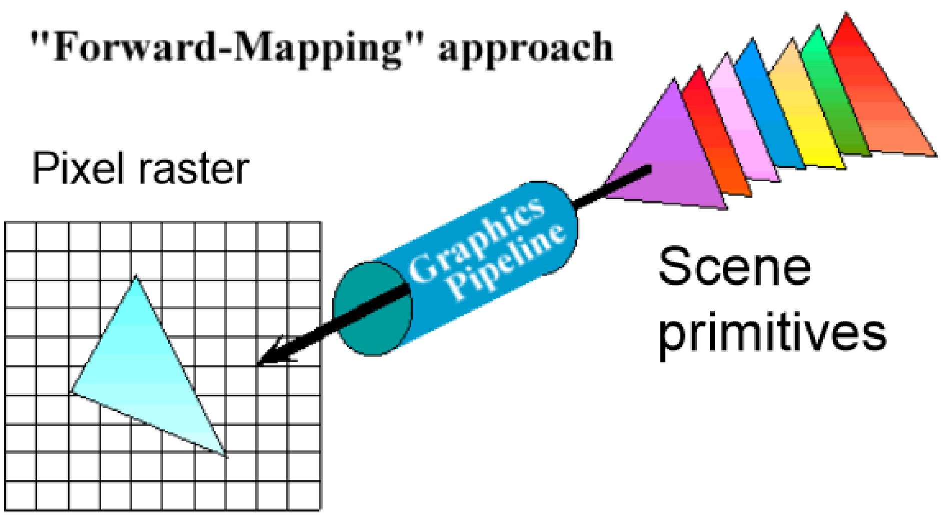
Advanced image synthesis

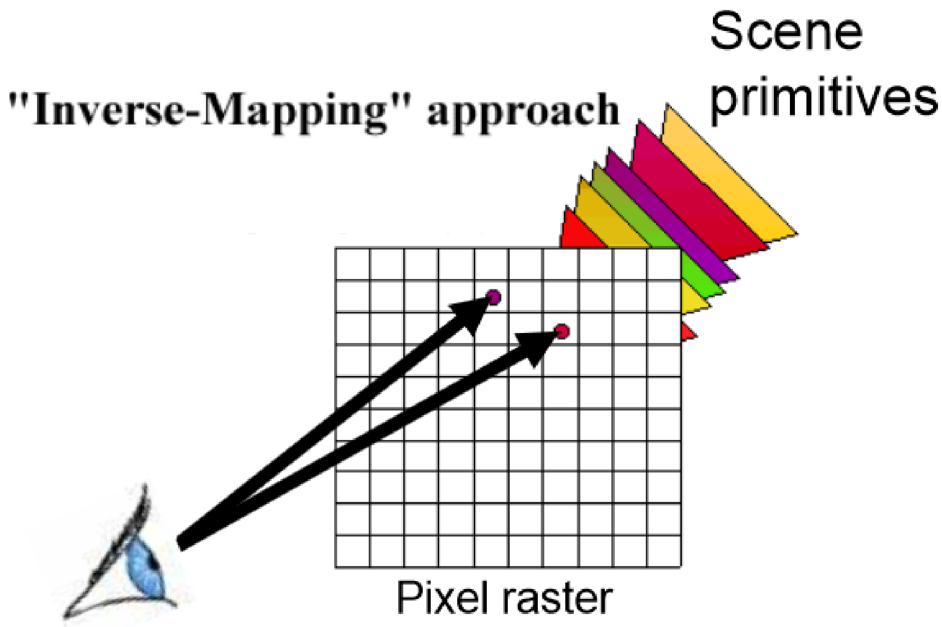
Romain Vergne – 2014/2015

Rasterization VS ray-casting

- For each triangle
  - Project triangle to image plane
  - For each pixel
    - Check pixel in triangle
    - Resolve visibility with z-buffer

- For each pixel
  - Compute pixel ray
  - For each triangle
    - Check ray-triangle intersection
    - Get closest intersection



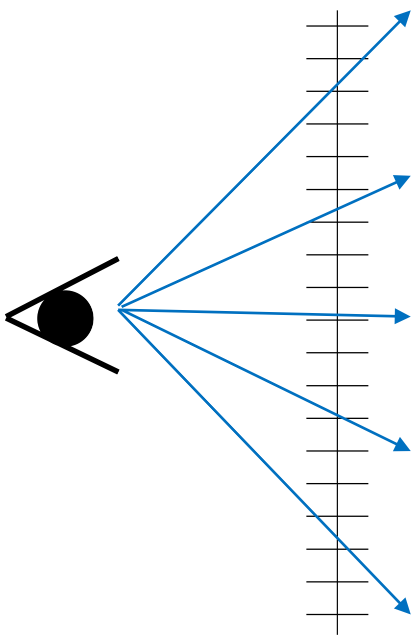


Eye ray and camera

- Persective

r = (x*u,aspect*y*v,D*w), normalized

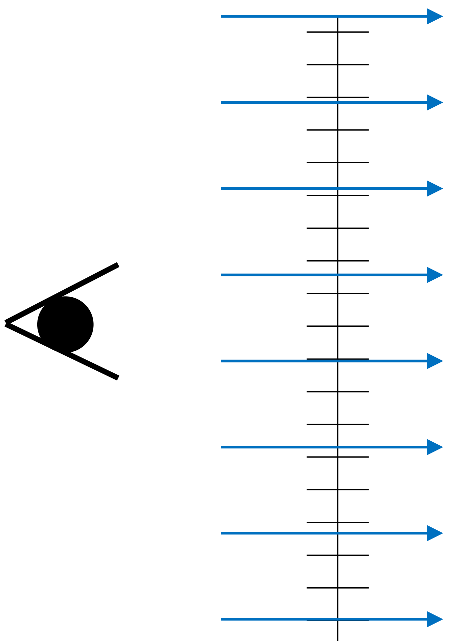
P(t) = e + t*r



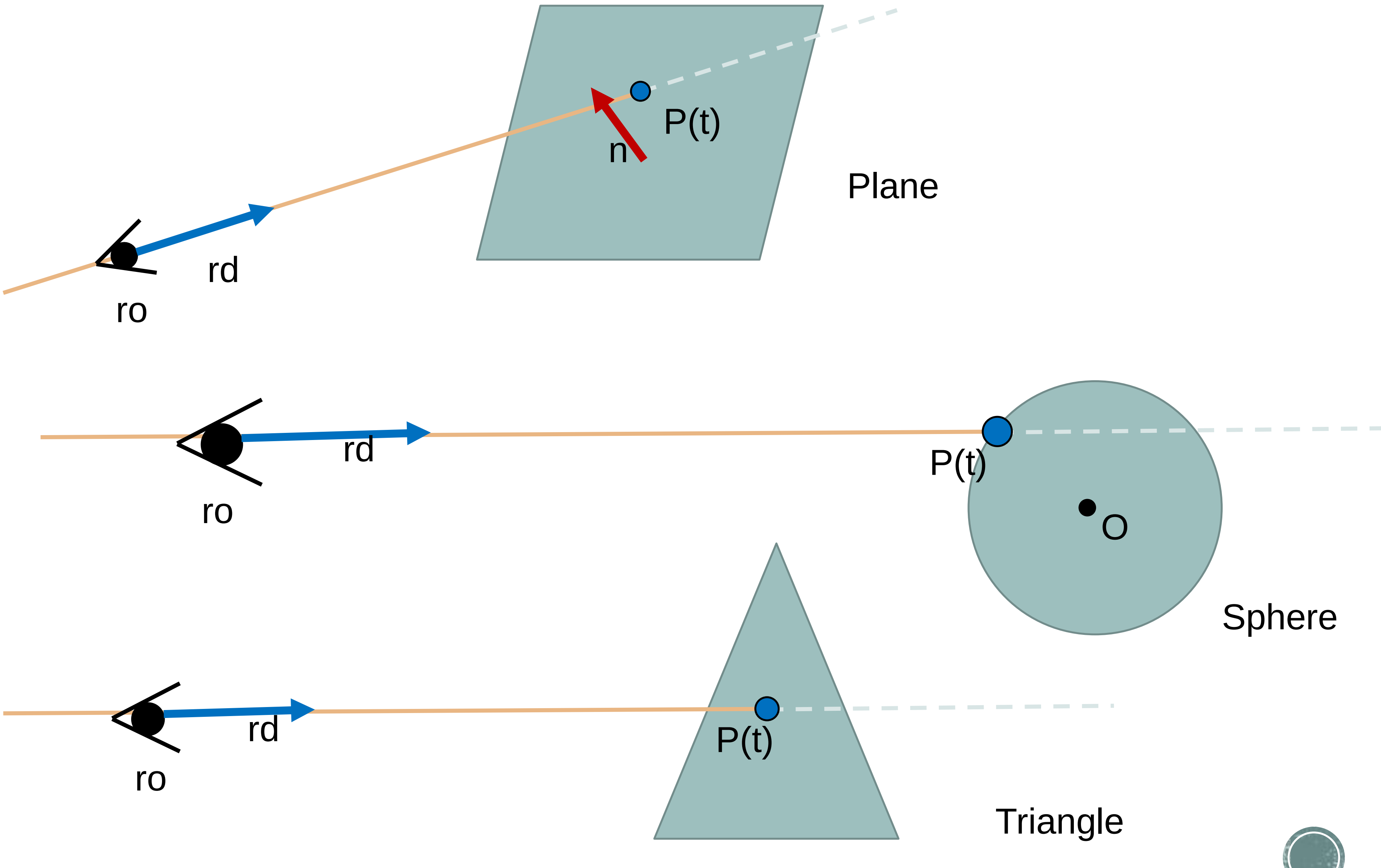
- Orthographic

P(t) = o + t*w

o = e + x*size*u + y*size*v



Ray-plane intersection

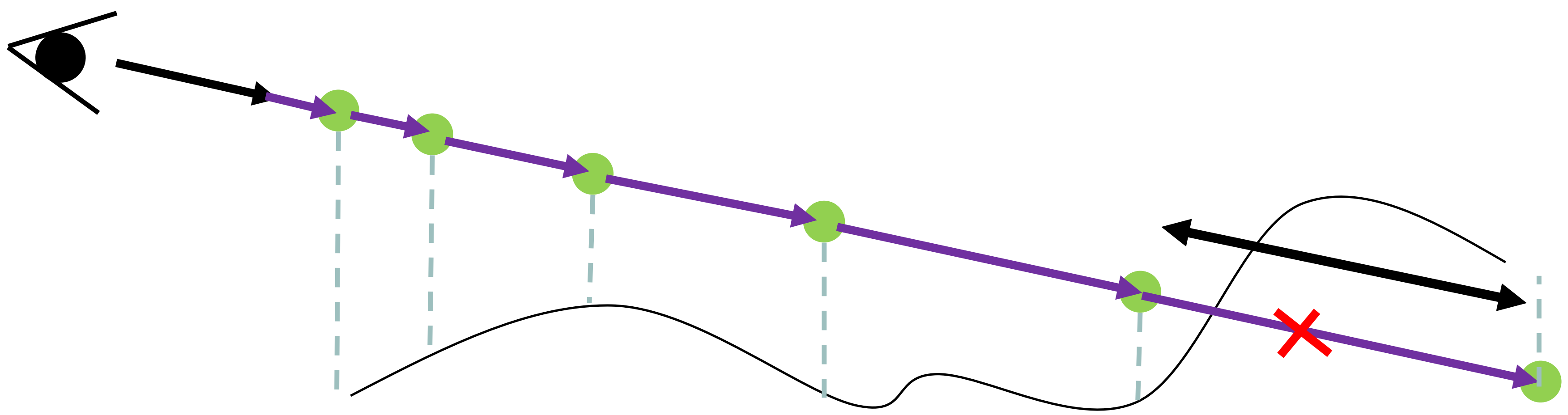


Ray marching height-fields

$P(t) = r_o + r_d * t$

Optimizations:

- Interpolate between the 2 last positions
- Increase deltaT with distance from eye
- See: http://www.iquilezles.org



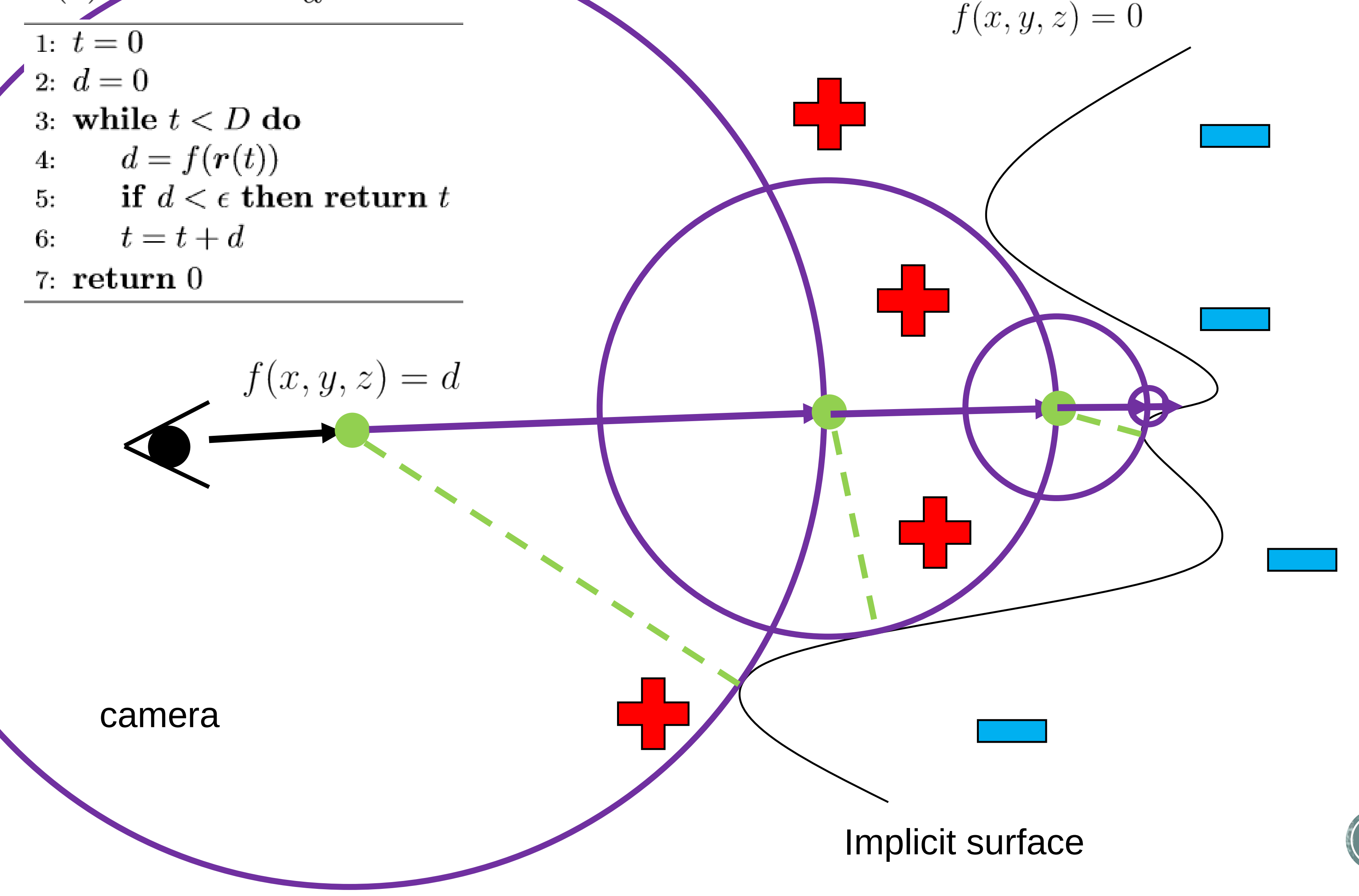
camera

Height field $f(x, z) = y$

Ray marching implicit surfaces

$$P(t) = r_o + r_d * t$$

```
t = 0
d = 0
while t < D do
    d = f(r(t))
    if d < ε then return t
    t = t + d
return 0
```

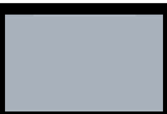


Common implicit surfaces

Sphere - signed

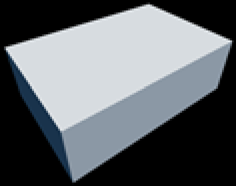
```
float sdSphere( vec3 p, float s )
{
  return length(p)-s;
}
```

Plane - signed



```
float sdPlane( vec3 p, vec4 n )
{
  // n must be normalized
  return dot(p,n.xyz) + n.w;
}
```

Box - signed



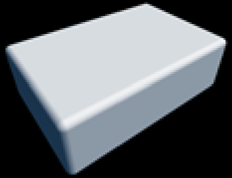
```
float sdBox( vec3 p, vec3 b )
{
  vec3 d = abs(p) - b;
  return min(max(d.x,max(d.y,d.z)),0.0) +
         length(max(d,0.0));
}
```

Torus - signed

```
float sdTorus( vec3 p, vec2 t )
{
  vec2 q = vec2(length(p.xz)-t.x,p.y);
  return length(q)-t.y;
}
```

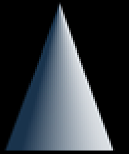
Round Box - unsigned

```
float udRoundBox( vec3 p, vec3 b, float r )
{
  return length(max(abs(p)-b,0.0))-r;
}
```



Cone - signed

```
float sdCone( vec3 p, vec2 c )
{
    // c must be normalized
    float q = length(p.xy);
    return dot(c,vec2(q,p.z));
}
```

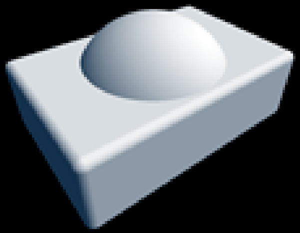


Complete list: http://www.iquilezles.org/www/articles/distfunctions/distfunctions.htm

Distance operations

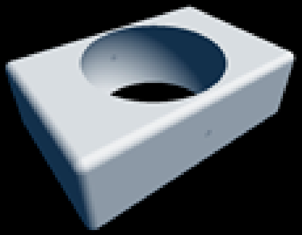
Union

```
float opU( float d1, float d2 )
{
    return min(d1,d2);
}
```



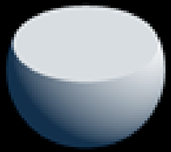
Substraction

```
float opS( float d1, float d2 )
{
    return max(-d1,d2);
}
```



Intersection

```
float opI( float d1, float d2 )
{
    return max(d1,d2);
}
```

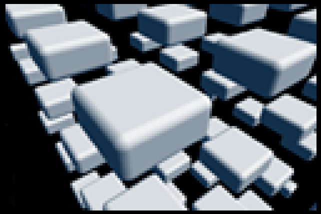


Complete list: http://www.iquilezles.org/www/articles/distfunctions/distfunctions.htm

Domain operations

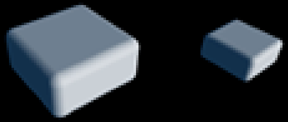
Repetition

```
float opRep( vec3 p, vec3 c )
{
    vec3 q = mod(p,c)-0.5*c;
    return primitve( q );
}
```



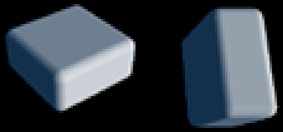
Scale

```
float opScale( vec3 p, float s )
{
    return primitive(p/s)*s;
}
```



Rotation/Translation

```
vec3 opTx( vec3 p, mat4 m )
{
    vec3 q = invert(m)*p;
    return primitive(q);
}
```

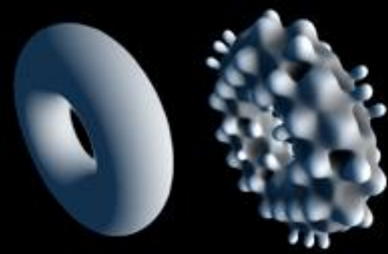


Complete list: http://www.iquilezles.org/www/articles/distfunctions/distfunctions.htm

Distance deformations

Displacement

```
float opDisplace( vec3 p )
{
    float d1 = primitive(p);
    float d2 = displacement(p);
    return d1+d2;
}
```

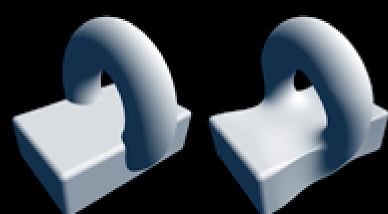


Twist

```
float opTwist( vec3 p )
{
    float c = cos(20.0*p.y);
    float s = sin(20.0*p.y);
    mat2  m = mat2(c,-s,s,c);
    vec3  q = vec3(m*p.xz,p.y);
    return primitive(q);
}
```

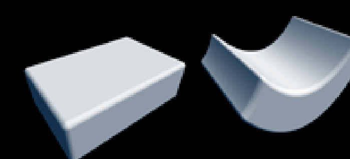
Blend

```
float opBlend( vec3 p )
{
    float d1 = primitiveA(p);
    float d2 = primitiveB(p);
    return smin( d1, d2 );
}
```



Cheap Bend

```
float opCheapBend( vec3 p )
{
    float c = cos(20.0*p.y);
    float s = sin(20.0*p.y);
    mat2  m = mat2(c,-s,s,c);
    vec3  q = vec3(m*p.xy,p.z);
    return primitive(q);
}
```

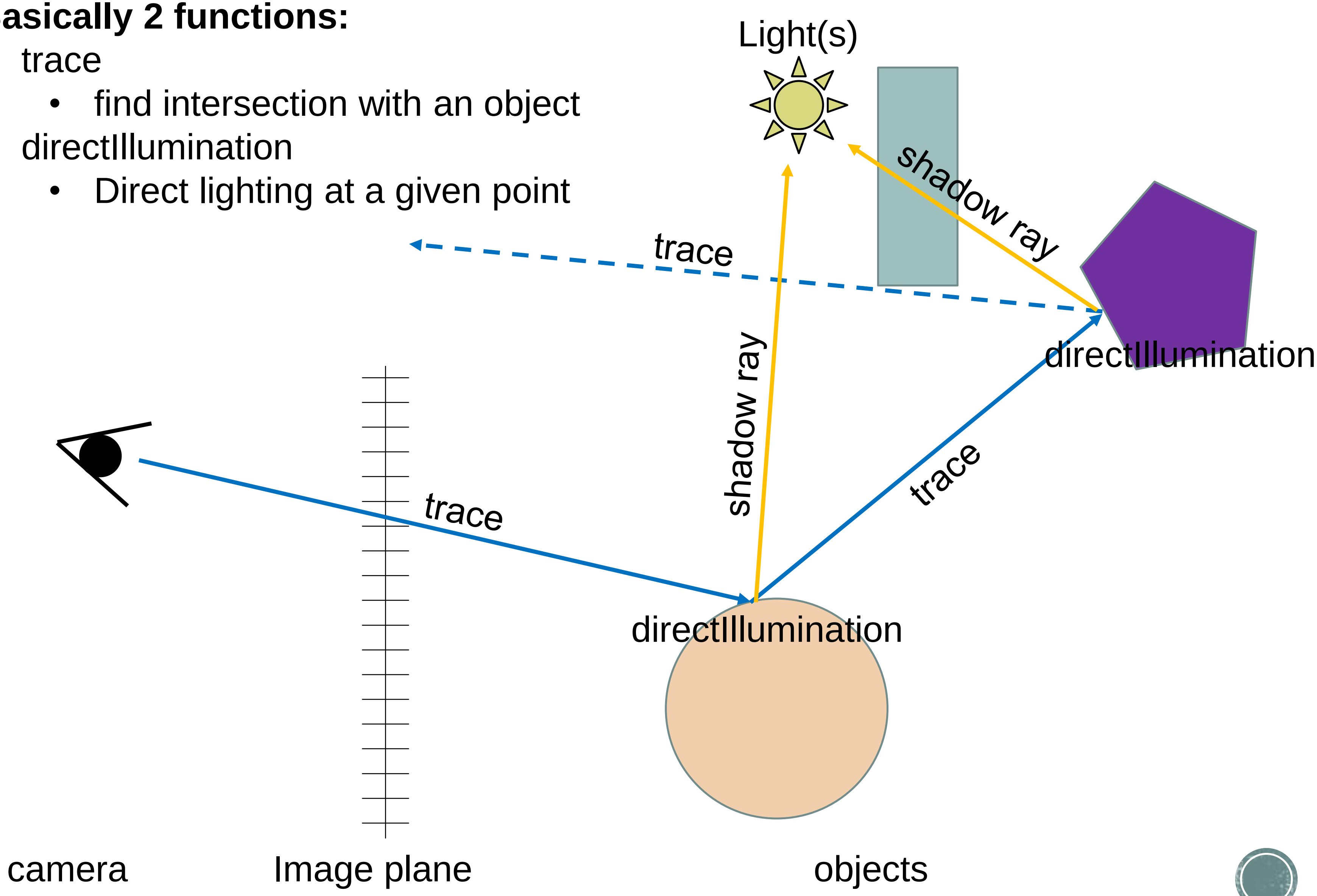


Complete list: http://www.iquilezles.org/www/articles/distfunctions/distfunctions.htm

Ray tracing

Basically 2 functions:

- trace
  - find intersection with an object
- directIllumination
  - Direct lighting at a given point



Ray tracing

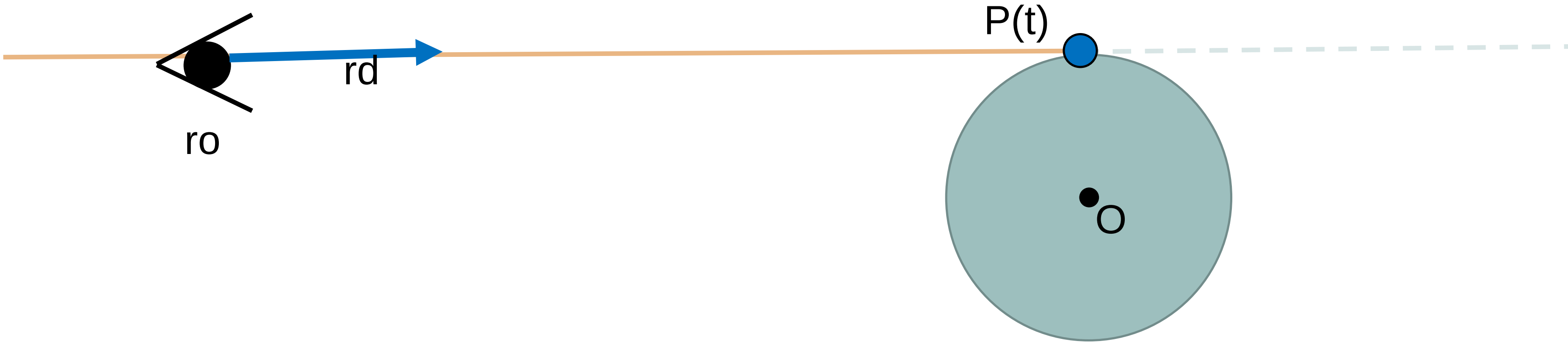
- color trace(ray) {
  - hit = intersectScene(ray)
  - if(hit) {
    - color = directIllumination(hit)
    - if hit is reflective
      - color += c_refl * trace(reflected ray)
    - if hit is transmissive
      - color += c_trans * trace(refracted ray)
  - } else
    - color = background_color
  - return color
- }

Ray tracing

- color directIllumination(hit) {
  - color = (0,0,0)
  - for each light L {
    - T = cast shadow ray to L
    - if hit is not shadowed by L
      - color += Ambient+diffuse+specular terms(L,hit)
  - }
  - return color
- }

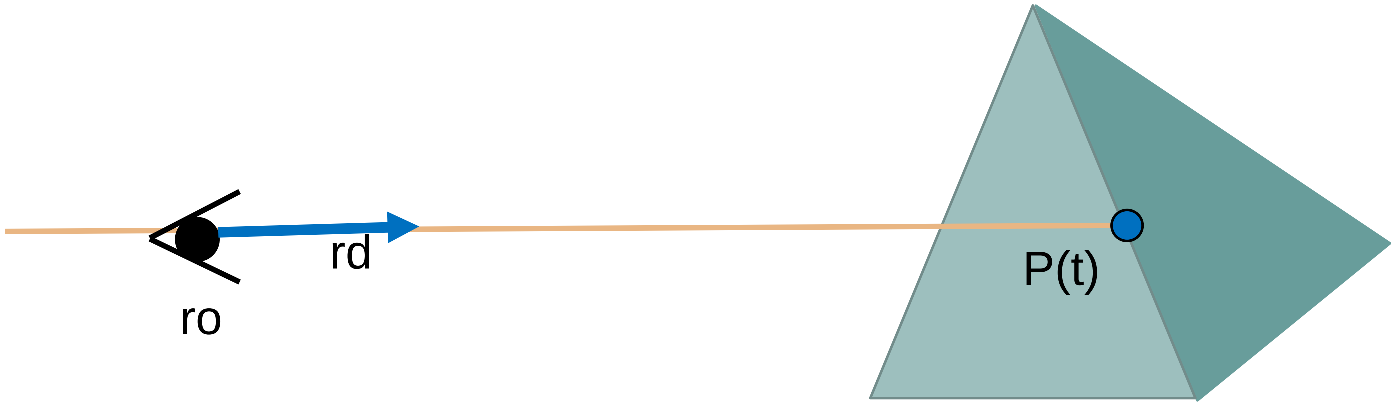
Precision

- Issues
  - Ray origin on an object surface
  - Grazing rays
- Floating point approximation



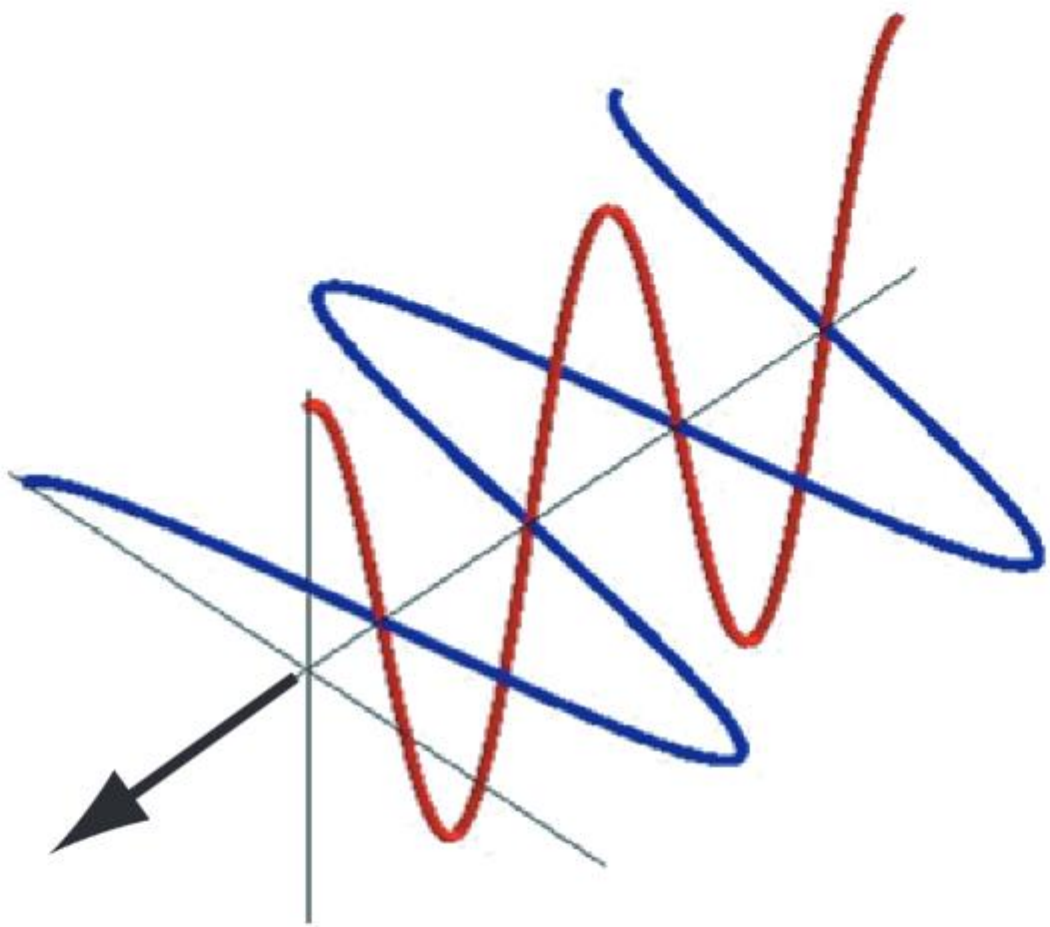
Precision

- Issues
  - Ray origin on an object surface
  - Grazing rays
- Floating point approximation
  - Must report intersection on triangles



Physics of shading

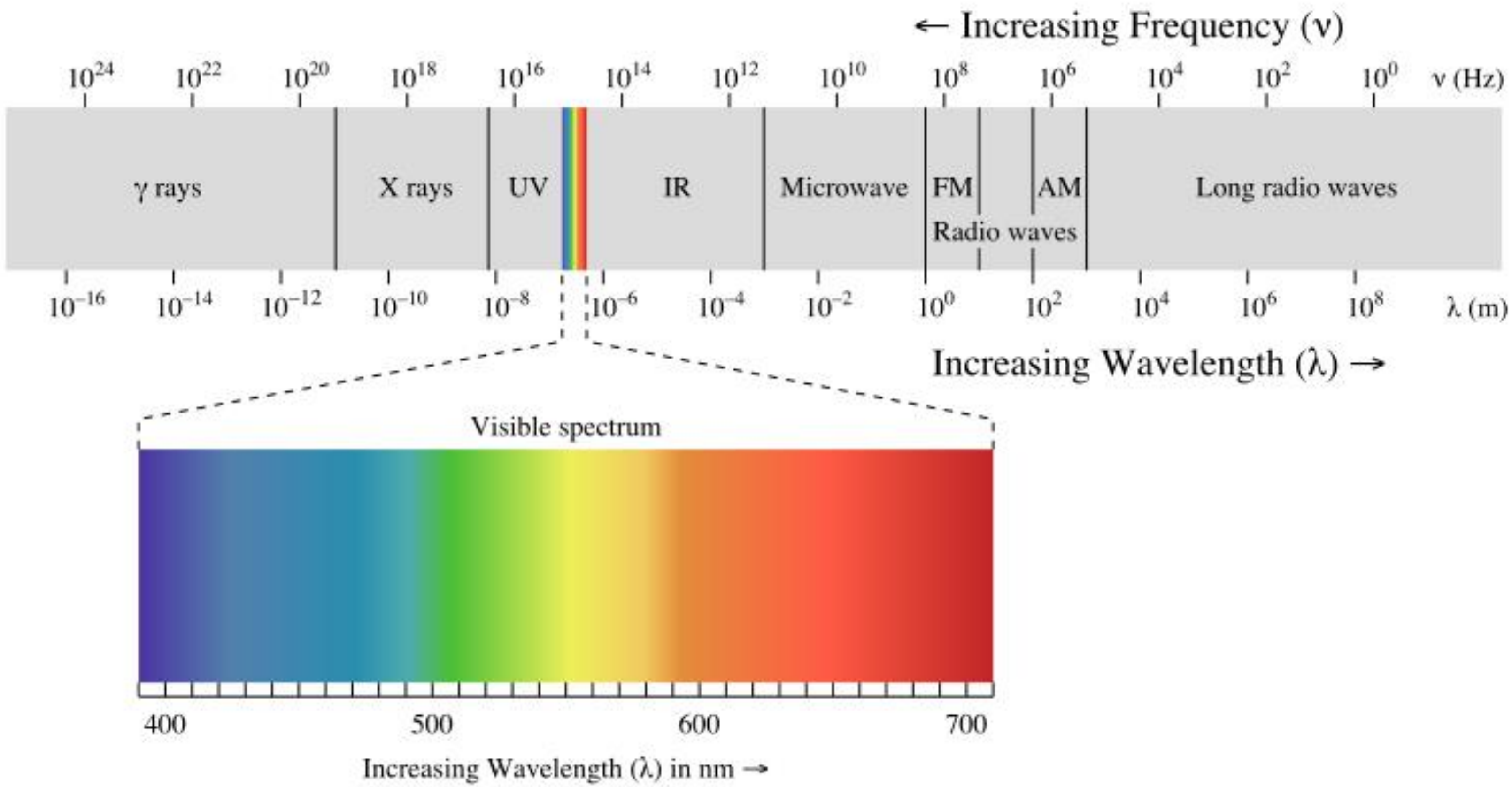
- Light: electromagnetic transverse wave



See Siggraph 2014 course by Naty Hoffman – almost everything from there!!
And images from « Real-time rendering » 3rd edition (A K Peters - 2008)

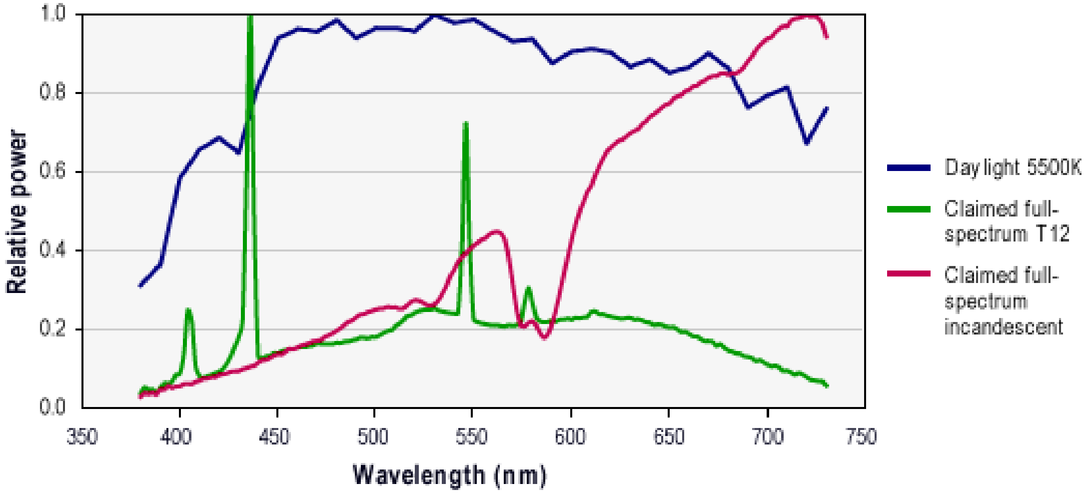
Physics of shading

- Visible light: between 400 and 700 nm



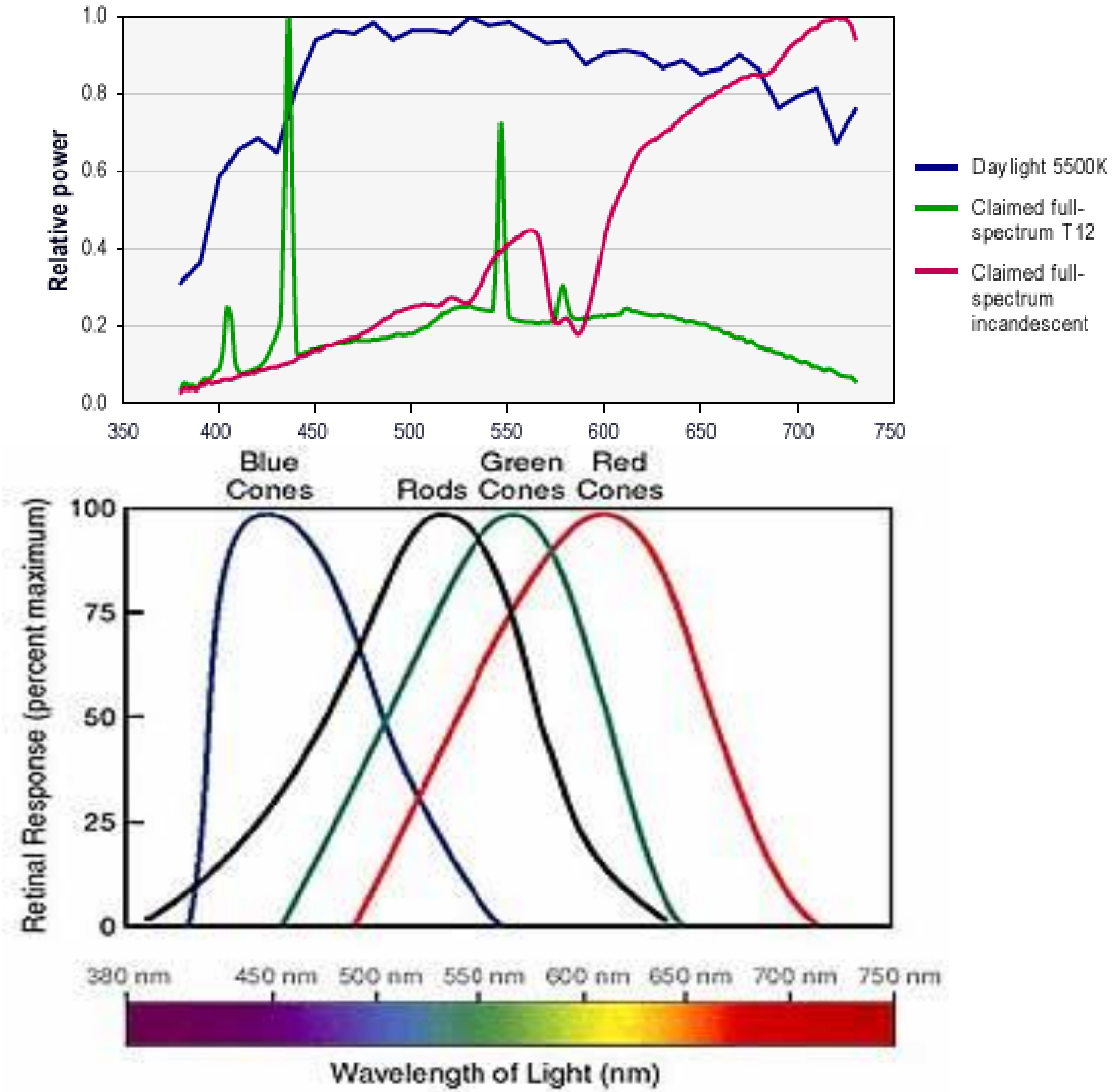
Physics of shading

- Visible light: between 400 and 700 nm



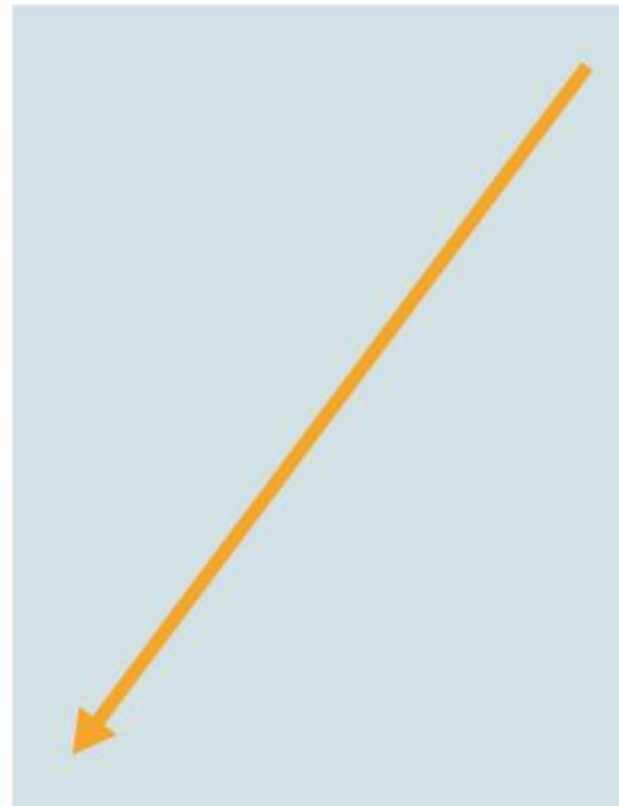
Physics of shading

- Visible light: between 400 and 700 nm



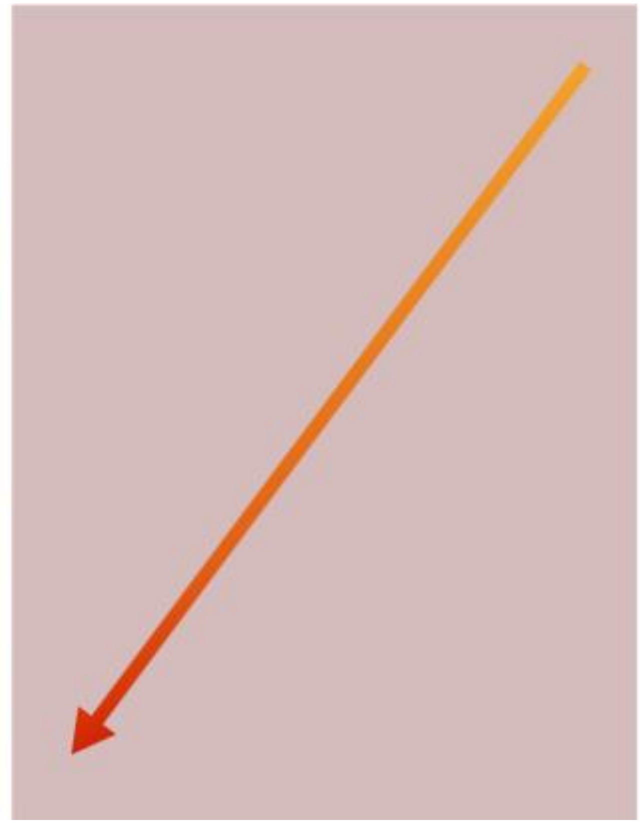
Physics of shading

- Light travels in straight line (homogeneous medium)



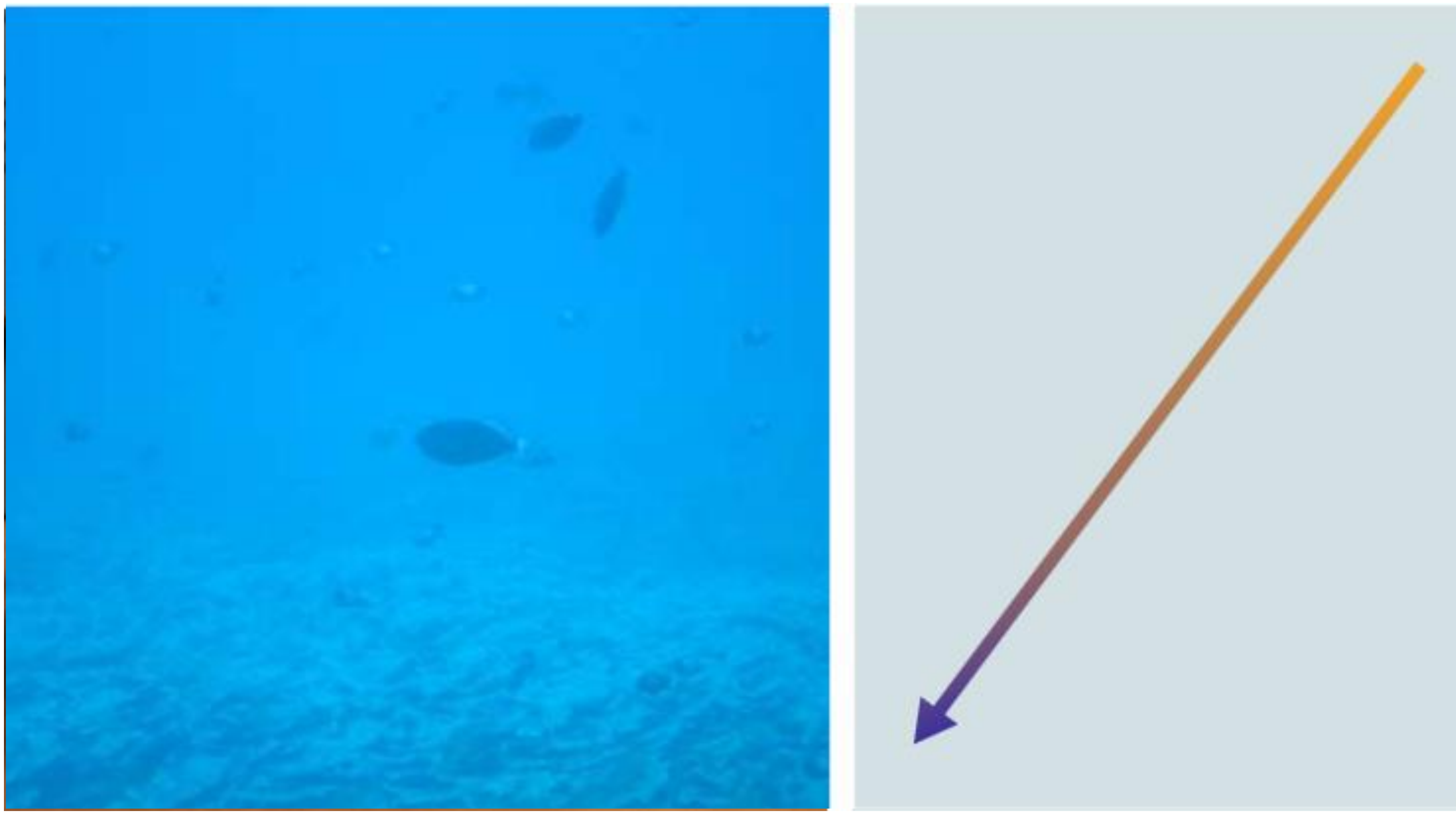
Physics of shading

- Absorption of parts of visible light



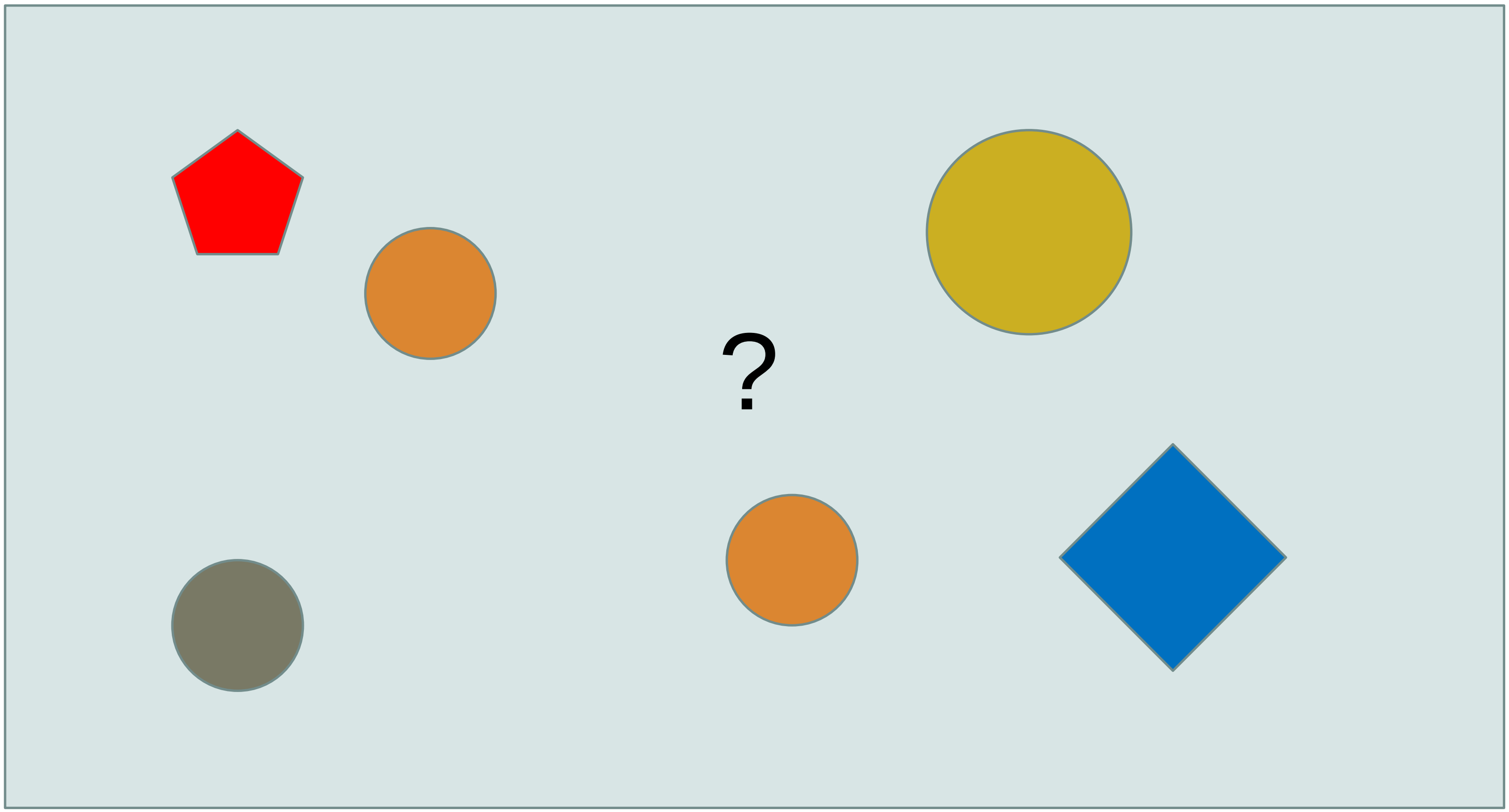
Physics of shading

- Absorption of parts of visible light



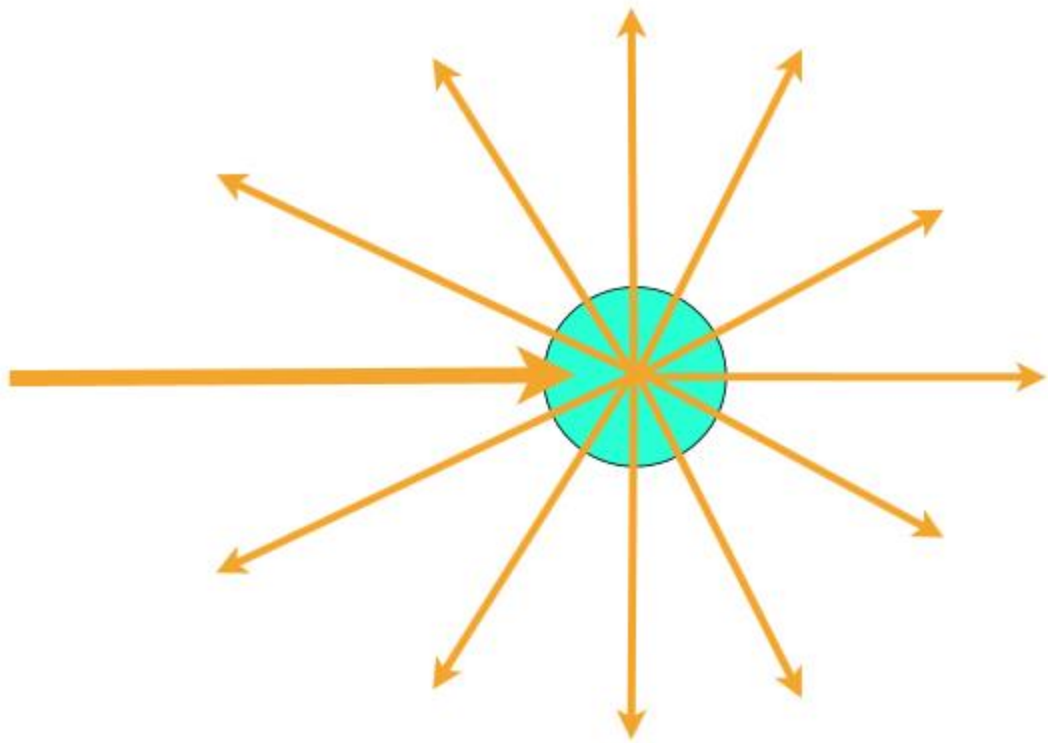
Physics of shading

- Non homogeneous media?



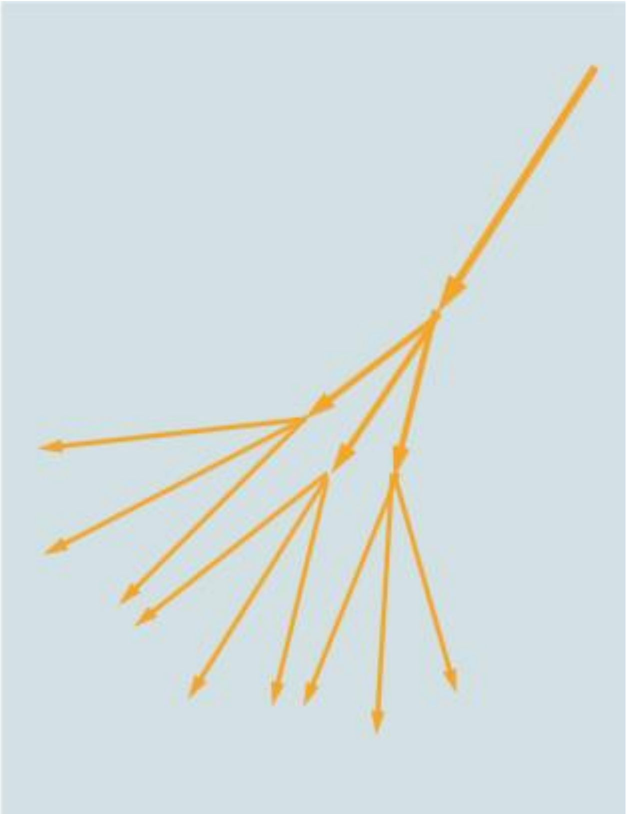
Physics of shading

- Non homogeneous media?
  - Scattering due changes in the index of refraction



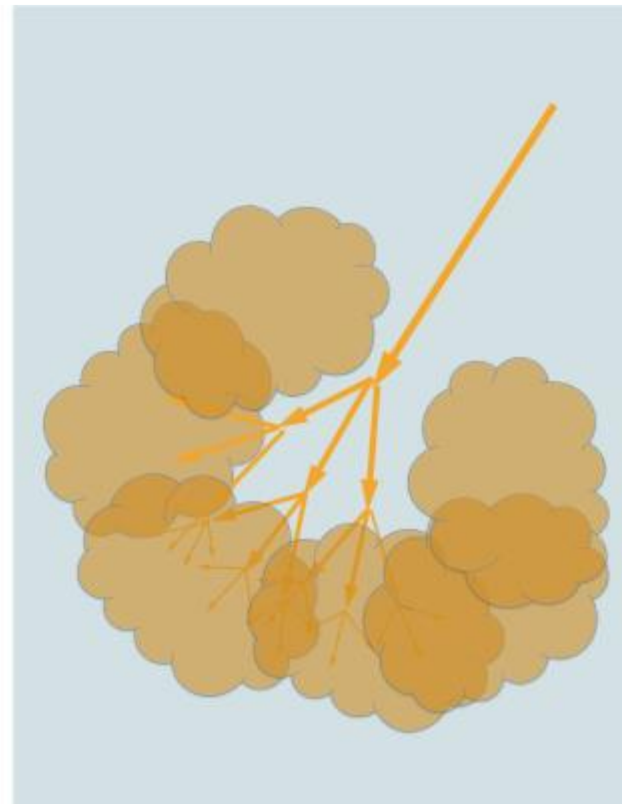
Physics of shading

- Non homogeneous media?
  - Scattering due changes in the index of refraction



Physics of shading

- Non homogeneous media?
  - Scattering due changes in the index of refraction

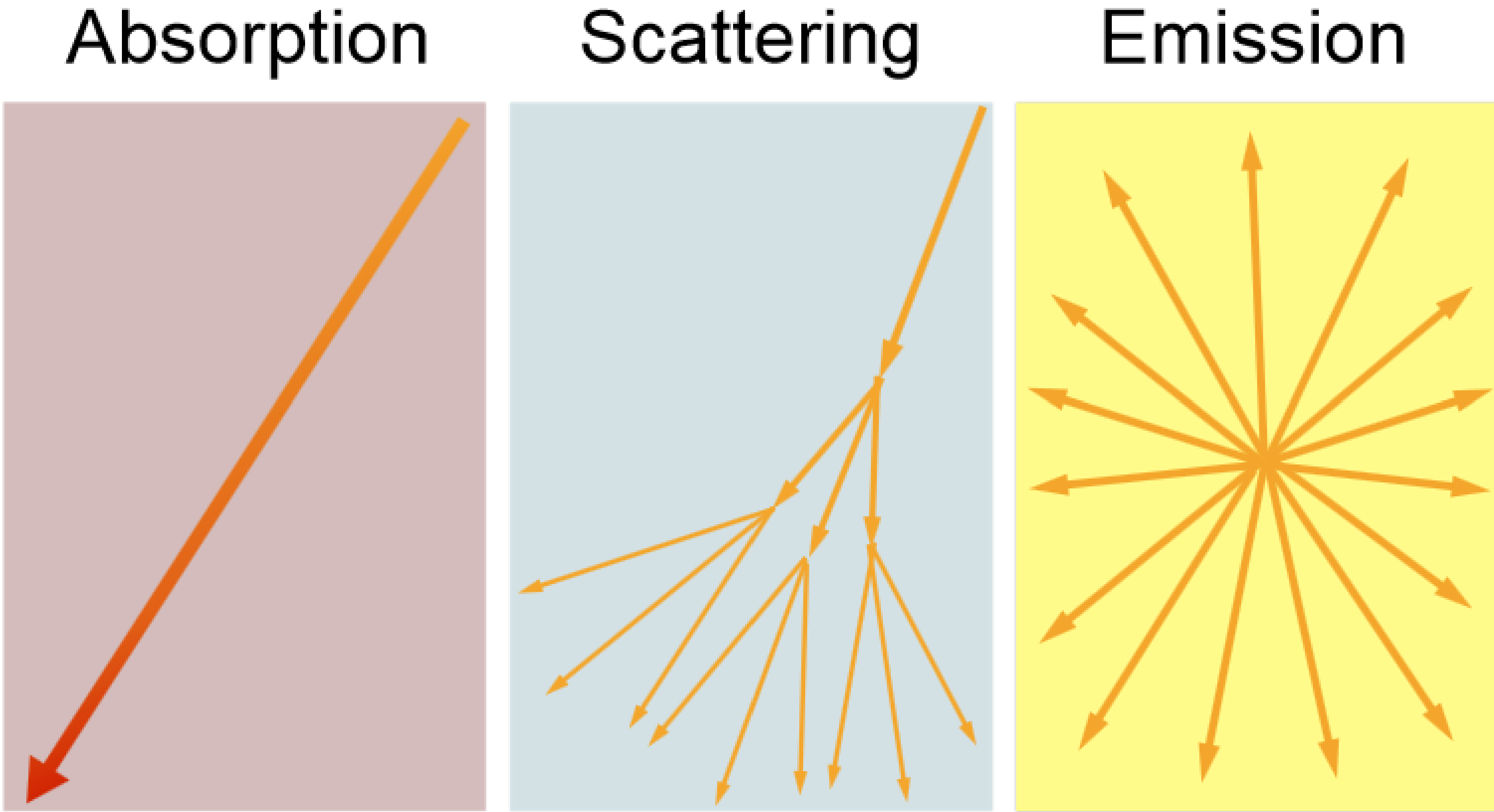


Physics of shading

- Non homogeneous media?
  - Scattering due changes in the index of refraction

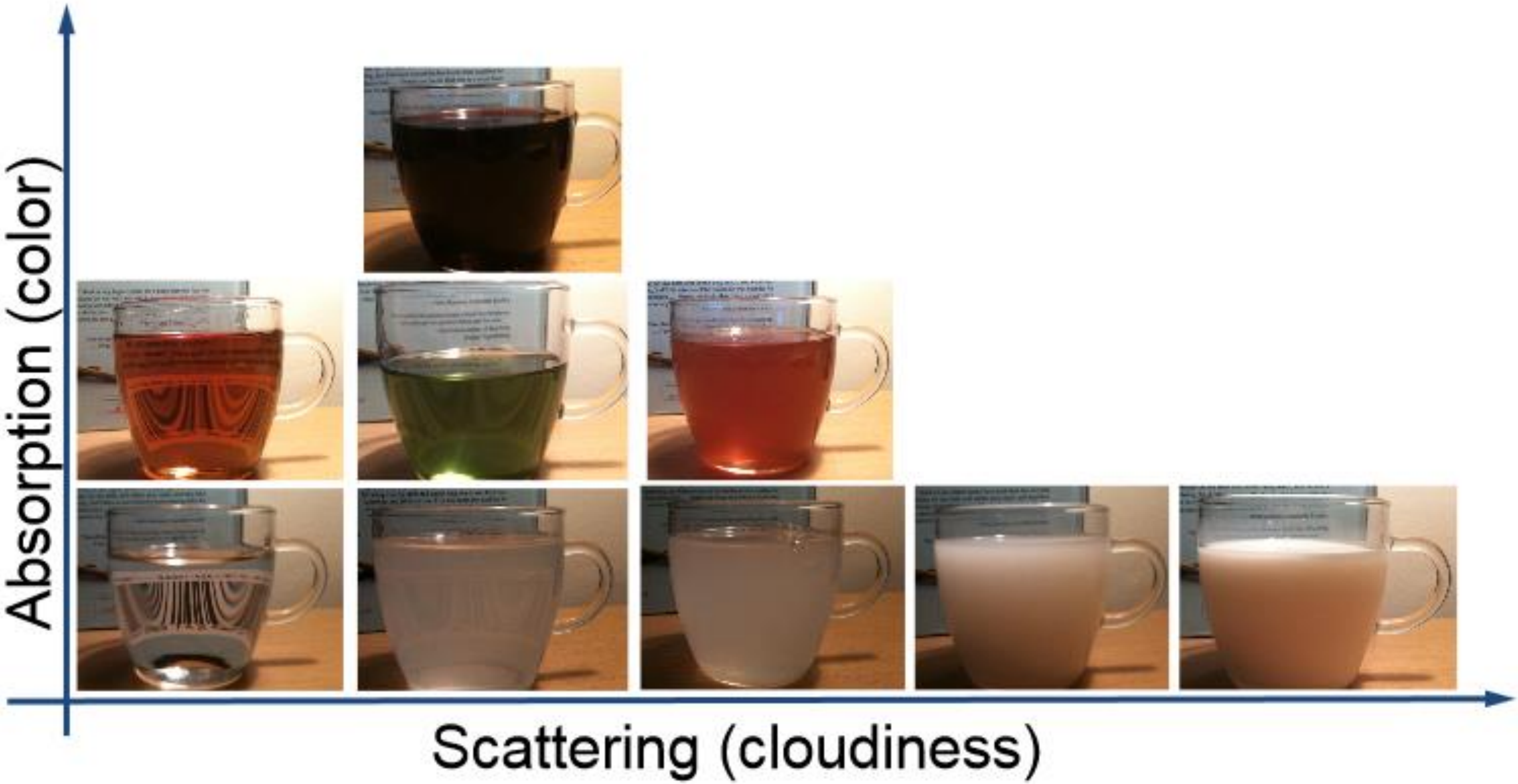
Physics of shading

- 3 Modes of light / matter interaction



Physics of shading

- 3 Modes of light / matter interaction



Physics of shading

- What about surfaces?








SIGGRAPH2014

Physics of shading

- Case of (optically) flat surface: Snell Descartes laws

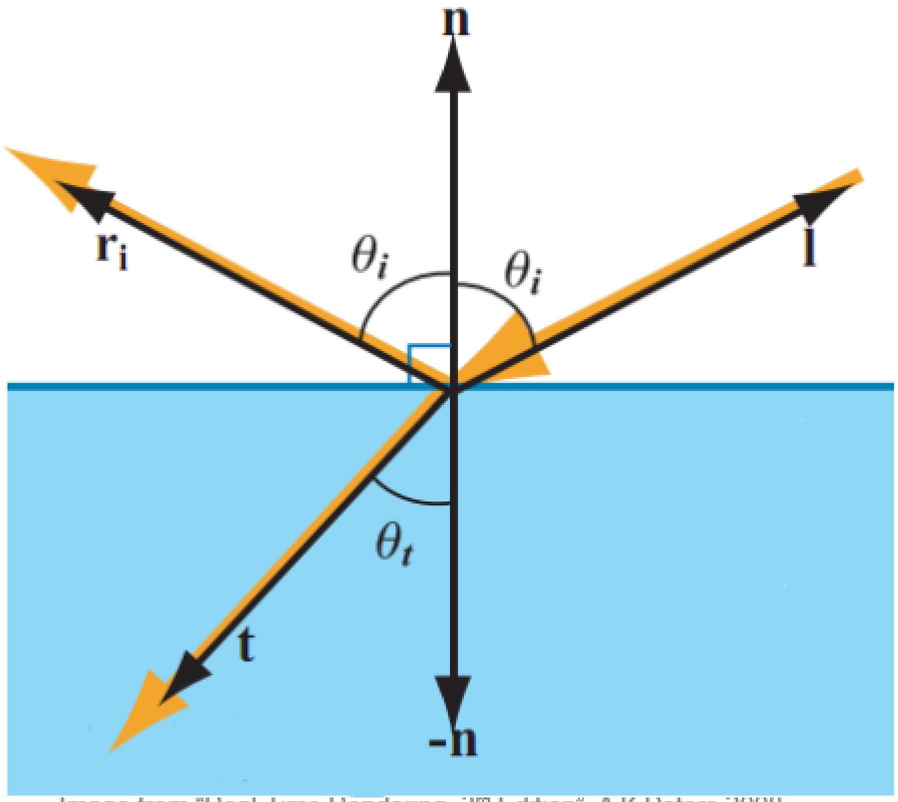
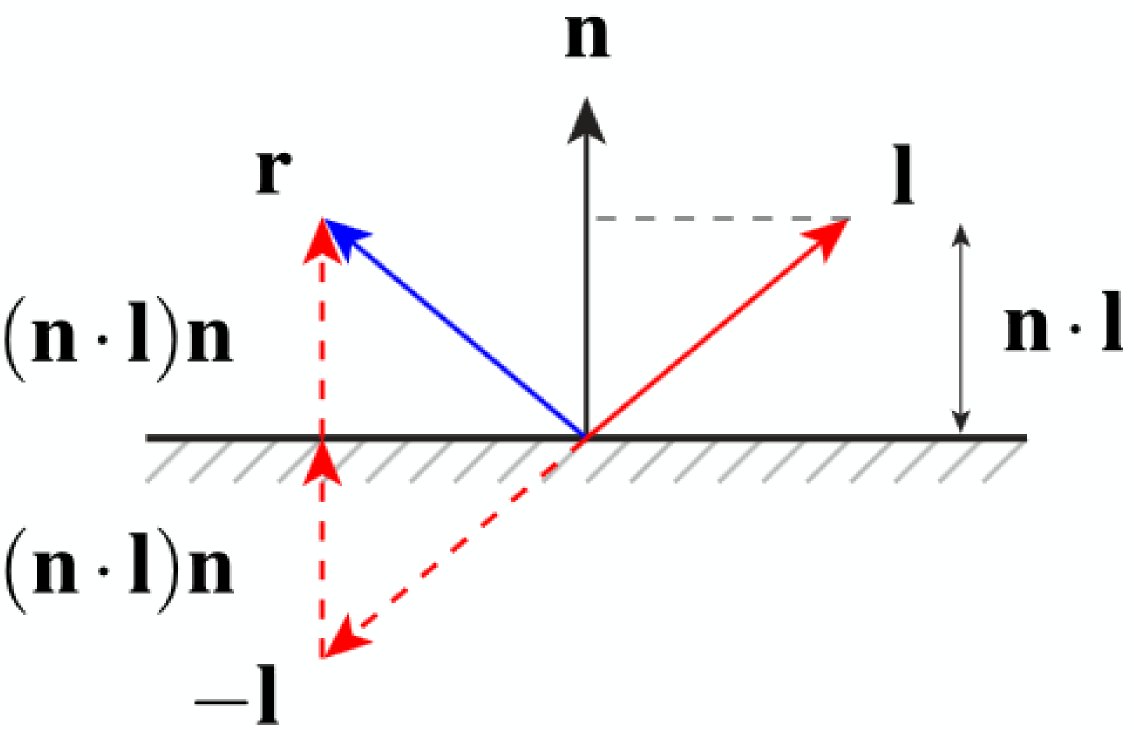


Image from "Real-Time Rendering, 3rd Edition", A K Peters 2008

Physics of shading

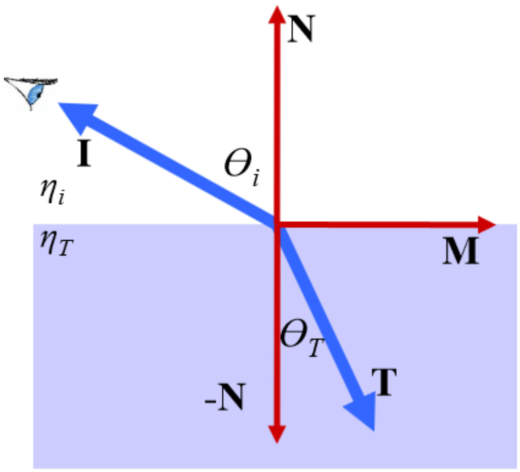
- Case of (optically) flat surface: Snell Descartes laws
  - Incident ray is reflected…



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

Physics of shading

- Case of (optically) flat surface: Snell Descartes laws
  - Incident ray is reflected…
  - … and refracted



T

$$n_i \sin\theta_i = n_T \sin\theta_T$$

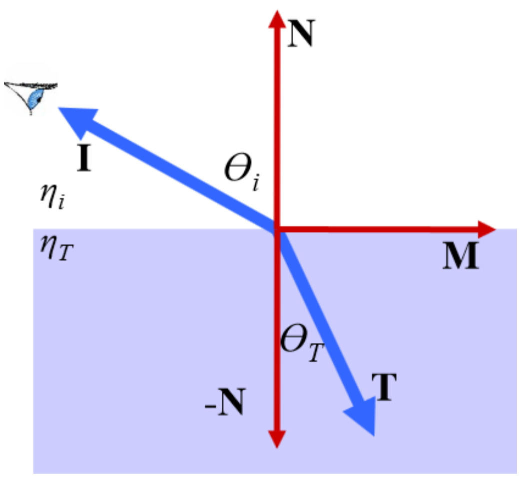
$$\frac{\sin\theta_T}{\sin\theta_i} = \frac{n_i}{n_T} = n_r$$

Physics of shading

- Case of (optically) flat surface: Snell Descartes laws
  - Incident ray is reflected…
  - … and refracted

$$\mathbf{I} = \mathbf{N}\cos\Theta_i - \mathbf{M}\sin\Theta_i$$

$$\mathbf{M} = (\mathbf{N}\cos\Theta_i - \mathbf{I}) / \sin\Theta_i$$

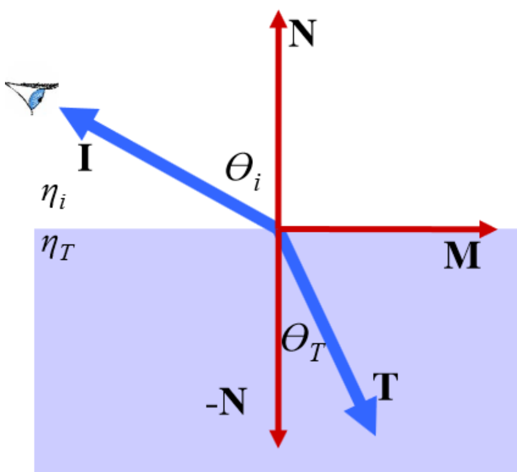


$$n_i \sin\theta_i = n_T \sin\theta_T$$

$$\frac{\sin\theta_T}{\sin\theta_i} = \frac{n_i}{n_T} = n_r$$

Physics of shading

- Case of (optically) flat surface: Snell Descartes laws
  - Incident ray is reflected…
  - … and refracted



$$n_i \sin\theta_i = n_T \sin\theta_T$$

$$\frac{\sin\theta_T}{\sin\theta_i} = \frac{n_i}{n_T} = n_r$$

$$\mathbf{I} = \mathbf{N}\cos\Theta_i - \mathbf{M}\sin\Theta_i$$

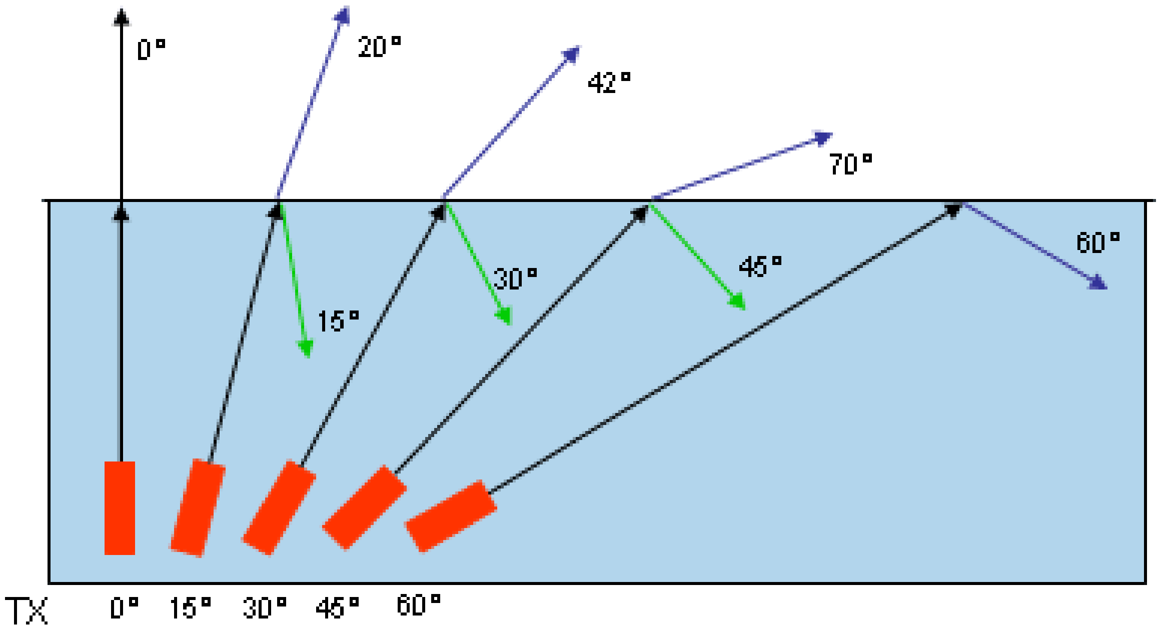
$$\mathbf{M} = (\mathbf{N}\cos\Theta_i - \mathbf{I}) / \sin\Theta_i$$

$$\begin{aligned}
\mathbf{T} &= -\mathbf{N}\cos\Theta_T + \mathbf{M}\sin\Theta_T \\
&= -\mathbf{N}\cos\Theta_T + (\mathbf{N}\cos\Theta_i - \mathbf{I})\sin\Theta_T / \sin\Theta_i \quad \textit{Plug M} \\
&= -\mathbf{N}\cos\Theta_T + (\mathbf{N}\cos\Theta_i - \mathbf{I})\,\eta_r \\
&= [\,\eta_r\cos\Theta_i - \cos\Theta_T\,]\,\mathbf{N} - \eta_r\mathbf{I} \\
&= [\,\eta_r\cos\Theta_i - \sqrt{1-\sin^2\Theta_T}\,]\,\mathbf{N} - \eta_r\mathbf{I} \\
&= [\,\eta_r\cos\Theta_i - \sqrt{1-\eta_r^2\sin^2\Theta_i}\,]\,\mathbf{N} - \eta_r\mathbf{I} \\
&= [\,\eta_r\cos\Theta_i - \sqrt{1-\eta_r^2(1-\cos^2\Theta_i)}\,]\,\mathbf{N} - \eta_r\mathbf{I} \\
&= [\,\eta_r(\mathbf{N}\cdot\mathbf{I}) - \sqrt{1-\eta_r^2(1-(\mathbf{N}\cdot\mathbf{I})^2)}\,]\,\mathbf{N} - \eta_r\mathbf{I}
\end{aligned}$$

let's get rid of the cos & sin

Physics of shading

- Case of (optically) flat surface: Snell Descartes laws
  - Incident ray is reflected…
  - … and refracted



Total internal reflection

Physics of shading

- Case of (optically) flat surface: Snell Descartes laws
  - Incident ray is reflected…
  - … and refracted

Fig. 3.7A The optical manhole. From under water, the entire celestial hemisphere is compressed into a circle only 97.2° across. The dark boundary defining the edges of the manhole is not sharp due to surface waves. The rays are analogous to the crepuscular type seen in hazy air, Section 1.9. (Photo by D. Granger)

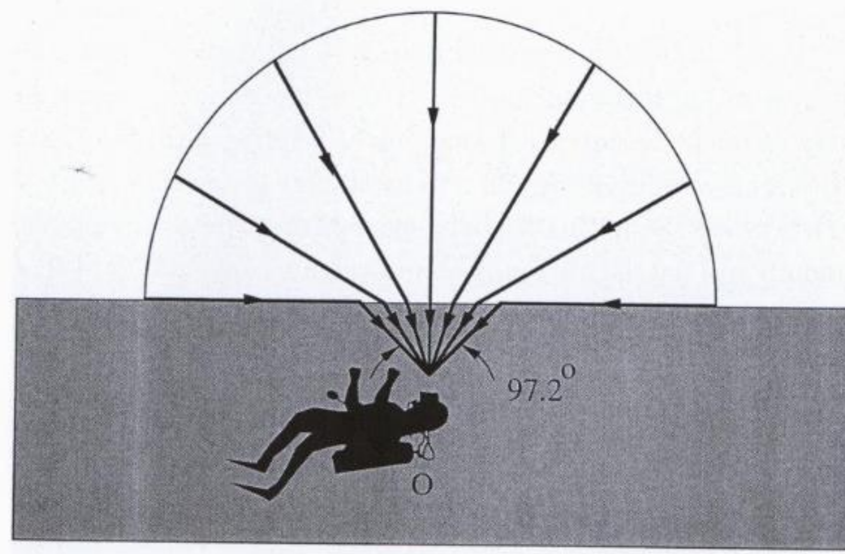
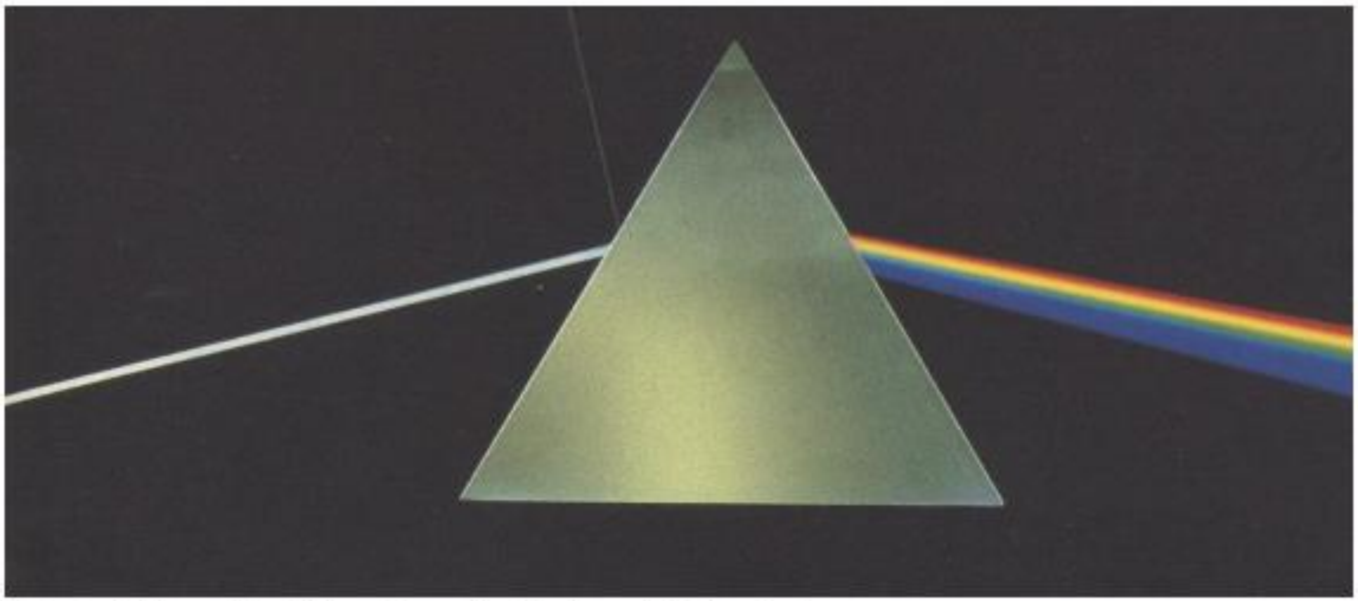


Fig. 3.7B The optical manhole. Light from the horizon (angle of incidence = 90°) is refracted downward at an angle of 48.6°. This compresses the sky into a circle with a diameter of 97.2° instead of its usual 180°.

Physics of shading

- Case of (optically) flat surface: Snell Descartes laws
  - Incident ray is reflected…
  - … and refracted

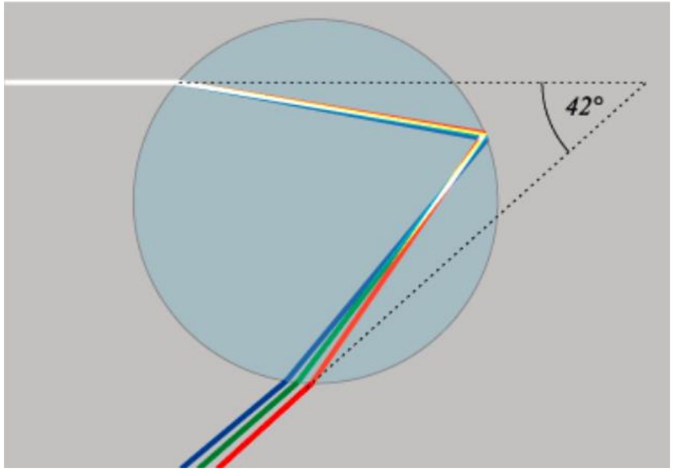


Pink Floyd, *The Dark Side of the Moon*

Refraction is wavelet dependent

Physics of shading

- Case of (optically) flat surface: Snell Descartes laws
  - Incident ray is reflected…
  - … and refracted



How rainbows work:

- Refraction + internal reflection + refraction
- Max for angle around 42 deg

Physics of shading

- Case of (optically) flat surface: Snell Descartes laws
  - Incident ray is reflected…
  - … and refracted
- The amount of reflection vs refraction
  - Controlled with Fresnel law (electromagnetic wave)

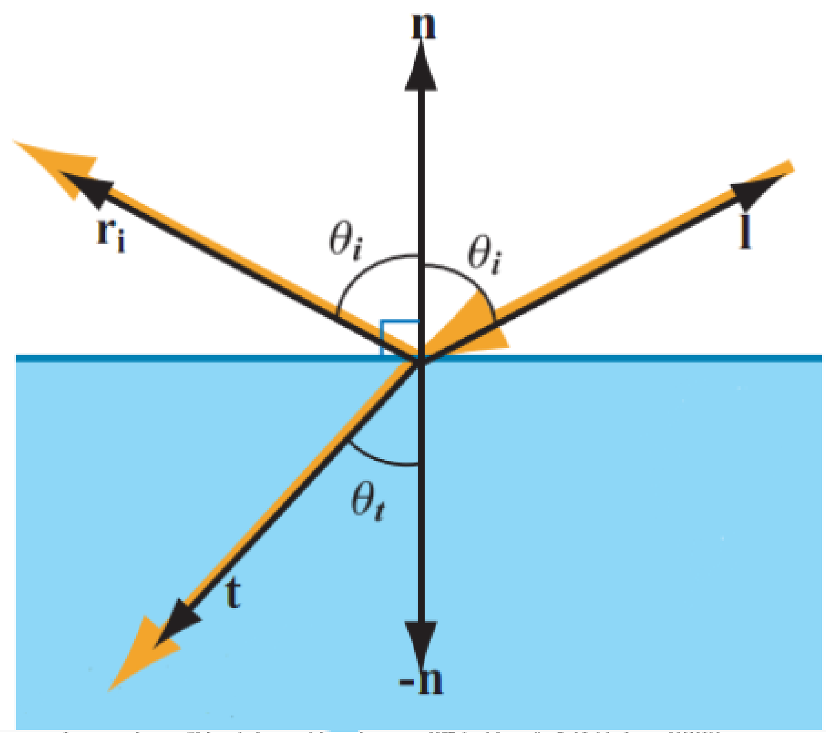


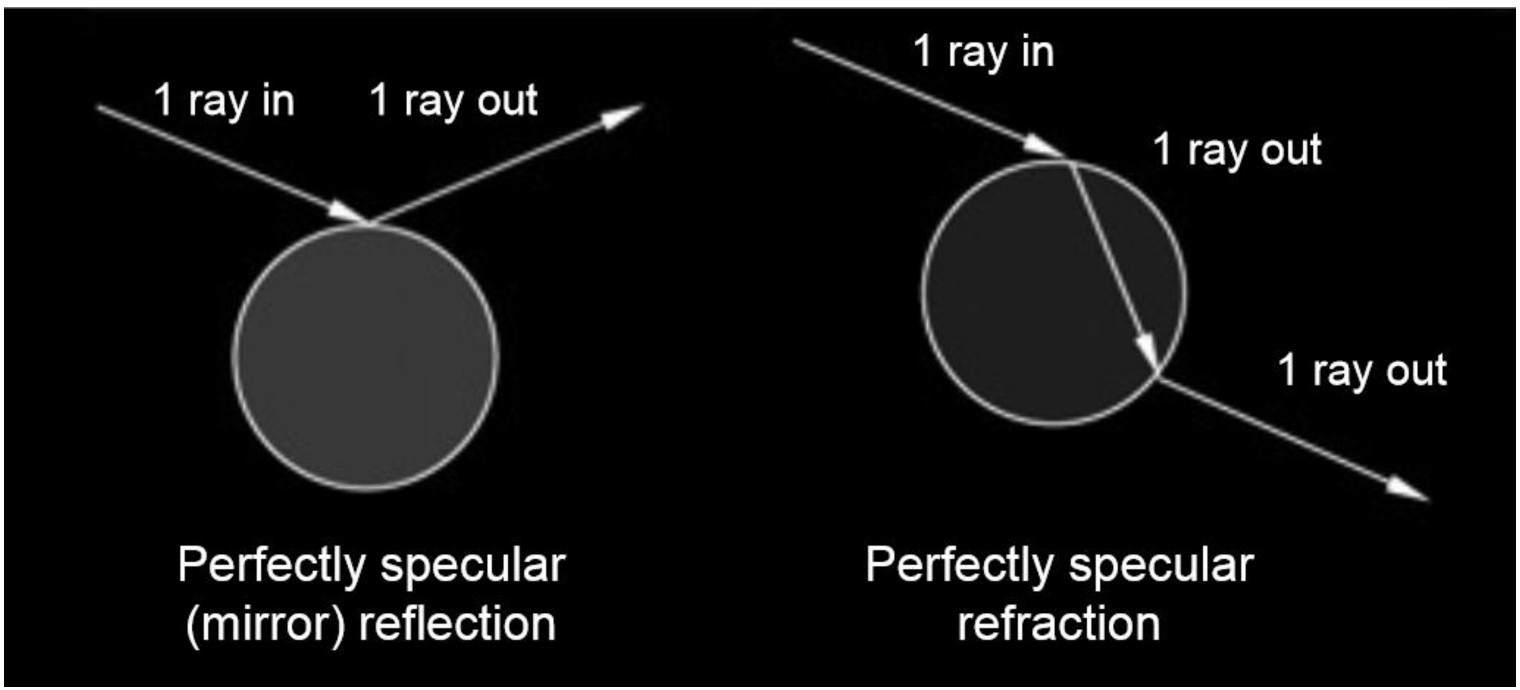
Image from "Real-Time Rendering, 3rd Edition", A K Peters 2008

Physics of shading

- Case of (optically) flat surface: Snell Descartes laws
  - Incident ray is reflected…
  - … and refracted
- The amount of reflection vs refraction
  - Controlled with Fresnel law (electromagnetic wave)

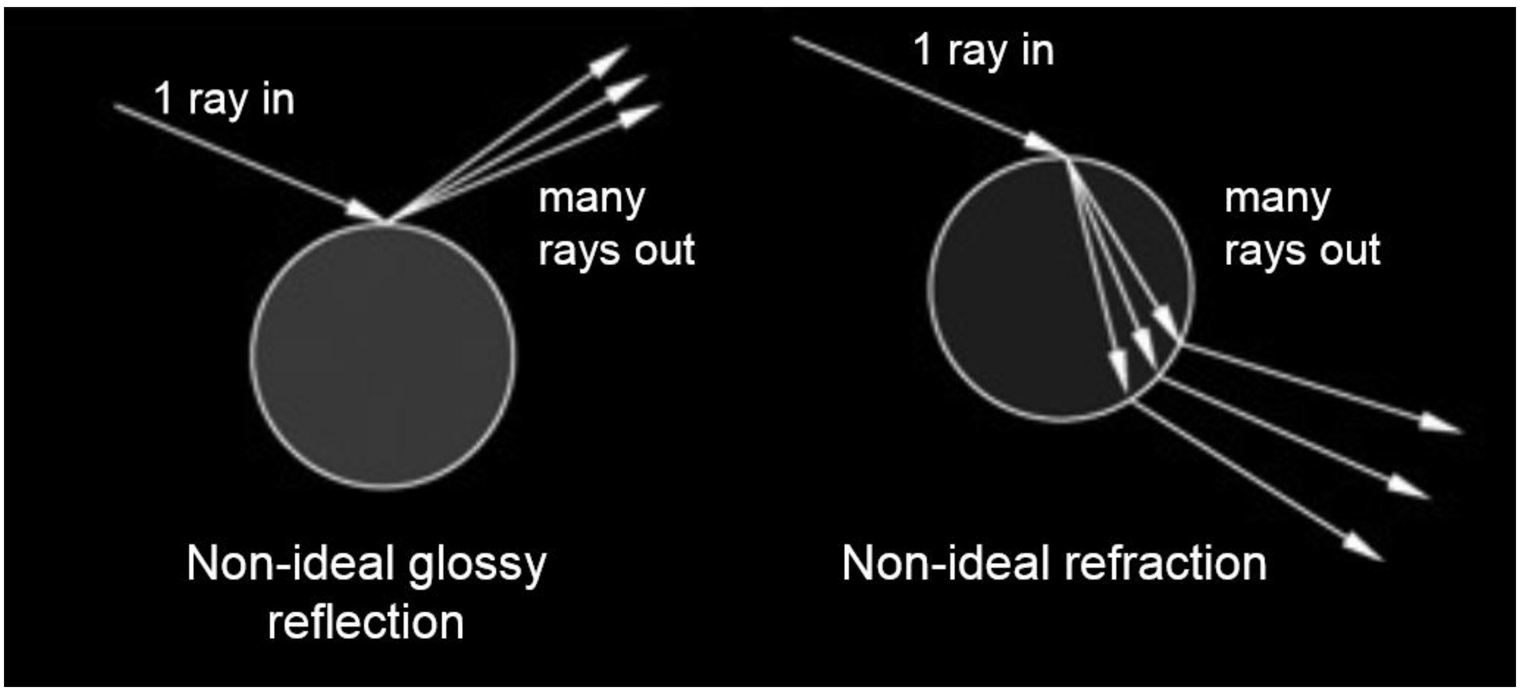
Physics of shading

- Case of (optically) flat surface: Snell Descartes laws
  - Incident ray is reflected…
  - … and refracted
- The amount of reflection vs refraction
  - Controlled with Fresnel law (electromagnetic wave)



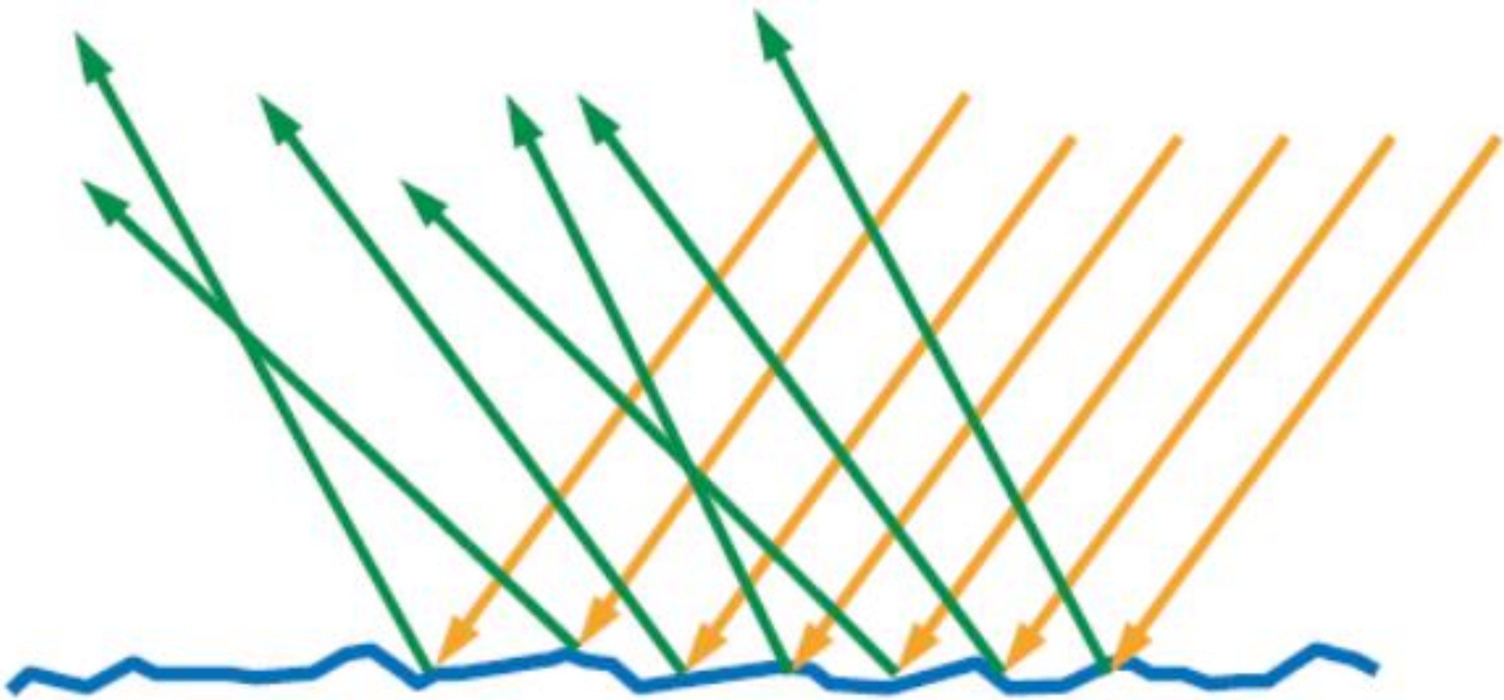
Physics of shading

- Case of (optically) flat surface: Snell Descartes laws
  - Incident ray is reflected…
  - … and refracted
- The amount of reflection vs refraction
  - Controlled with Fresnel law (electromagnetic wave)



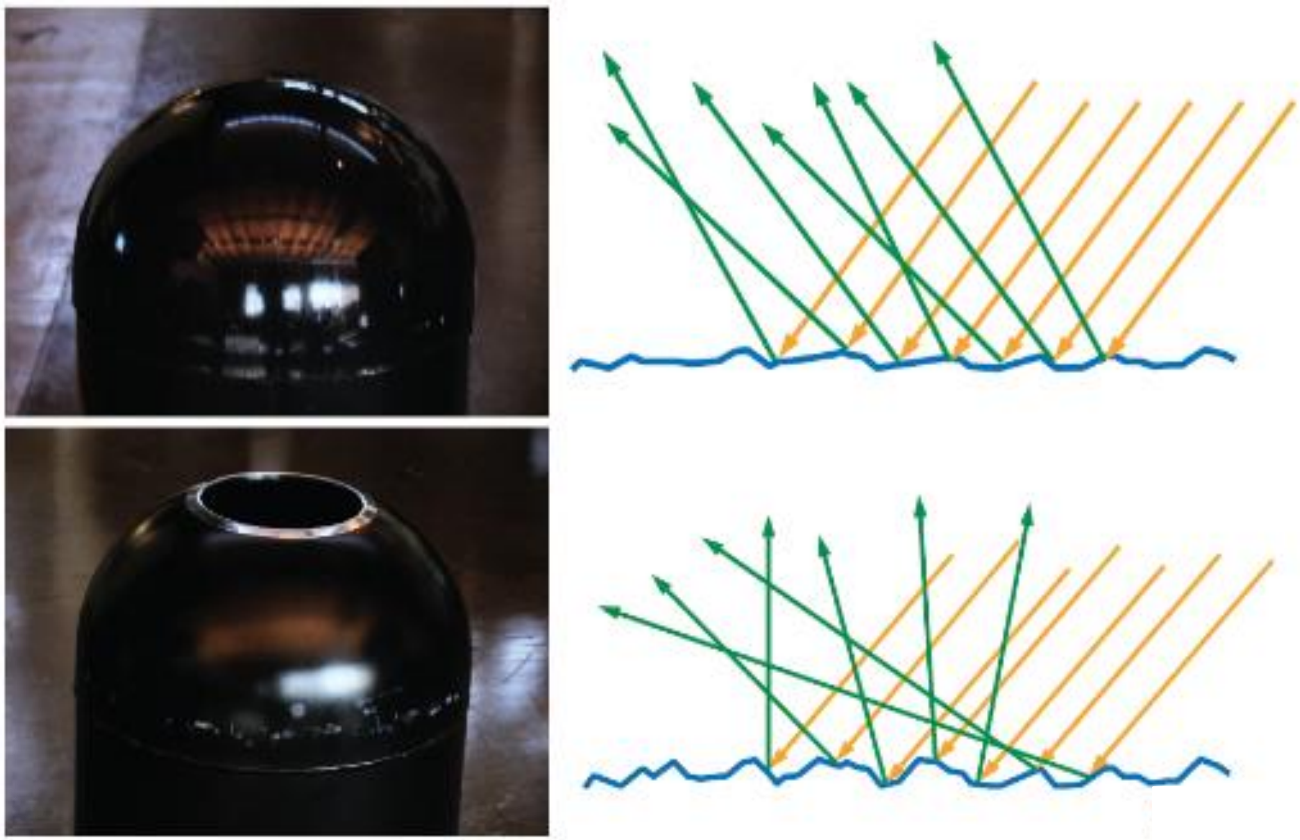
Micro geometry

- Microgeometry bumps
  - Bigger than light wavelength
  - But too small to be visible!
  - Agregate of all response



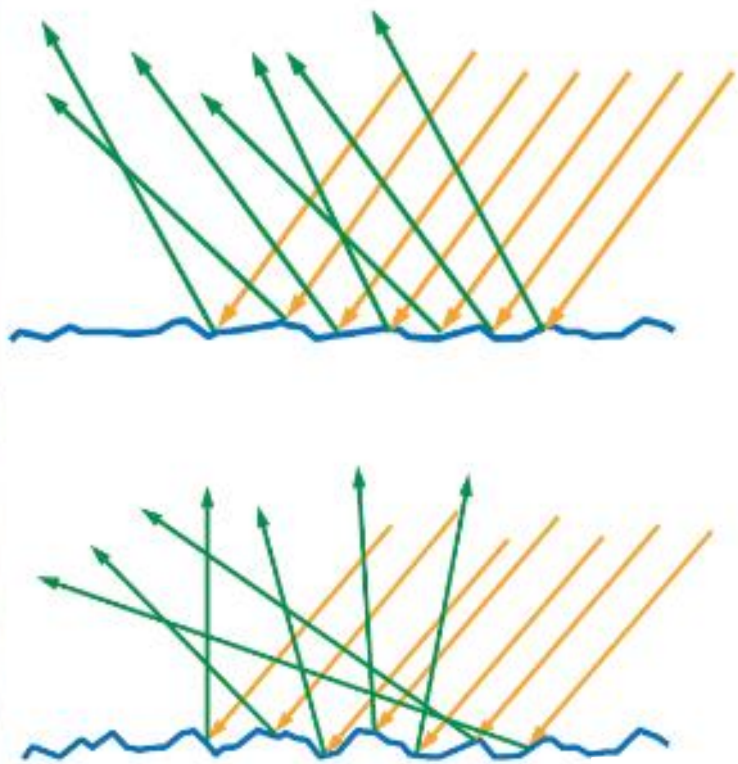
Micro geometry

- Microgeometry bumps
  - Bigger than light wavelength
  - But too small to be visible!
  - Agregate of all response



Micro geometry

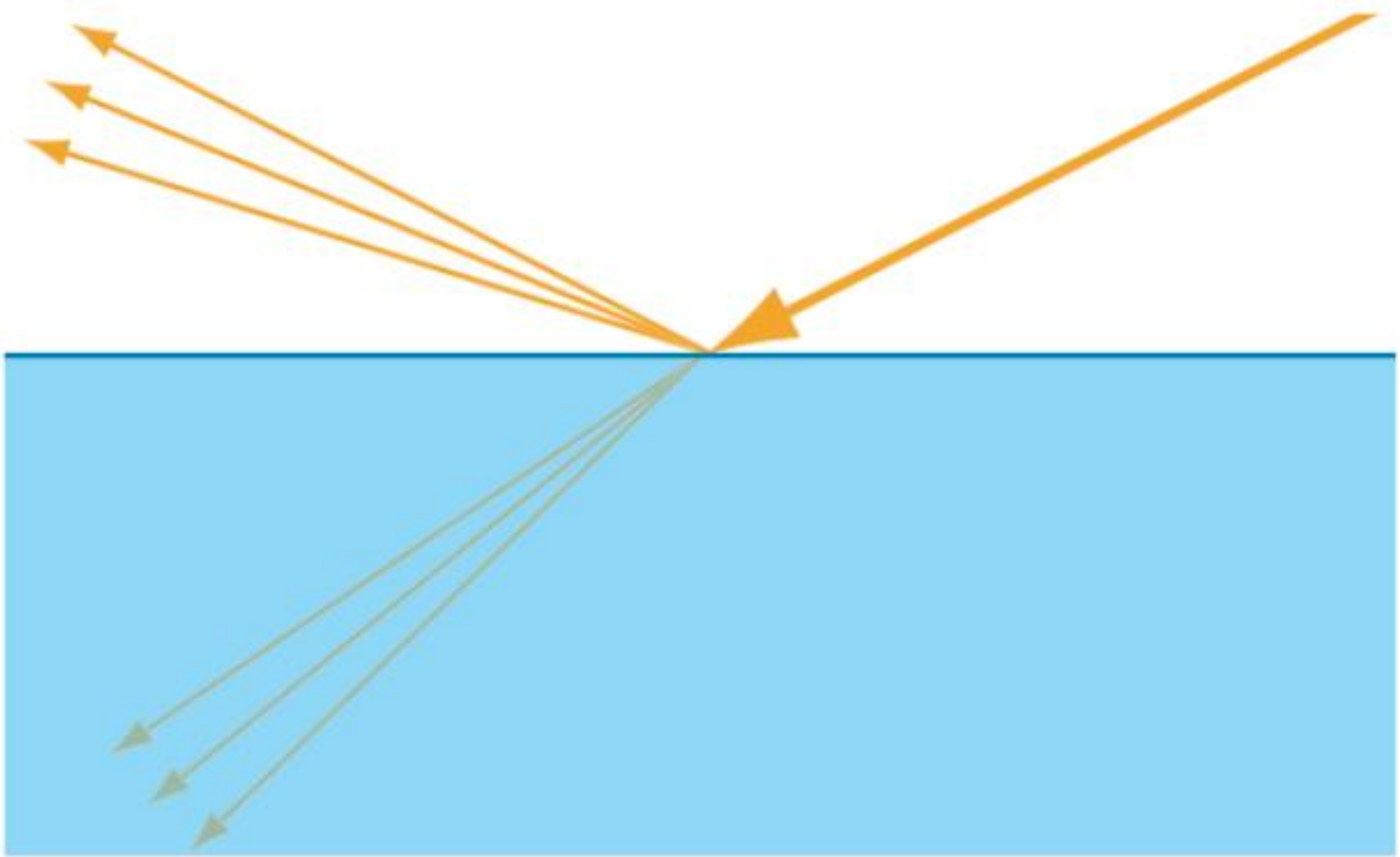
- Microgeometry bumps
  - Bigger than light wavelength
  - But too small to be visible!
  - Agregate of all response



~Mirror

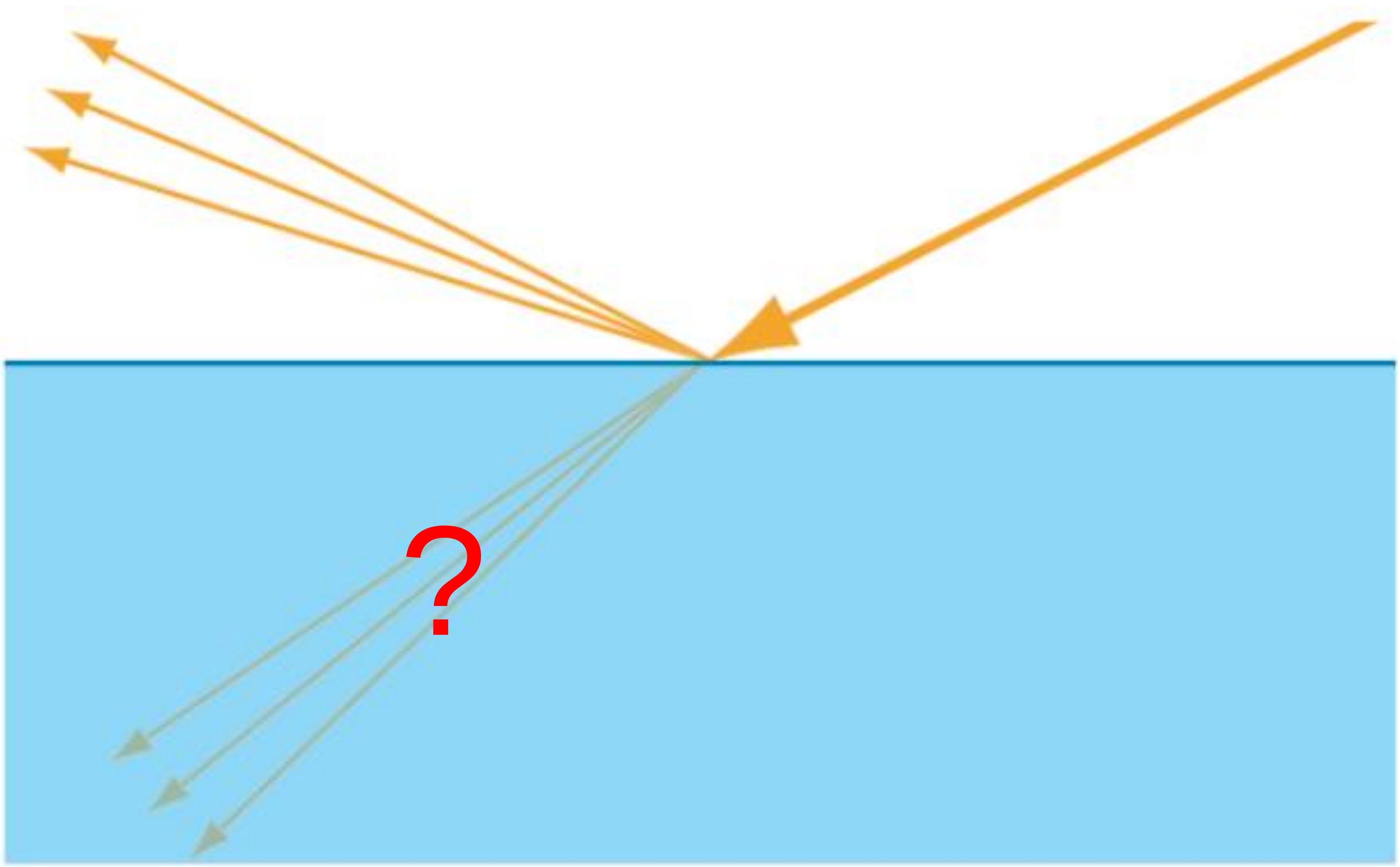
~Rough

Macroscopic view



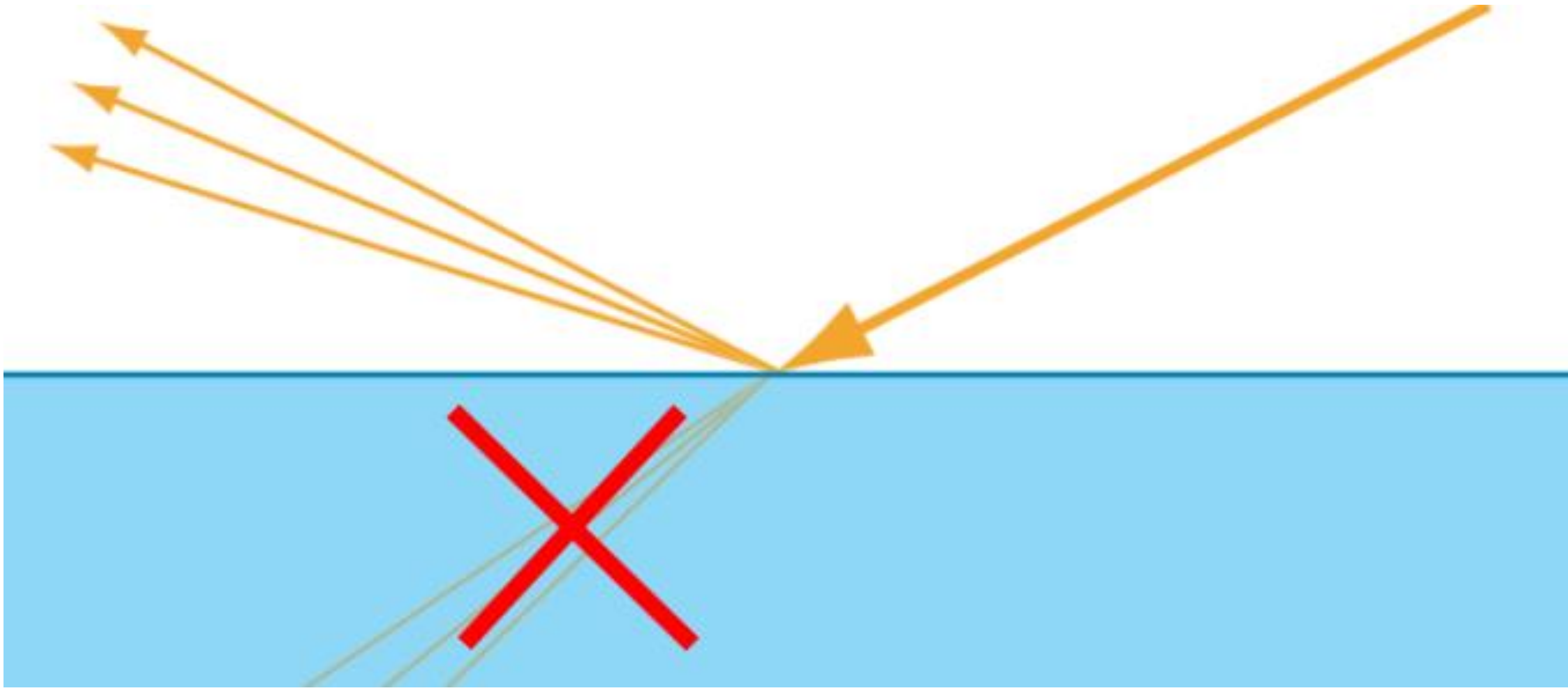
Macroscopic view

- Refractions?



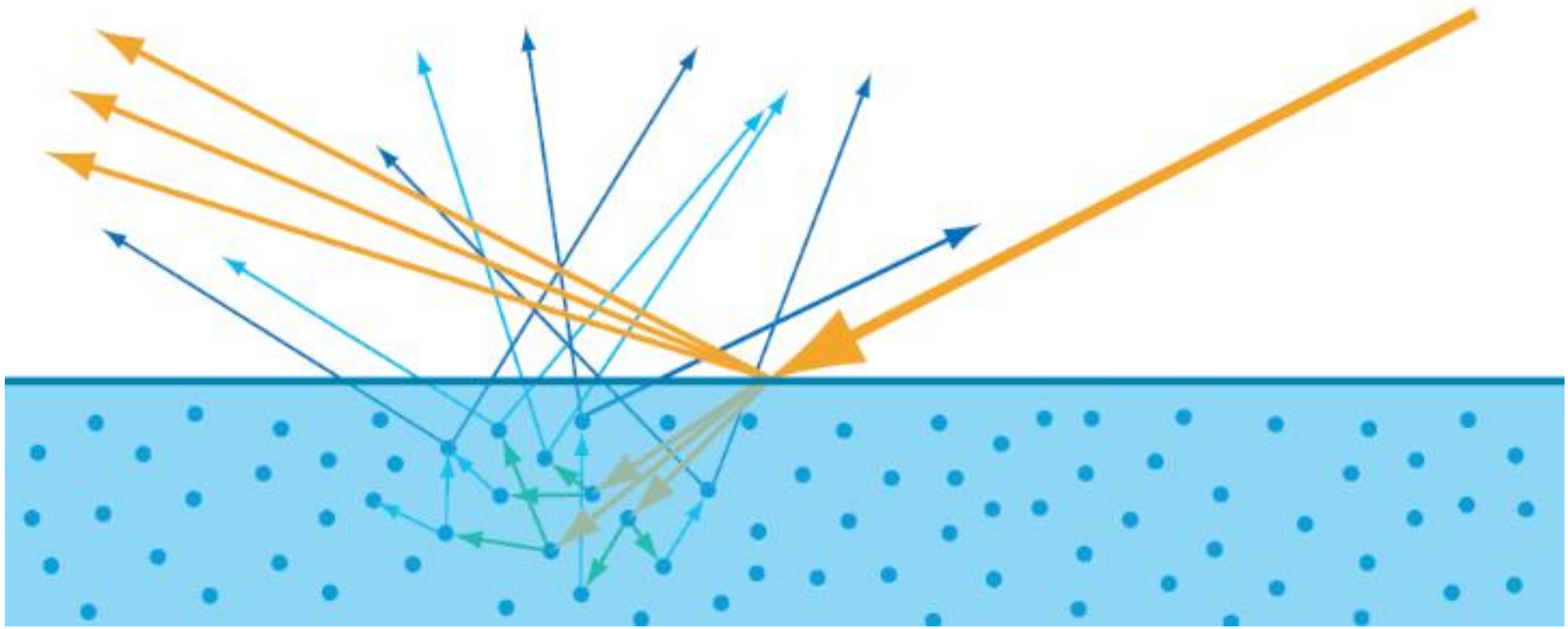
Metals

- Refracted light immediatly absorbed by free electrons



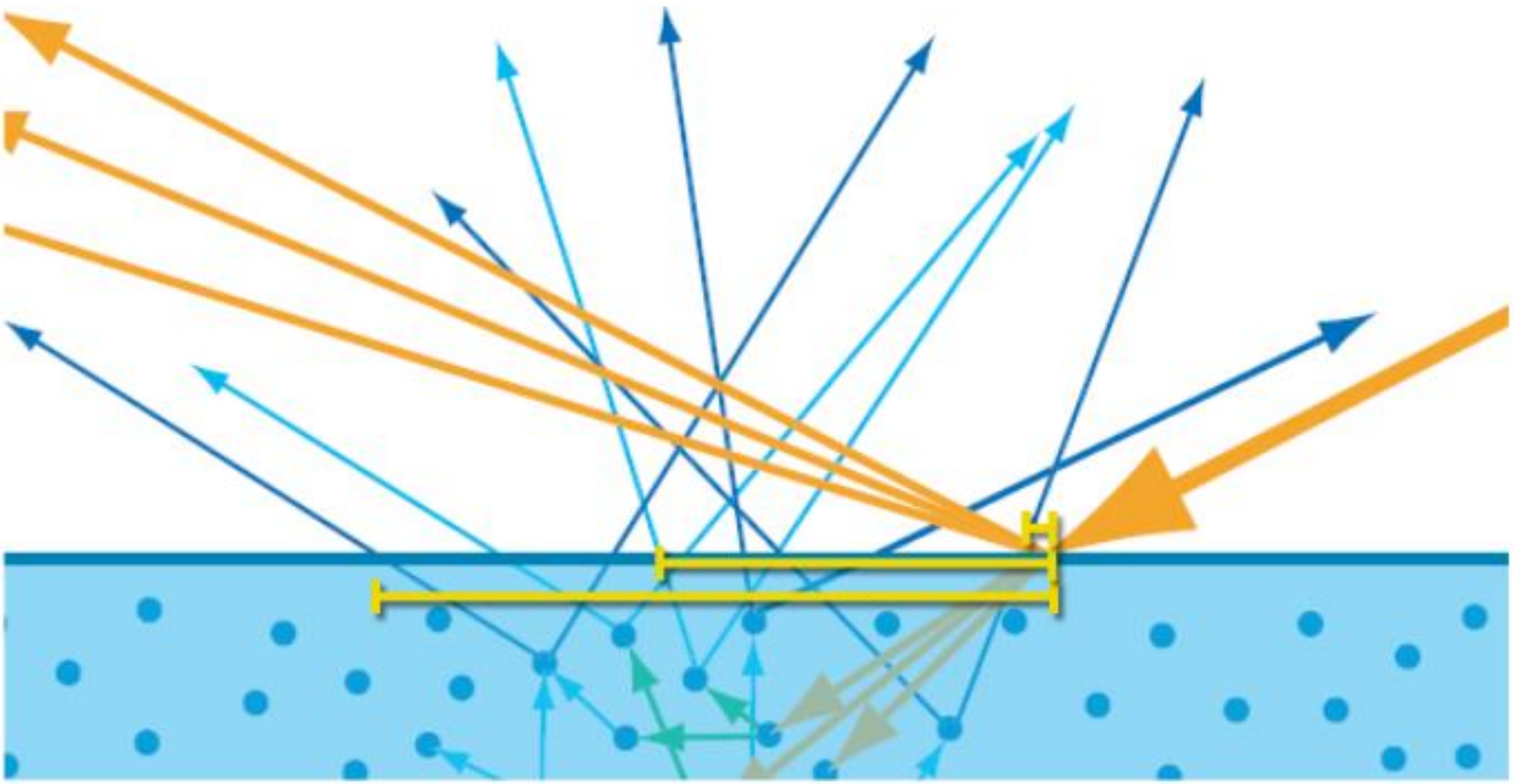
Non-metals

- Behave like regular participating media
  - Light is scattered (enough) are re-emited



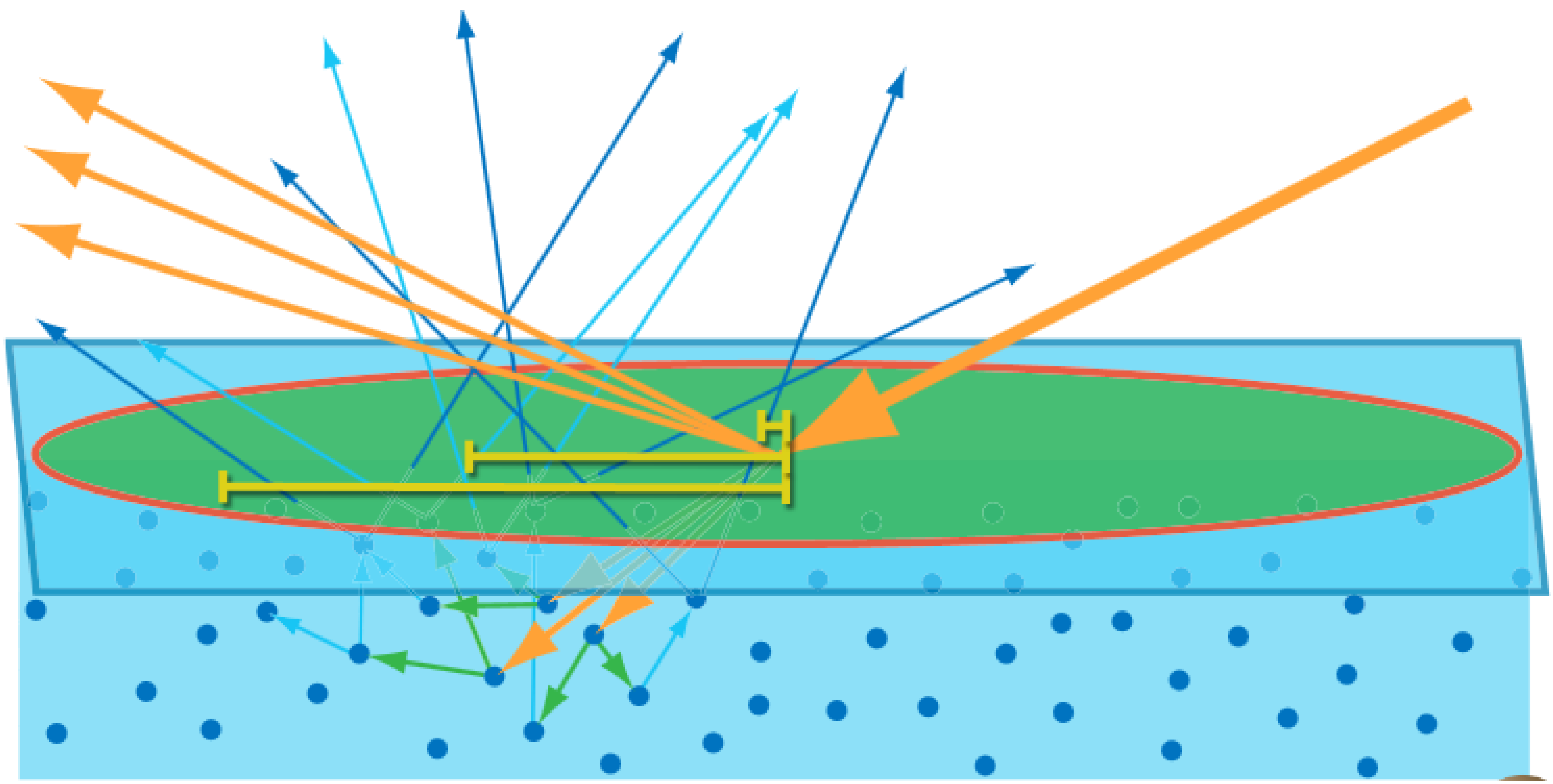
Non-metals

- Behave like regular participating media
  - Light is scattered (enough) are re-emited
  - Distance depends on particle densities



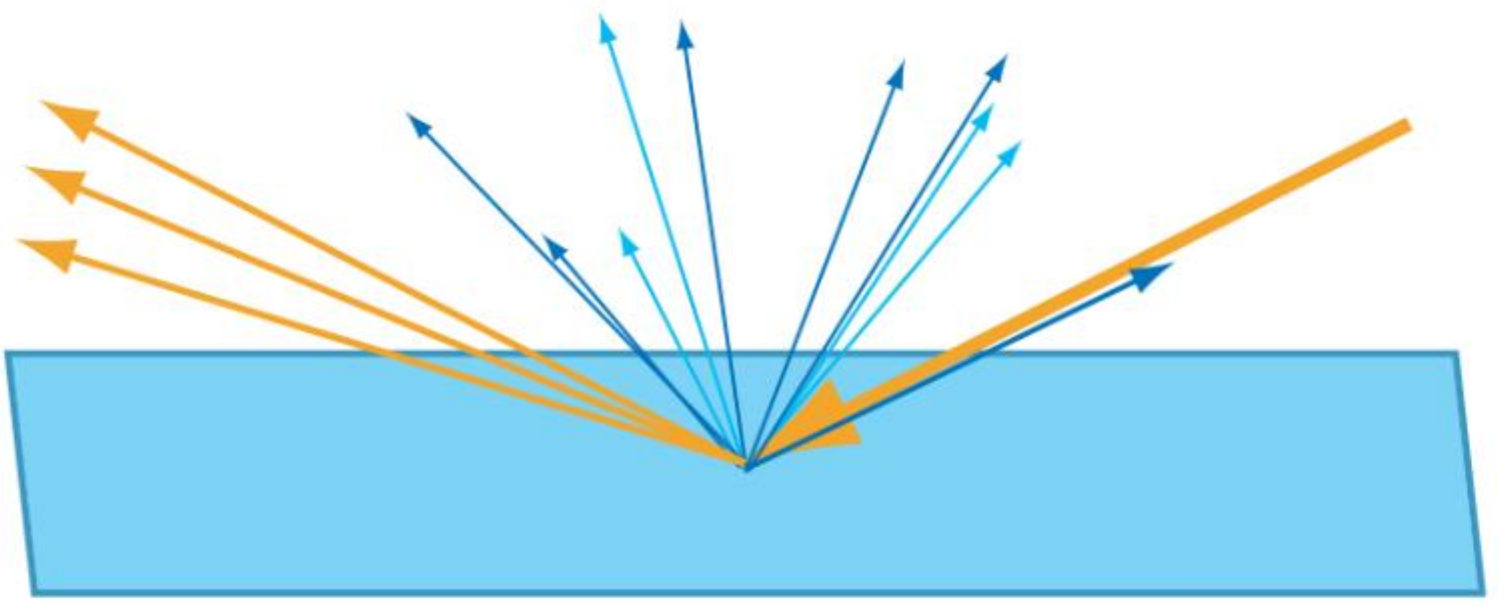
Non-metals

- Behave like regular participating media
  - Light is scattered (enough) are re-emited
  - Distance depends on particle densities
    - Assume = 0 if shading area (surface point) large enough
    - Must be taken into account otherwise



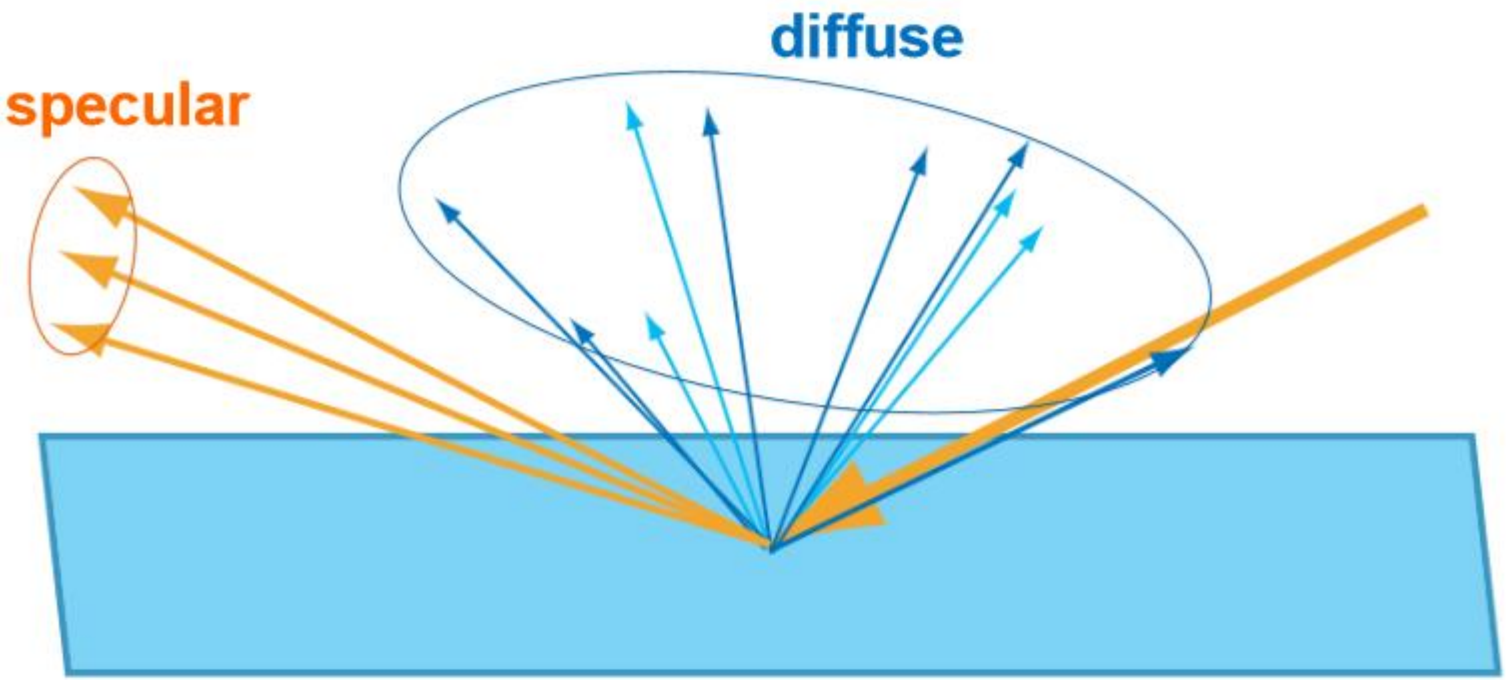
Non-metals

- Behave like regular participating media
  - Light is scattered (enough) are re-emited
  - Distance depends on particle densities
    - Assume = 0 if shading area (surface point) large enough
    - Must be taken into account otherwise



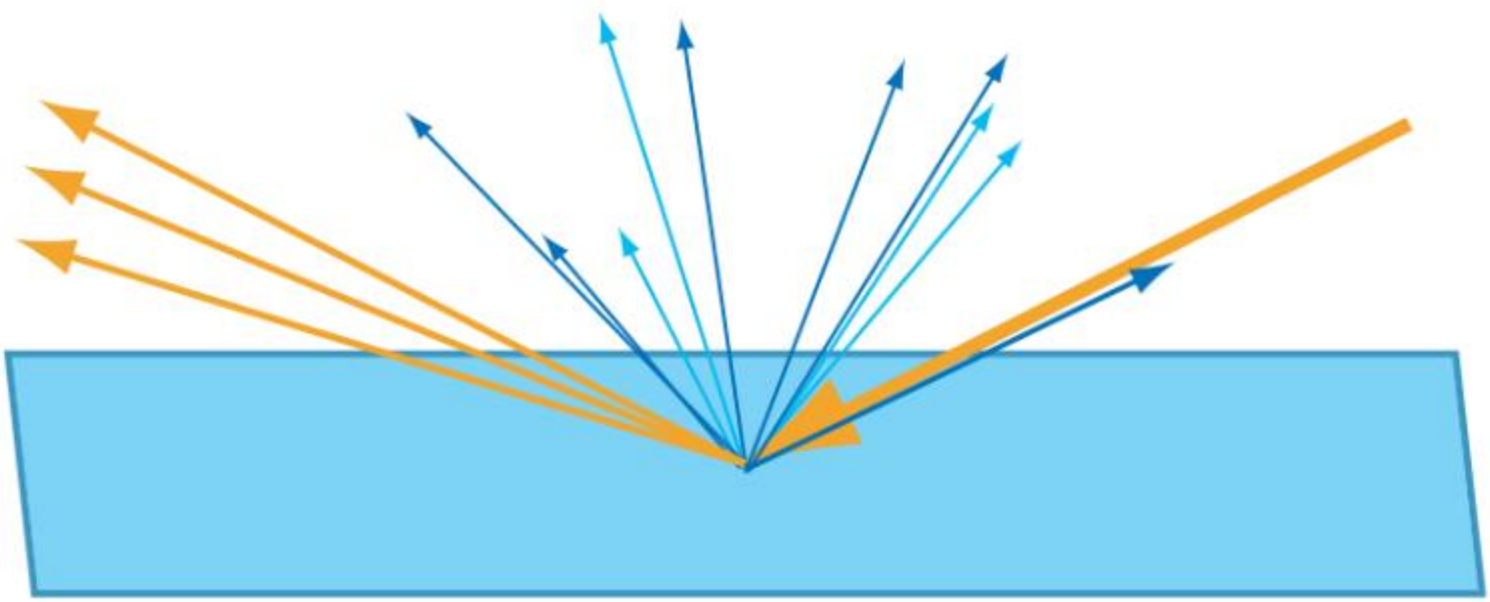
Non-metals

- Behave like regular participating media
  - Light is scattered (enough) are re-emited
  - Distance depends on particle densities
    - Assume = 0 if shading area (surface point) large enough
    - Must be taken into account otherwise



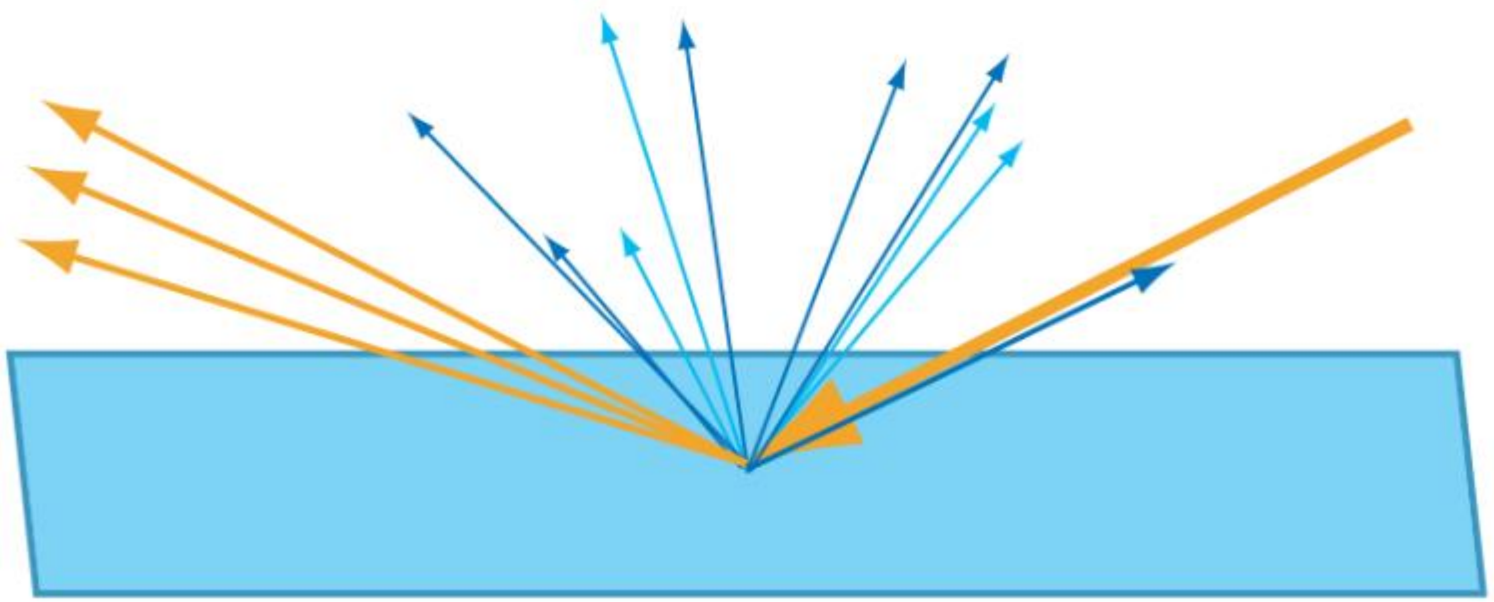
Physics → Maths

- Radiance
  - radiometric quantity used mesure the amount of light along a single ray
  - Spectral quantity (RGB in practice), Watt per steradian per square meter



Physics → Maths

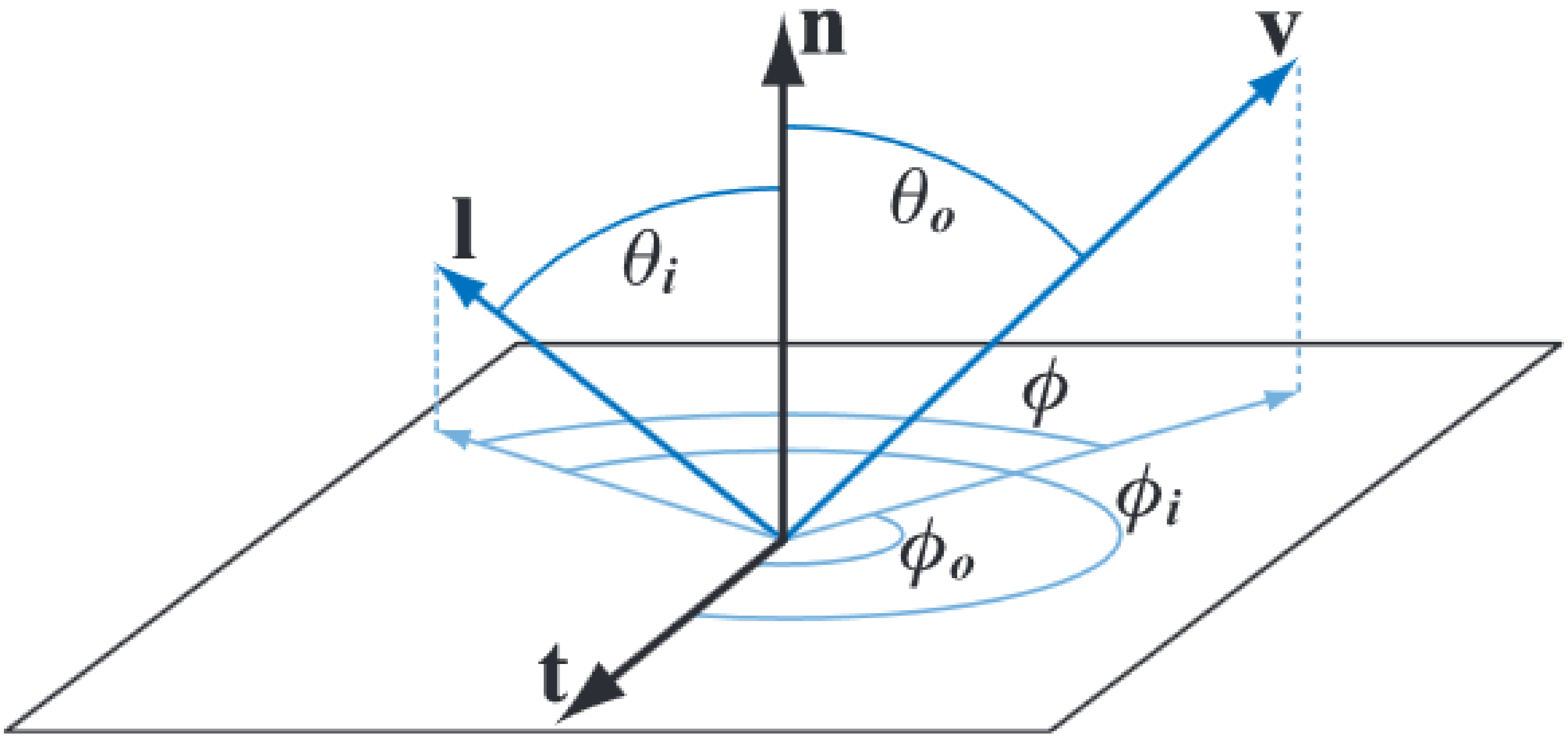
- Radiance
  - radiometric quantity used mesure the amount of light along a single ray
  - Spectral quantity (RGB in practice), Watt per steradian per square meter
- If shading can be handled locally, light response depends on
  - Light direction
  - View direction



Physics → Maths

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

$$f(\mathbf{l}, \mathbf{v})$$



Physics → Maths

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

$$f(\mathbf{l}, \mathbf{v})$$

$$L_o(\mathbf{v}) = \int_\Omega f(\mathbf{l}, \mathbf{v}) \otimes L_i(\mathbf{l})(\mathbf{n} \cdot \mathbf{l}) d\omega_i$$

Reflectance equation

Physics → Maths

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

$$f(\mathbf{l}, \mathbf{v})$$

$$L_o(\mathbf{v}) = \int_\Omega f(\mathbf{l}, \mathbf{v}) \otimes L_i(\mathbf{l})(\mathbf{n} \cdot \mathbf{l}) d\omega_i$$

Outgoing radiance

Reflectance equation

Physics → Maths

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

$$f(\mathbf{l}, \mathbf{v})$$

$$L_o(\mathbf{v}) = \int_\Omega f(\mathbf{l}, \mathbf{v}) \otimes L_i(\mathbf{l})(\mathbf{n} \cdot \mathbf{l}) d\omega_i$$

Outgoing radiance

Ingoing radiance

Reflectance equation

Physics → Maths

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

$$f(\mathbf{l}, \mathbf{v})$$

$$L_o(\mathbf{v}) = \int_\Omega f(\mathbf{l}, \mathbf{v}) \otimes L_i(\mathbf{l})(\mathbf{n} \cdot \mathbf{l}) d\omega_i$$

Outgoing radiance

Ingoing radiance

Surface orientation

Reflectance equation

Physics → Maths

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

$$f(\mathbf{l}, \mathbf{v})$$

$$L_o(\mathbf{v}) = \int_\Omega f(\mathbf{l}, \mathbf{v}) \otimes L_i(\mathbf{l})(\mathbf{n} \cdot \mathbf{l}) d\omega_i$$

Outgoing radiance

BRDF

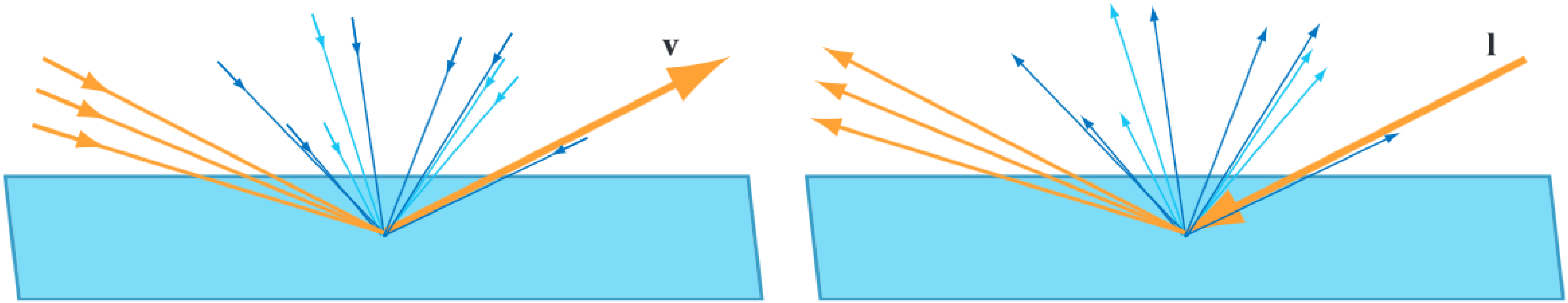
Ingoing radiance

Surface orientation

Reflectance equation

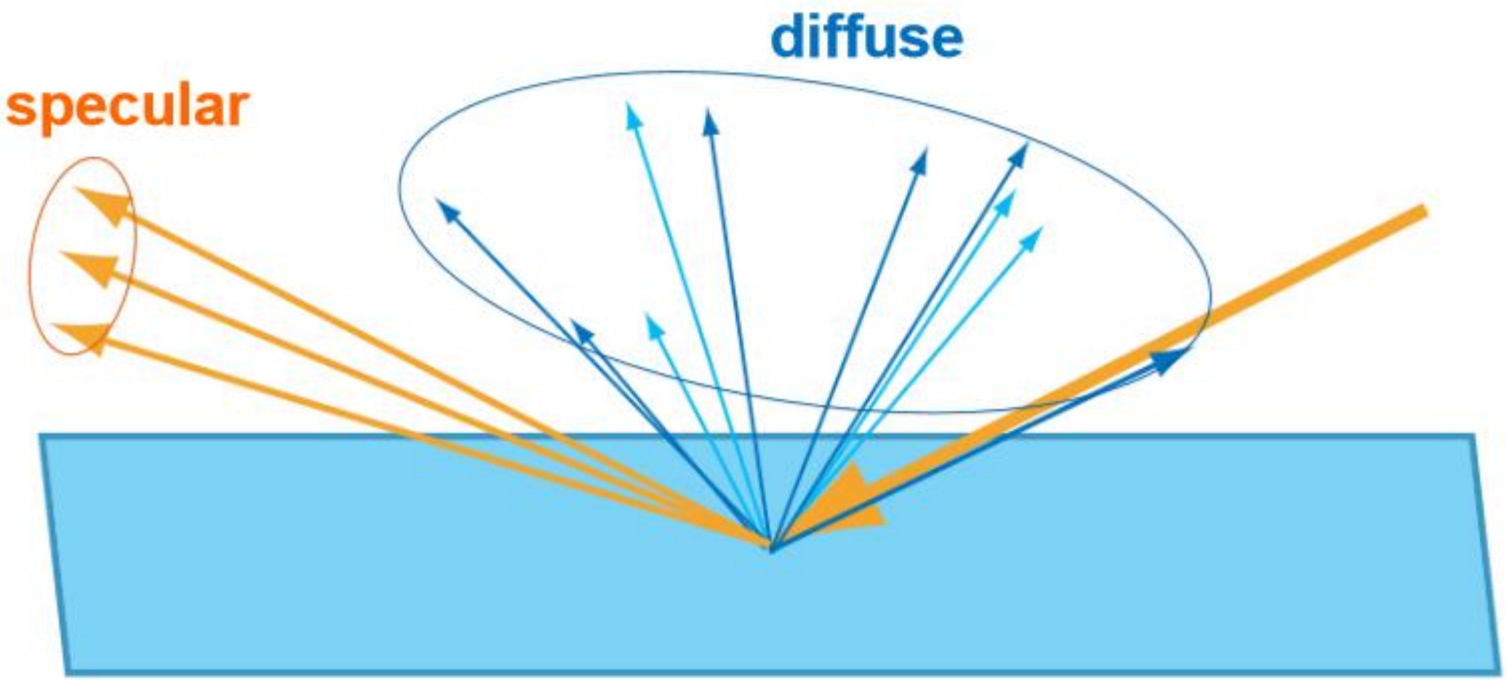
BRDF: intuition

- 2 possible interpretations
  - Given outgoing view → relative contributions of incoming light
  - Given incoming light direction → distribution of outgoing light



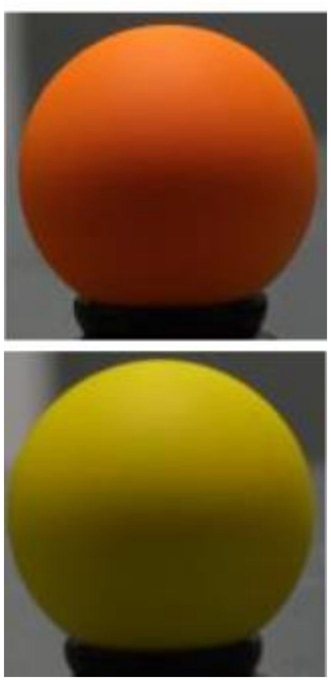
BRDF: intuition

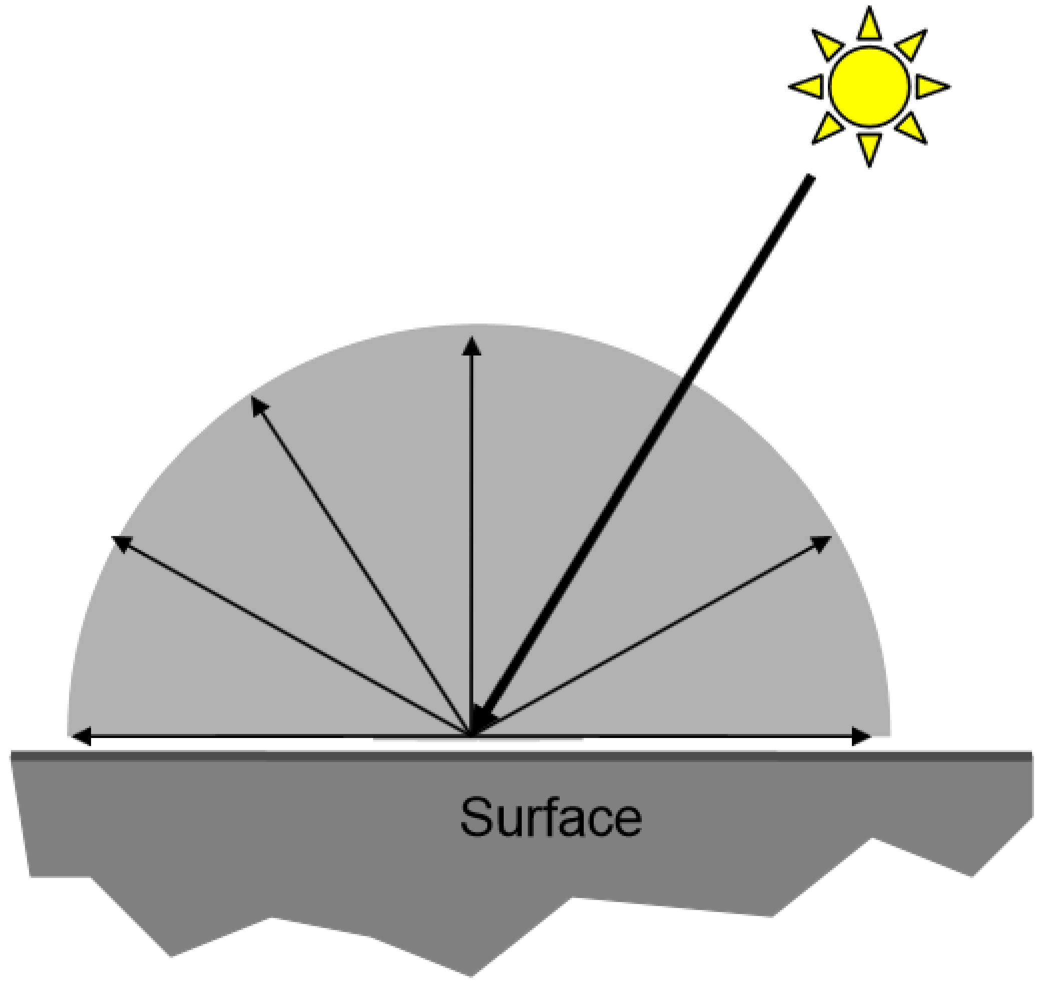
- Phenomena handled separatly:
  - Diffuse term
  - Specular term



BRDF: intuition

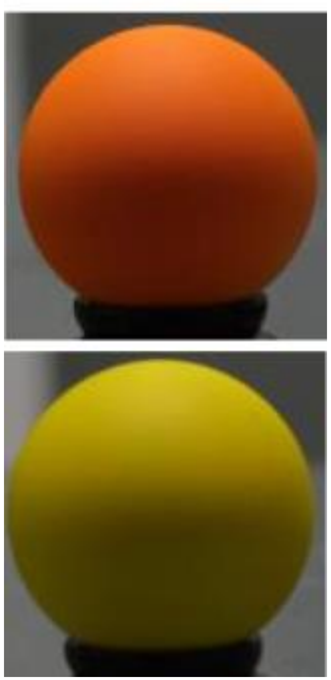
- Ideal diffuse reflectance (matte materials)
  - Assume surface reflects equally in all directions

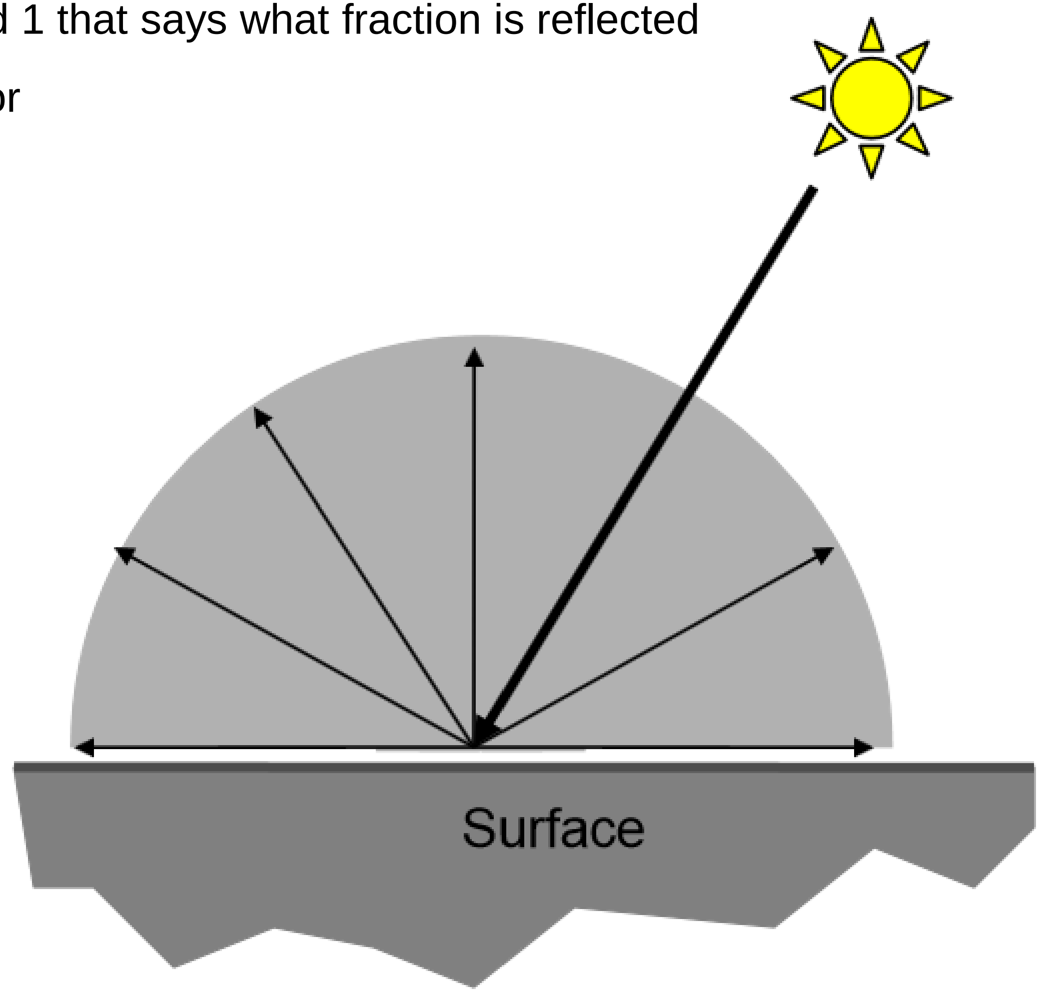




BRDF: intuition

- Ideal diffuse reflectance (matte materials)
  - Assume surface reflects equally in all directions
  - Coefficient between 0 and 1 that says what fraction is reflected
  - Usually called diffuse color

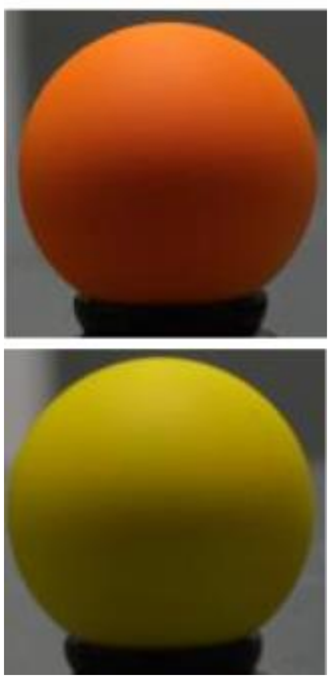


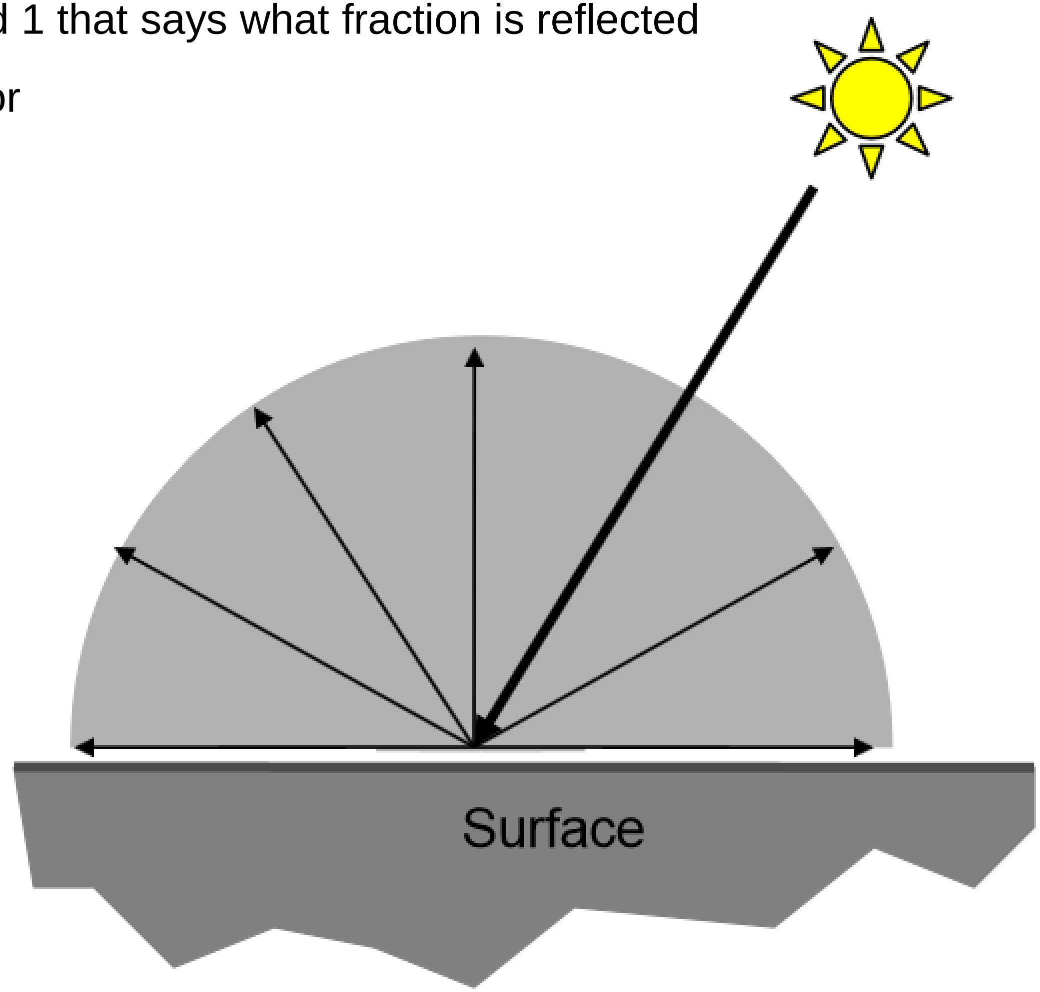


BRDF: intuition

- Ideal diffuse reflectance (matte materials)
  - Assume surface reflects equally in all directions
  - Coefficient between 0 and 1 that says what fraction is reflected
  - Usually called diffuse color

$$f(\mathbf{l}, \mathbf{v}) = const$$

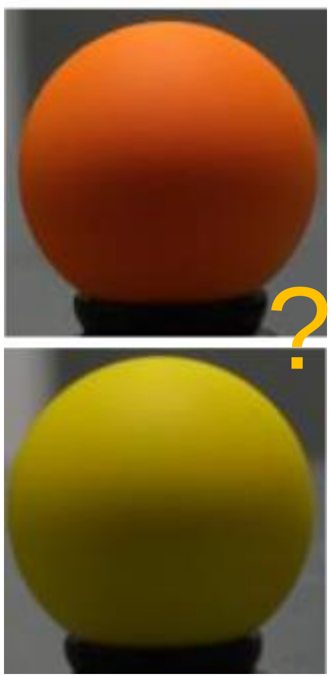




BRDF: intuition

- Ideal diffuse reflectance (matte materials)
  - Assume surface reflects equally in all directions
  - Coefficient between 0 and 1 that says what fraction is reflected
  - Usually called diffuse color

$$f(\mathbf{l}, \mathbf{v}) = const$$

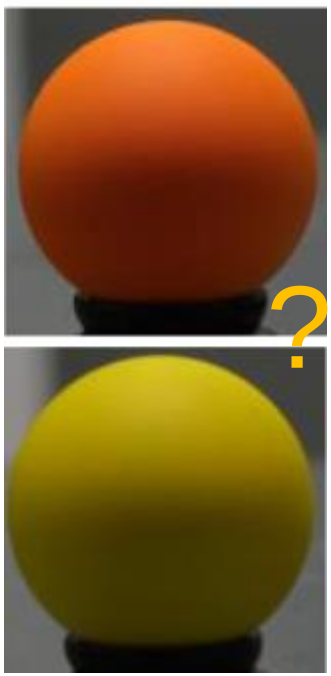


Why does color change?

BRDF: intuition

- Ideal diffuse reflectance (matte materials)
  - Assume surface reflects equally in all directions
  - Coefficient between 0 and 1 that says what fraction is reflected
  - Usually called diffuse color

$$f(\mathbf{l}, \mathbf{v}) = const$$



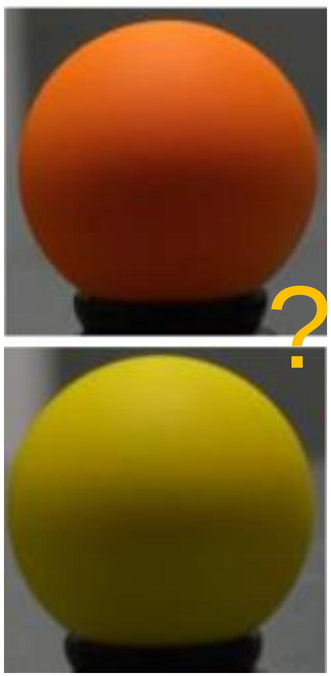
Why does color change?

$$L_o(\mathbf{v}) = \int_\Omega f(\mathbf{l}, \mathbf{v}) \otimes L_i(\mathbf{l}) \boxed{(\mathbf{n} \cdot \mathbf{l})} d\omega_i$$

BRDF: intuition

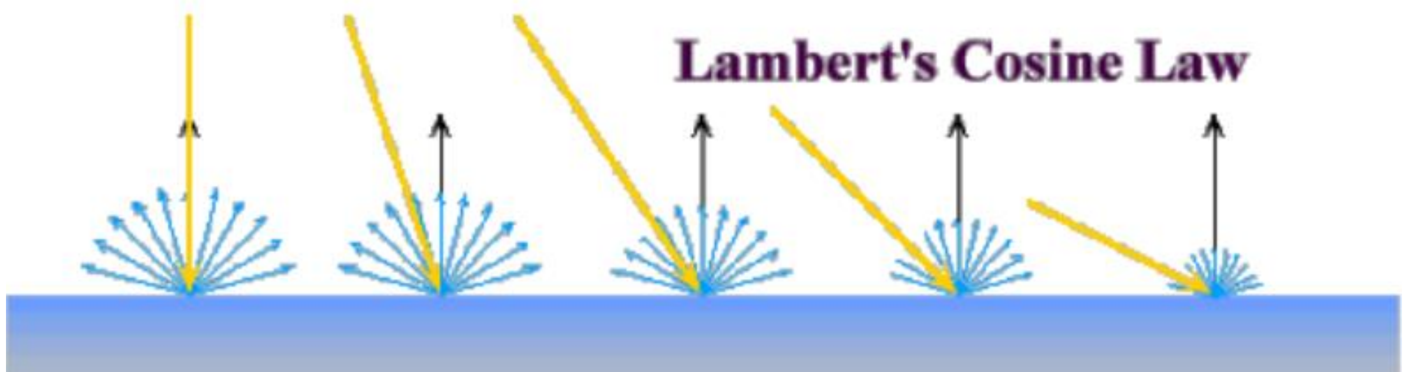
- Ideal diffuse reflectance (matte materials)
  - Assume surface reflects equally in all directions
  - Coefficient between 0 and 1 that says what fraction is reflected
  - Usually called diffuse color

$$f(\mathbf{l}, \mathbf{v}) = const$$



Why does color change?

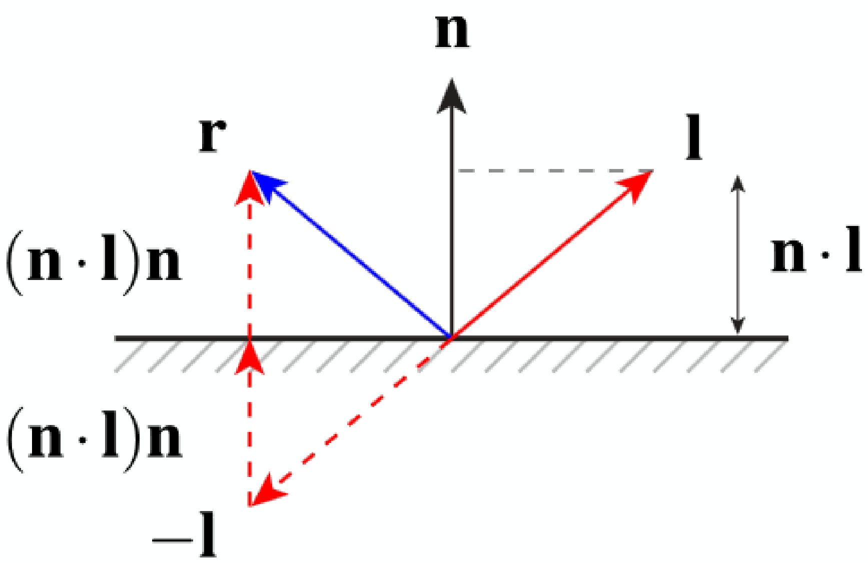
$$L_o(\mathbf{v}) = \int_\Omega f(\mathbf{l}, \mathbf{v}) \otimes L_i(\mathbf{l}) \boxed{(\mathbf{n} \cdot \mathbf{l})} d\omega_i$$



BRDF: intuition

- Ideal specular reflectance (mirror materials)
  - Delta dirac in the reflected direction
  - Not usefull for point lights… better for reflections of other surfaces

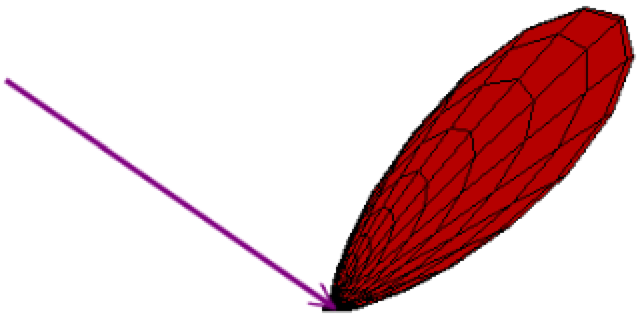
$$f(\mathbf{l}, \mathbf{v}) = dirac$$

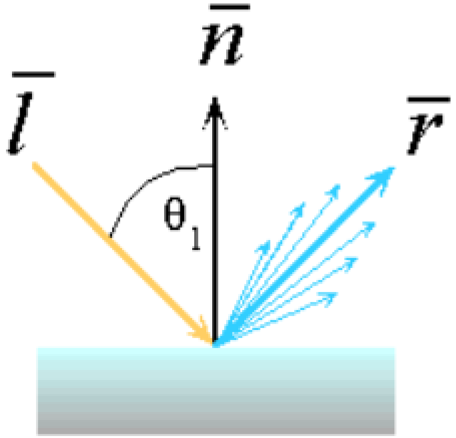


$$\mathbf{r} = 2(\mathbf{n}\cdot\mathbf{l})\mathbf{n} - \mathbf{l}$$

BRDF: intuition

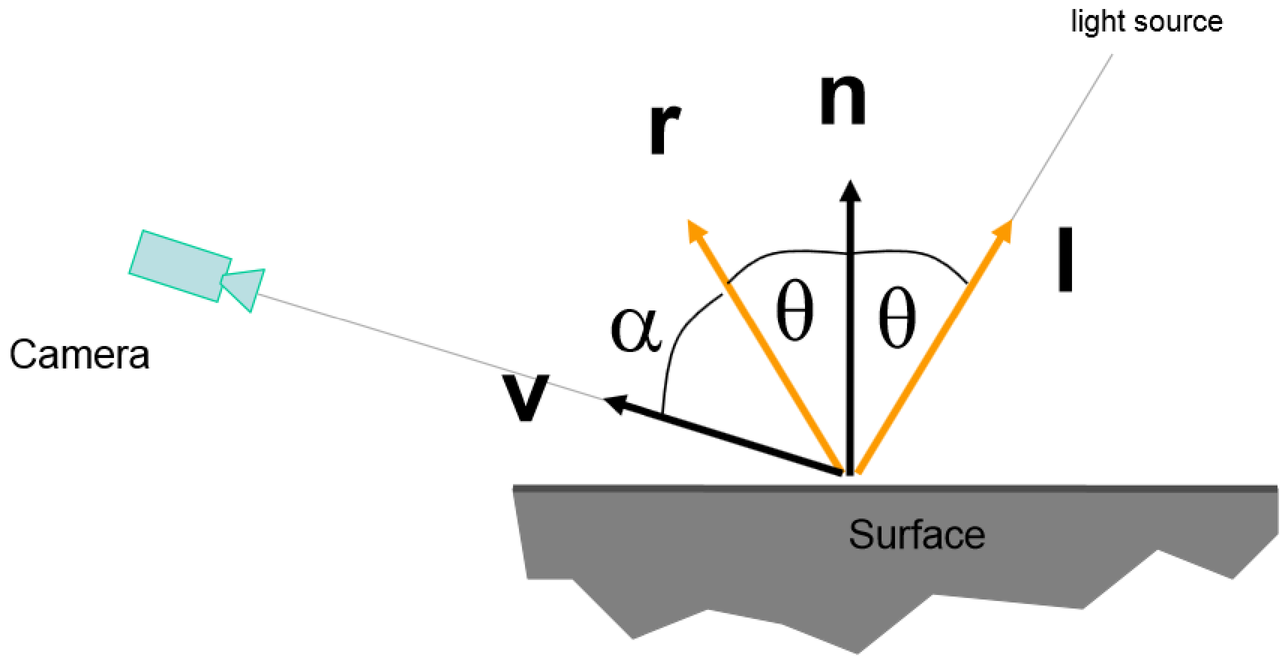
- Non ideal reflectors (glossy material)
  - Expect most of reflected light to travel in the direction of the ideal mirror ray
  - Some of the light should also be reflected slightly offsetted from the mirror ray
  - As we move farther and farther from the mirror ray, we expect to see less light reflected





BRDF: Phong model

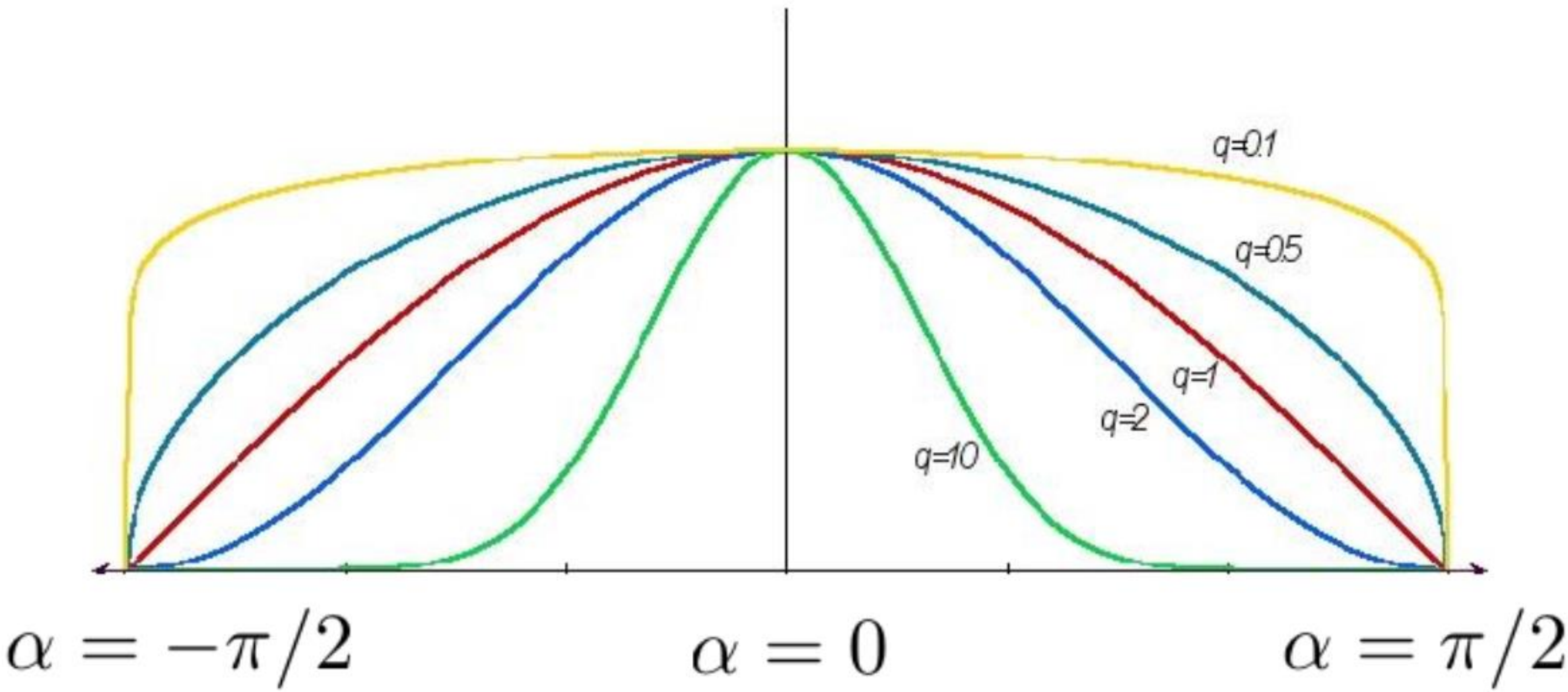
- Reflection depends on the angle between the ideal reflection and the view vectors



$$L_o = \; k_s\,(\cos\alpha)^q\,\frac{L_i}{r^2} \; = k_s\,(\boldsymbol{v}\cdot\boldsymbol{r})^q\,\frac{L_i}{r^2}$$

BRDF: Phong model

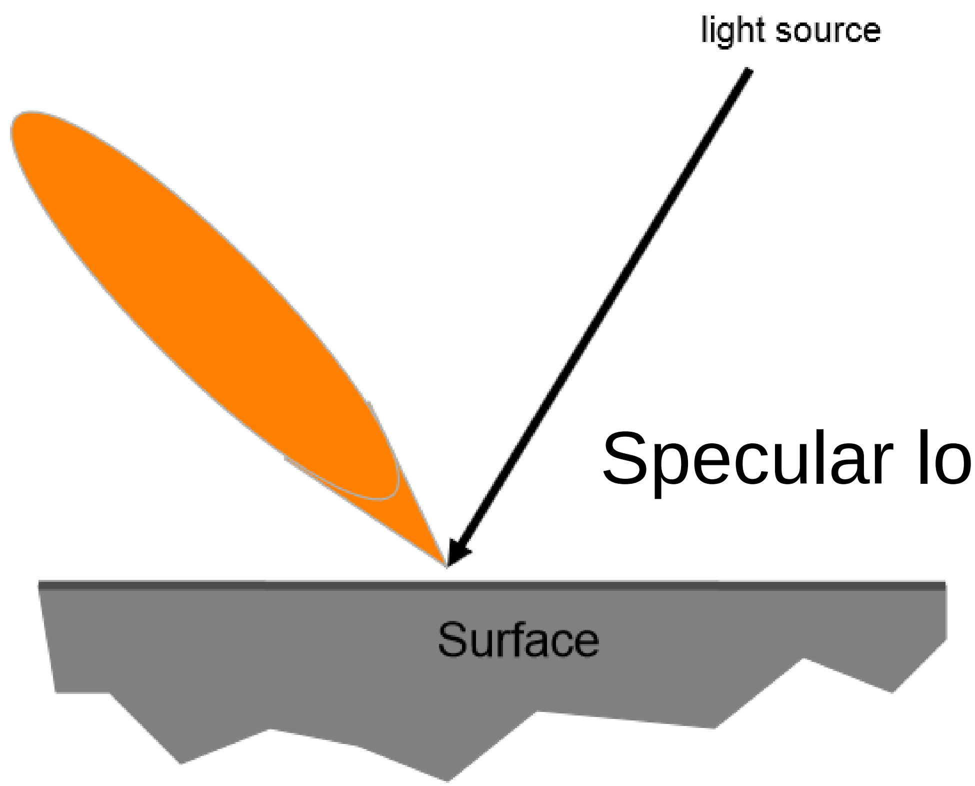
- Reflection depends on the angle between the ideal reflection and the view vectors



$$L_o = k_s (\cos\alpha)^q \frac{L_i}{r^2} = k_s (\boldsymbol{v} \cdot \boldsymbol{r})^q \frac{L_i}{r^2}$$

BRDF: Phong model

- Reflection depends on the angle between the ideal reflection and the view vectors

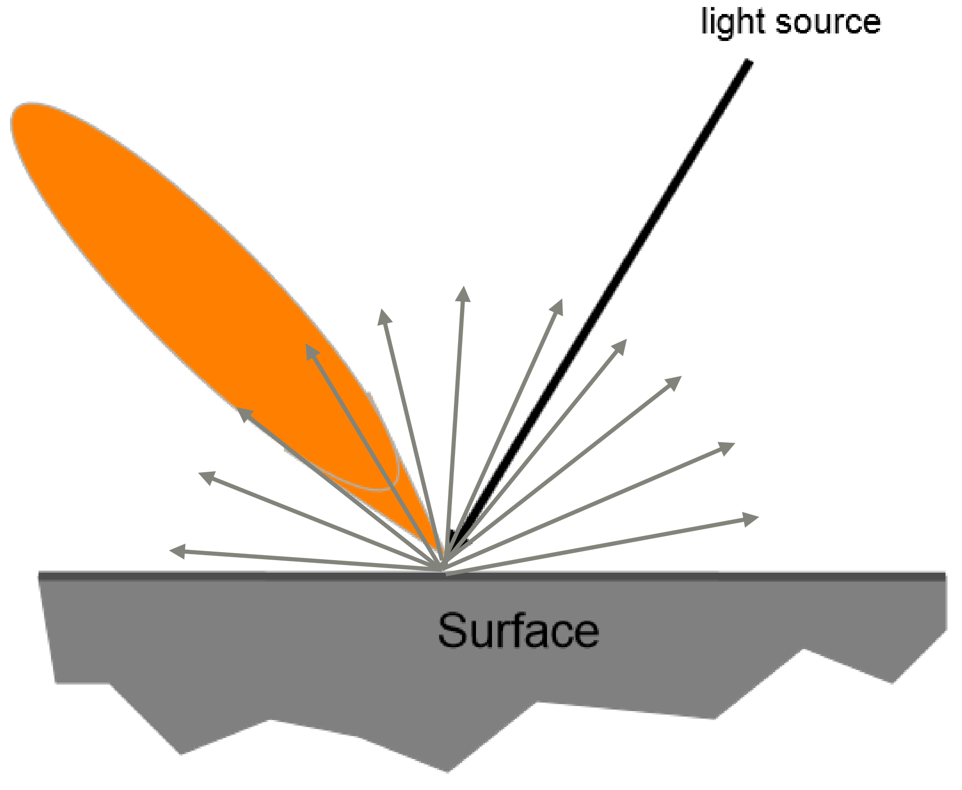


Specular lobe $(\boldsymbol{r} \cdot \boldsymbol{v})^q$

$$L_o = \; k_s \, (\cos\alpha)^q \, \frac{L_i}{r^2} \; = k_s \, (\boldsymbol{v} \cdot \boldsymbol{r})^q \, \frac{L_i}{r^2}$$

BRDF: Phong model

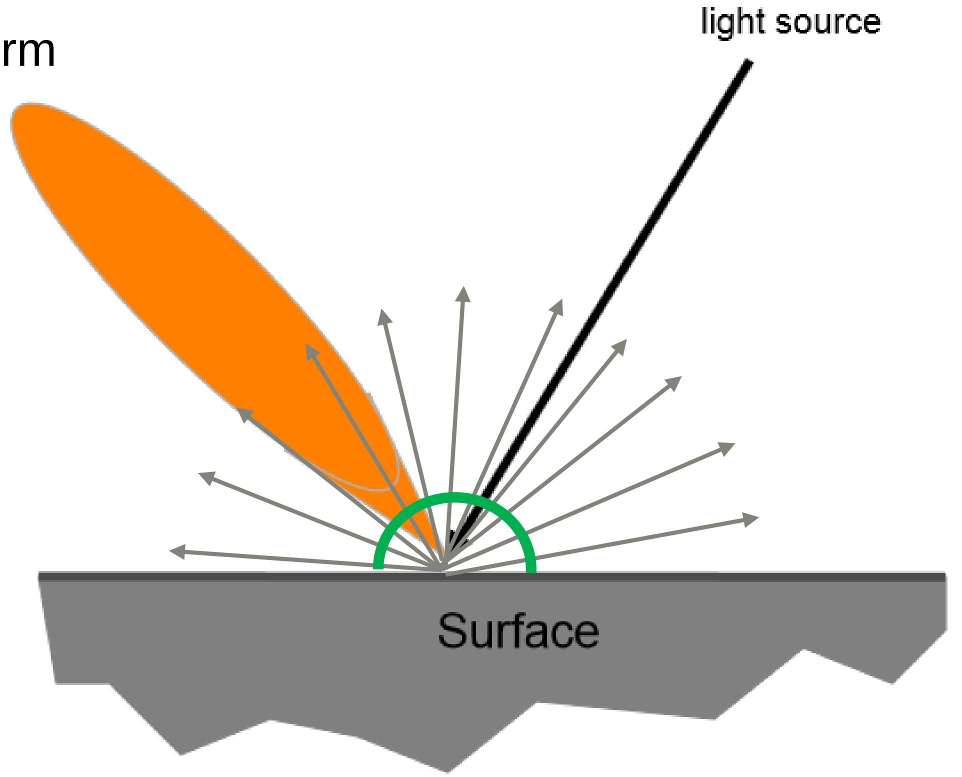
- Reflection depends on the angle between the ideal reflection and the view vectors
- + ideal diffuse reflection



$$L_o = \left[k_d \left(\boldsymbol{n} \cdot \boldsymbol{l} \right) + k_s \left(\boldsymbol{v} \cdot \boldsymbol{r} \right)^q \right] \frac{L_i}{r^2}$$

BRDF: Phong model

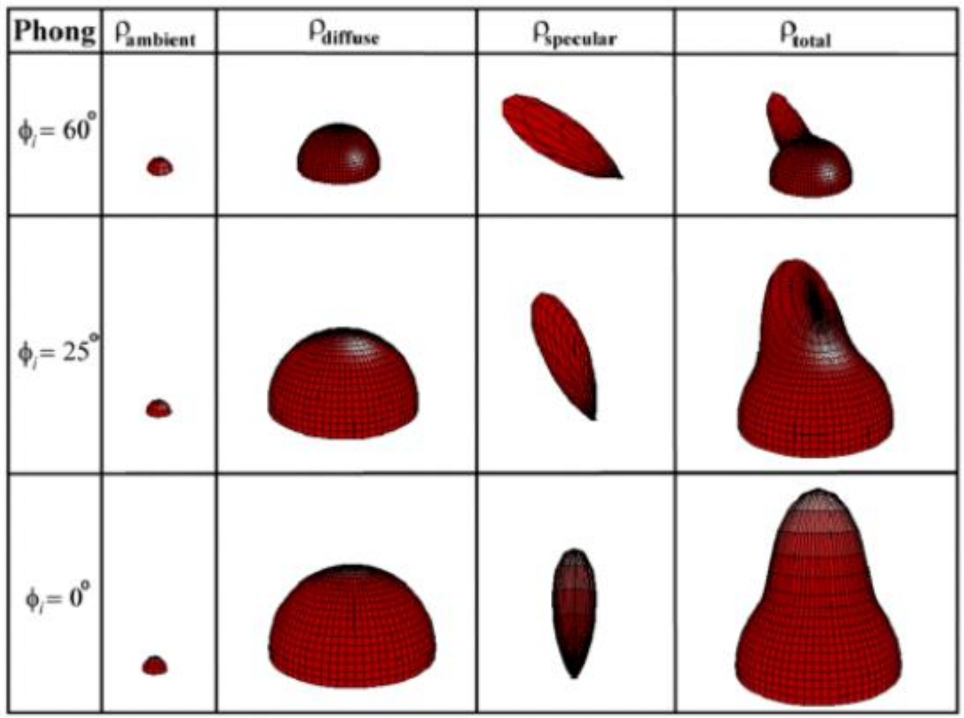
- Reflection depends on the angle between the ideal reflection and the view vectors
- + ideal diffuse reflection
- + ambient term



$$L_o = \left[k_a + k_d\left(\boldsymbol{n}\cdot\boldsymbol{l}\right) + k_s\left(\boldsymbol{v}\cdot\boldsymbol{r}\right)^q\right]\frac{L_i}{r^2}$$

BRDF: Phong model

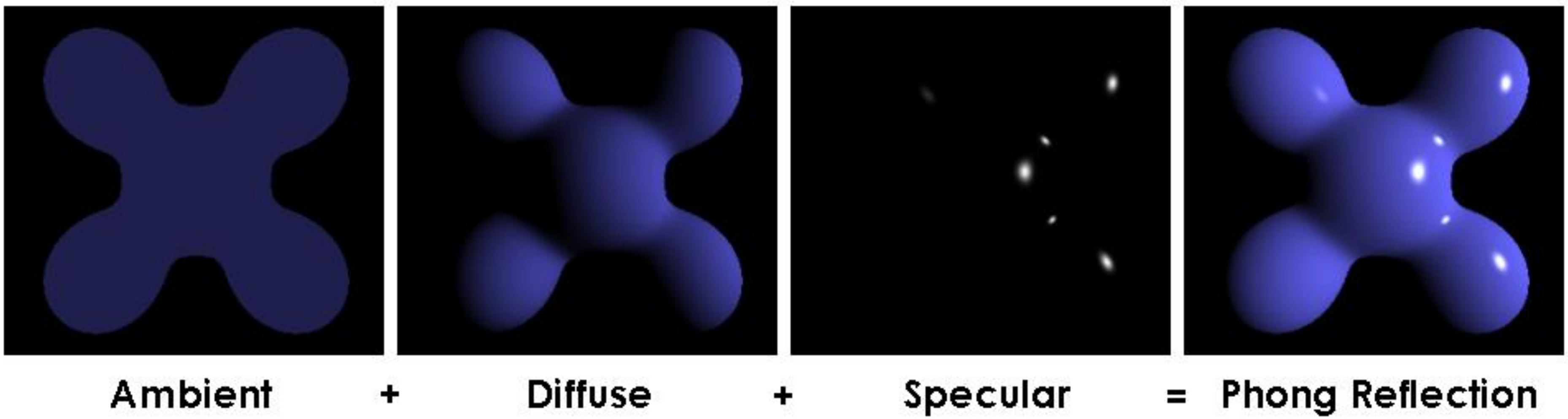
- Reflection depends on the angle between the ideal reflection and the view vectors
- + ideal diffuse reflection
- + ambient term



$$L_o = \left[k_a + k_d \left(\boldsymbol{n} \cdot \boldsymbol{l} \right) + k_s \left(\boldsymbol{v} \cdot \boldsymbol{r} \right)^q \right] \frac{L_i}{r^2}$$

BRDF: Phong model

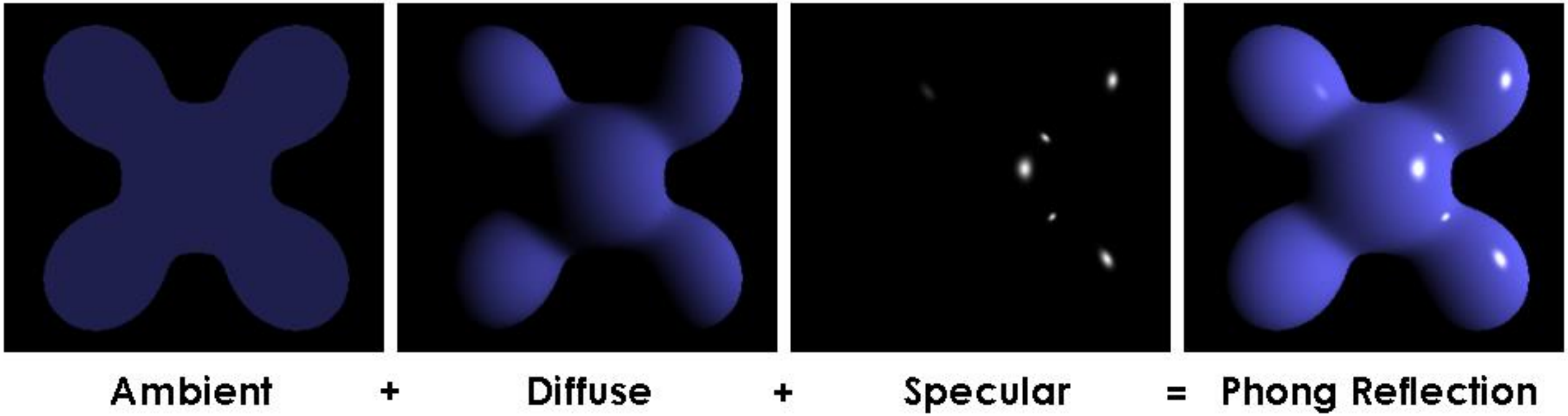
- Reflection depends on the angle between the ideal reflection and the view vectors
- + ideal diffuse reflection
- + ambient term



$$L_o = \left[k_a + k_d\,(\boldsymbol{n}\cdot\boldsymbol{l}) + k_s\,(\boldsymbol{v}\cdot\boldsymbol{r})^q\right]\frac{L_i}{r^2}$$

BRDF: Phong model

- Problems:
  - Does not conserve energy (may reflect more than it receives)
  - Not conform to BRDF model (cosine)
  - Ambient is a total hack



$$L_o = \left[k_a + k_d \left(\boldsymbol{n} \cdot \boldsymbol{l} \right) + k_s \left(\boldsymbol{v} \cdot \boldsymbol{r} \right)^q \right] \frac{L_i}{r^2}$$

Physically plausible BRDFs

$$L_o(\mathbf{v}) = \int_{\Omega} f(\mathbf{l}, \mathbf{v}) \otimes L_i(\mathbf{l})(\mathbf{n} \cdot \mathbf{l}) d\omega_i$$

Physically plausible BRDFs

$$L_o(\mathbf{v}) = \int_\Omega f(\mathbf{l},\mathbf{v}) \otimes L_i(\mathbf{l})(\mathbf{n}\cdot\mathbf{l})d\omega_i$$

- Positivity $f(\mathbf{l},\mathbf{v}) >= 0$

Physically plausible BRDFs

$$L_o(\mathbf{v}) = \int_\Omega f(\mathbf{l},\mathbf{v}) \otimes L_i(\mathbf{l})(\mathbf{n}\cdot\mathbf{l})d\omega_i$$

- Positivity $f(\mathbf{l},\mathbf{v}) >= 0$
- Reciprocity $f(\mathbf{l},\mathbf{v}) = f(\mathbf{v},\mathbf{l})$

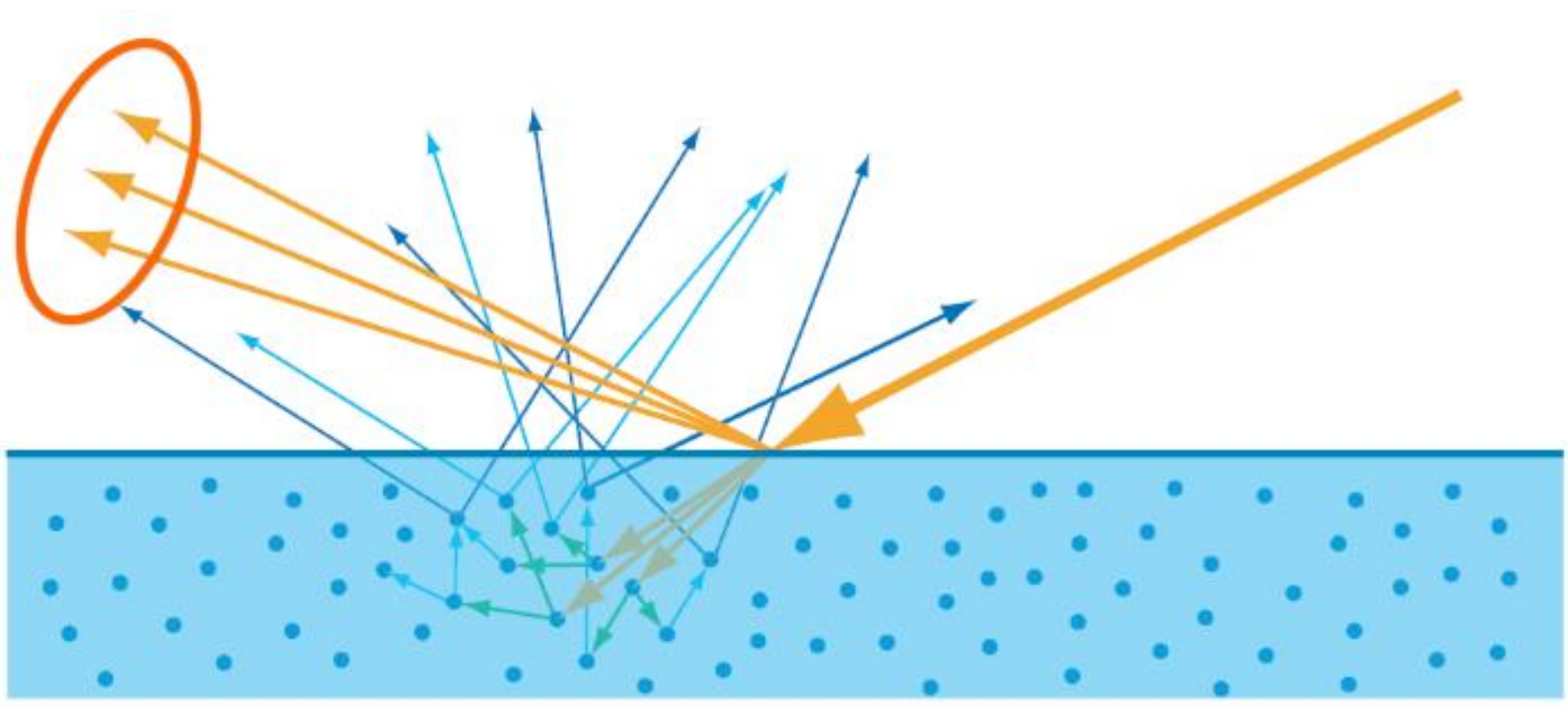
Physically plausible BRDFs

$$L_o(\mathbf{v}) = \int_\Omega f(\mathbf{l},\mathbf{v}) \otimes L_i(\mathbf{l})(\mathbf{n}\cdot\mathbf{l})d\omega_i$$

- Positivity $$f(\mathbf{l},\mathbf{v}) >= 0$$
- Reciprocity $$f(\mathbf{l},\mathbf{v}) = f(\mathbf{v},\mathbf{l})$$
- Energy conservation $$\forall \mathbf{l}, \int_\Omega f(\mathbf{l},\mathbf{v})(\mathbf{n}\cdot\mathbf{v})\, d\omega_o \leq 1$$

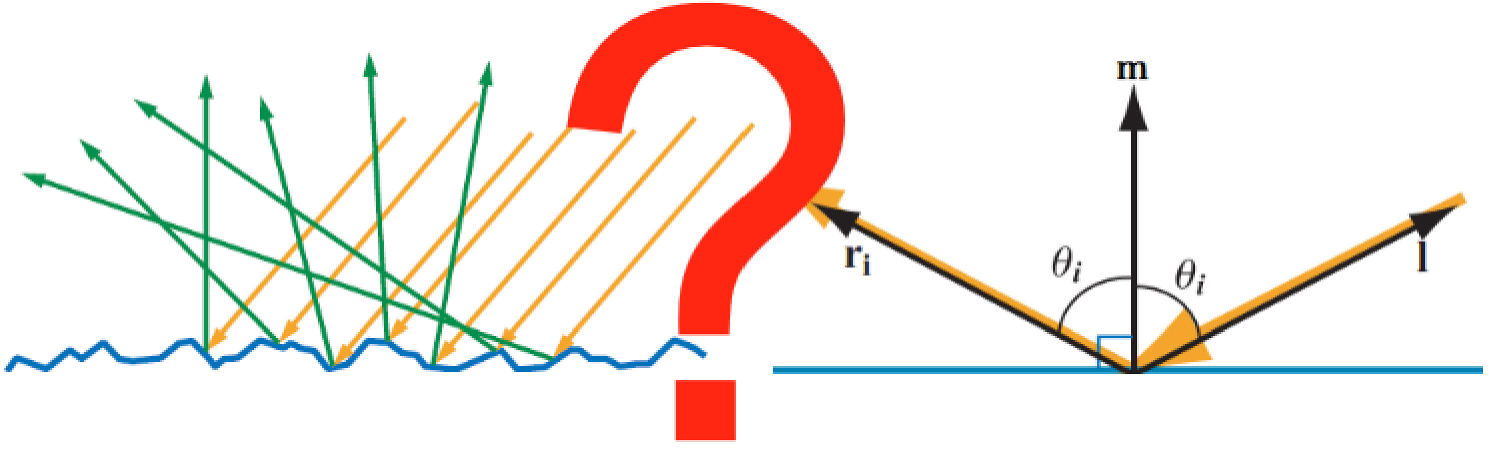
Microfacet theory

- Surface reflection (specular term)



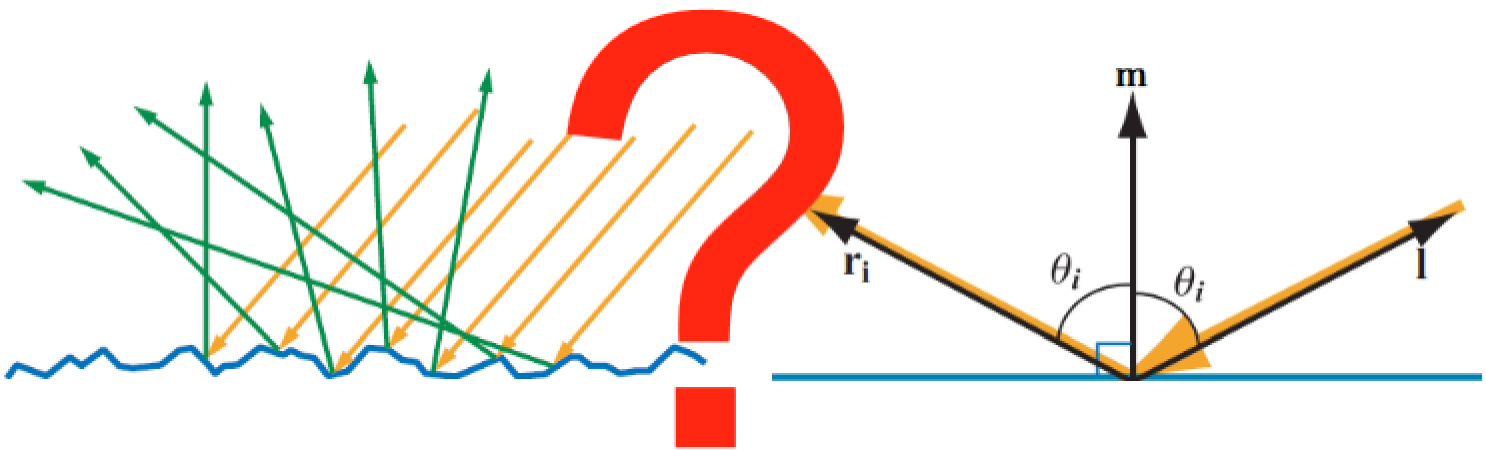
Microfacet theory

- Derive BRDF from non optically flat surfaces
  - Details too small to be visible
  - But large compared to light wavelength



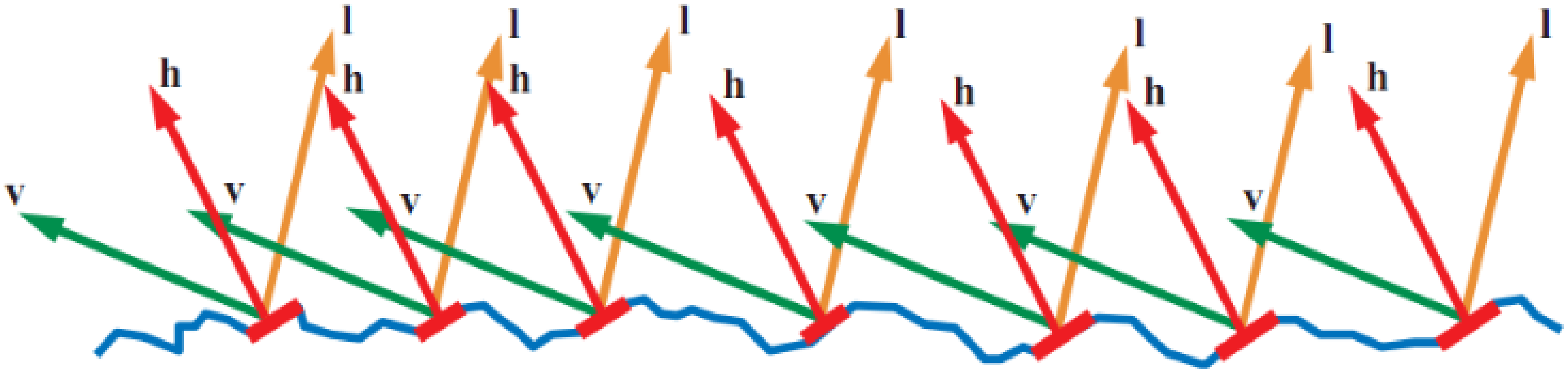
Microfacet theory

- Derive BRDF from non optically flat surfaces
  - Details too small to be visible
  - But large compared to light wavelength
- Each facet considered as a perfect mirror
  - Reflection depends on light direction and microfacet normal



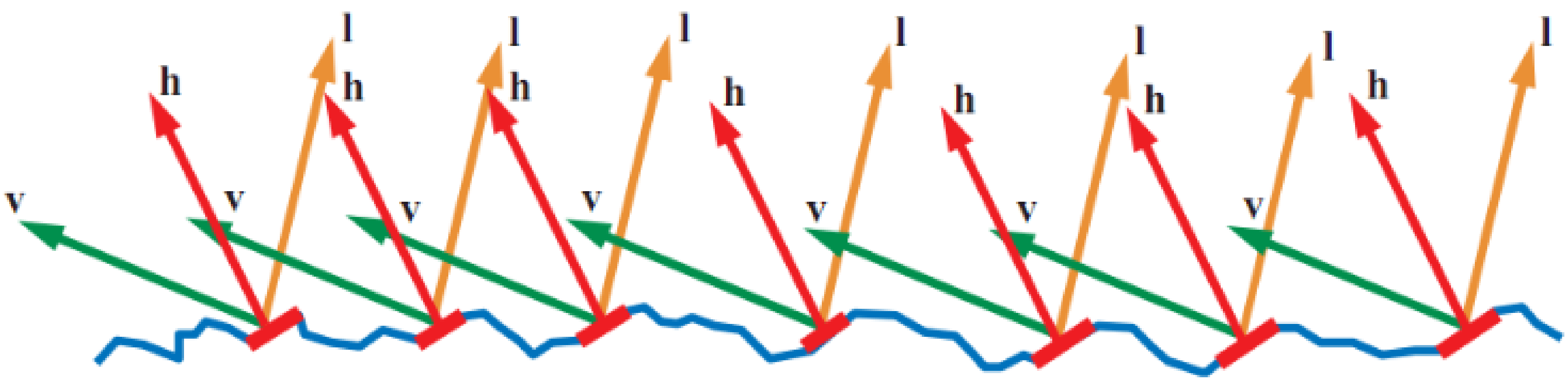
Microfacet theory

- Half vector
  - →microfact normal



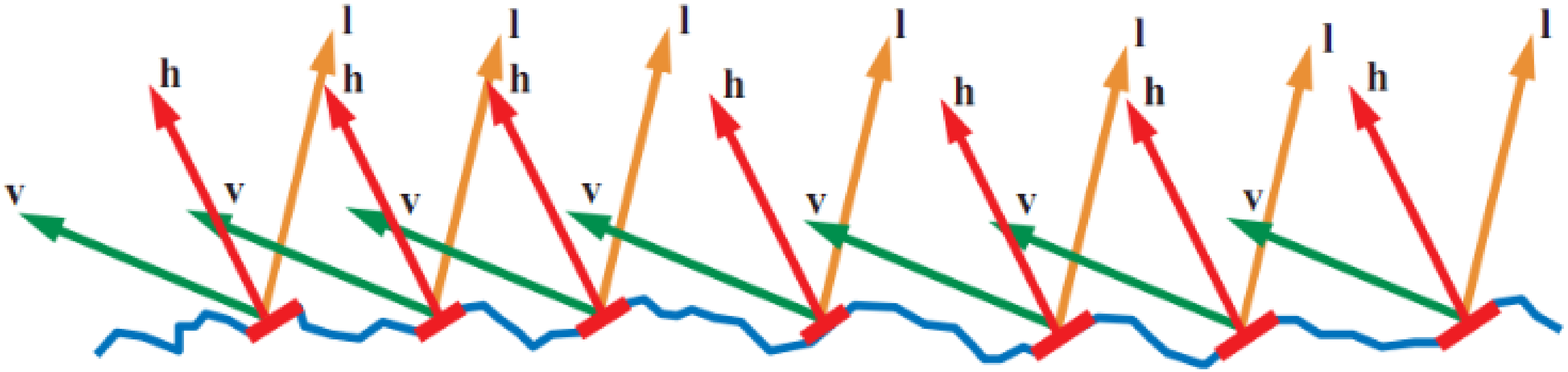
Microfacet theory

- Half vector
  - →microfact normal
  - →Only microfacets having their normals halfway between the view and light direction will reflect something!



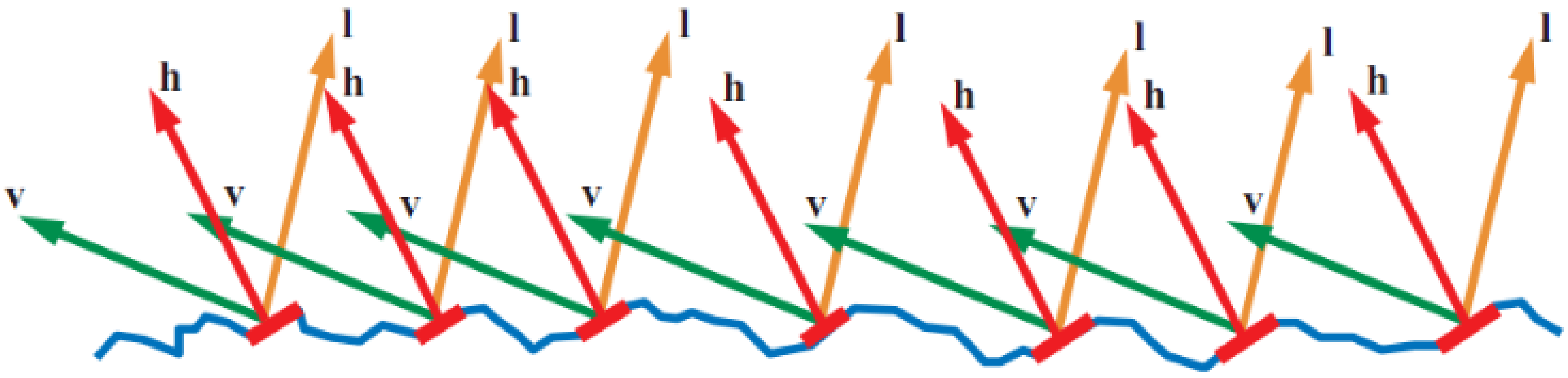
Microfacet theory

- Half vector
  - →microfact normal
  - →Only microfacets having their normals halfway between the view and light direction will reflect something!
  - →Parametrized by h: give me the percent number of facets having this orientation



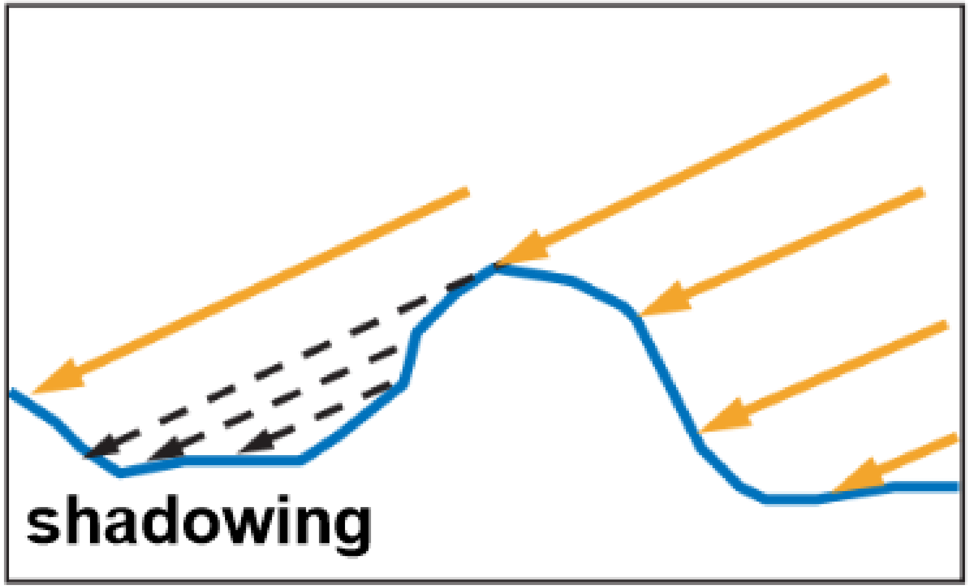
Microfacet theory

- Shadowing and masking
  - Not all microfacets oriented by a given h will contribute…
  - Some will be blocked by other microfacets from either



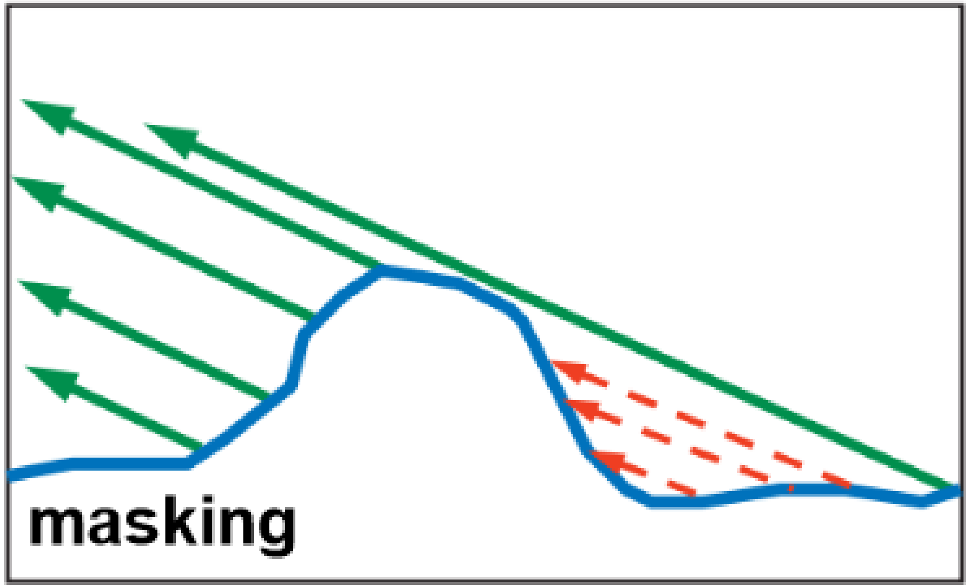
Microfacet theory

- Shadowing and masking
  - Not all microfacets oriented by a given h will contribute…
  - Some will be blocked by other microfacets from either
    - The light direction (shadowing)



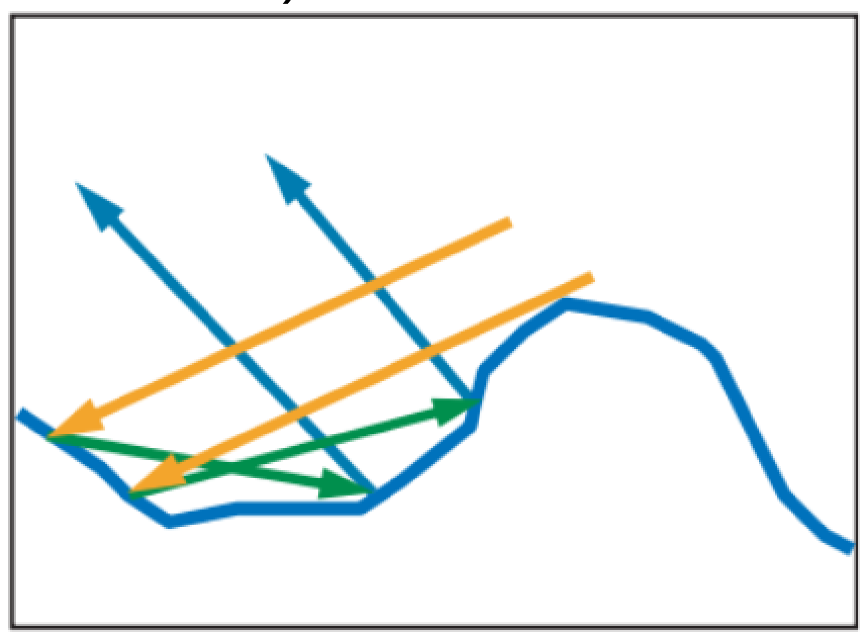
Microfacet theory

- Shadowing and masking
  - Not all microfacets oriented by a given h will contribute…
  - Some will be blocked by other microfacets from either
    - The light direction (shadowing)
    - The view direction (masking)



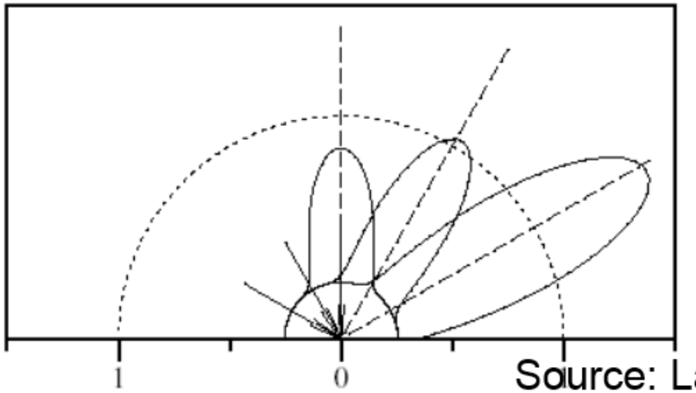
Microfacet theory

- Shadowing and masking
  - Not all microfacets oriented by a given h will contribute…
  - Some will be blocked by other microfacets from either
    - The light direction (shadowing)
    - The view direction (masking)
  - Not completely true (interreflections)
    - Microfacet limitation…



Microfacet theory

- Fresnel effect
  - Increase specularity near grazing angles



Source: Lafortune et al. 97

Microfacet theory

- Summary
  - Fresnel effect

$$f(\mathbf{l}, \mathbf{v}) = \quad F(\mathbf{l}, \mathbf{h})$$

Microfacet theory

- Summary
  - Fresnel effect
  - Masking shadowing

$$f(\mathbf{l}, \mathbf{v}) = \frac{F(\mathbf{l}, \mathbf{h}) G(\mathbf{l}, \mathbf{v}, \mathbf{h})}{}$$

Microfacet theory

- Summary
  - Fresnel effect
  - Masking shadowing
  - Amount of microfacets at a particular orientation

$$f(\mathbf{l}, \mathbf{v}) = F(\mathbf{l}, \mathbf{h})G(\mathbf{l}, \mathbf{v}, \mathbf{h})D(\mathbf{h})$$

Microfacet theory

- Summary
  - Fresnel effect
  - Masking shadowing
  - Amount of microfacets at a particular orientation

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

correction factor for quantities being transformed between the microgeometry local space and the overall macrosurface

→ forshortening

Microfacet theory

- Summary (cook-terrance model)
  - Fresnel effect
  - Masking shadowing
  - Amount of microfacets at a particular orientation

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

correction factor for quantities being transformed between the microgeometry local space and the overall macrosurface

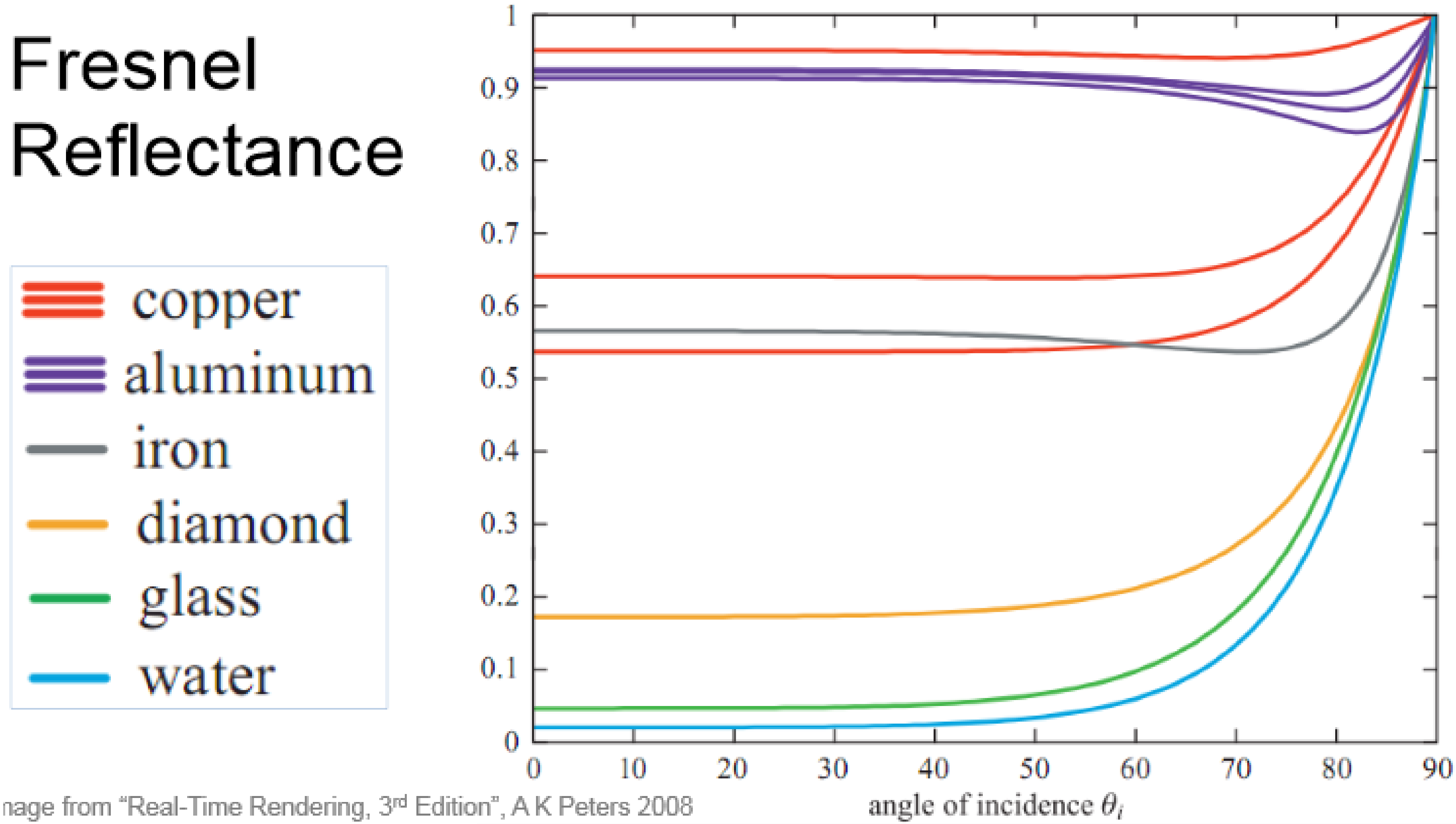
Fresnel reflectance

- Fraction of incoming light that is reflected
- In this case:
  - How much of the light hitting the relevant microfacets is reflected

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

Fresnel reflectance

- Fraction of incoming light that is reflected



nage from “Real-Time Rendering, 3rd Edition”, A K Peters 2008

Fresnel reflectance

- Fraction of incoming light that is reflected

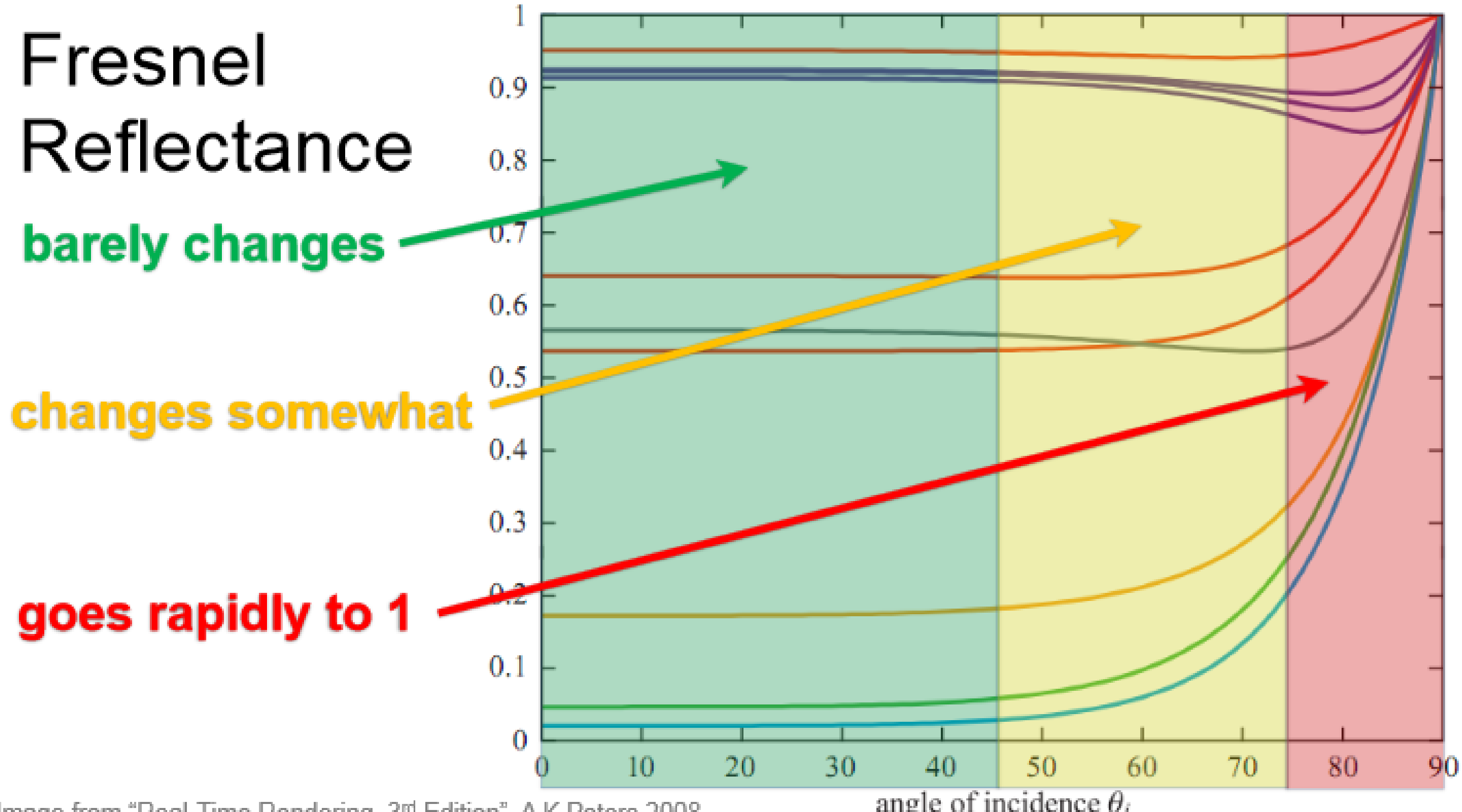


Image from "Real-Time Rendering, 3rd Edition", A K Peters 2008

Fresnel reflectance

- Fraction of incoming light that is reflected

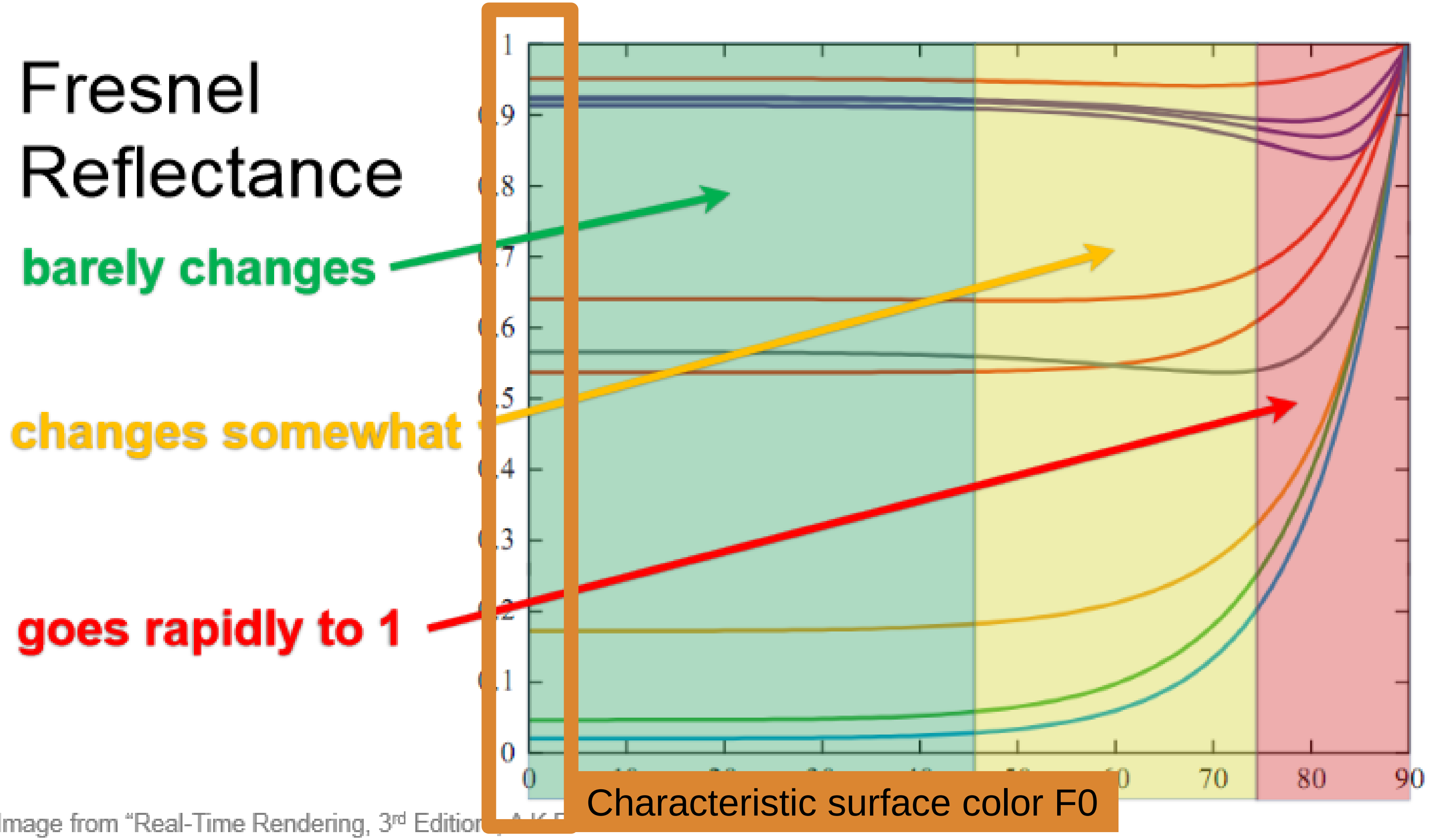
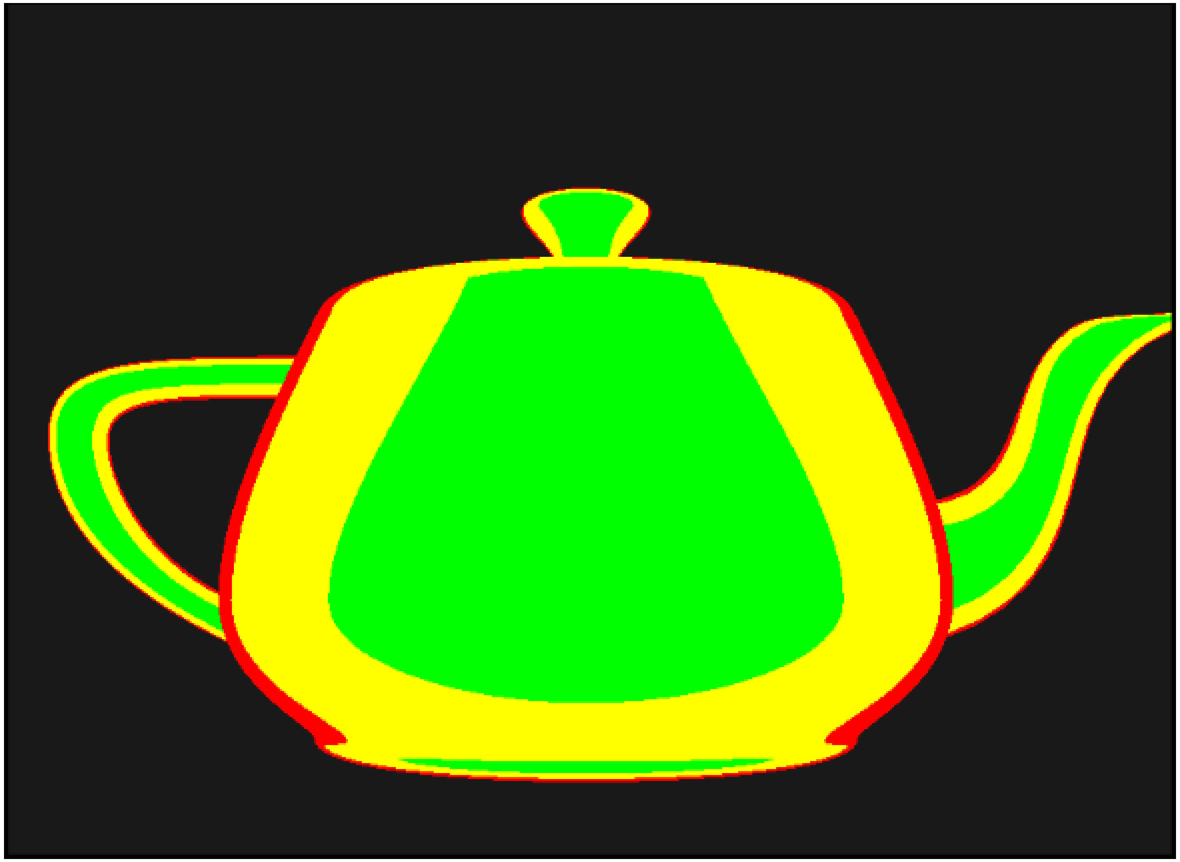


Image from "Real-Time Rendering, 3rd Edition", A K P

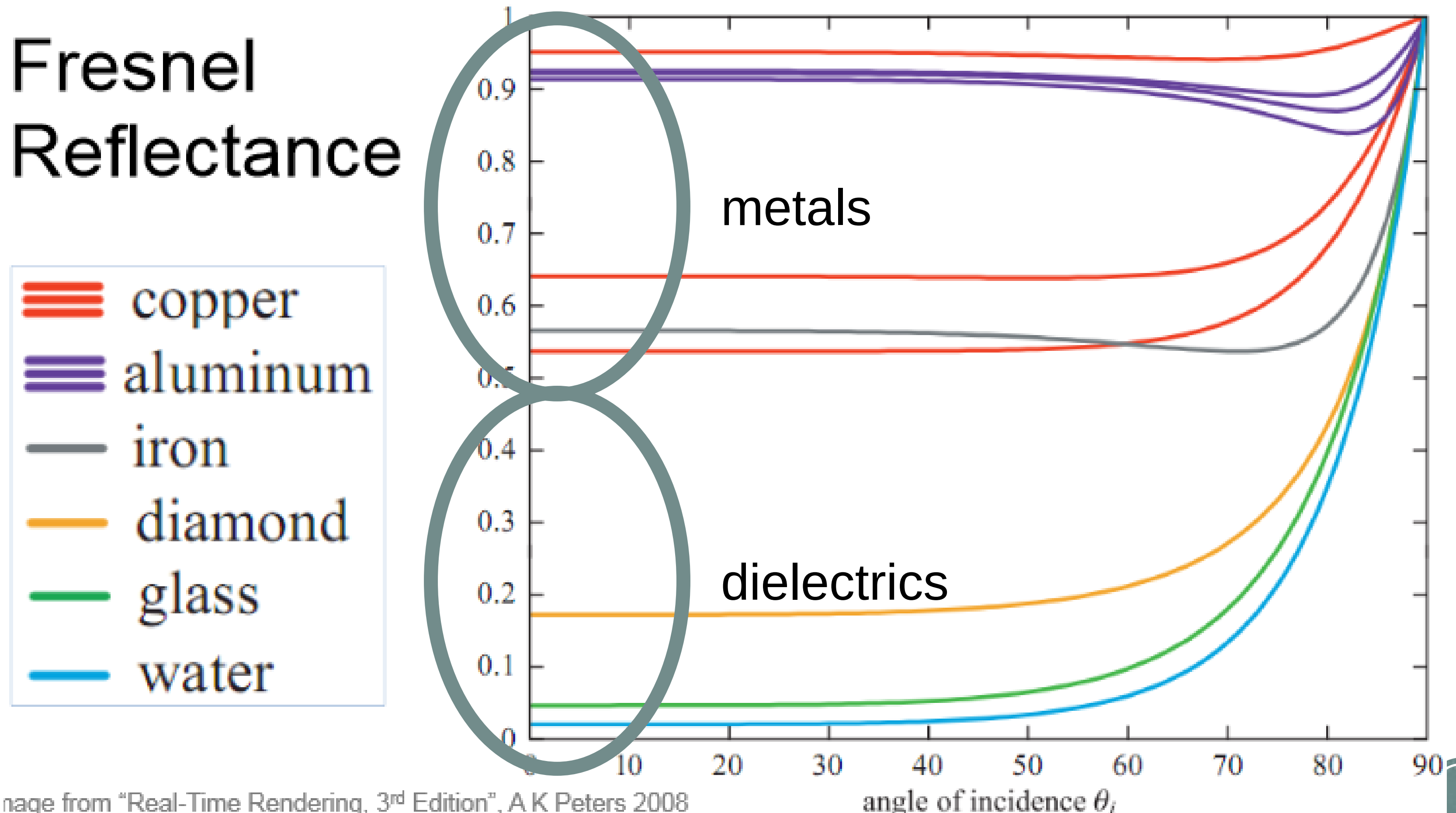
Fresnel reflectance

- Fraction of incoming light that is reflected
  - Mainly affect edges



Fresnel reflectance

- Fraction of incoming light that is reflected
  - Mainly affect edges



nage from “Real-Time Rendering, 3rd Edition”, A K Peters 2008

Fresnel reflectance

- Fraction of incoming light that is reflected
  - Mainly affect edges
- Schlick approximation
  - Accurate, cheap and parametrized by F0

$$F_{\text{Schlick}}(F_0, \mathbf{l}, \mathbf{n}) = F_0 + (1 - F_0)(1 - (\mathbf{l} \cdot \mathbf{n}))^5$$

  - For microfacet models

$$F_{\text{Schlick}}(F_0, \mathbf{l}, \mathbf{h}) = F_0 + (1 - F_0)(1 - (\mathbf{l} \cdot \mathbf{h}))^5$$

Normal Distribution Function

- Statistical distribution of orientation h
  - Determine size, brightness and shape of specular highlight

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

Normal Distribution Function

- Statistical distribution of orientation h
  - Determine size, brightness and shape of specular highlight

$$D_p(\mathbf{m}) = \frac{\alpha_p + 2}{2\pi}(\mathbf{n}\cdot\mathbf{m})^{\alpha_p}$$

$$D_{uabc}(\mathbf{m}) = \frac{1}{(1 + \alpha_{abc1}(1 - (\mathbf{n}\cdot\mathbf{m})))^{\alpha_{abc2}}}$$

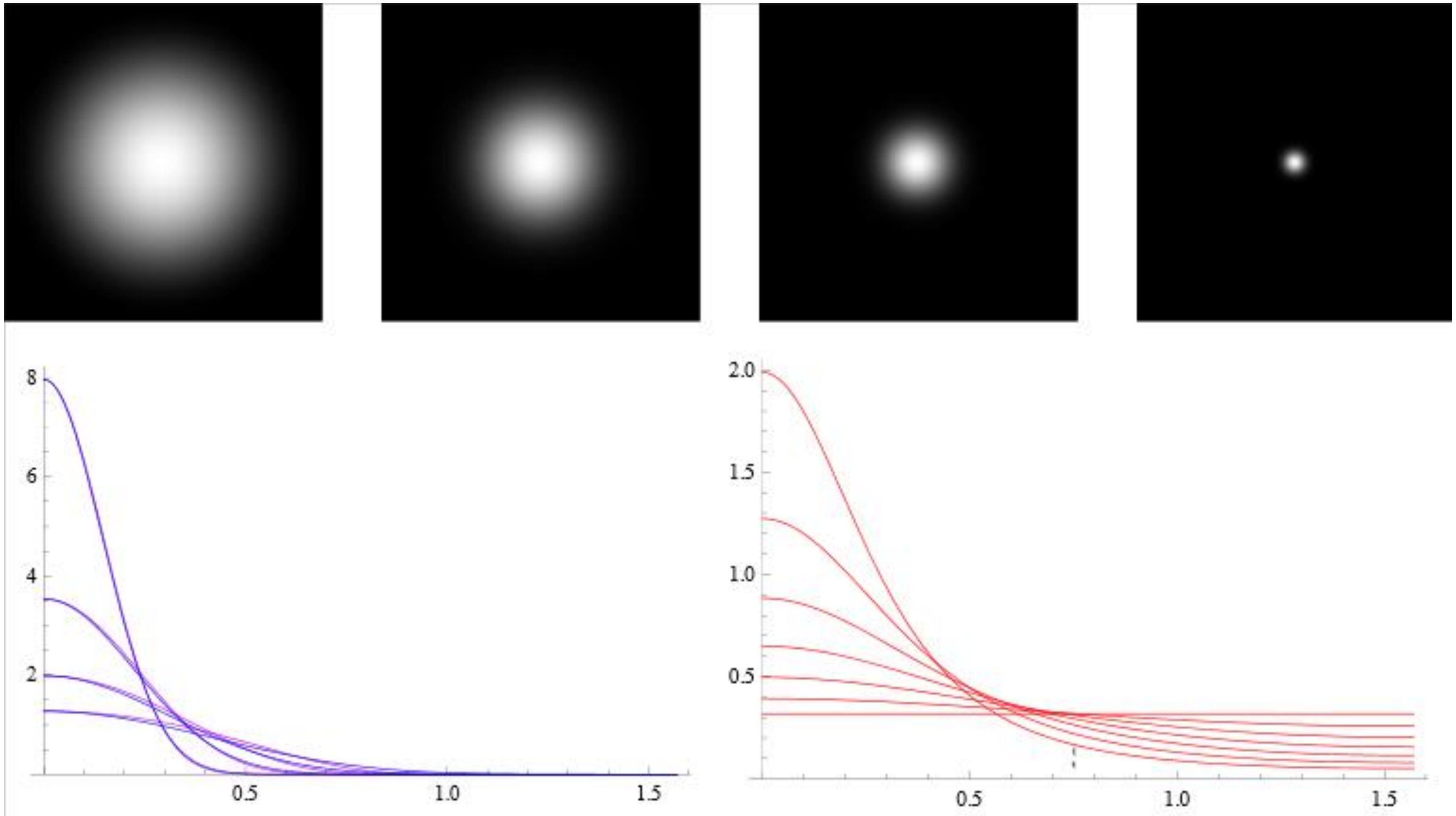
$$D_{tr}(\mathbf{m}) = \frac{\alpha_{tr}^2}{\pi\left((\mathbf{n}\cdot\mathbf{m})^2(\alpha_{tr}^2 - 1) + 1\right)^2}$$

$$D_b(\mathbf{m}) = \frac{1}{\pi\alpha_b^2(\mathbf{n}\cdot\mathbf{m})^4} e^{-\left(\frac{1-(\mathbf{n}\cdot\mathbf{m})^2}{\alpha_b^2(\mathbf{n}\cdot\mathbf{m})^2}\right)}$$

$$D_{\mathrm{sgd}}(\mathbf{m}) = \frac{\mathrm{p22}\left[\frac{1-(\mathbf{n}\cdot\mathbf{m})^2}{(\mathbf{n}\cdot\mathbf{m})^2}\right]}{\pi(\mathbf{n}\cdot\mathbf{m})^4}$$

Normal Distribution Function

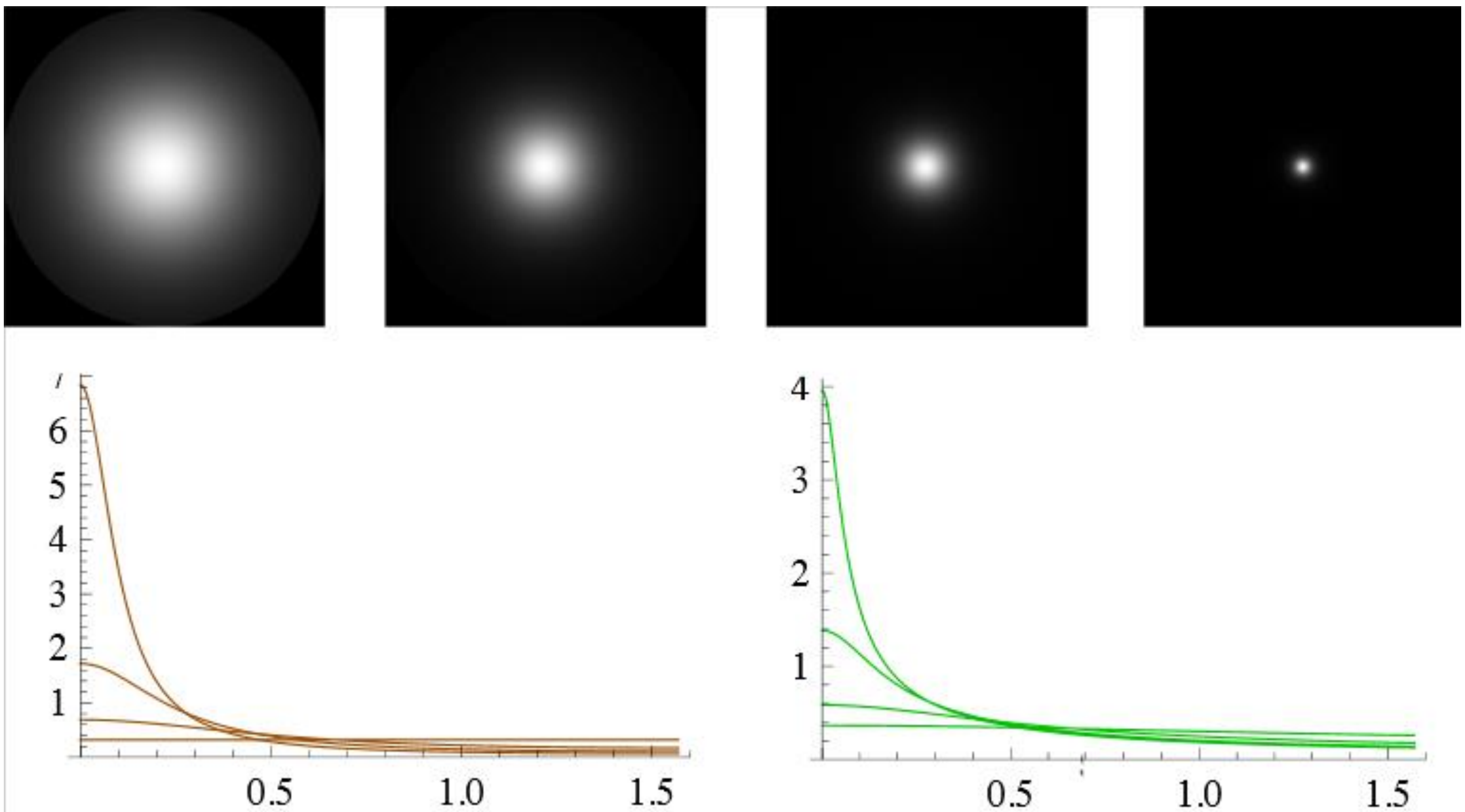
- Statistical distribution of orientation h
  - Determine size, brightness and shape of specular highlight



Gaussian shapes

Normal Distribution Function

- Statistical distribution of orientation h
  - Determine size, brightness and shape of specular highlight



Spiky shapes

Normal Distribution Function

- Statistical distribution of orientation h
  - Determine size, brightness and shape of specular highlight
- Commonly used NDFs
  - Phong distribution

$$D_p(\mathbf{m}) = \frac{\alpha_p + 2}{2\pi}(\mathbf{n} \cdot \mathbf{m})^{\alpha_p}$$

Normal Distribution Function

- Statistical distribution of orientation h
  - Determine size, brightness and shape of specular highlight
- Commonly used NDFs
  - Phong distribution

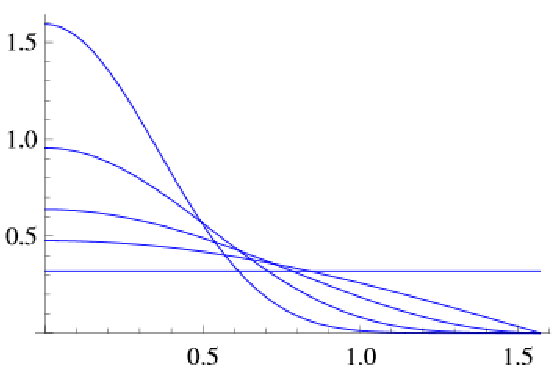
$$D_p(\mathbf{m}) = \frac{\alpha_p + 2}{2\pi}(\mathbf{n} \cdot \mathbf{m})^{\alpha_p}$$

Normalization factor: $$(\mathbf{v} \cdot \mathbf{n}) = \int_{\Theta} D(\mathbf{m})(\mathbf{v} \cdot \mathbf{m})\, d\omega_m$$

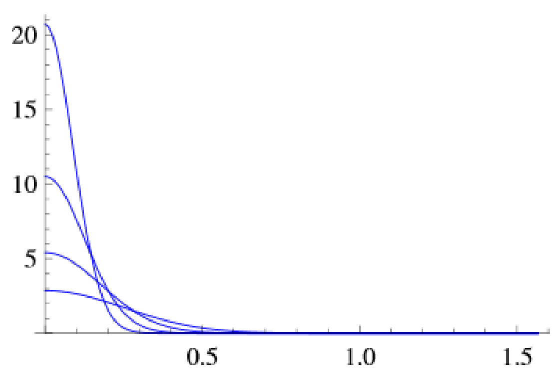
Normal Distribution Function

- Statistical distribution of orientation h
  - Determine size, brightness and shape of specular highlight
- Commonly used NDFs
  - Phong distribution

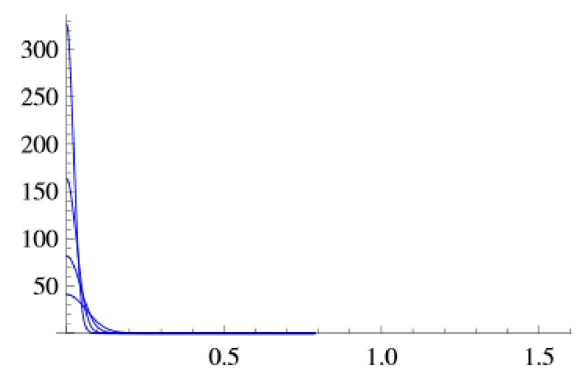
$$D_p(\mathbf{m}) = \frac{\alpha_p + 2}{2\pi}(\mathbf{n} \cdot \mathbf{m})^{\alpha_p}$$



0 → 8



16 → 128



256 → 2048

Normal Distribution Function

- Statistical distribution of orientation h
  - Determine size, brightness and shape of specular highlight
- Commonly used NDFs
  - Phong distribution
  - Beckmann distribution

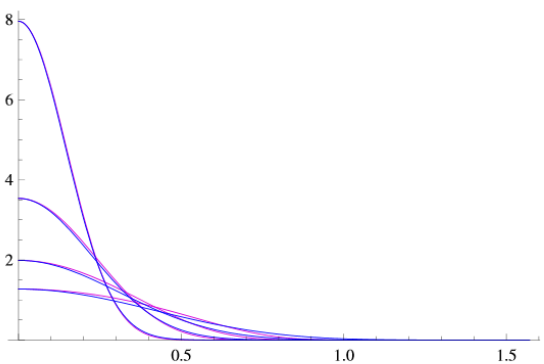
$$D_b(\mathbf{m}) = \frac{1}{\pi\alpha_b^2(\mathbf{n}\cdot\mathbf{m})^4} e^{-\left(\frac{1-(\mathbf{n}\cdot\mathbf{m})^2}{\alpha_b^2(\mathbf{n}\cdot\mathbf{m})^2}\right)}$$

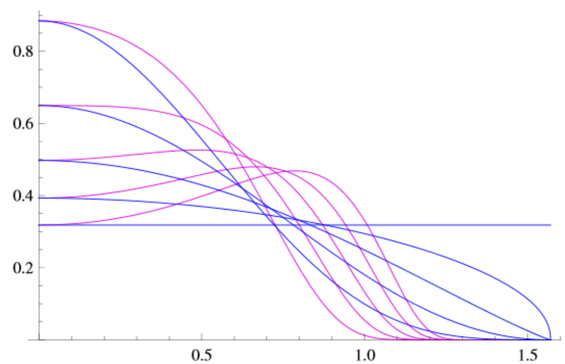
0 → 8 16 → 128 256 → 2048

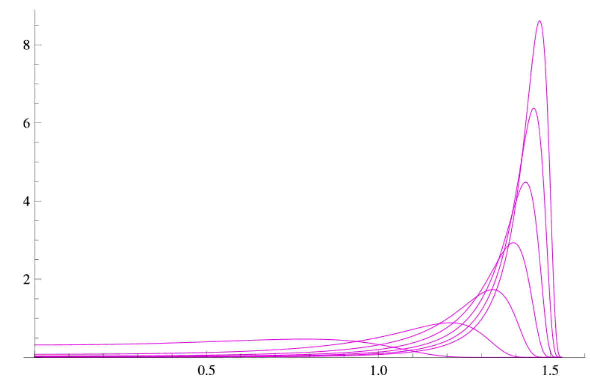
Normal Distribution Function

- Statistical distribution of orientation h
  - Determine size, brightness and shape of specular highlight
- Commonly used NDFs
  - Phong distribution
  - Beckmann distribution

$$D_b(\mathbf{m}) = \frac{1}{\pi \alpha_b^2 (\mathbf{n}\cdot\mathbf{m})^4} e^{-\left(\frac{1-(\mathbf{n}\cdot\mathbf{m})^2}{\alpha_b^2(\mathbf{n}\cdot\mathbf{m})^2}\right)}$$







$$\alpha_p = 2\alpha_b^{-2} - 2$$

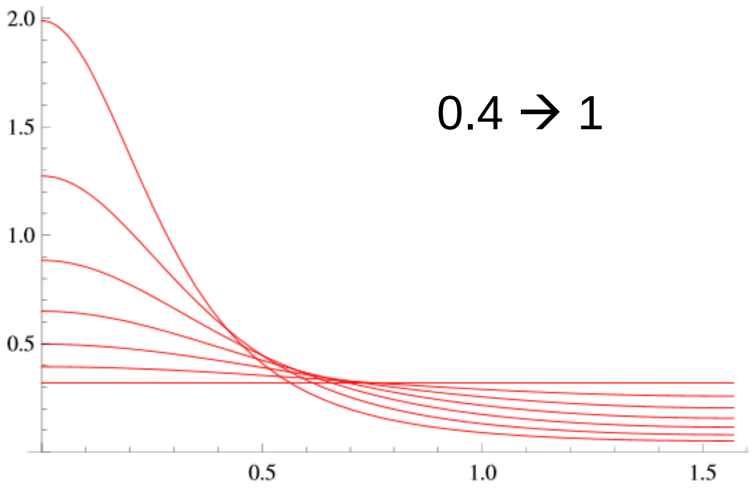
1→ 7

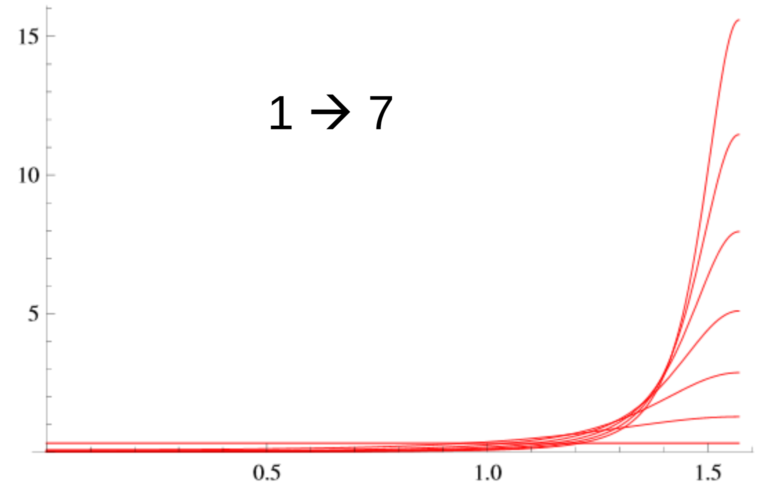
Super rough

Normal Distribution Function

- Statistical distribution of orientation h
  - Determine size, brightness and shape of specular highlight
- Commonly used NDFs
  - Phong distribution
  - Beckmann distribution
  - GGX distribution

$$D_{\mathrm{tr}}(\mathbf{m}) = \frac{\alpha_{\mathrm{tr}}^2}{\pi\left((\mathbf{n}\cdot\mathbf{m})^2\left(\alpha_{\mathrm{tr}}^2 - 1\right) + 1\right)^2}$$

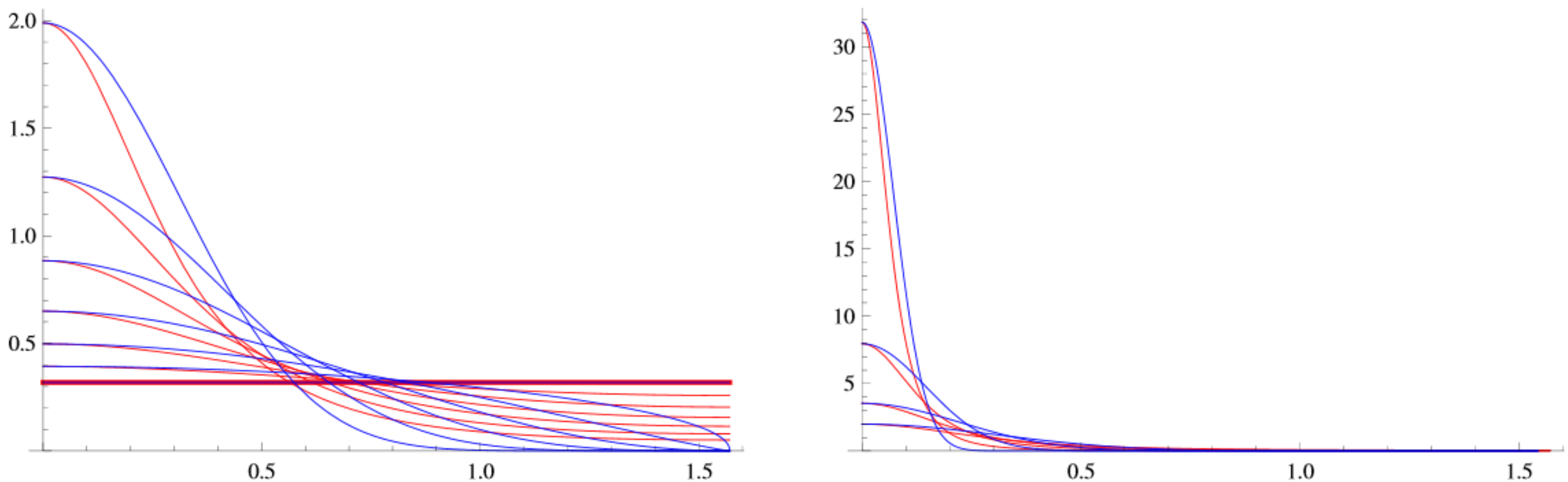




Normal Distribution Function

- Statistical distribution of orientation h
  - Determine size, brightness and shape of specular highlight
- Commonly used NDFs
  - Phong distribution
  - Beckmann distribution
  - GGX distribution

$$D_{\mathrm{tr}}(\mathbf{m}) = \frac{\alpha_{\mathrm{tr}}^2}{\pi\left((\mathbf{n}\cdot\mathbf{m})^2\left(\alpha_{\mathrm{tr}}^2-1\right)+1\right)^2}$$



Phong comparison

Normal Distribution Function

- Statistical distribution of orientation h
  - Determine size, brightness and shape of specular highlight
- Commonly used NDFs
  - Phong distribution
  - Beckmann distribution
  - GGX distribution
  - And many others…

Normal Distribution Function

- Statistical distribution of orientation h
  - Determine size, brightness and shape of specular highlight
- Commonly used NDFs
  - Phong distribution
  - Beckmann distribution
  - GGX distribution
  - And many others…
- Choice of NDF?
  - Depends on evaluation cost (applications)
  - Material properties (rough, isotropic, etc)
  - Artistic controls

Geometry function

- Shadowing and masking
- Probability that points with given microfacet normal
  - is visible from light
  - And from view

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

Geometry function

- Shadowing and masking
- Probability that points with given microfacet normal
  - is visible from light
  - And from view
- Commonly used geometry functions
  - No visibility $G_{\text{implicit}}(\mathbf{l}, \mathbf{v}, \mathbf{m}) = (\mathbf{n} \cdot \mathbf{l_c})(\mathbf{n} \cdot \mathbf{v})$

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

Geometry function

- Shadowing and masking
- Probability that points with given microfacet normal
  - is visible from light
  - And from view
- Commonly used geometry functions
  - No visibility $G_{\text{implicit}}(\mathbf{l}, \mathbf{v}, \mathbf{m}) = (\mathbf{n} \cdot \mathbf{l}_\mathbf{c})(\mathbf{n} \cdot \mathbf{v})$

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

Geometry function

- Shadowing and masking
- Probability that points with given microfacet normal
  - is visible from light
  - And from view
- Commonly used geometry functions
  - No visibility $G_{\text{implicit}}(\mathbf{l}, \mathbf{v}, \mathbf{m}) = (\mathbf{n} \cdot \mathbf{l_c})(\mathbf{n} \cdot \mathbf{v})$
  - Cook-Terrance $G_{\text{ct}}(\mathbf{l}, \mathbf{v}, \mathbf{h}) = \min\left(1, \frac{2(\mathbf{n} \cdot \mathbf{h})(\mathbf{n} \cdot \mathbf{v})}{(\mathbf{v} \cdot \mathbf{h})}, \frac{2(\mathbf{n} \cdot \mathbf{h})(\mathbf{n} \cdot \mathbf{l})}{(\mathbf{v} \cdot \mathbf{h})}\right)$

Geometry function

- Shadowing and masking
- Probability that points with given microfacet normal
  - is visible from light
  - And from view
- Commonly used geometry functions
  - No visibility $G_{\text{implicit}}(\mathbf{l}, \mathbf{v}, \mathbf{m}) = (\mathbf{n} \cdot \mathbf{l_c})(\mathbf{n} \cdot \mathbf{v})$
  - Cook-Terrance $G_{\text{ct}}(\mathbf{l}, \mathbf{v}, \mathbf{h}) = \min\left(1, \frac{2(\mathbf{n} \cdot \mathbf{h})(\mathbf{n} \cdot \mathbf{v})}{(\mathbf{v} \cdot \mathbf{h})}, \frac{2(\mathbf{n} \cdot \mathbf{h})(\mathbf{n} \cdot \mathbf{l})}{(\mathbf{v} \cdot \mathbf{h})}\right)$
  - Smith $G(\mathbf{l}, \mathbf{v}, \mathbf{h}) = G_1(\mathbf{l})G_1(\mathbf{v})$ depends on NDF

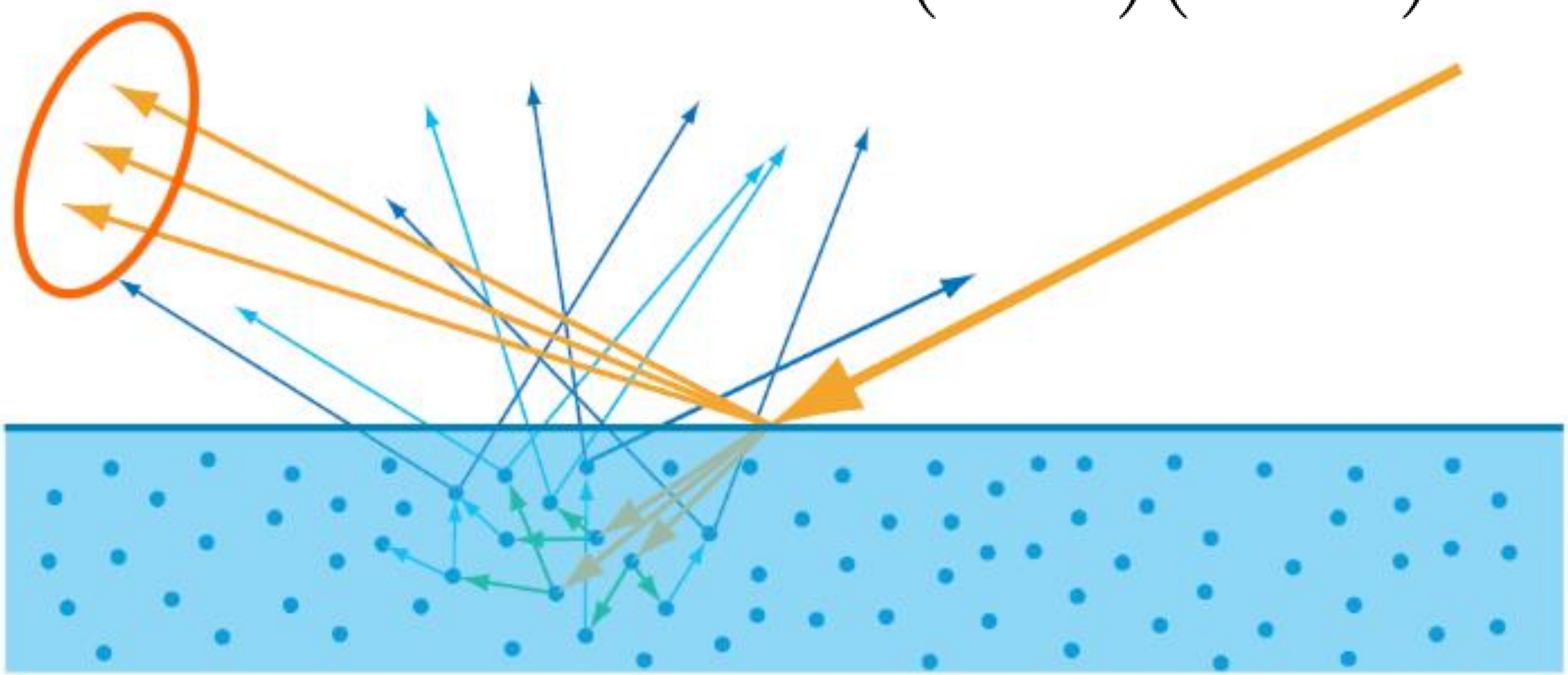
Geometry function

- Shadowing and masking
- Probability that points with given microfacet normal
  - is visible from light
  - And from view
- Commonly used geometry functions
  - No visibility $G_{\text{implicit}}(\mathbf{l}, \mathbf{v}, \mathbf{m}) = (\mathbf{n} \cdot \mathbf{l_c})(\mathbf{n} \cdot \mathbf{v})$
  - Cook-Terrance $G_{\text{ct}}(\mathbf{l}, \mathbf{v}, \mathbf{h}) = \min\left(1, \frac{2(\mathbf{n} \cdot \mathbf{h})(\mathbf{n} \cdot \mathbf{v})}{(\mathbf{v} \cdot \mathbf{h})}, \frac{2(\mathbf{n} \cdot \mathbf{h})(\mathbf{n} \cdot \mathbf{l})}{(\mathbf{v} \cdot \mathbf{h})}\right)$
  - Smith $G(\mathbf{l}, \mathbf{v}, \mathbf{h}) = G_1(\mathbf{l}) G_1(\mathbf{v})$ depends on NDF
- More about the masking shadowing function:
  - Understanding the Masking-Shadowing Function in Microfacet-Based BRDFs [Heitz - JCGT 2014]

Microfacet theory

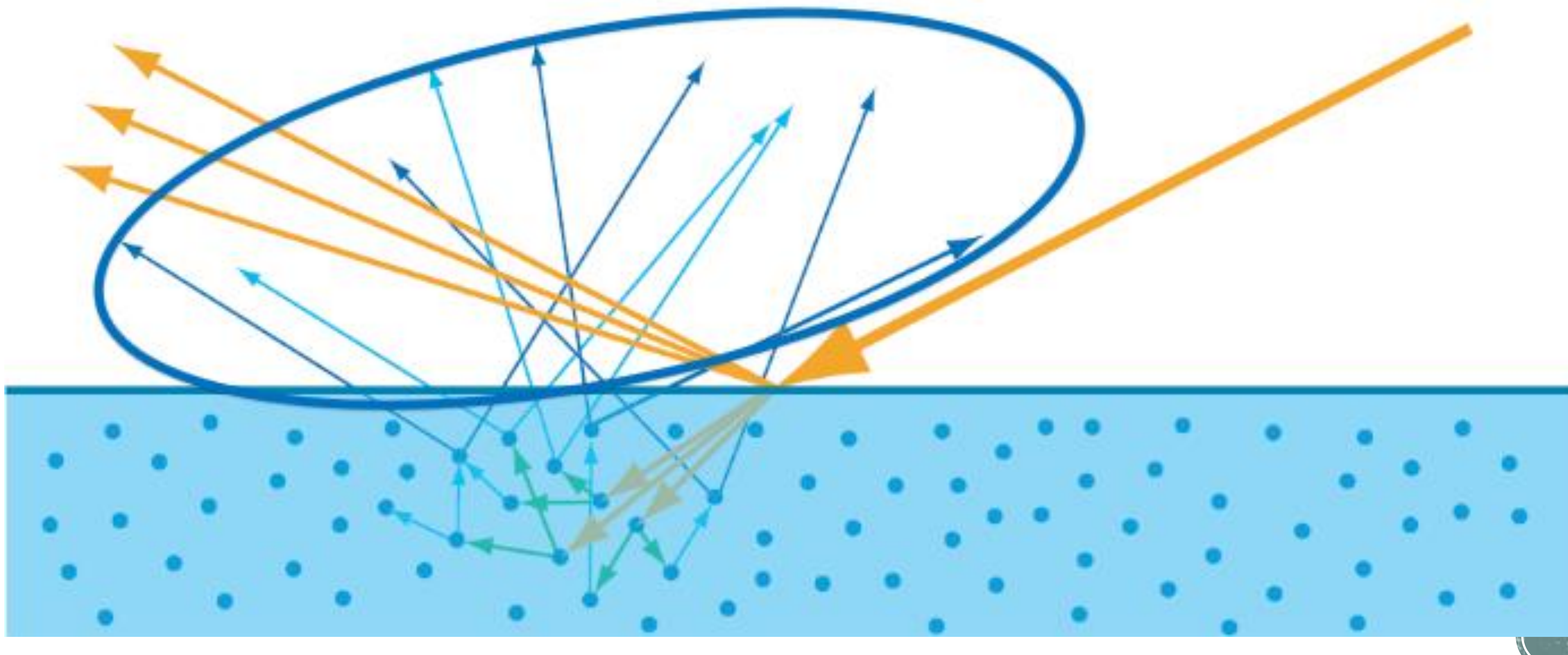
- Surface reflection (specular term)

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



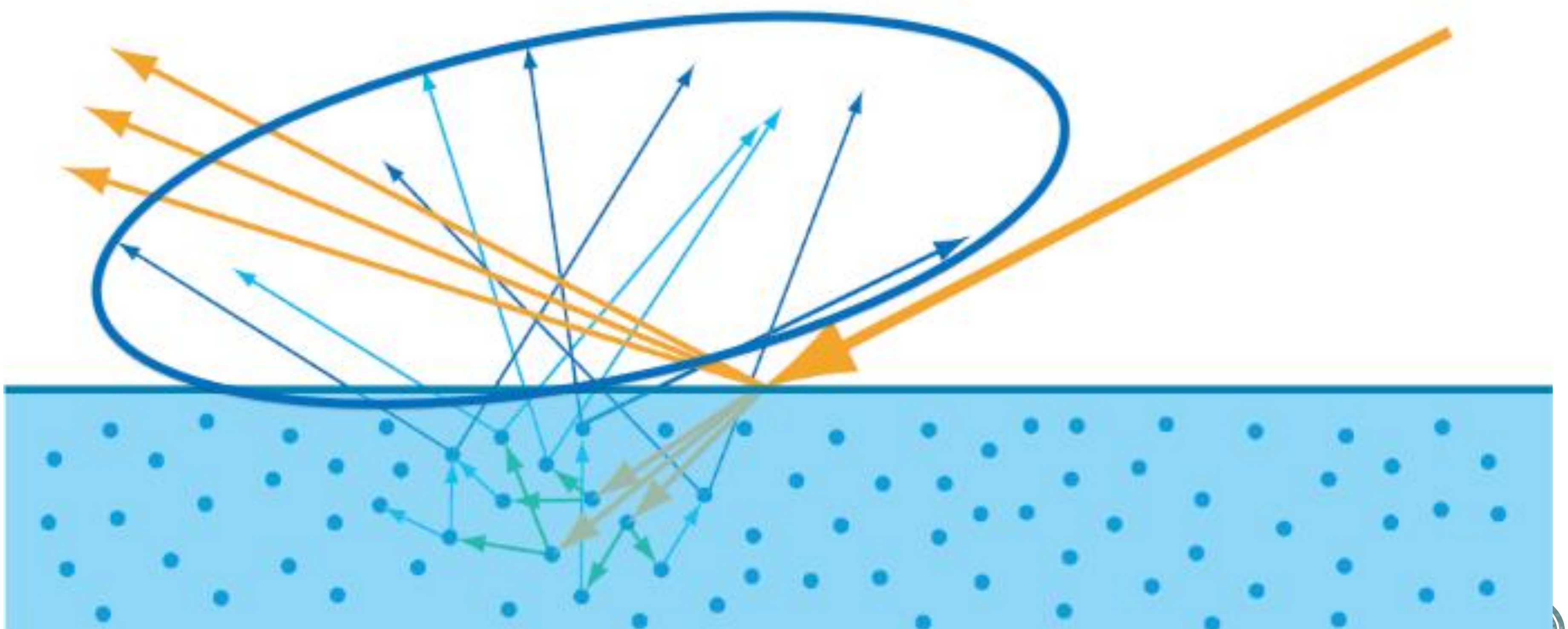
Microfacet theory

- Subsurface reflection (diffuse term)



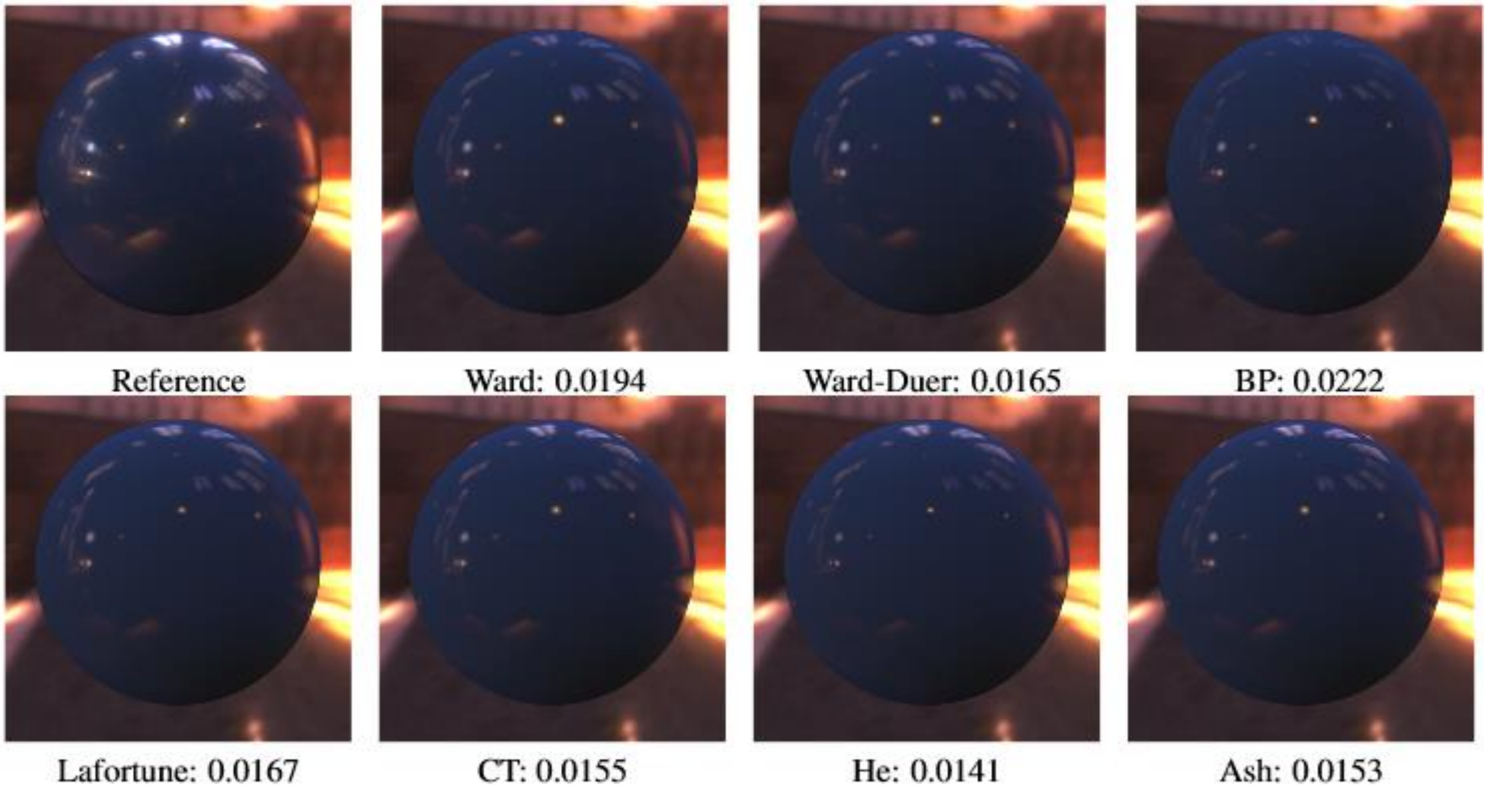
Microfacet theory

- Subsurface reflection (diffuse term)
  - Constant: $f_{\mathrm{Lambert}}(\mathbf{l}, \mathbf{v}) = \frac{\mathbf{c}_{\mathrm{diff}}}{\pi}$



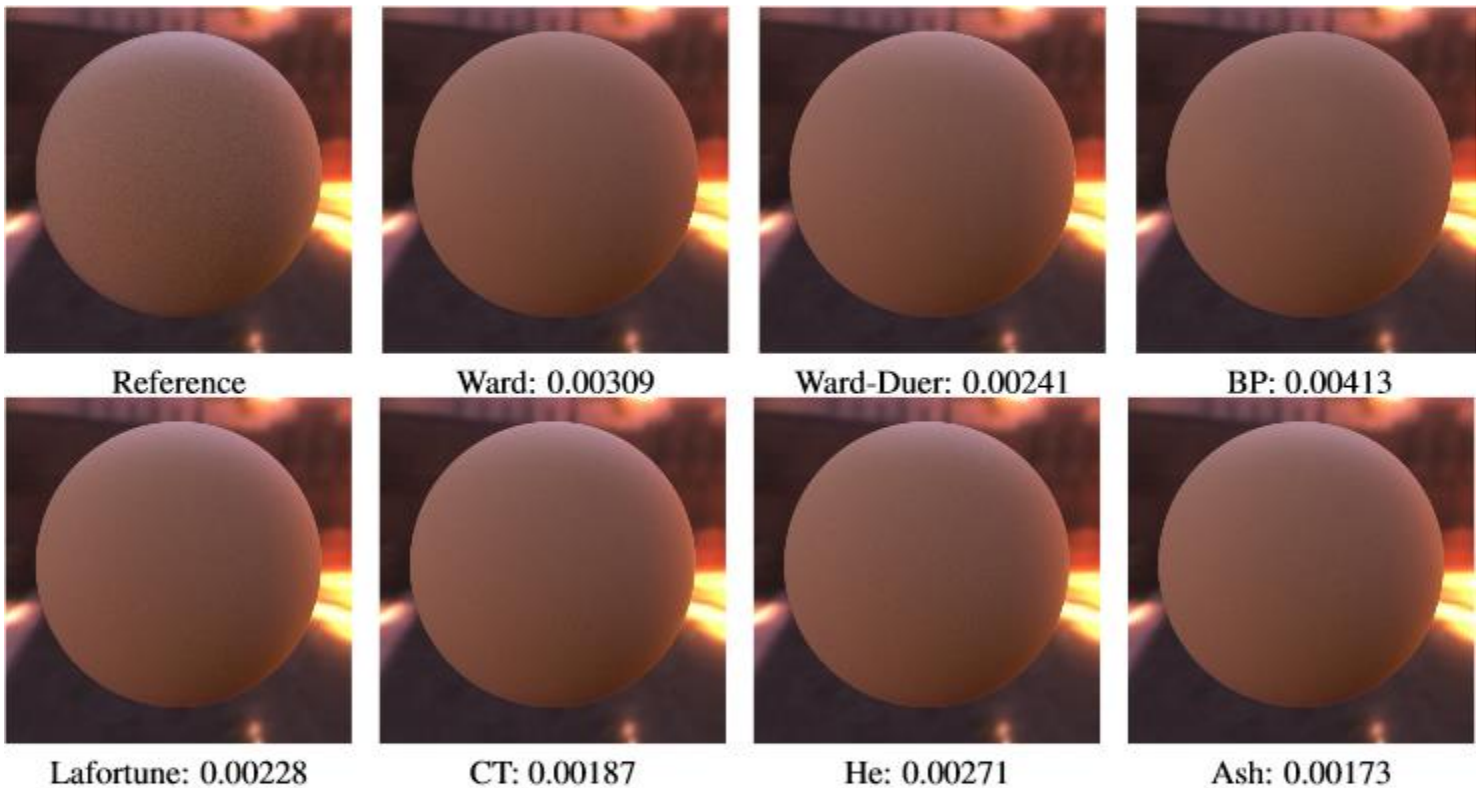
BRDF comparison

- Ngan et al. 2005



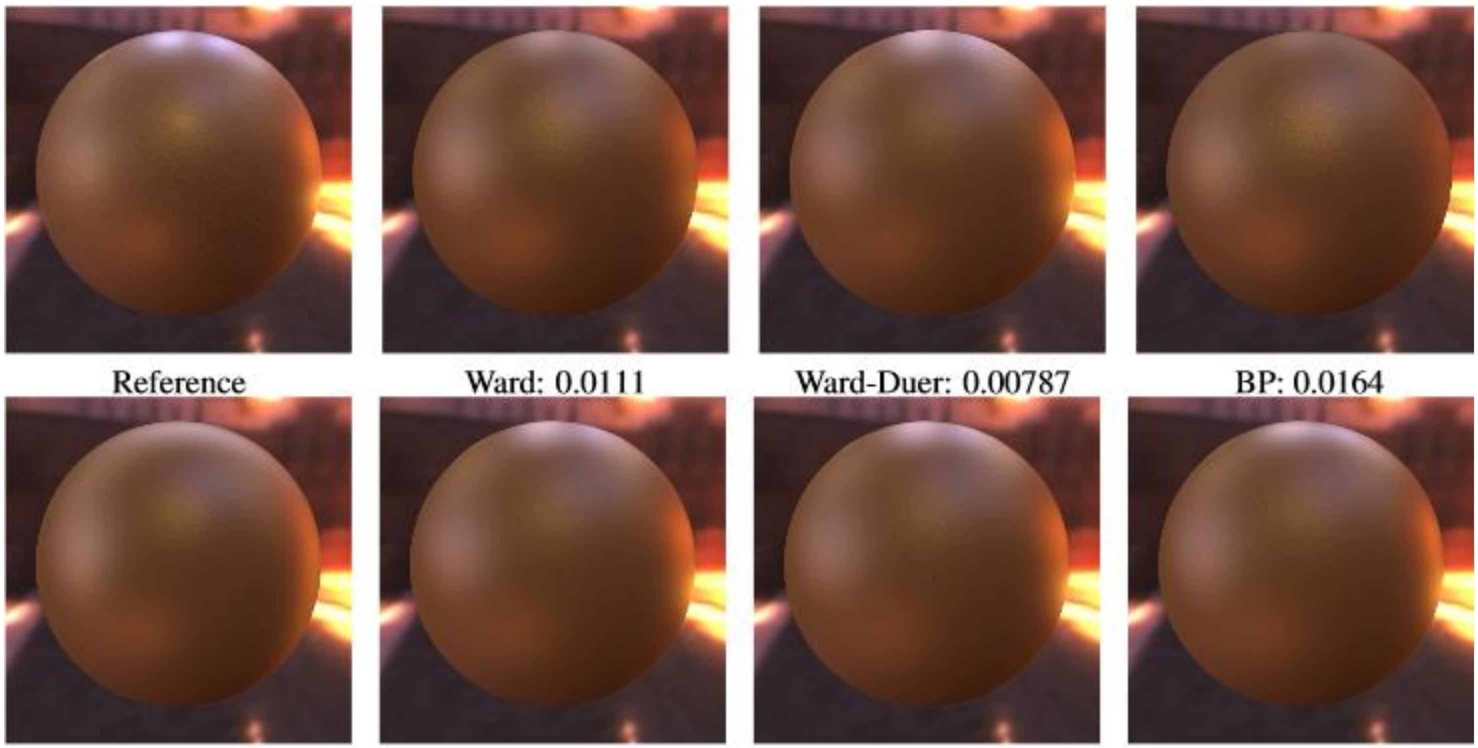
BRDF comparison

- Ngan et al. 2005



BRDF comparison

- Ngan et al. 2005

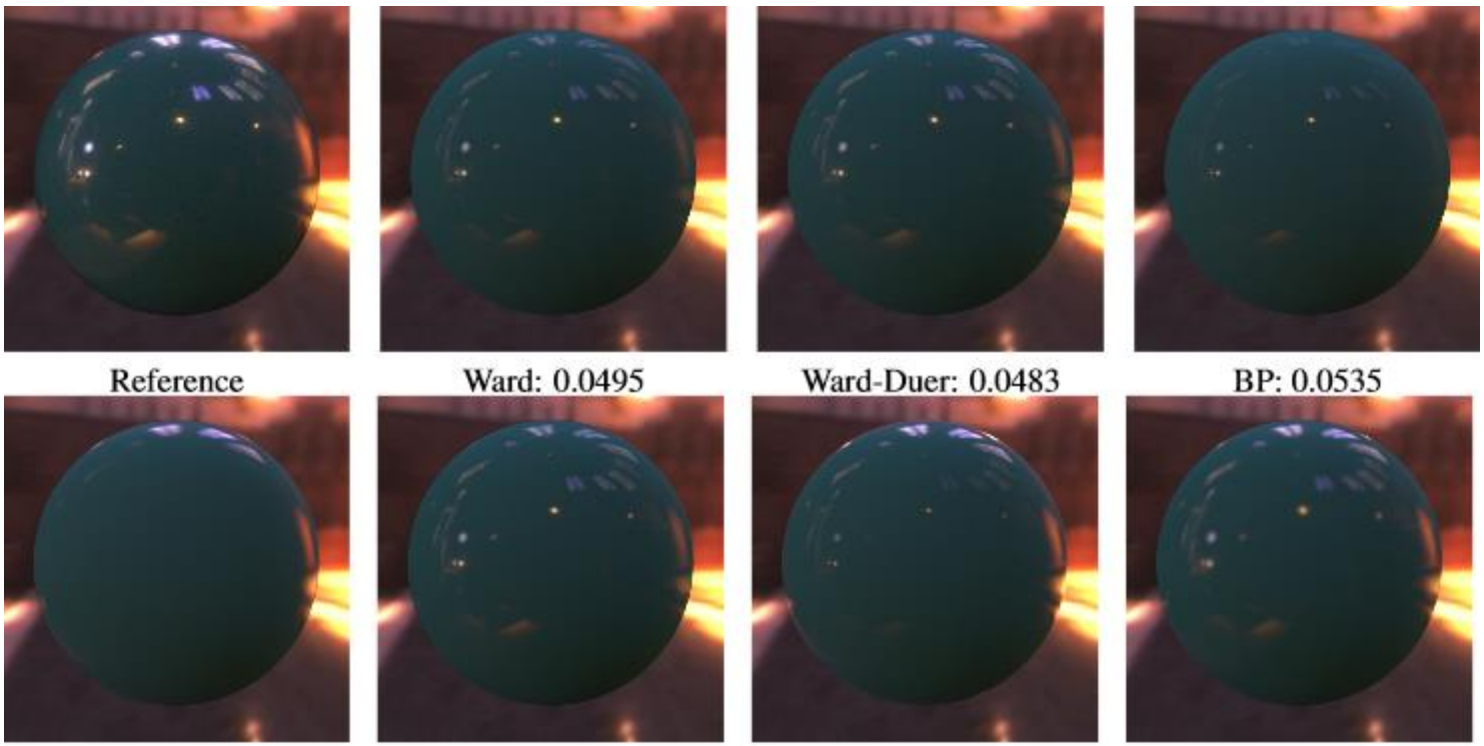


Reference | Ward: 0.0111 | Ward-Duer: 0.00787 | BP: 0.0164

Lafortune: 0.0132 | CT: 0.00771 | He: 0.00740 | Ash: 0.00699

BRDF comparison

- Ngan et al. 2005

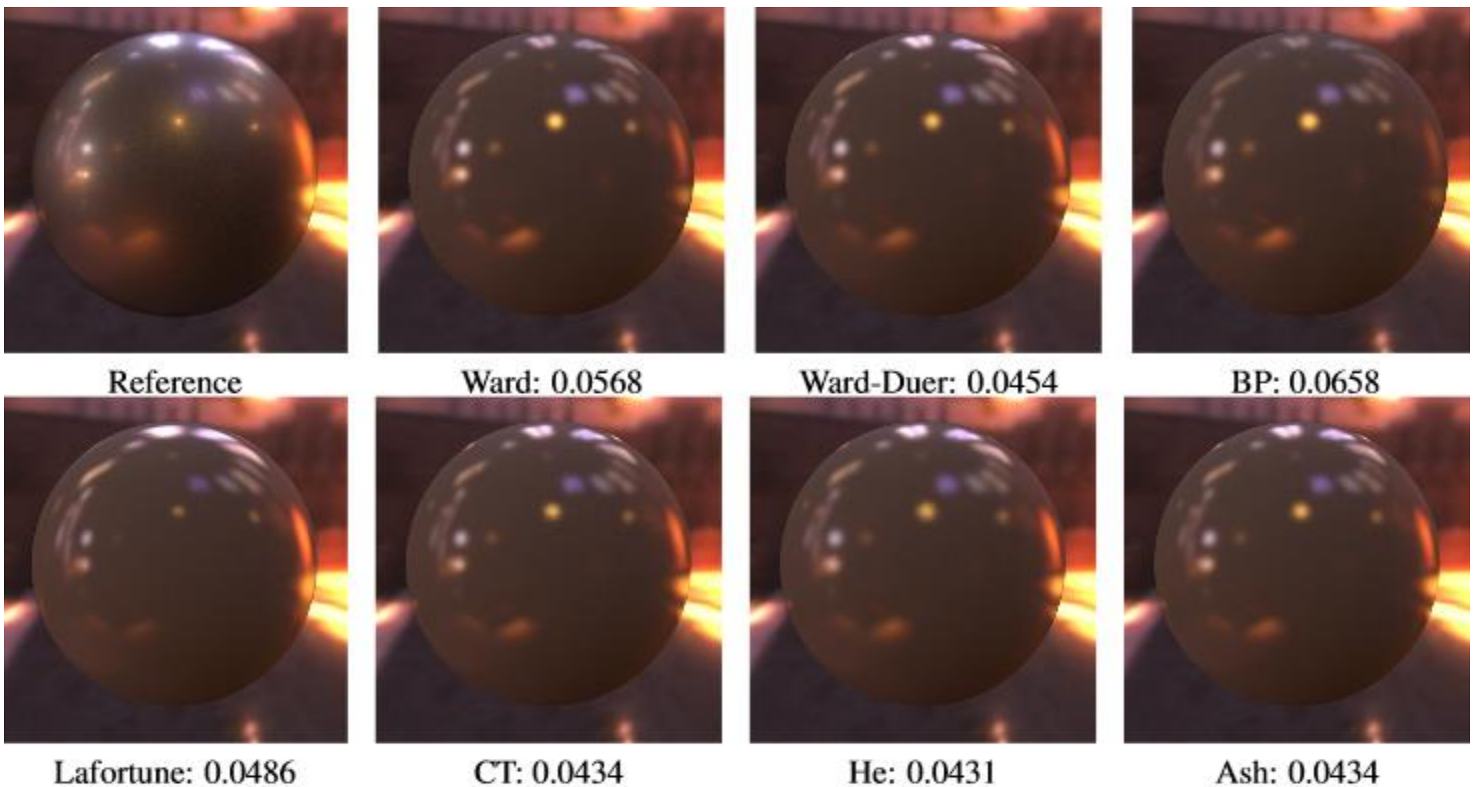


Reference | Ward: 0.0495 | Ward-Duer: 0.0483 | BP: 0.0535

Lafortune: 0.0482 | CT: 0.0483 | He: 0.0379 | Ash: 0.0463

BRDF comparison

- Ngan et al. 2005



What are we missing?

- BRDF: Bidirectionnal Reflectance Distribution Function
  - 4D or 3D for isotropic materials $f(\mathbf{l}, \mathbf{v})$

What are we missing?

- BRDF: Bidirectionnal Reflectance Distribution Function
  - 4D or 3D for isotropic materials $f(\mathbf{l}, \mathbf{v})$
- BTDF: Bidirectionnal Transmission Distribution Function
  - Same as BRDF for opposite side of surface

What are we missing?

- BRDF: Bidirectionnal Reflectance Distribution Function
  - 4D or 3D for isotropic materials $f(\mathbf{l}, \mathbf{v})$
- BTDF: Bidirectionnal Transmission Distribution Function
  - Same as BRDF for opposite side of surface
- SVBRDF: Spatially Varying BRDF
  - 6D: changes over the surface position $f(\mathbf{l}, \mathbf{v}, \mathbf{x})$

What are we missing?

- BRDF: Bidirectionnal Reflectance Distribution Function
  - 4D or 3D for isotropic materials $f(\mathbf{l}, \mathbf{v})$
- BTDF: Bidirectionnal Transmission Distribution Function
  - Same as BRDF for opposite side of surface
- SVBRDF: Spatially Varying BRDF
  - 6D: changes over the surface position $f(\mathbf{l}, \mathbf{v}, \mathbf{x})$
- BSSRDF: Bidirectionnal Surface Scattering DF
  - 8D: light exits at another location $f(\mathbf{l}, \mathbf{x}_{\mathbf{l}}, \mathbf{v}, \mathbf{x}_{\mathbf{v}})$

What are we missing?

- BRDF: Bidirectionnal Reflectance Distribution Function
  - 4D or 3D for isotropic materials $f(\mathbf{l}, \mathbf{v})$
- BTDF: Bidirectionnal Transmission Distribution Function
  - Same as BRDF for opposite side of surface
- SVBRDF: Spatially Varying BRDF
  - 6D: changes over the surface position $f(\mathbf{l}, \mathbf{v}, \mathbf{x})$
- BSSRDF: Bidirectionnal Surface Scattering DF
  - 8D: light exits at another location $f(\mathbf{l}, \mathbf{x}_\mathbf{l}, \mathbf{v}, \mathbf{x}_\mathbf{v})$
- BSDF: Bidirectionnal Scattering Distribution Function
  - XD: General formulation $f(\mathbf{l}, \mathbf{x}_\mathbf{l}, \lambda_\mathbf{l}, \mathbf{v}, \mathbf{x}_\mathbf{v}, \lambda_\mathbf{v})$

References

- MIT:
  - http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-837-computer-graphics-fall-2012/lecture-notes/
- Standford:
  - http://candela.stanford.edu/cs348b-14/doku.php
- Siggraph:
  - http://blog.selfshadow.com/publications/s2014-shading-course/
  - http://blog.selfshadow.com/publications/s2013-shading-course/
- Image synthesis & OpenGL:
  - http://romain.vergne.free.fr/blog/?page_id=97
- Path tracing and global illum:
  - http://www.graphics.stanford.edu/courses/cs348b-01/course29.hanrahan.pdf
  - http://web.cs.wpi.edu/~emmanuel/courses/cs563/write_ups/zackw/realistic_raytracing.html
- GLSL / Shadertoy:
  - https://www.opengl.org/documentation/glsl/
  - https://www.shadertoy.com/
  - http://www.iquilezles.org/
- http://fileadmin.cs.lth.se/cs/Education/EDAN30/lectures/L2-rt.pdf
- http://csokavar.hu/raytrace/imm6392.pdf
- http://web.cs.wpi.edu/~emmanuel/courses/cs563/write_ups/zackw/realistic_raytracing.html