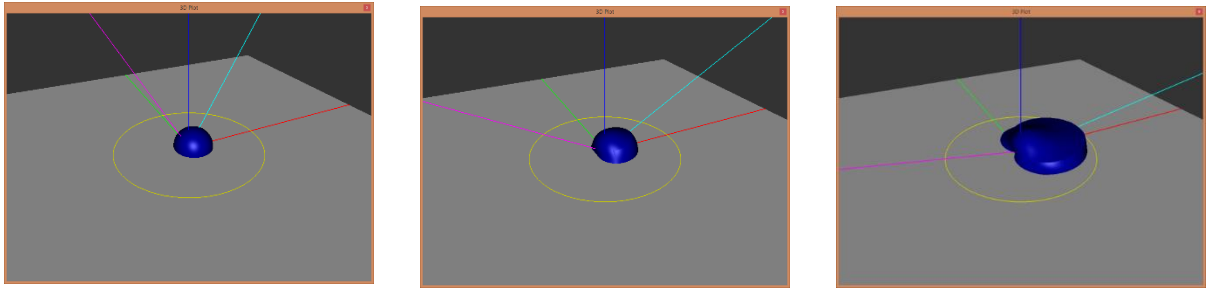


Images from [Ashikhmin, Shirley 2000]

[Oren, Nayar 1994]

- models retro reflection

$\theta_i = 30°$ $\theta_i = 60°$ $\theta_i = 80°$



Photograph Lambertian Oren-Nayar

Images from [Oren, Nayar 1994]

Implementation in GLSL

- BRDF:

outgoing radiance

$$f(\mathbf{l}, \mathbf{v}) = \frac{dL_o(\mathbf{v})}{dE_i(\mathbf{l})}$$

- irradiance from light source:

light source intensity

irradiance incident on surface

incoming irradiance

$$E_L = \frac{I_L}{r^2}$$

squared distance to light source

- radiance towards viewer:

$$L_o(\mathbf{v}) = \sum_{k=1}^{n} f(\mathbf{l}_k, \mathbf{v}) \circ E_{L_k} \cos\theta_{i_k}$$

sum up contributions of each light source

Example: Blinn-Phong Vertex Shader

```glsl
#version 330

uniform mat4x4 PV;  // view projection matrix

layout(location = 0) in vec3 vertex_position;
layout(location = 1) in vec3 vertex_normal;

out vec3 p;
out vec3 normal;

void main()
{
    gl_Position = PV * vec4(vertex_position, 1.0f);
    p = vertex_position;
    normal = vertex_normal;
}
```

Example: Phong Fragment Shader

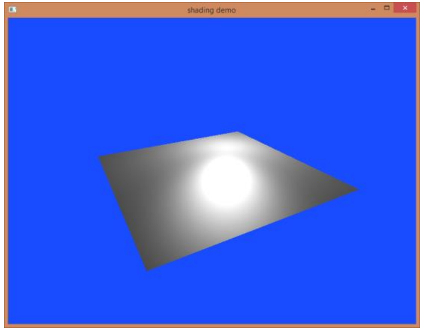
```
#version 330

uniform vec3 camera_position;
uniform vec3 light_direction;

uniform vec3 B_L;

uniform vec3 c_d;
uniform vec3 c_s;
uniform float m;

in vec3 p;
in vec3 normal;

layout(location = 0) out vec4 fragment_color;

void main()
{
	vec3 n = normalize(normal);  // renormalize interpolated normal
	vec3 v = normalize(camera_position - p);
	vec3 l =-light_direction;
	vec3 r = 2.0f * n * dot(n, l) - l;

	float lambert = max(dot(n, l), 0.0f);
	float specular = pow(max(dot(v, r), 0.0f), m);

	fragment_color.rgb = (c_d *lambert + c_s * specular) * B_L;
	fragment_color.a = 1.0f;
}
```

Example: Blinn-Phong Fragment Shader

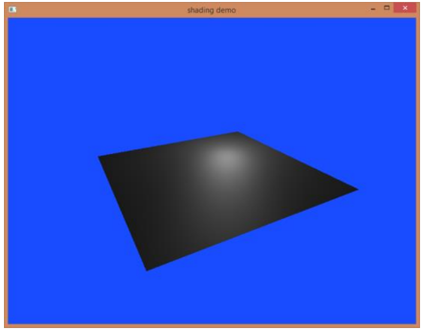
```
#version 330

const float pi = 3.14159265358979f;

uniform vec3 camera_position;
uniform vec3 light_direction;

uniform vec3 I_L;

uniform vec3 c_d;
uniform vec3 c_s;
uniform float m;

in vec3 p;
in vec3 normal;

layout(location = 0) out vec4 fragment_color;

void main()
{
        vec3 n = normalize(normal);  // renormalize interpolated normal
        vec3 v = normalize(camera_position – p);
        vec3 l =-light_direction;
        vec3 h = normalize(v + l);

        float lambert = max(dot(n, l), 0.0f);
        float specular = (m + 8) / (8 * pi) * pow(max(dot(n, h), 0.0f), m);

        fragment_color.rgb = (c_d / pi + c_s * specular) * lambert * I_L;
        fragment_color.a = 1.0f;
}
```

Results

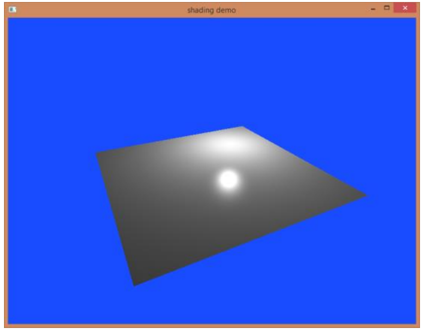
Phong, $m = 8$



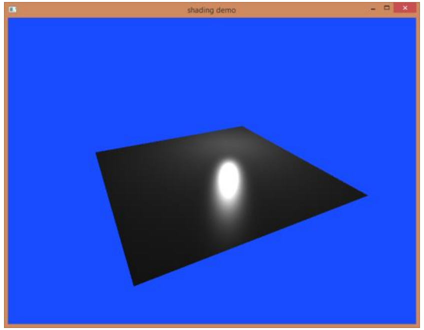
Blinn-Phong, $m = 8$



Phong, $m = 100$



Blinn-Phong, $m = 100$



References

[Akenine-Möller et al. 2008] Tomas Akenine-Möller, Eric Haines, Naty Hoffman (2008). Real-Time Rendering. 3rd ed. AK Peters.

[Lengyel 2004] Eric Lengyel (2004). Mathematics for 3D Game Programming & Computer Graphics. 2nd ed. Charles River Media.

[Phong 1975] Bui Tuong Phong (1975). “Illumination for computer generated pictures”. *Communications of the ACM*, vol. 18, no. 5.

[Blinn 1977] James F. Blinn (1977). “Models of light reflection for computer synthesized pictures”. Proc. 4th annual conference on computer graphics and interactive techniques: 192–198.

[Schlick 1994] Christophe Schlick. (1994). “An Inexpensive BRDF Model for Physically-based Rendering”. Computer Graphics Forum, vol. 13, no. 3: 233–246.

[Torrance, Sparrow 1967] K. E. Torrance, E. M. Sparrow (1967). “Theory for Off-specular Reflection From Roughened Surfaces”. *Journal of the Optical Society of America*, vol. 57, no. 9: 32–41.

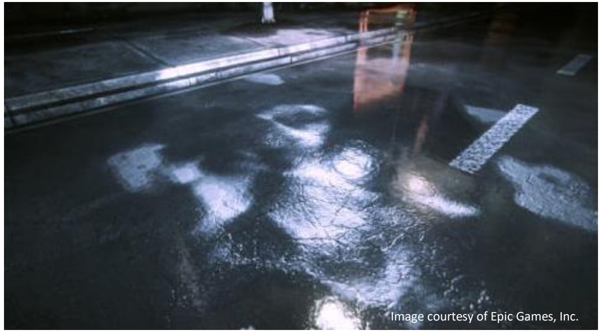
[Beckmann, Spizzichino 1963] Petr Beckmann, André Spizzichino (1963). The scattering of electromagnetic waves from rough surfaces, Pergamon Press.

[Ashikhmin, Shirley 2000] Michael Ashikhmin, Peter Shirley (2000). “An Anisotropic Phong BRDF Model”. Journal of Graphics Tools, vol. 7, no. 4: 61–68.

[Oren, Nayar 1994] Michael Oren, Shree K. Nayar (1994). “Generalization of Lambert's reflectance model”. Proc. 21st annual conference on computer graphics and interactive techniques: 239–246.

Energy Conservation

Specular tweaked for glossy



Specular tweaked for dull

Energy conserving