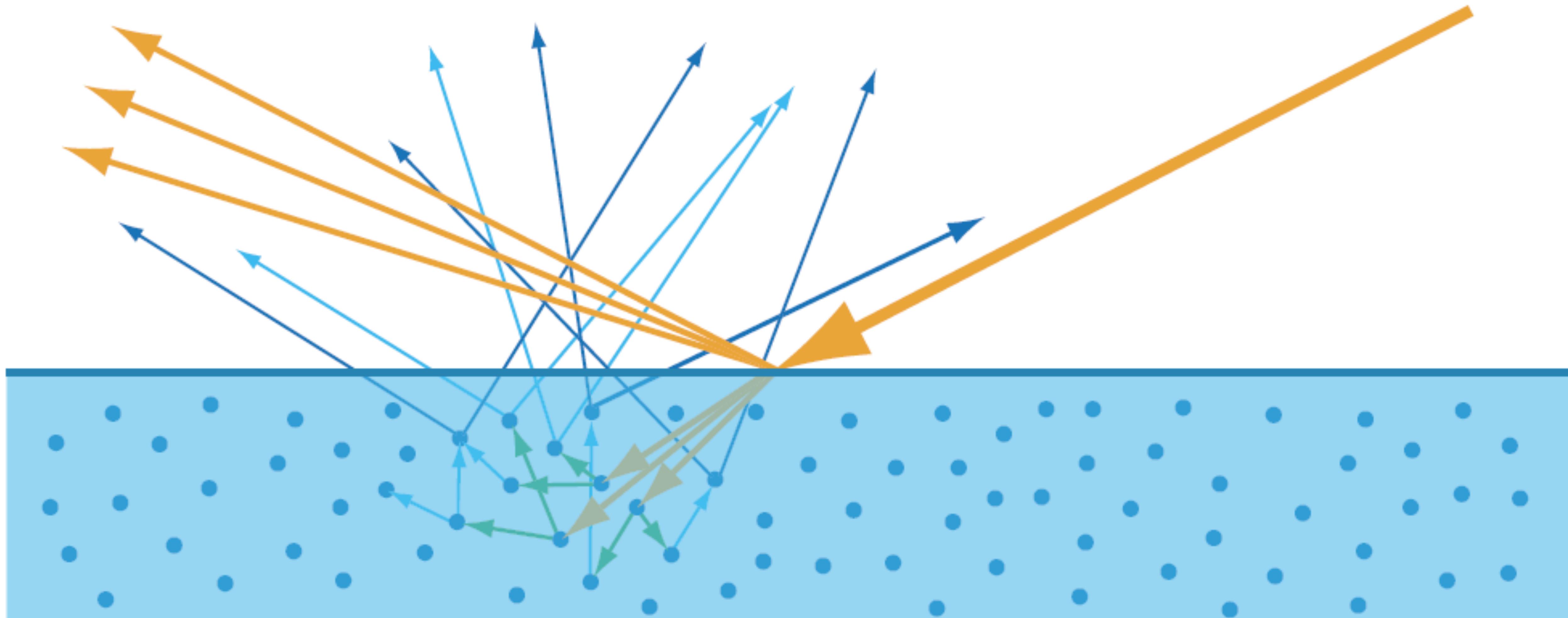


Background: Physics and Math of Shading

Naty Hoffman
2K



Hi. Over the next 25 minutes or so I'll be going from the physics underlying shading, to the math used to describe it in the kind of shading models we'll see in the rest of the course.

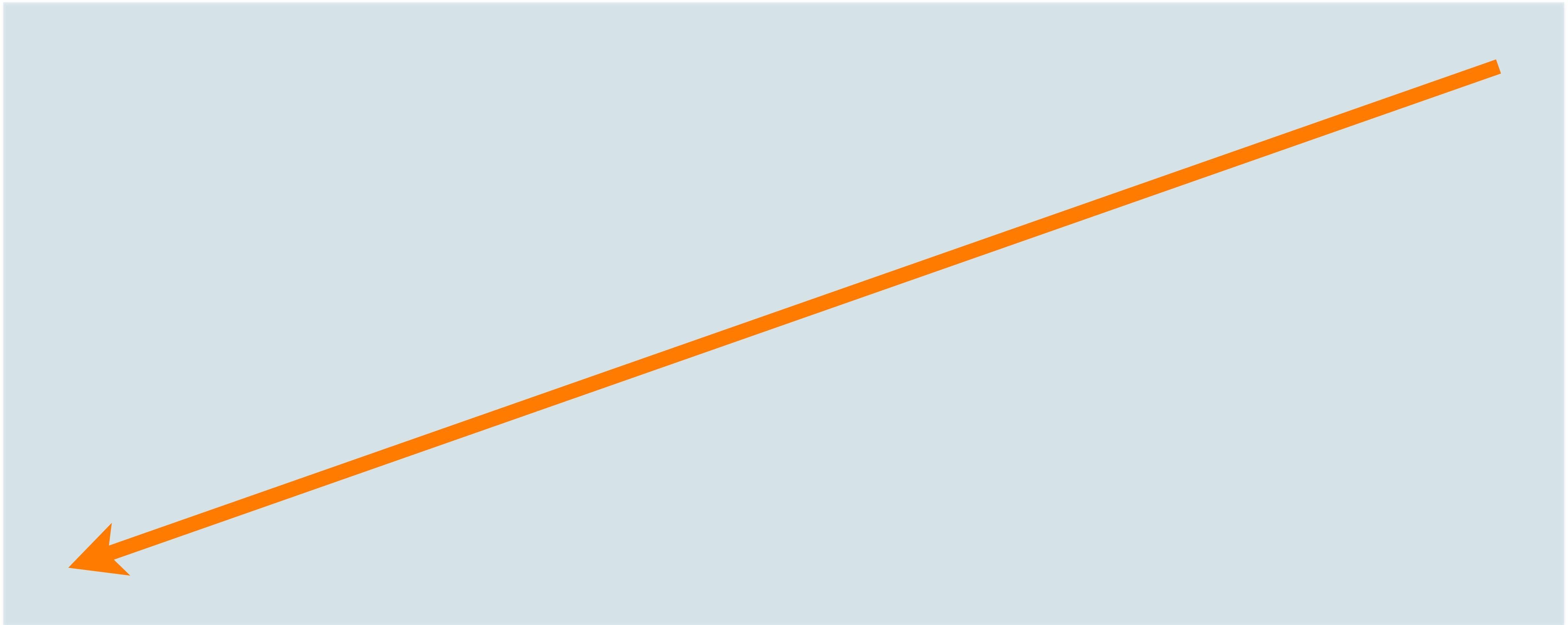


SIGGRAPH 2012



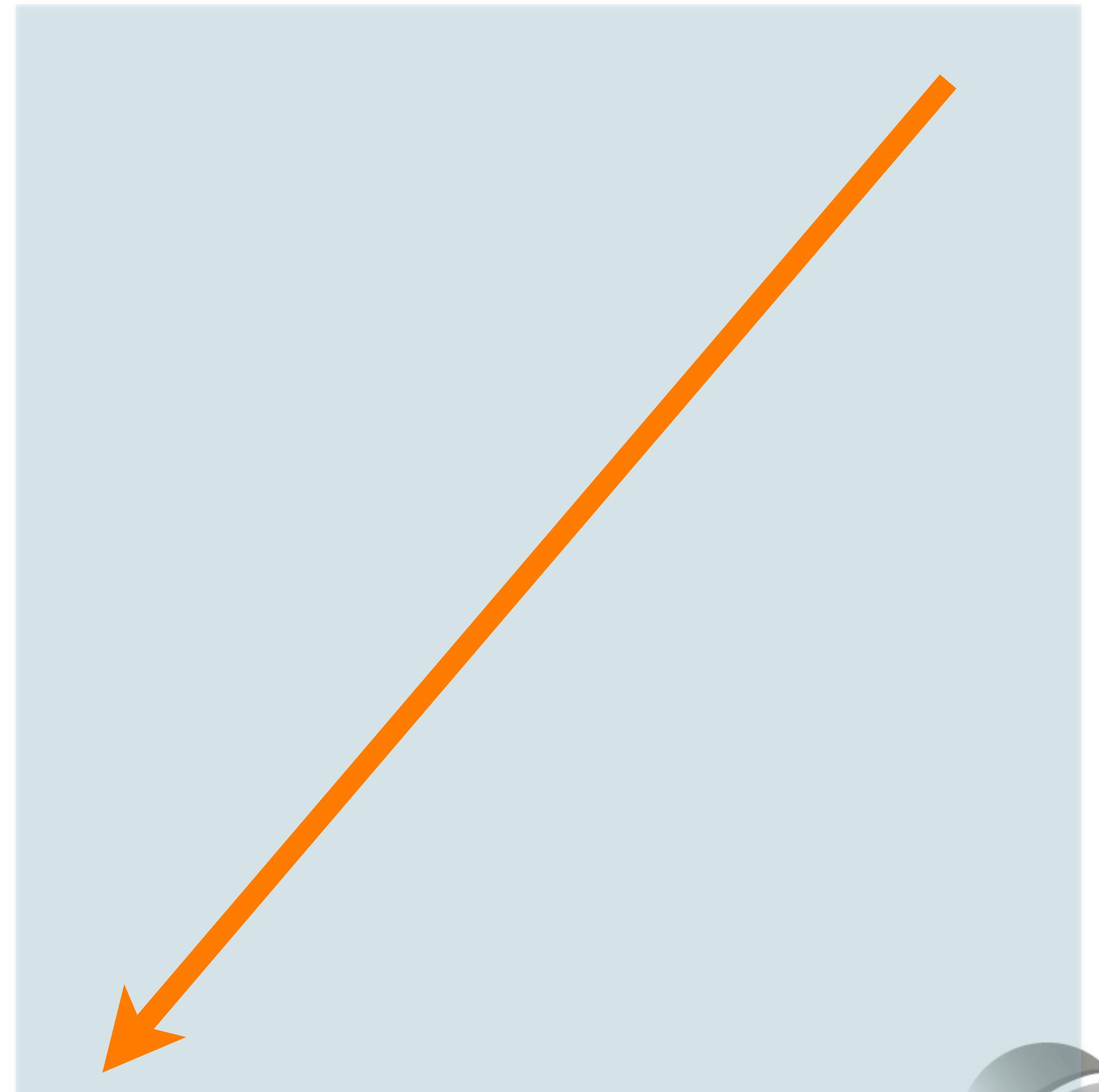
We'll start with the physics of what happens when light interacts with matter.

Light Through Homogeneous Medium



The simplest case is light propagating through a homogeneous medium, or one that is the same everywhere. In this case light moves in a straight line.

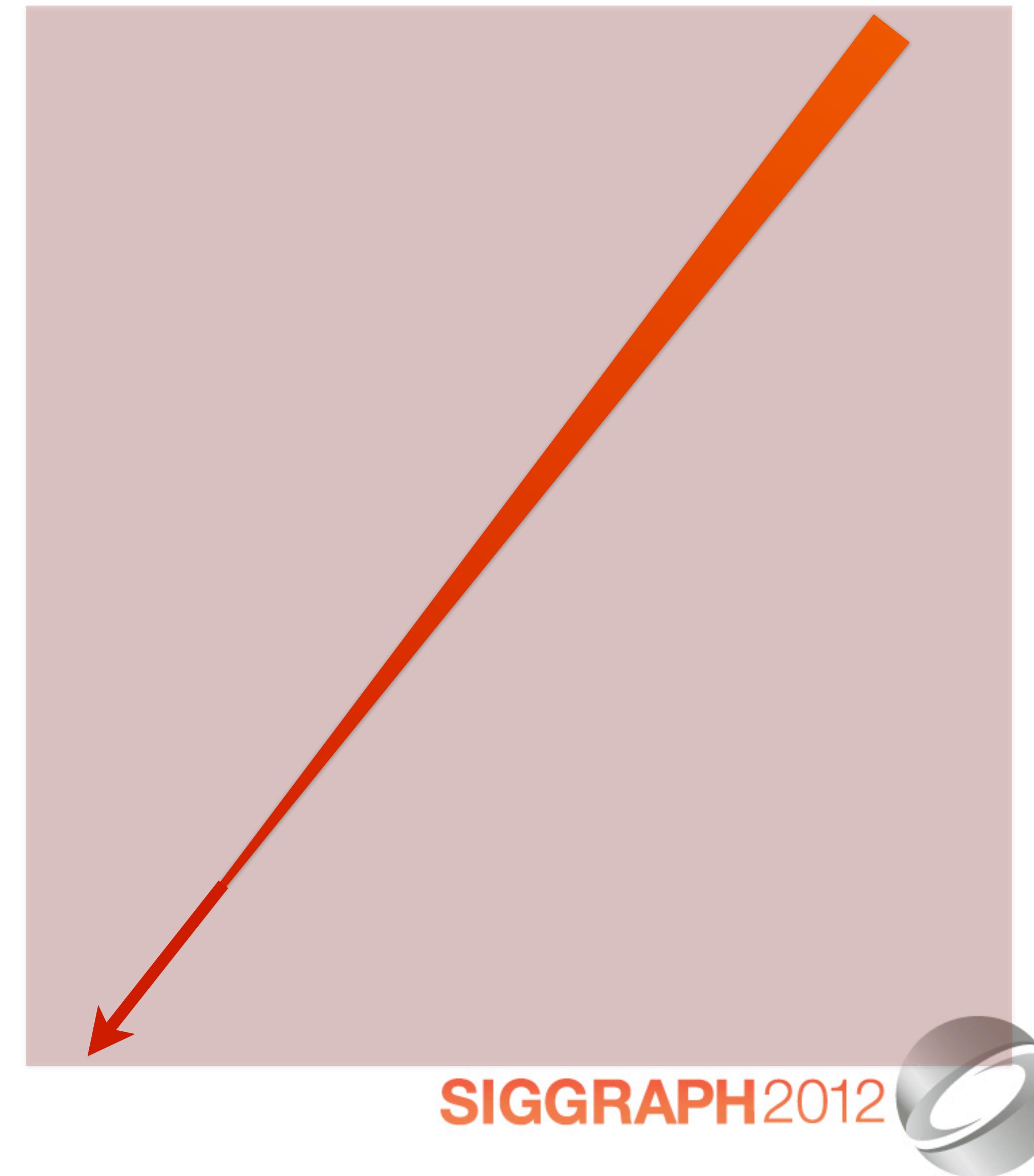
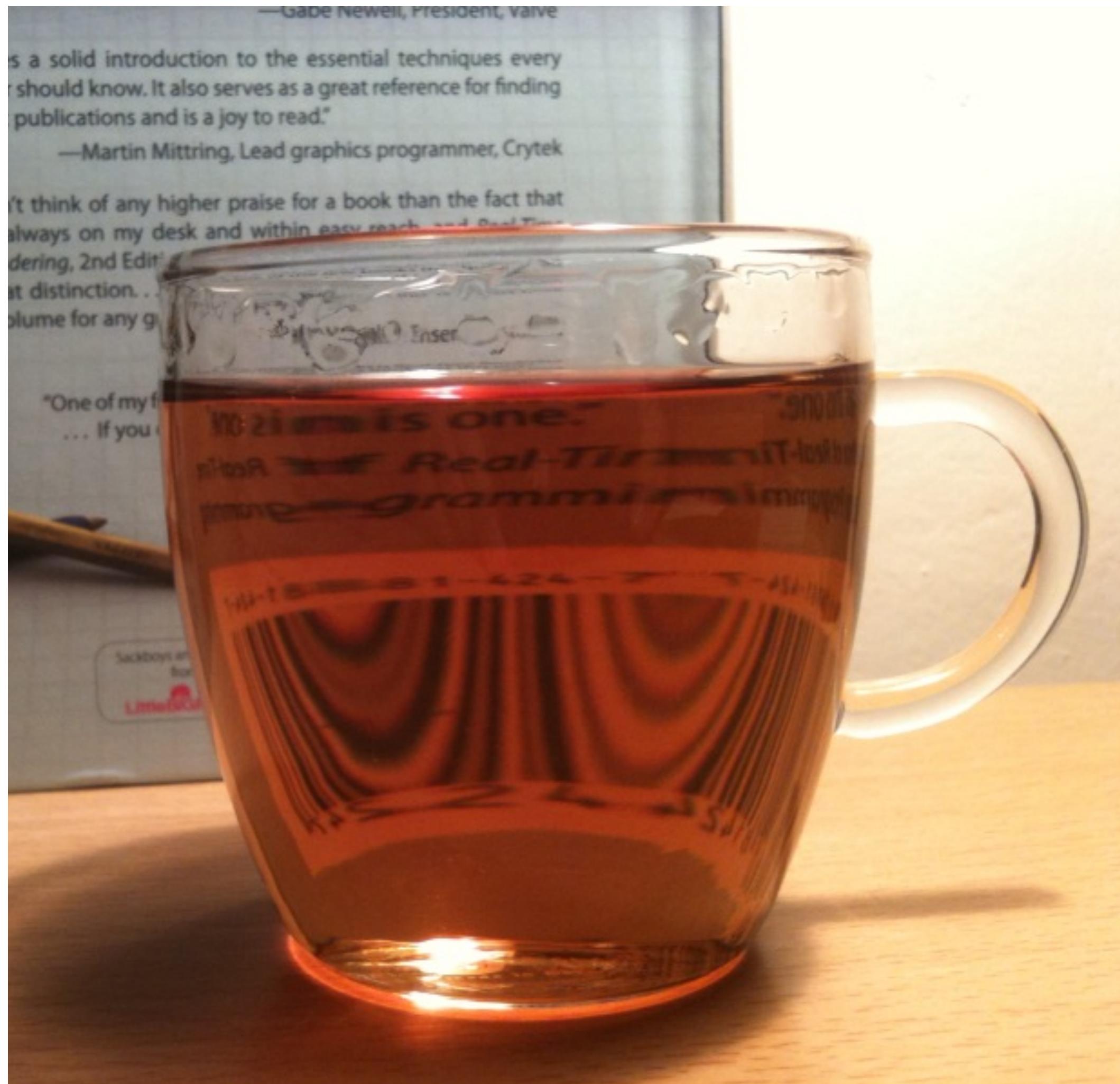
Transparent media



SIGGRAPH 2012

Some homogeneous media (like clean water or glass) barely absorb light in the visible spectrum.

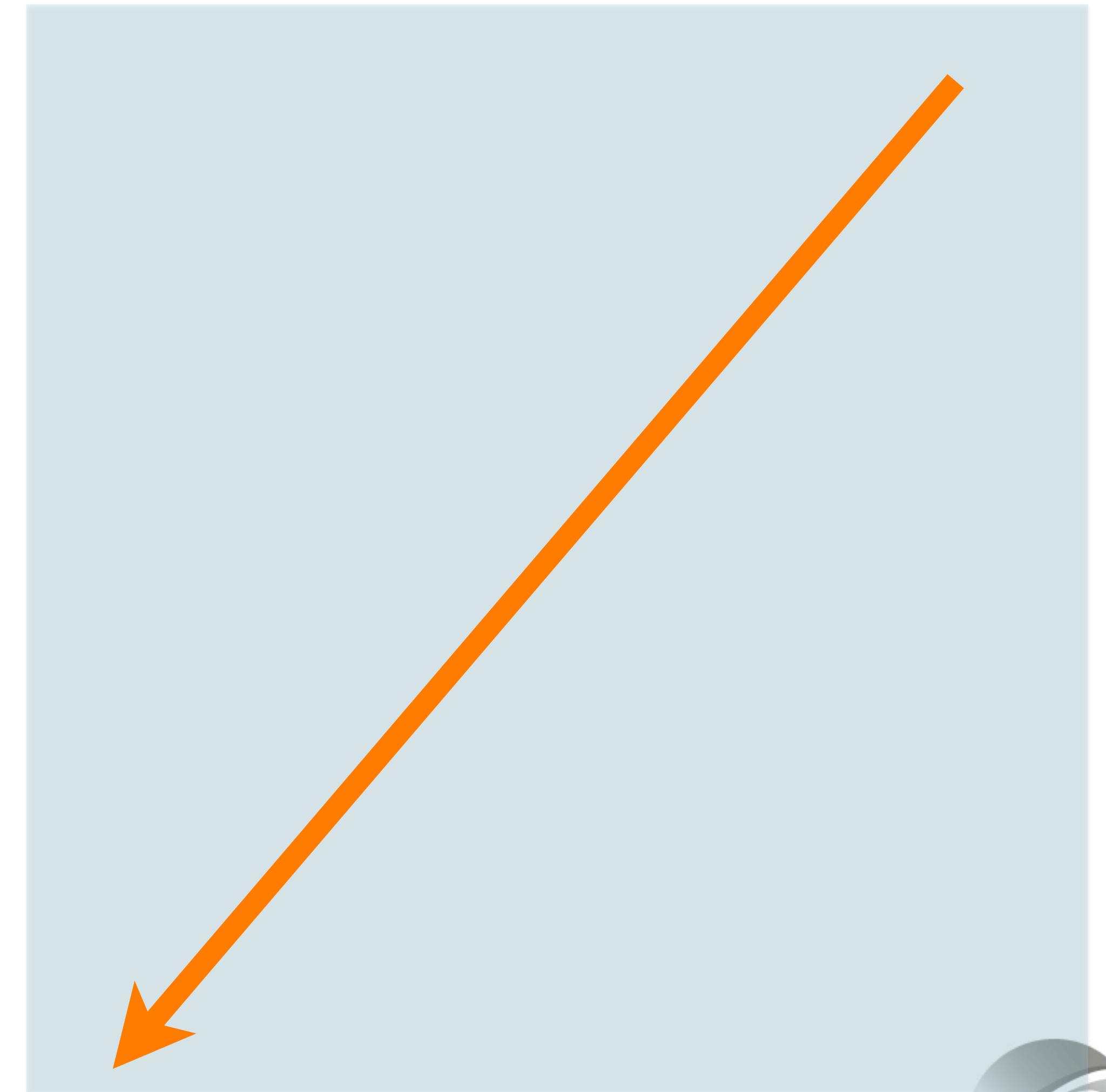
Clear Absorbent Media



SIGGRAPH 2012

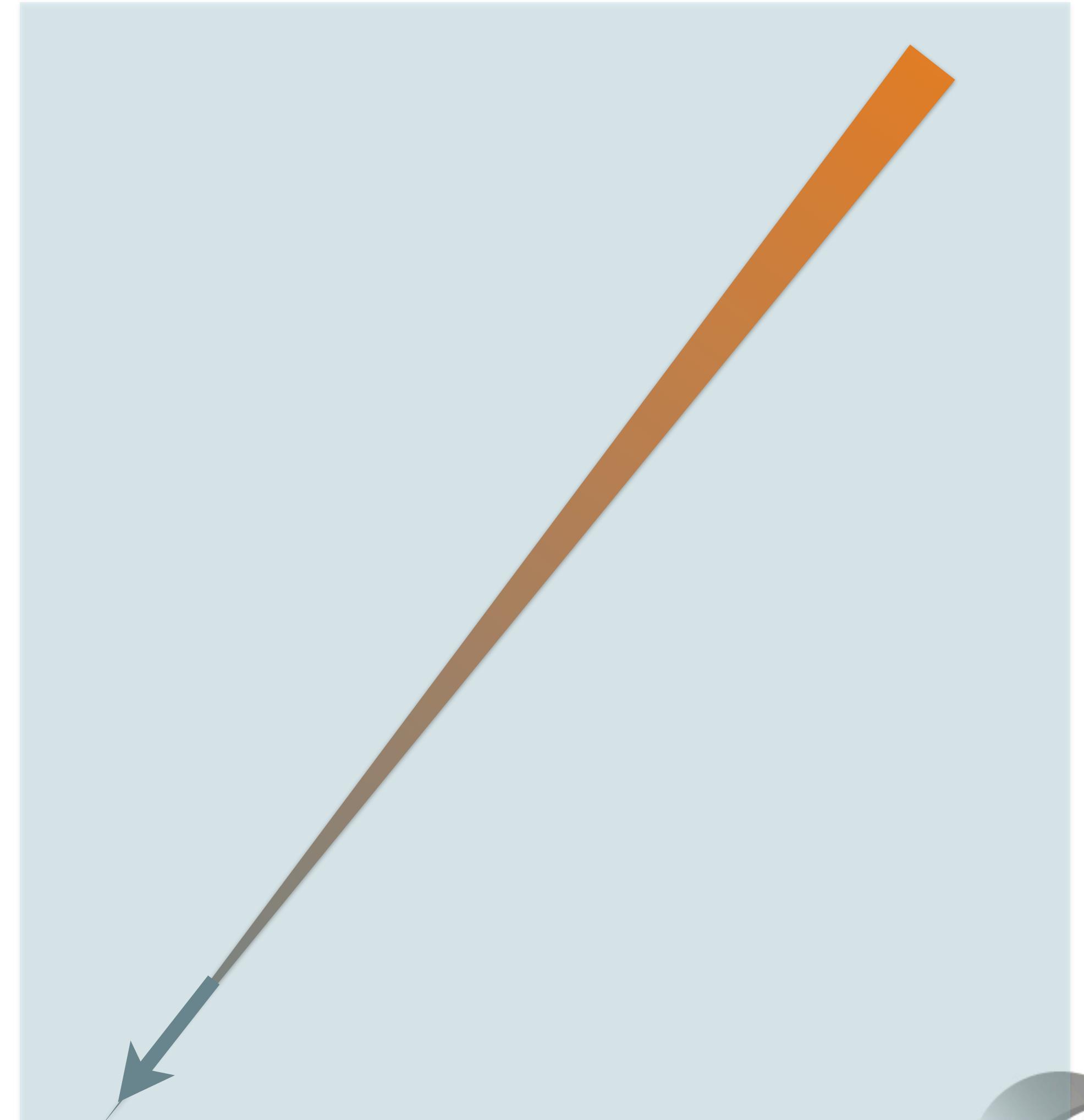
Other homogeneous media noticeably absorb light in the visible spectrum - in this case more blue light than red is absorbed.

Scale



Scale is important. For example, water does absorb a little light in the red end of the visible spectrum, but it's not noticeable over a few inches.

Scale

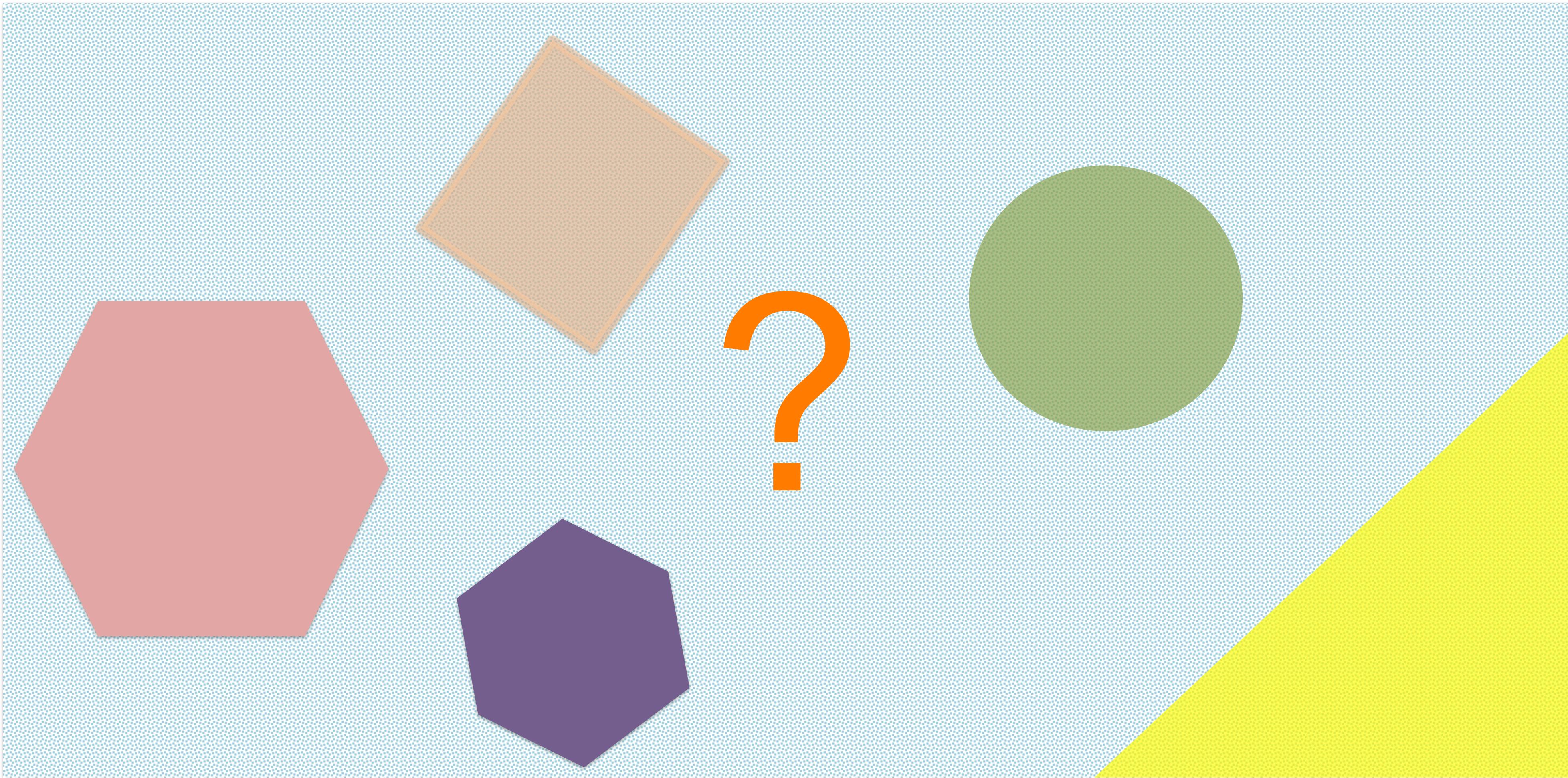


SIGGRAPH 2012



But this absorption is quite significant over distances of dozens of yards.

What if the medium isn't homogeneous?

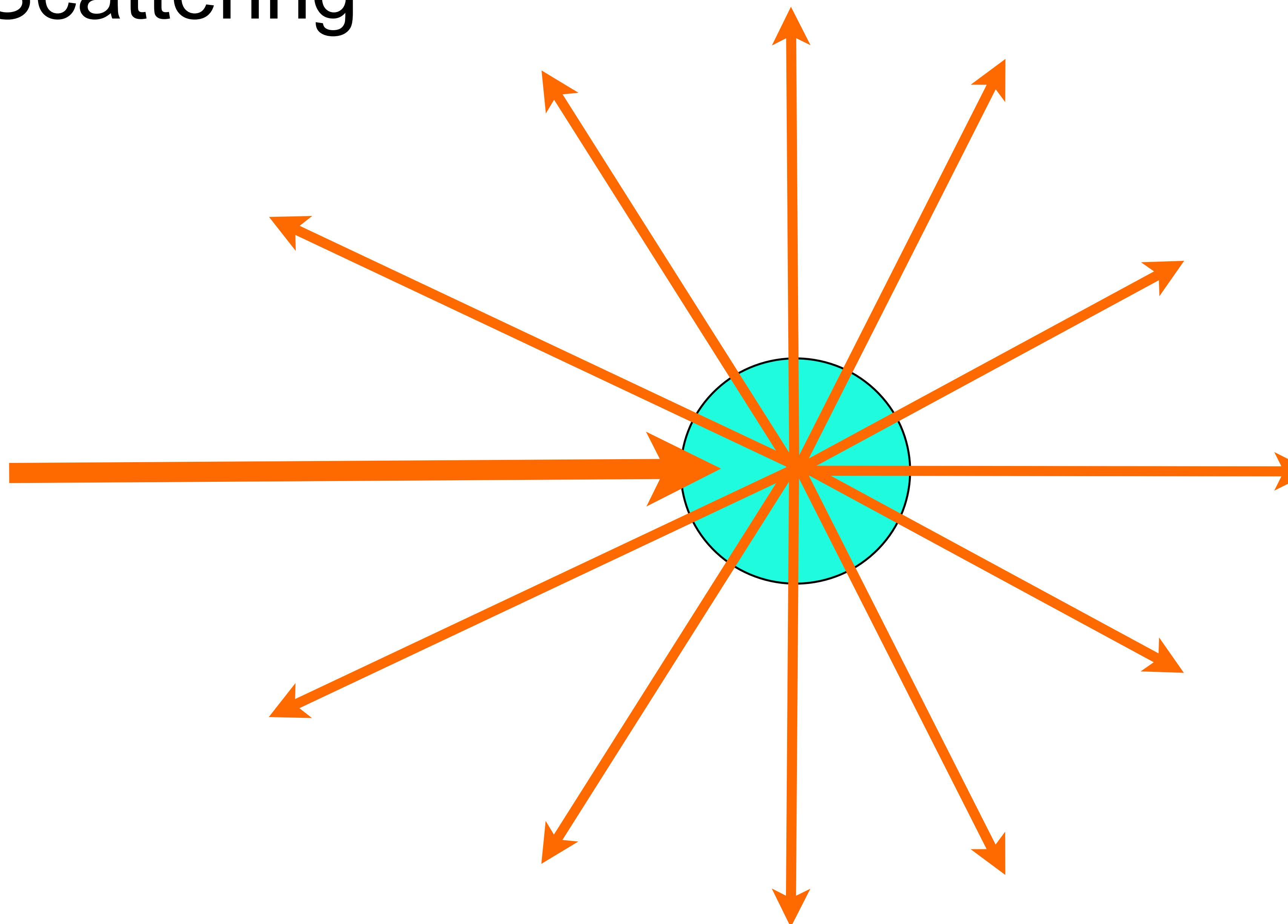


SIGGRAPH 2012



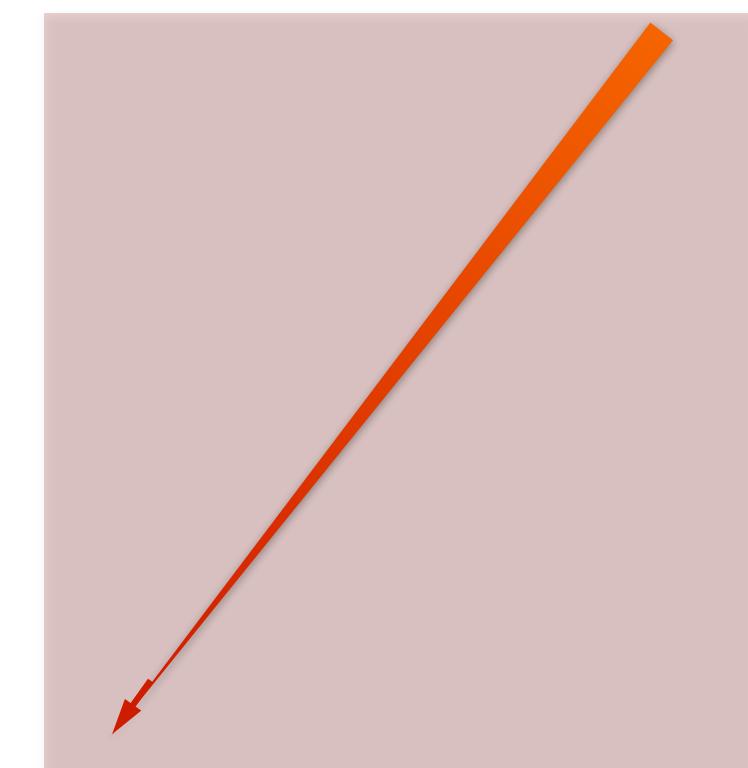
That means the index of refraction (the property of matter that affects light) changes. Will it still move in a straight line?

Scattering

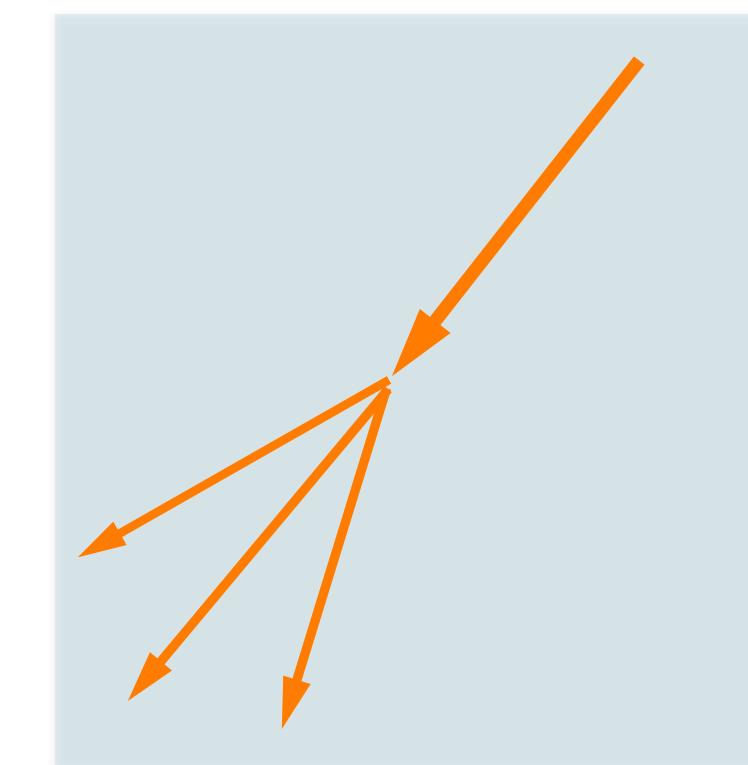


Interactions of Light and Matter

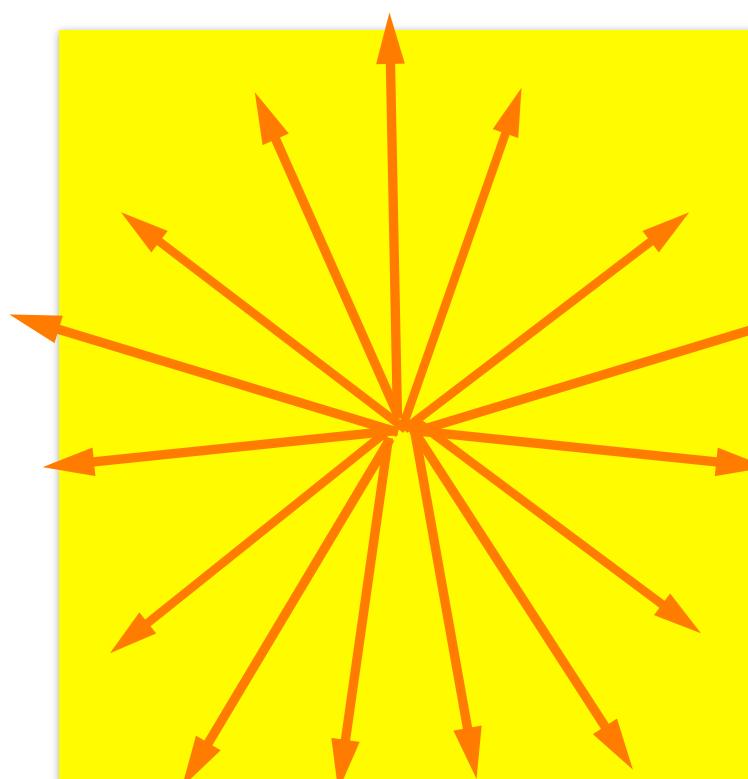
- Absorption: intensity / color changes, direction unchanged



- Scattering: direction changes, intensity & color unchanged

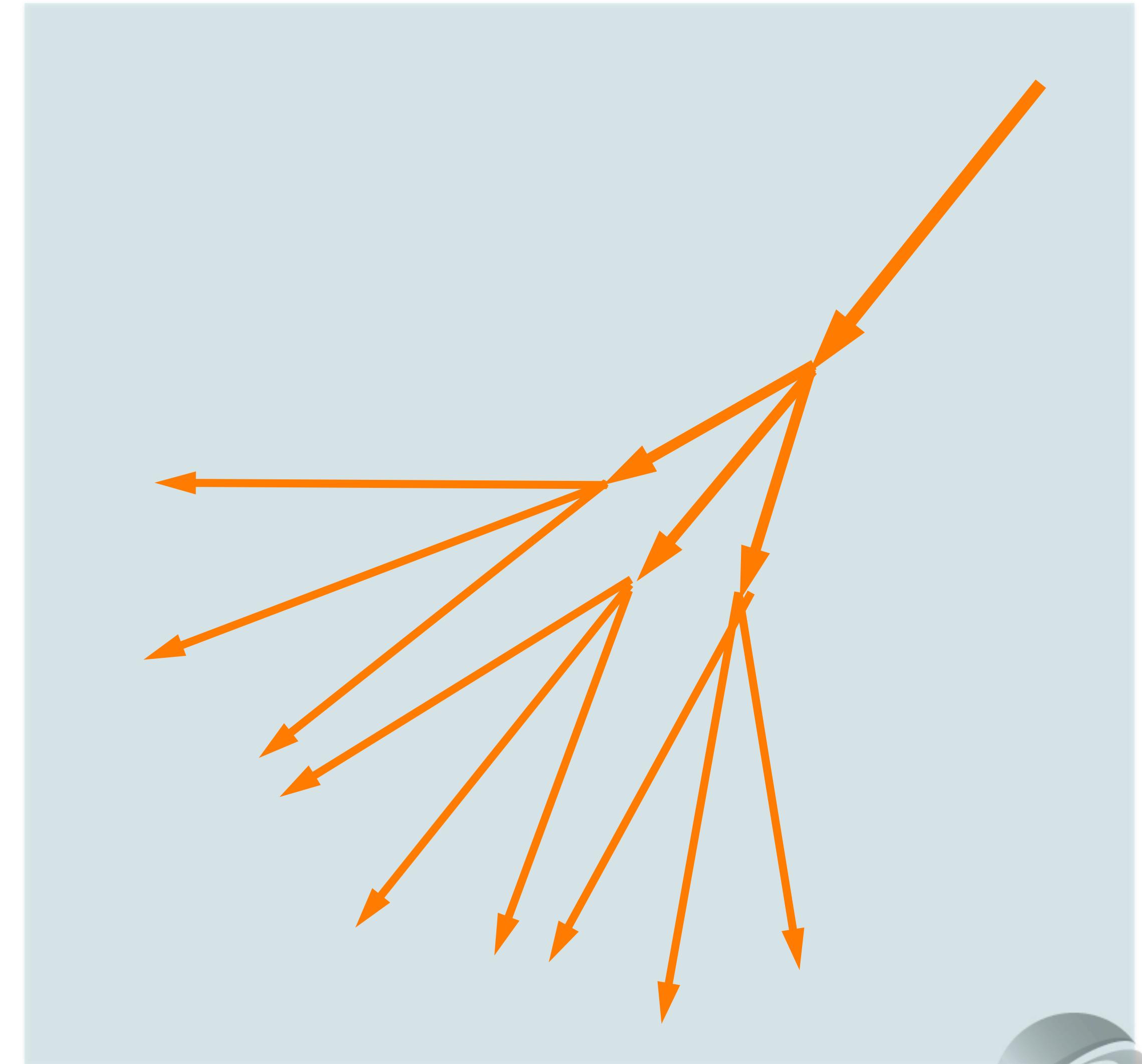


- Emission (new light created; not relevant for most materials)



To summarize, we have three basic modes of light / matter interaction: absorption, scattering, and emission (the last one will not be discussed further).

Cloudy Media



SIGGRAPH 2012

Cloudy media cause enough scattering to somewhat randomize the light direction.

Scale Matters For Scattering Too

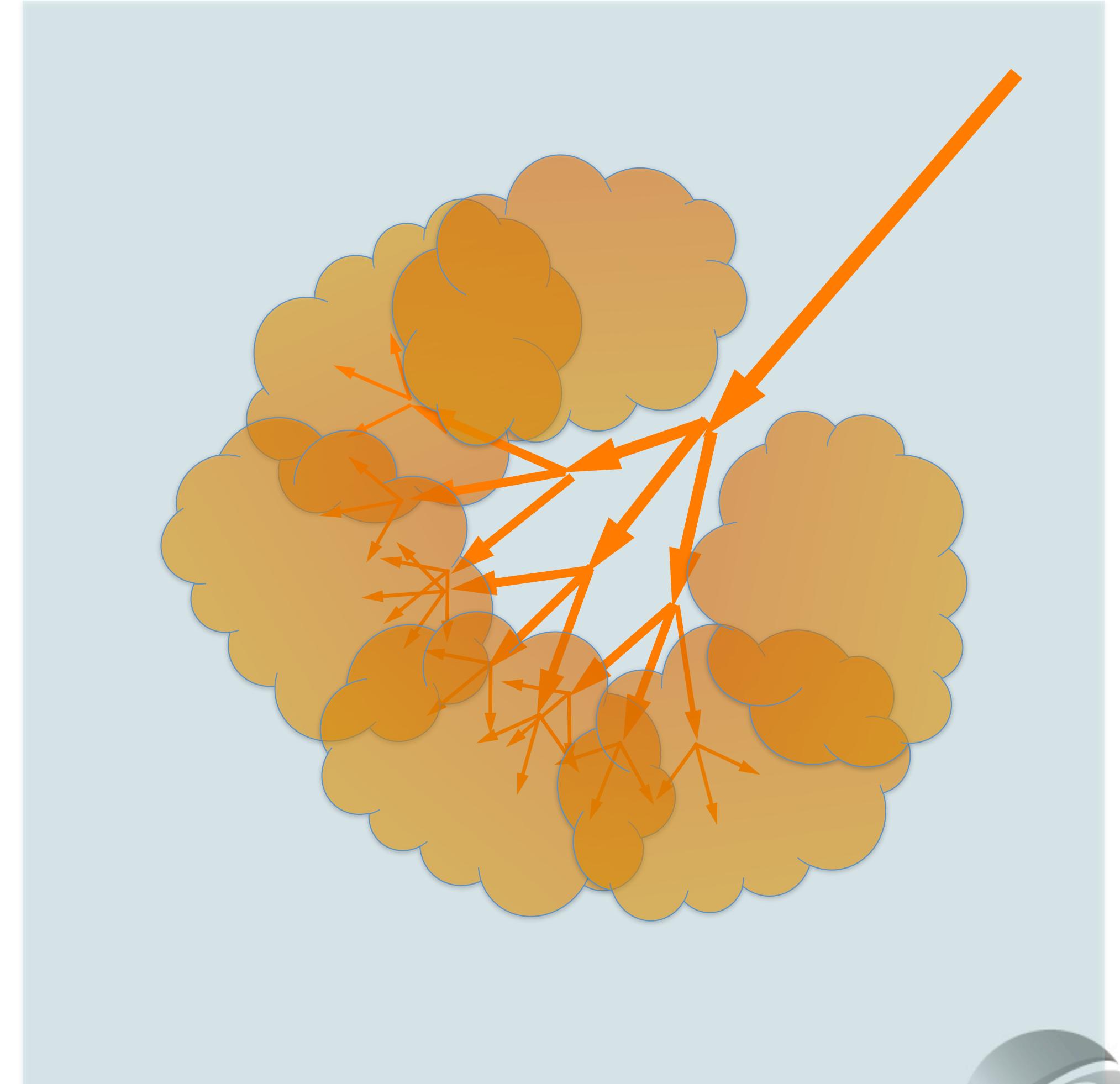


SIGGRAPH 2012



Like absorption, scale matters for scattering: air doesn't scatter noticeably over a few yards, but it does over a scale of miles.

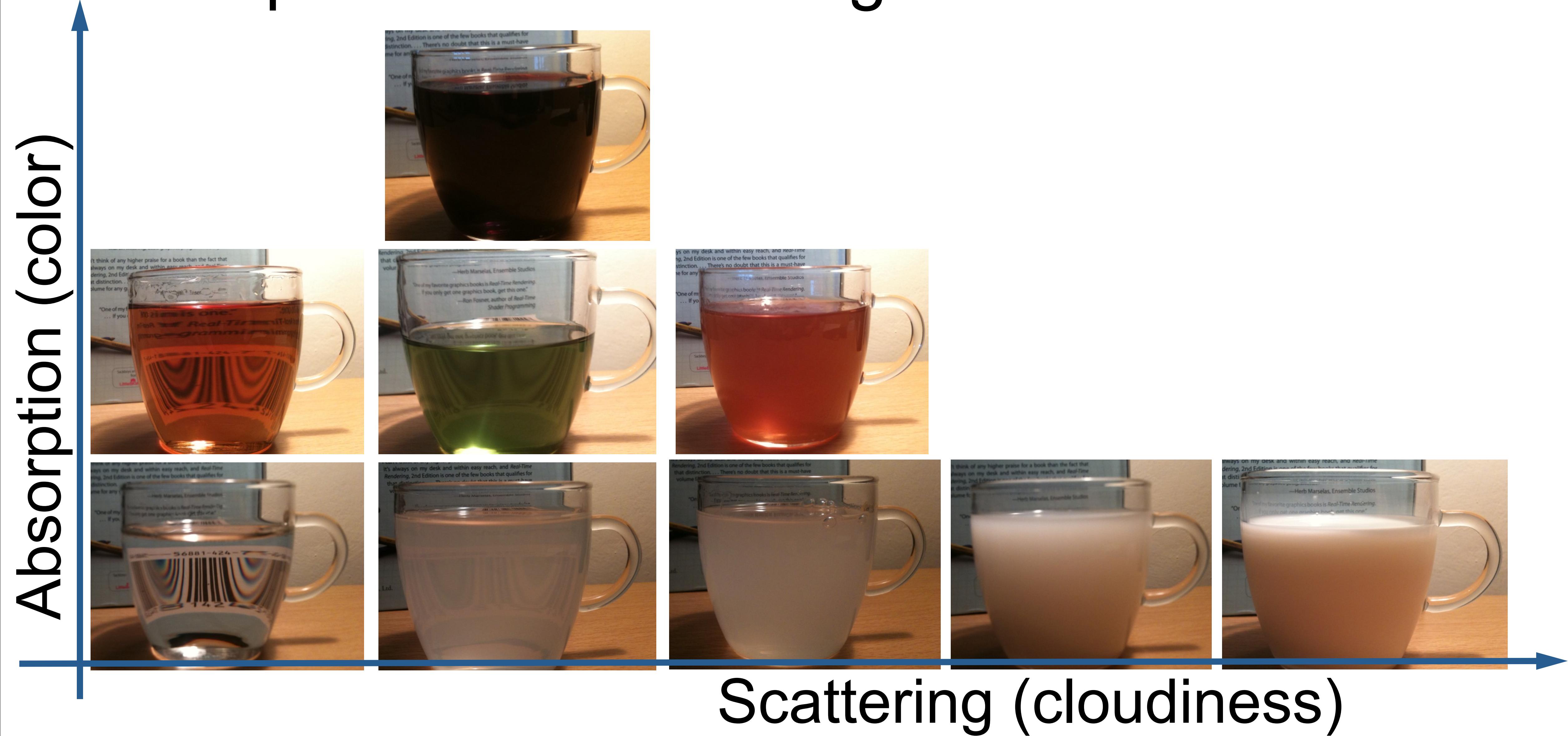
Opaque or Translucent Media



SIGGRAPH 2012

Opaque (or translucent) media cause enough scattering to completely randomize the light direction.

Absorption and Scattering



The appearance of a homogeneous medium is determined by the combination of its absorption and scattering properties.

Scattering at a Plane: Reflection & Refraction

- This special scattering splits light in two directions: reflection and refraction
- Refracted light may be absorbed, scattered under the surface

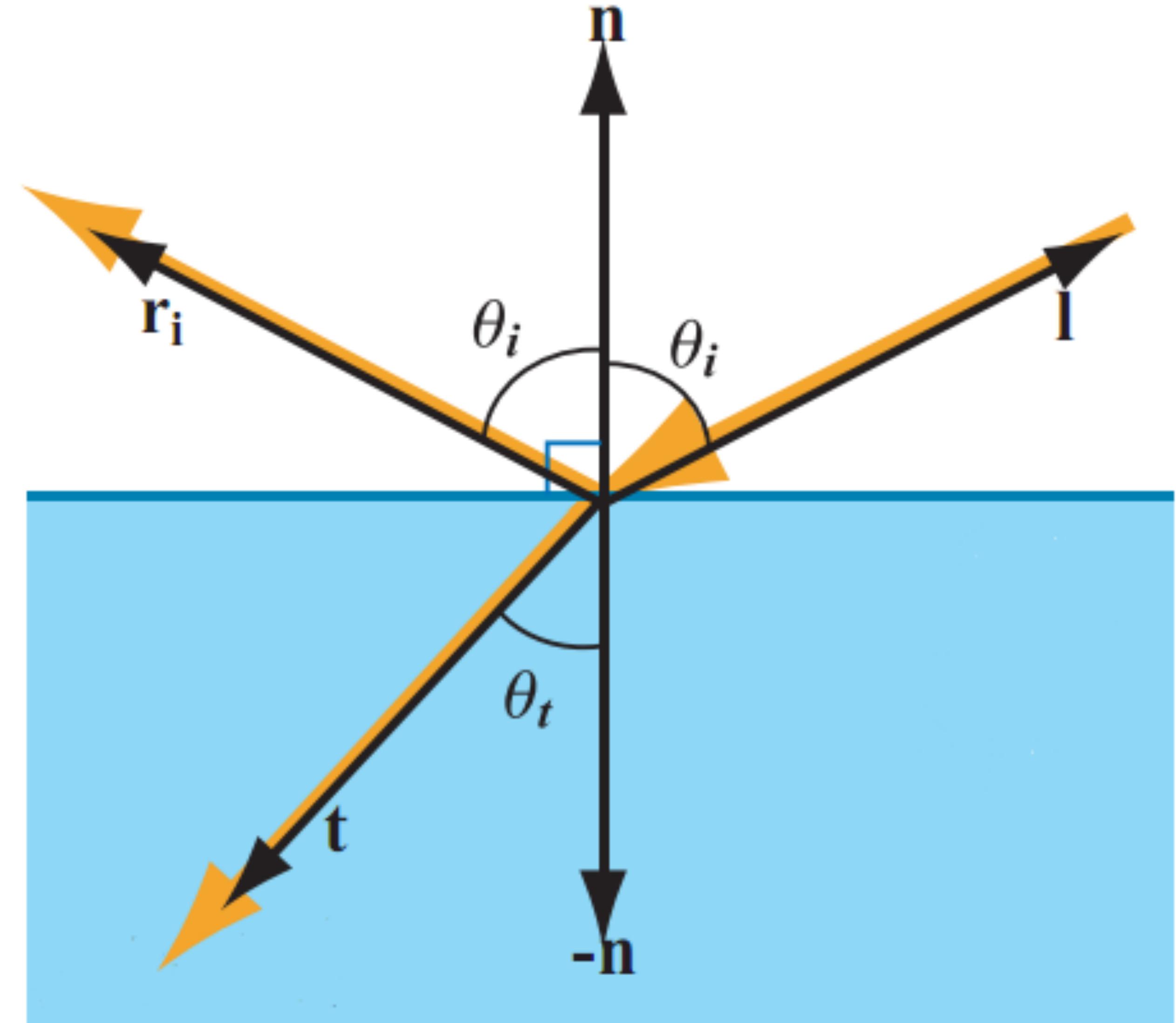
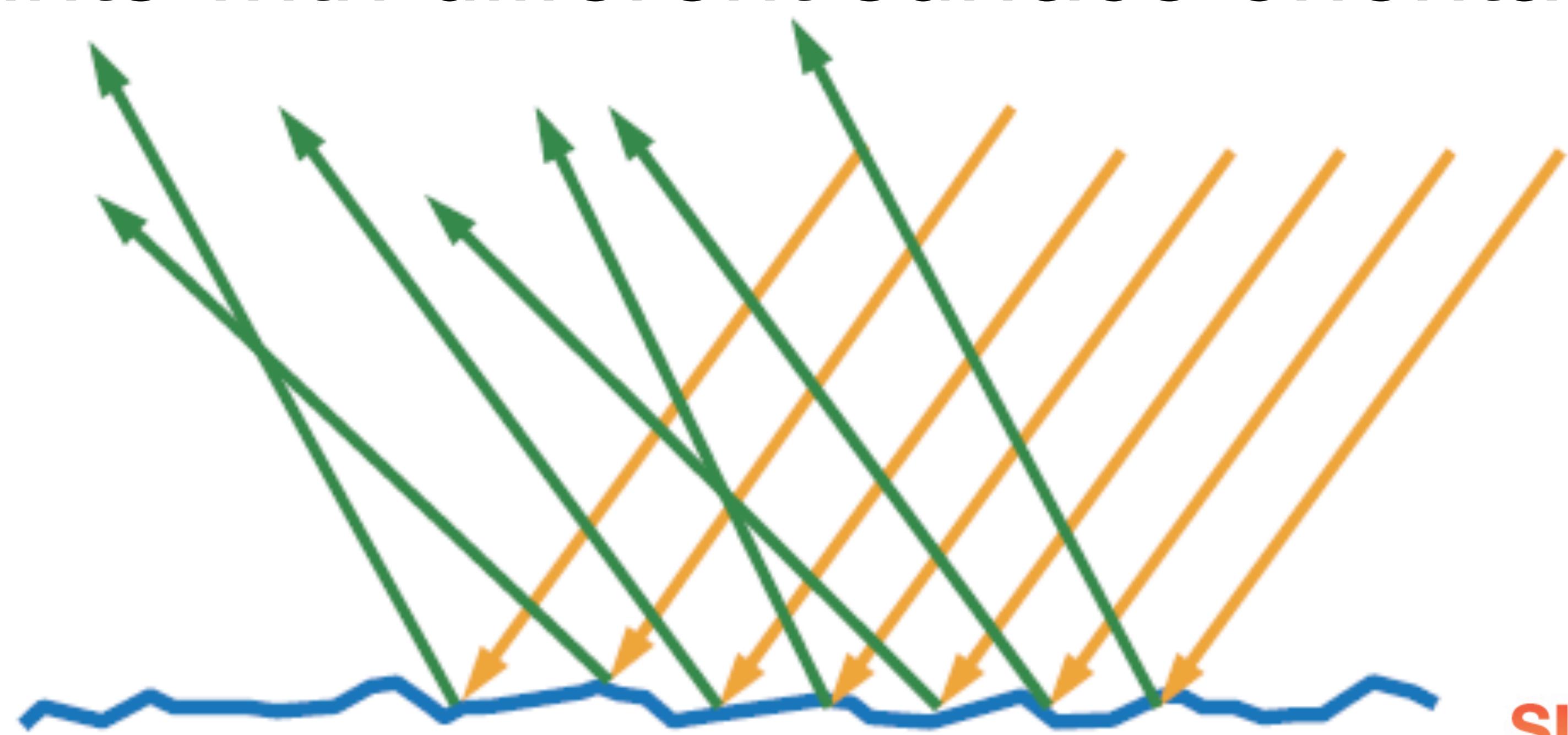


Image from "Real-Time Rendering, 3rd Edition", A K Peters 2008

Important scattering special case: a perfectly flat (up to the size of light waves) plane between two indices of refraction. Instead of splitting up into an arbitrary number of scattering directions (as would happen in the general case), in this case light splits into exactly two directions: reflection and refraction.

Non-Optically-Flat Surfaces: Microgeometry

- Each surface point reflects light in one direction
- Surface appearance is the aggregate result of many points with different surface orientations



SIGGRAPH 2012

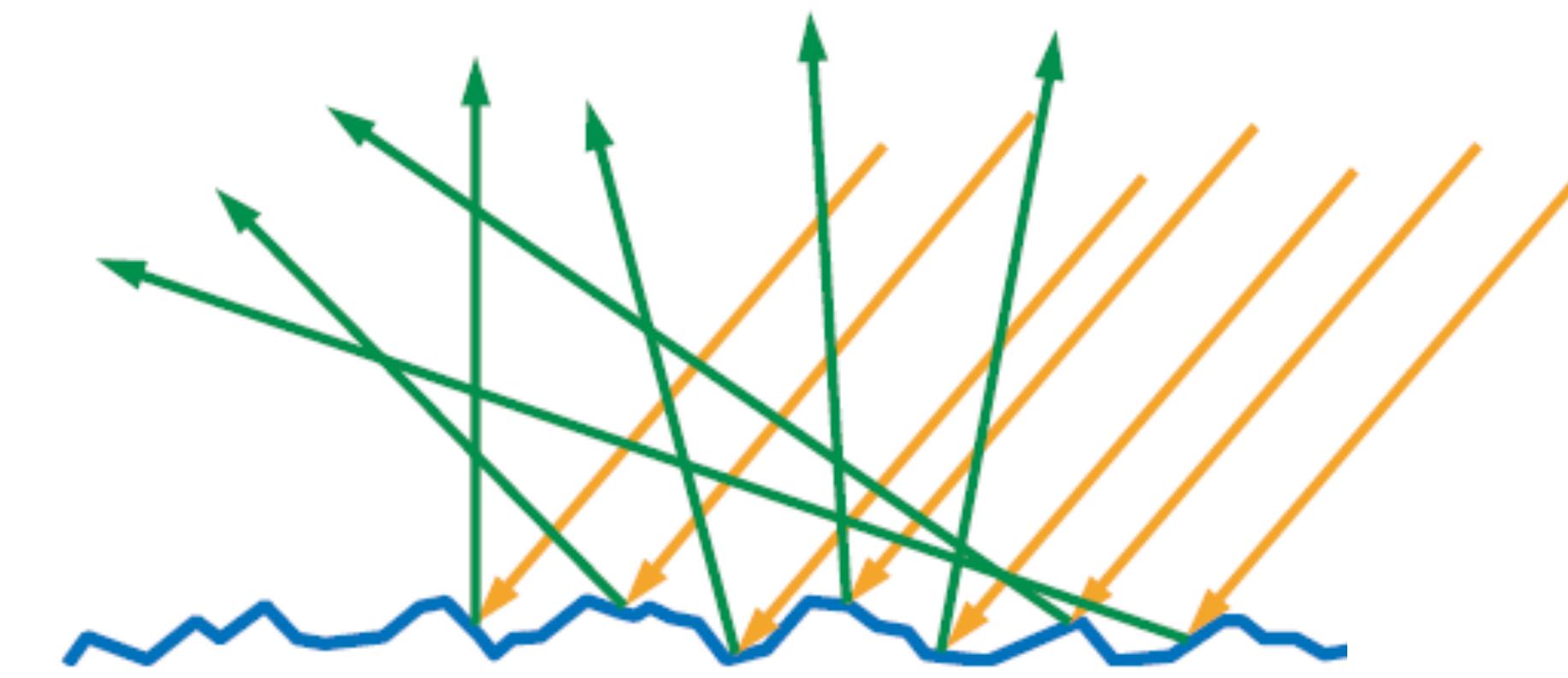
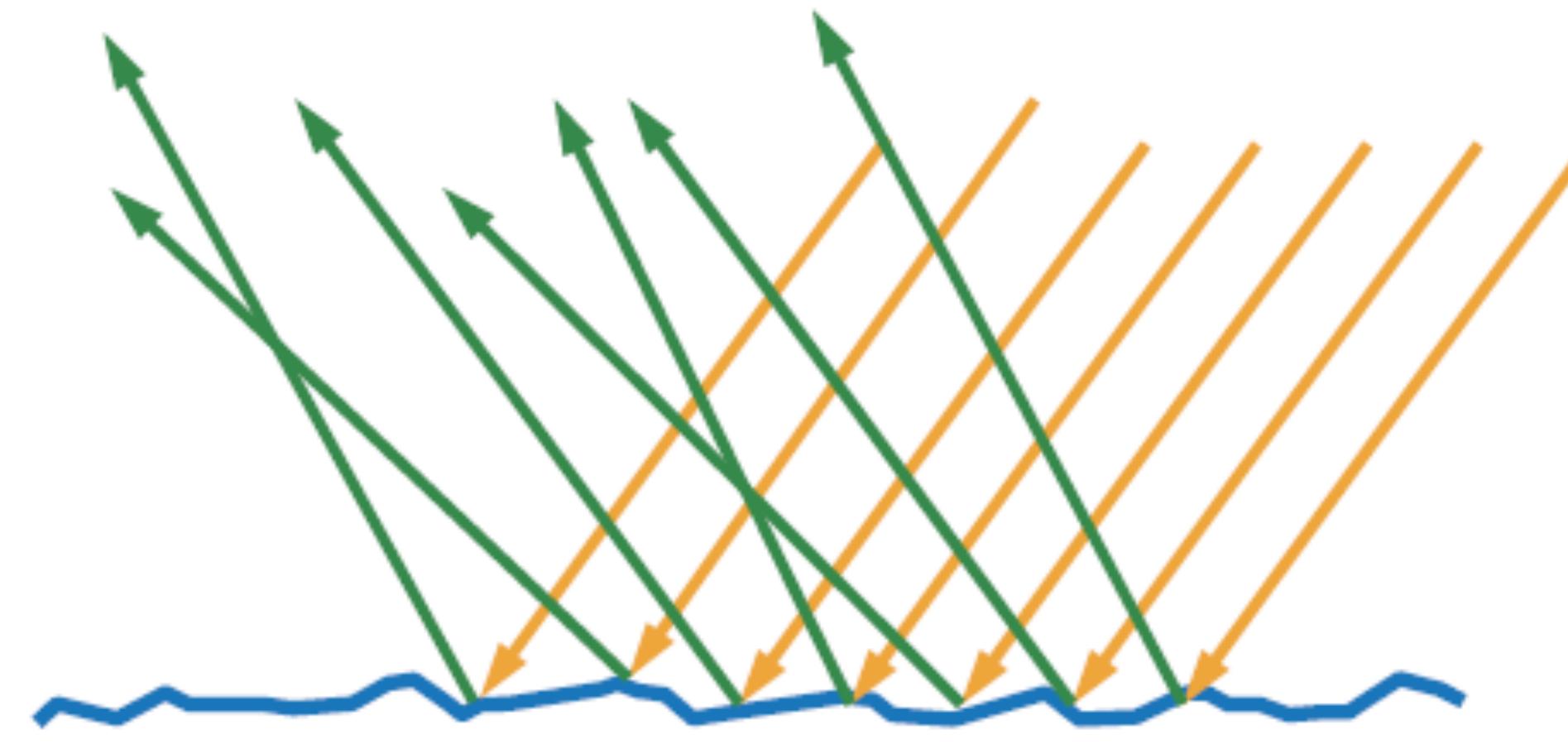


Image from "Real-Time Rendering, 3rd Edition", A K Peters 2008

16

A surface which is flat up to the size of light wavelengths (e.g. high-end telescope optics) is *optically flat* - however, most surfaces have microgeometry (bumps bigger than a light wavelength but smaller than a pixel) that affect light reflection.

Rougher = Blurrier Reflections



SIGGRAPH 2012

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

17

Both of these surfaces look equally smooth to the naked eye; however under a microscope we can see the difference in roughness. The surface on the top is only slightly bumpy; the incoming light rays hit bits of the surface that are angled slightly differently and get reflected to slightly different outgoing directions. The surface on the bottom is very rugged on the microscopic scale - each incoming light ray hits a piece of surface that is oriented very differently, spreading the outgoing light rays over a wide cone of directions. We can see the results in the differences between the reflections off the two objects.

Macroscopic View

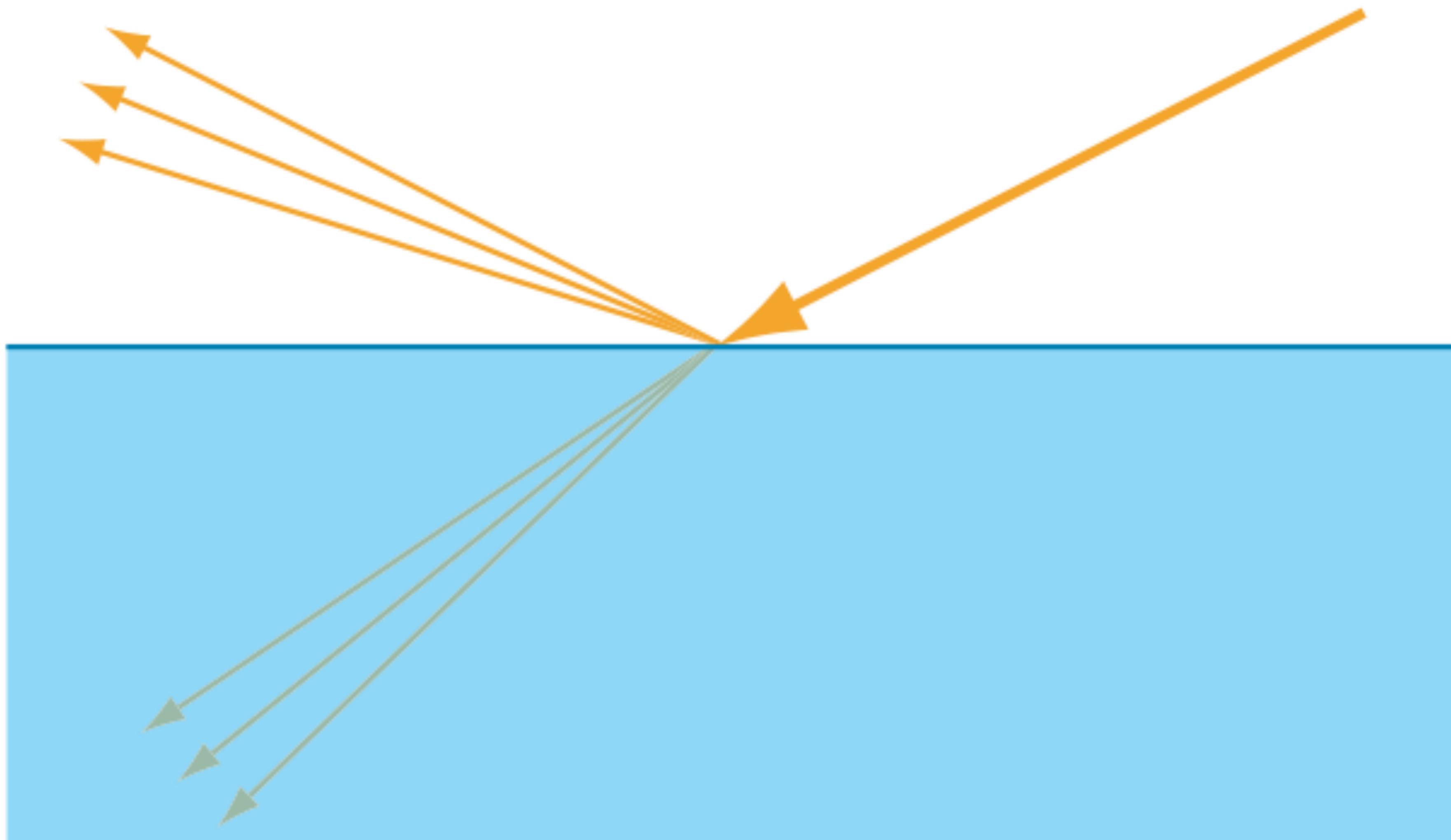
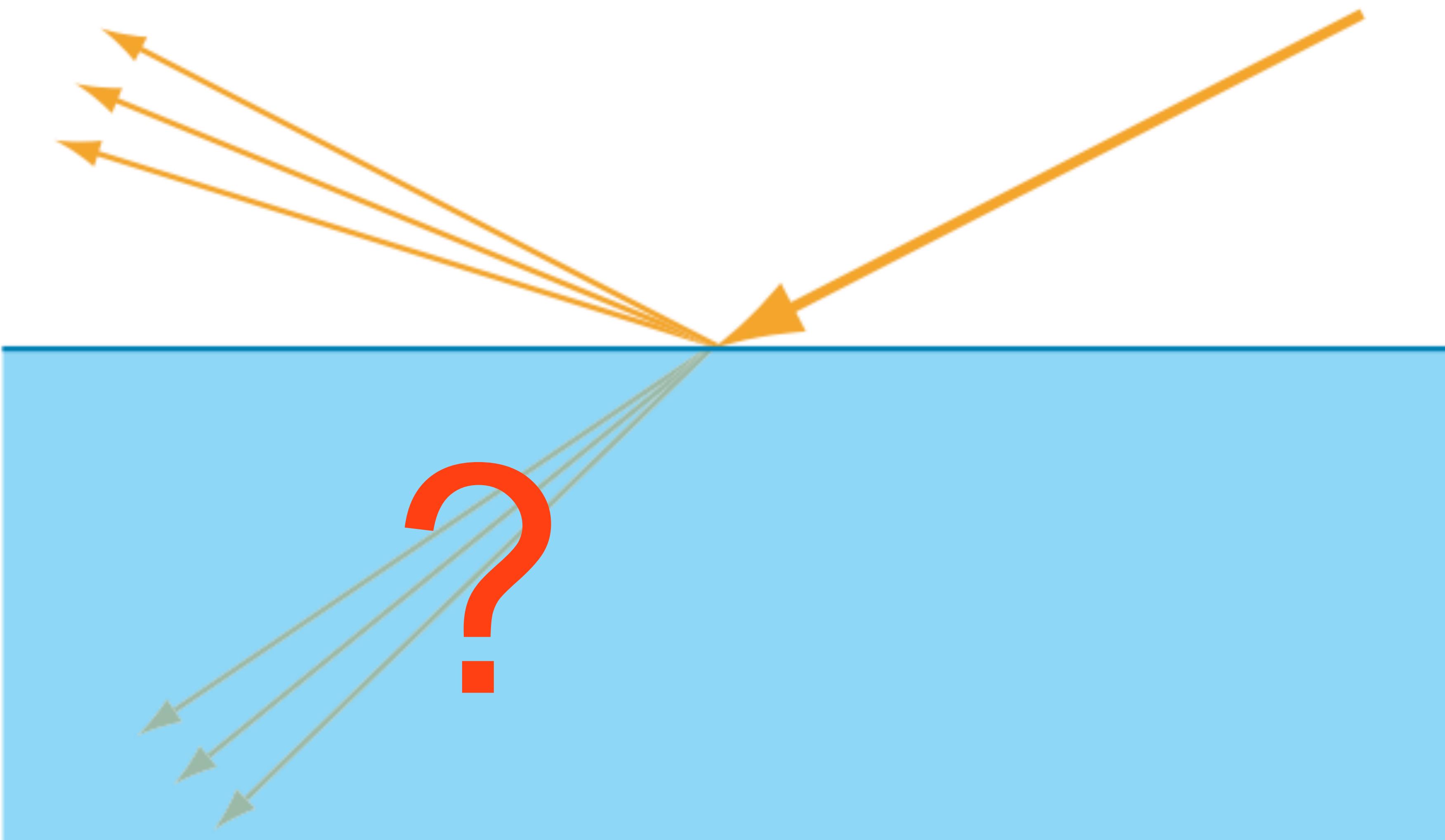


Image from "Real-Time Rendering, 3rd Edition", A K Peters 2008

SIGGRAPH 2012



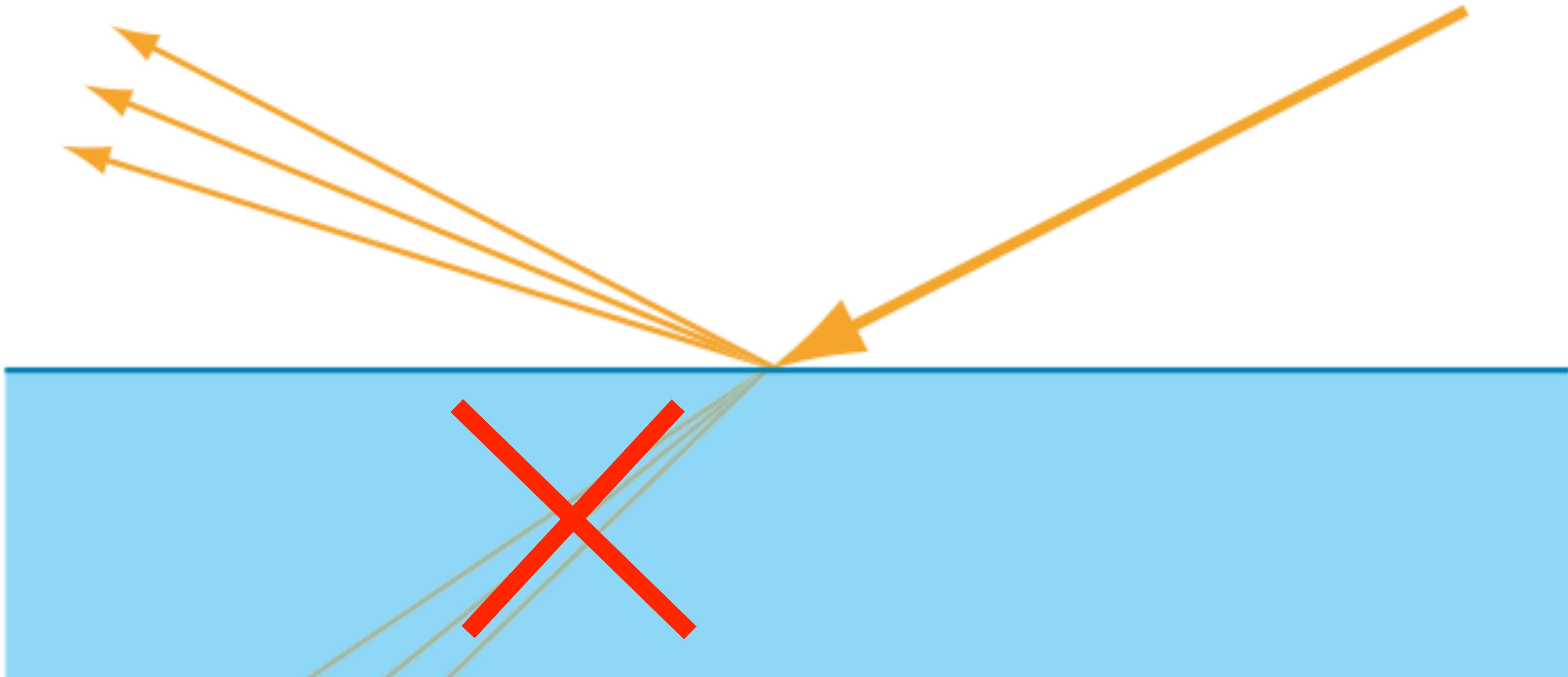
In the macroscopic view, we treat the microgeometry statistically and view the surface as reflecting (and refracting) light in multiple directions.



SIGGRAPH 2012

What happens to the refracted light? It depends what kind of material the object is made of.

Metals



Non-Metals (Insulators, Dielectrics)

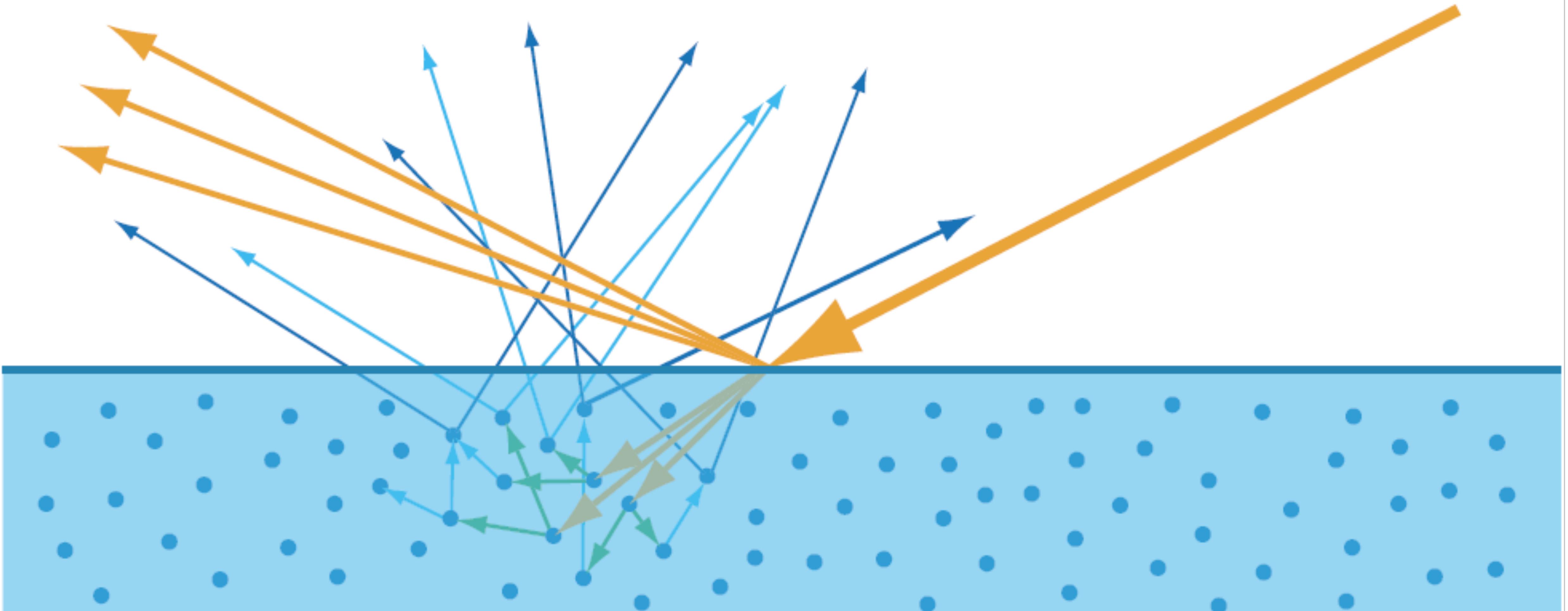


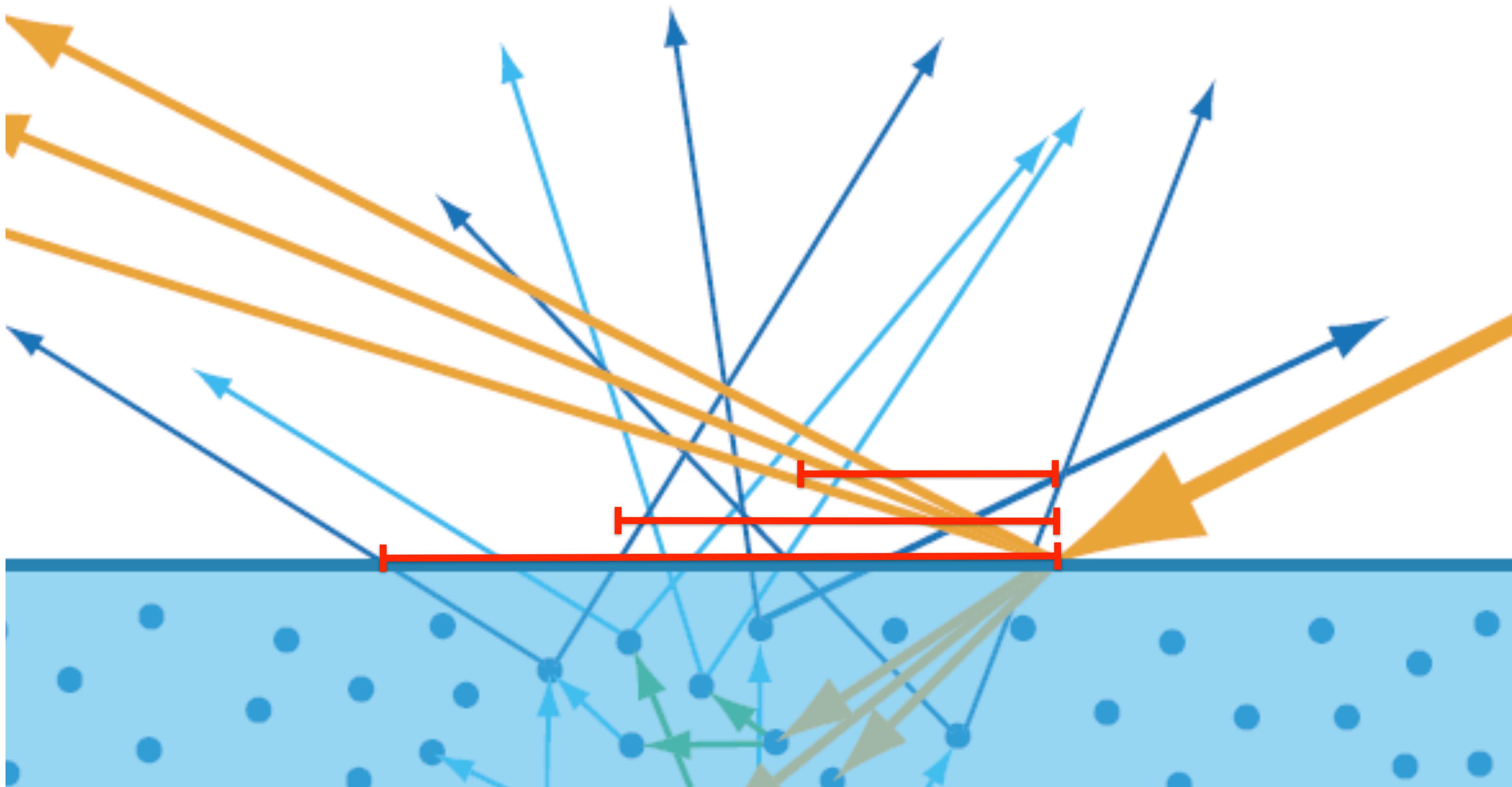
Image from "Real-Time Rendering, 3rd Edition", A K Peters 2008

SIGGRAPH 2012



Non-metals behave like those cups of liquid we saw earlier. They cause various degrees of absorption and scattering; the refracted light gets scattered and absorbed inside the material and some of it comes back out.

Scale and Subsurface Scattering



SIGGRAPH 2012



The re-emitted light comes out at varying distances from the entry point. The distribution of distances depends on the density and properties of the scattering particles.

Scale and Subsurface Scattering

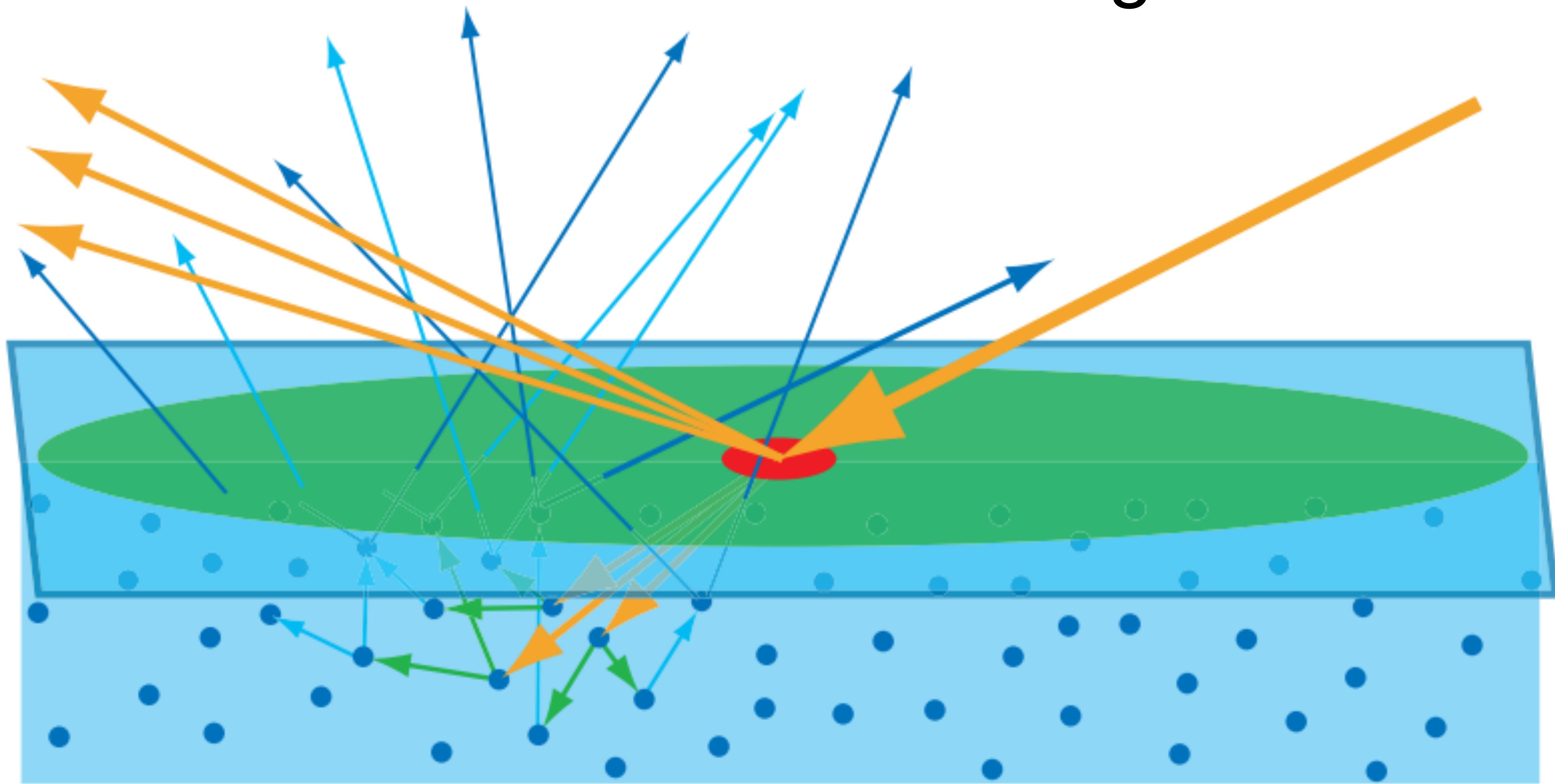


Image from “Real-Time Rendering, 3rd Edition”, A K Peters 2008

23

If the pixel size (or distance between samples) is large (green circle) compared to the entry-exit distances, we can assume the distances are effectively zero.

Scale and Subsurface Scattering

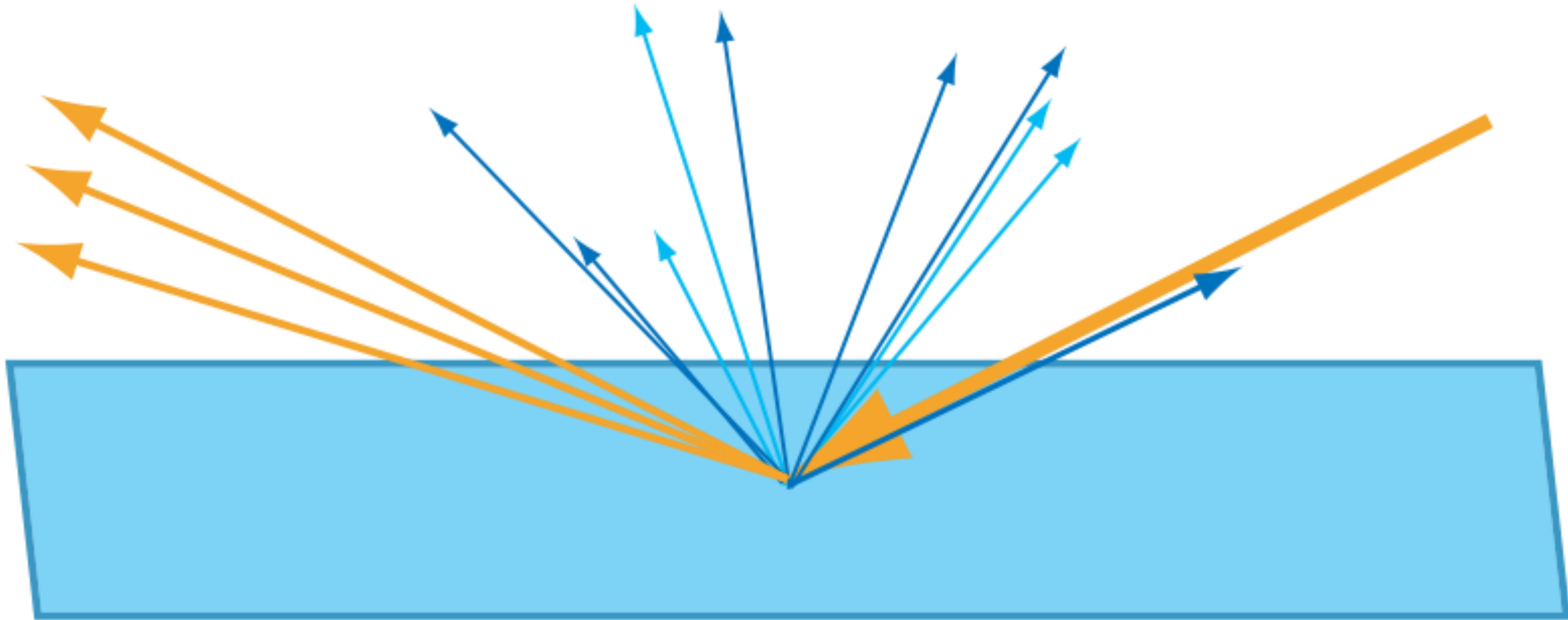


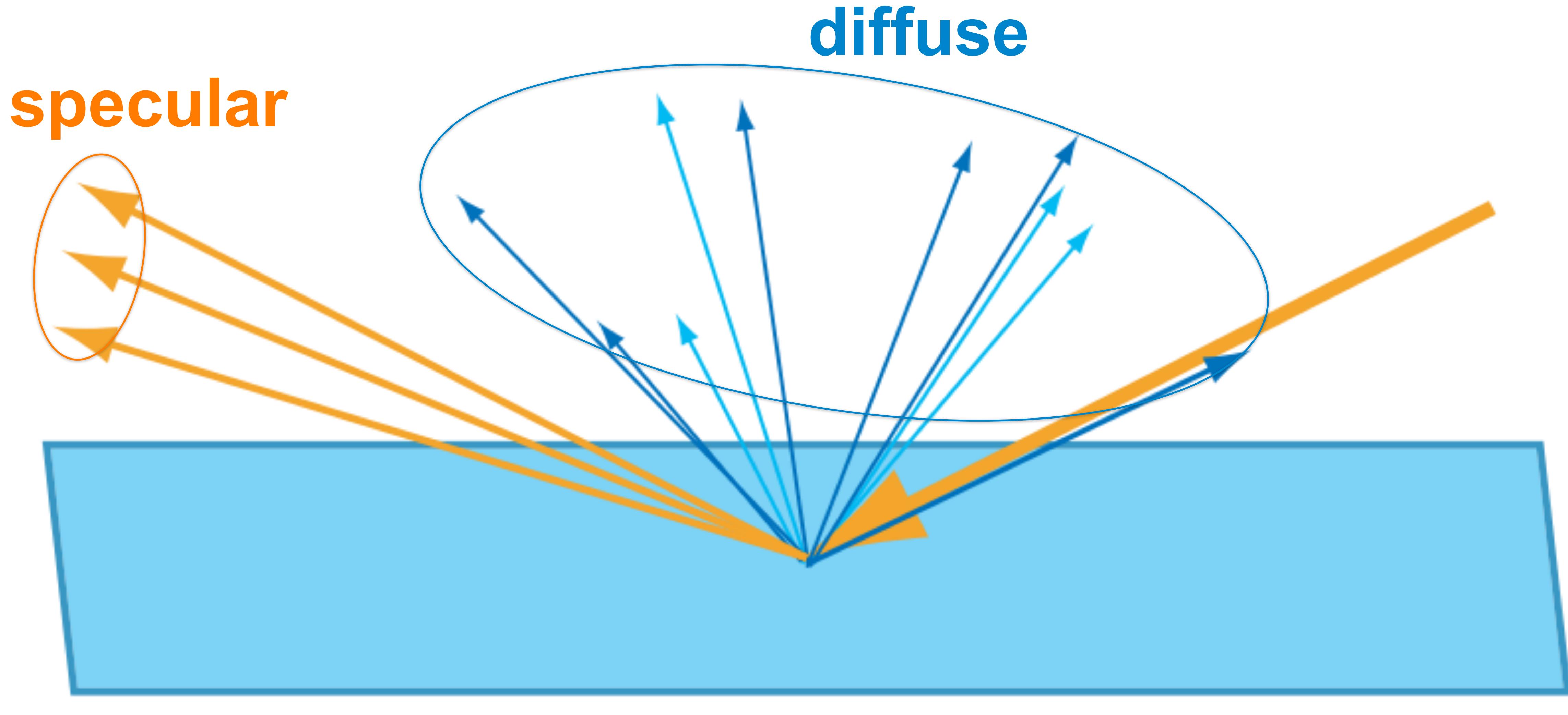
Image from "Real-Time Rendering, 3rd Edition", A K Peters 2008

SIGGRAPH 2012



By ignoring the entry-to-exit distance, we can then compute all shading locally at a single point. The shaded color is only affected by light hitting that surface point.

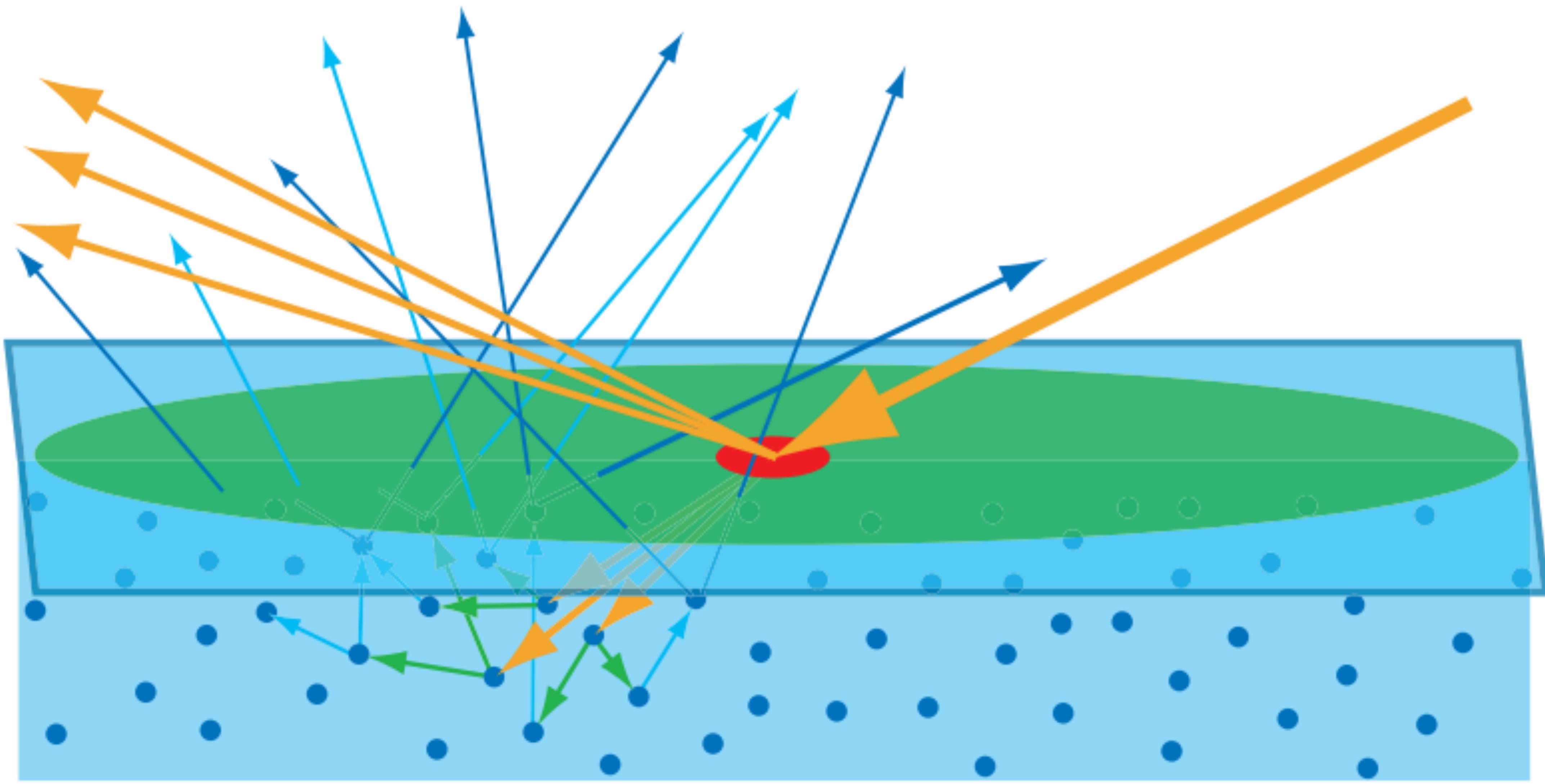
Shading Terms



25

It is convenient to split these two very different light-material interactions (surface reflection and refraction-absorption-scattering) into different shading terms - the familiar “specular” and “diffuse”.

Scale and Subsurface Scattering



26

If the pixel is small (red circle) compared to the entry-exit distances, you can't ignore the distances and need specialized rendering techniques. These are referred to as "subsurface scattering", but diffuse is always subsurface scattering - the difference is the scale. So distant skin or marble can use normal rendering techniques, and plastic in extreme close up would need "subsurface scattering" techniques.

Math of Physically-Grounded Shading

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

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

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



Math of physical shading - to turn light physics into a mathematical model, the first step is to quantify light as a number.

Radiance

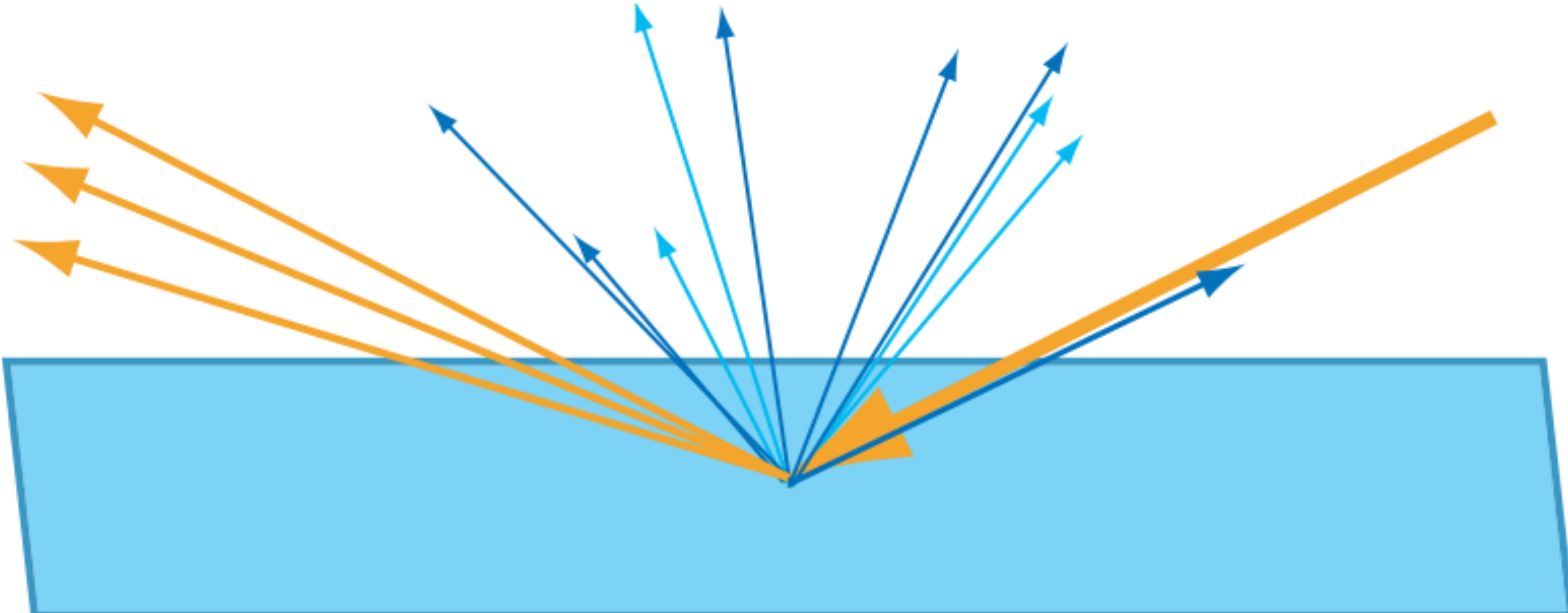
Single Ray

Spectral/RGB



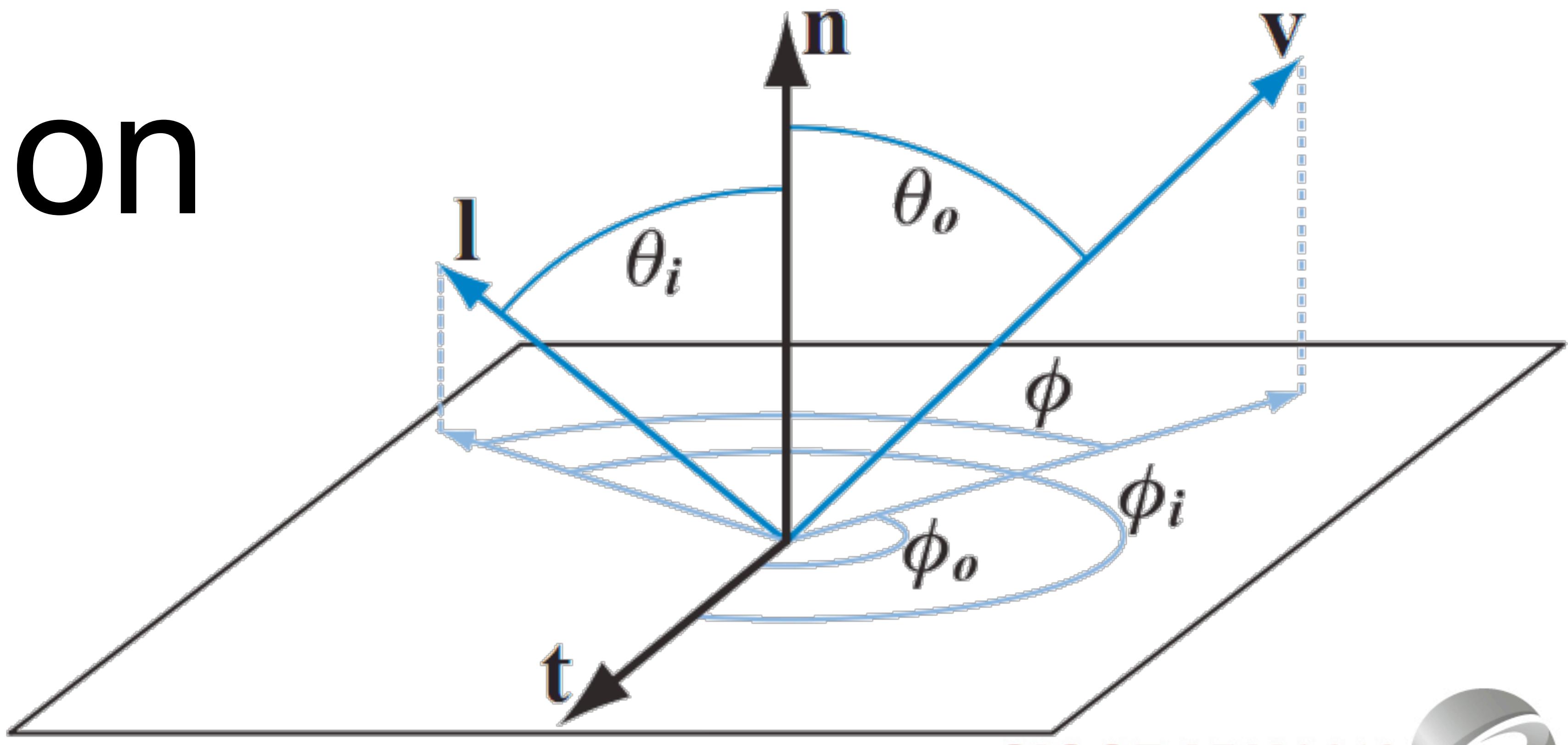
Radiometry is the measurement of light. Of the various radiometric quantities, we'll use *radiance*, which measures the intensity of light along a single ray. Radiance is spectral (varies with wavelength) - it's technically a continuous spectral power distribution but for production purposes it's represented as an RGB value.

Local Shading



Bidirectional Reflectance Distribution Function

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



SIGGRAPH 2012

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

30

We represent this variation with the *BRDF*, a function of light and view direction. In principle, the BRDF is a 4D or 3D function. In practice, a varying number of angles are used to compute it. The BRDF is only defined for light and view vectors above the macroscopic surface. Backfacing light directions should be avoided or at least have their contribution multiplied by 0. Backfacing view directions are theoretically impossible for opaque surfaces but may occur with interpolated normals or normal mapping; these are typically handled either by clamping the $\mathbf{n} \cdot \mathbf{v}$ product to 0 or taking its absolute value to avoid negative values.

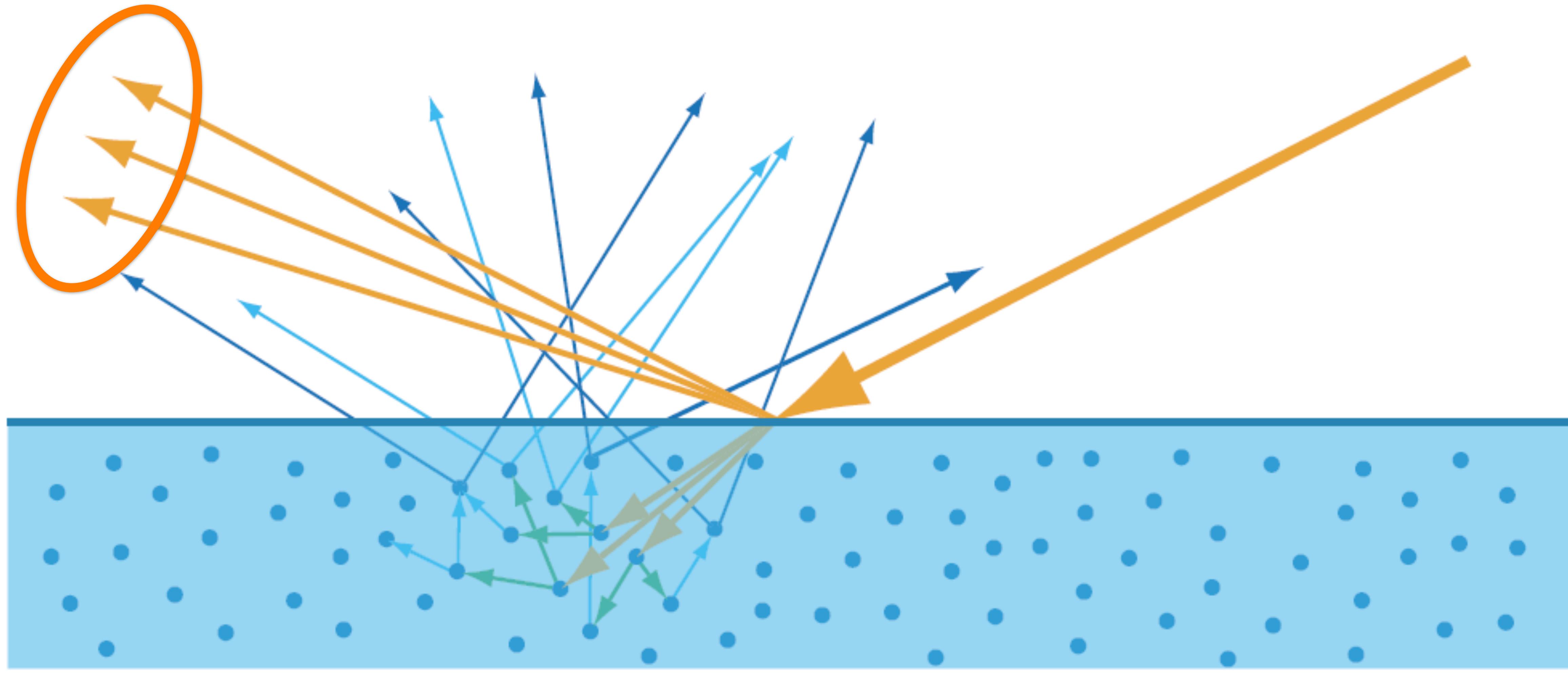
The Reflectance Equation

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



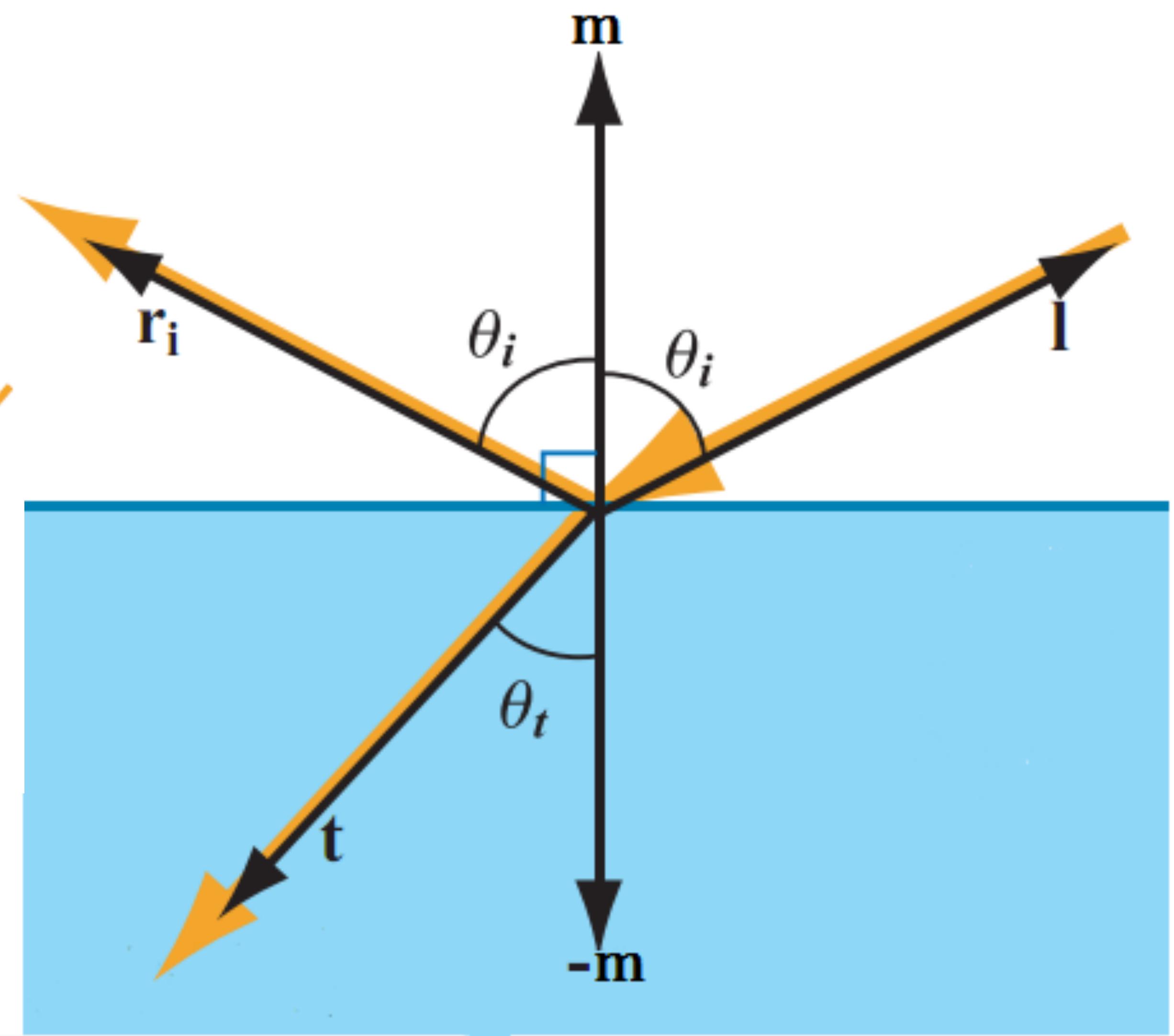
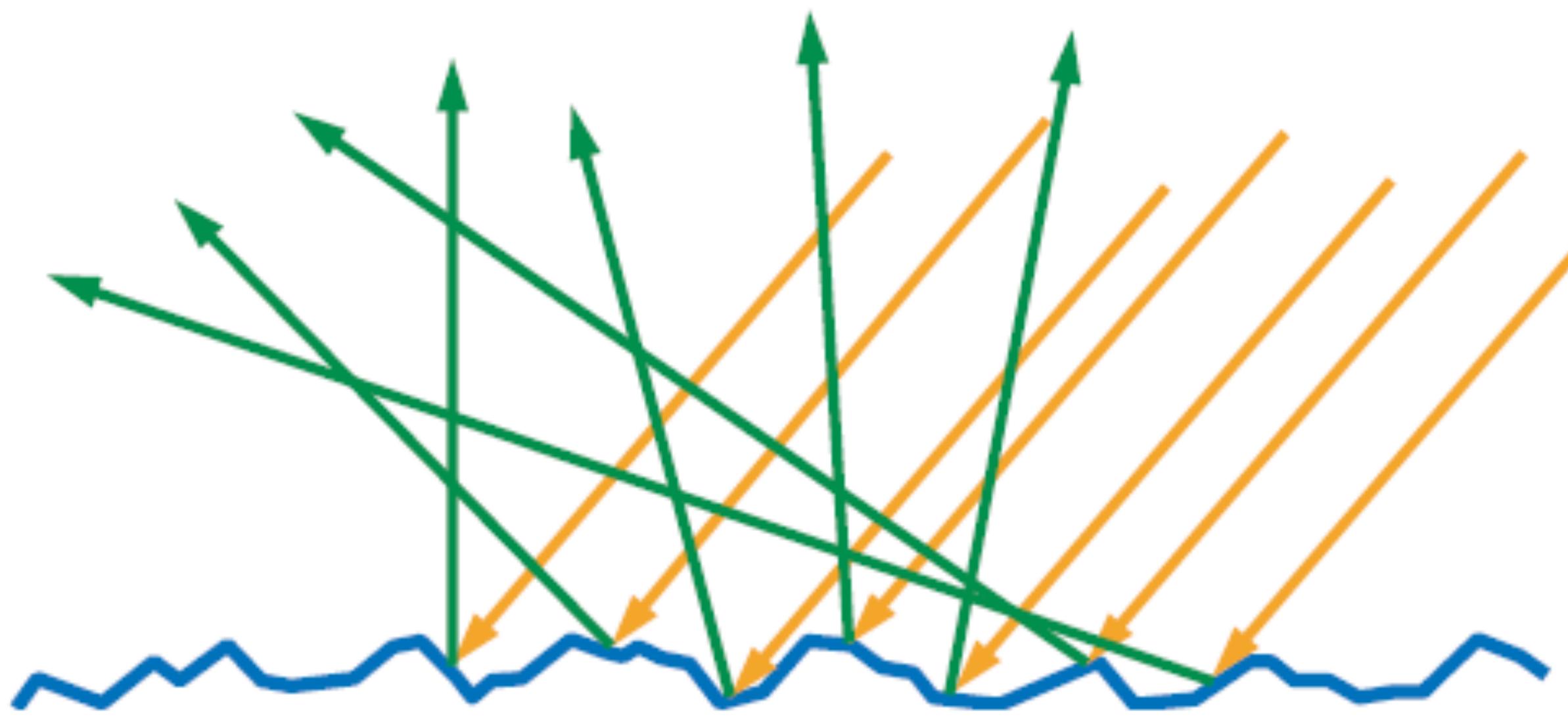
Outgoing radiance equals the integral (over all directions above the surface) of incoming radiance times BRDF and cosine factor; “X in circle” denotes component-wise vector (RGB) multiplication.

Surface Reflection (Specular Term)



We'll start by looking at the specular term.

Microfacet Theory

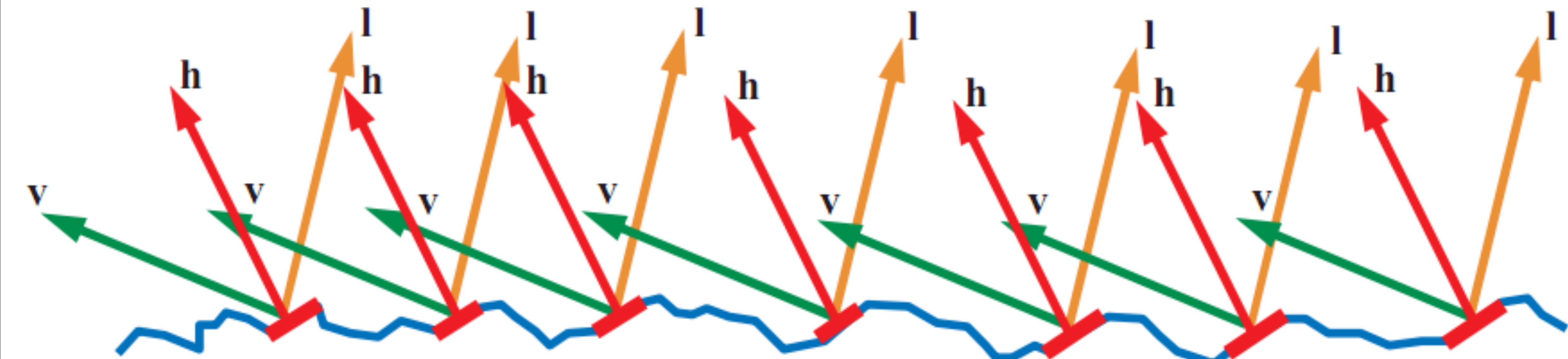


SIGGRAPH 2012



Microfacet theory is a way to derive BRDFs for surface (specular) reflection from general (non-optically flat) surfaces. It assumes the surface is composed of many *microfacets*. Each facet is a perfect mirror (optically flat), so it reflects each incoming ray of light into only one outgoing direction, which depends on the light direction \mathbf{I} and the microfacet normal \mathbf{m} .

The Half Vector

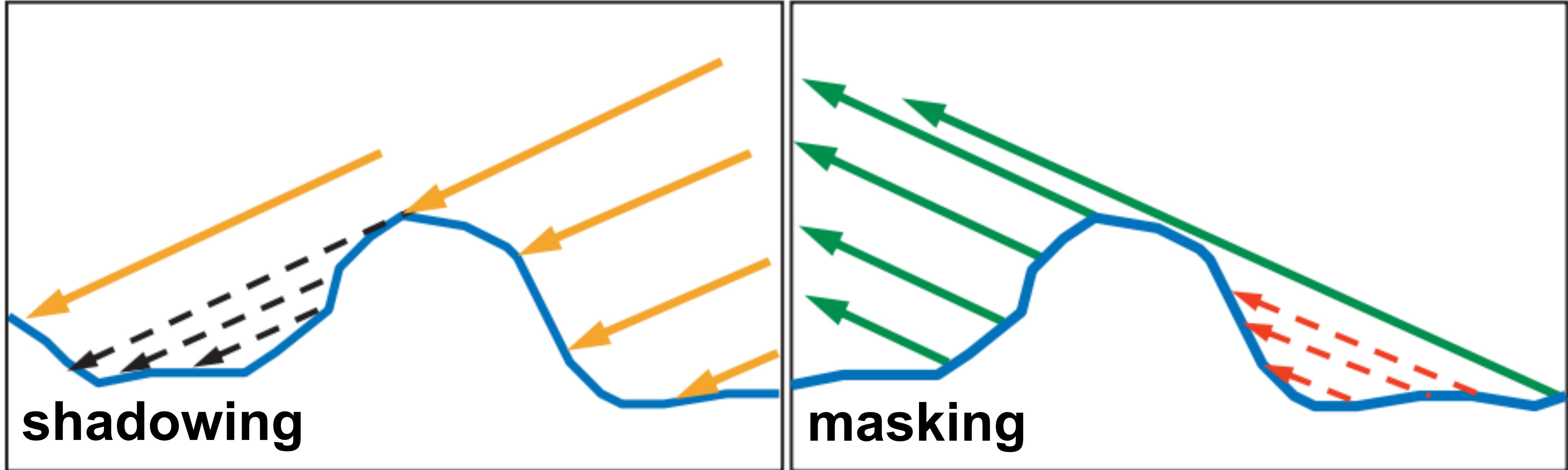


SIGGRAPH

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

Only those microfacets which happen to have their surface normal \mathbf{m} oriented exactly halfway between \mathbf{l} and \mathbf{v} will reflect any visible light - this direction is the half-vector \mathbf{h} .

Shadowing and Masking



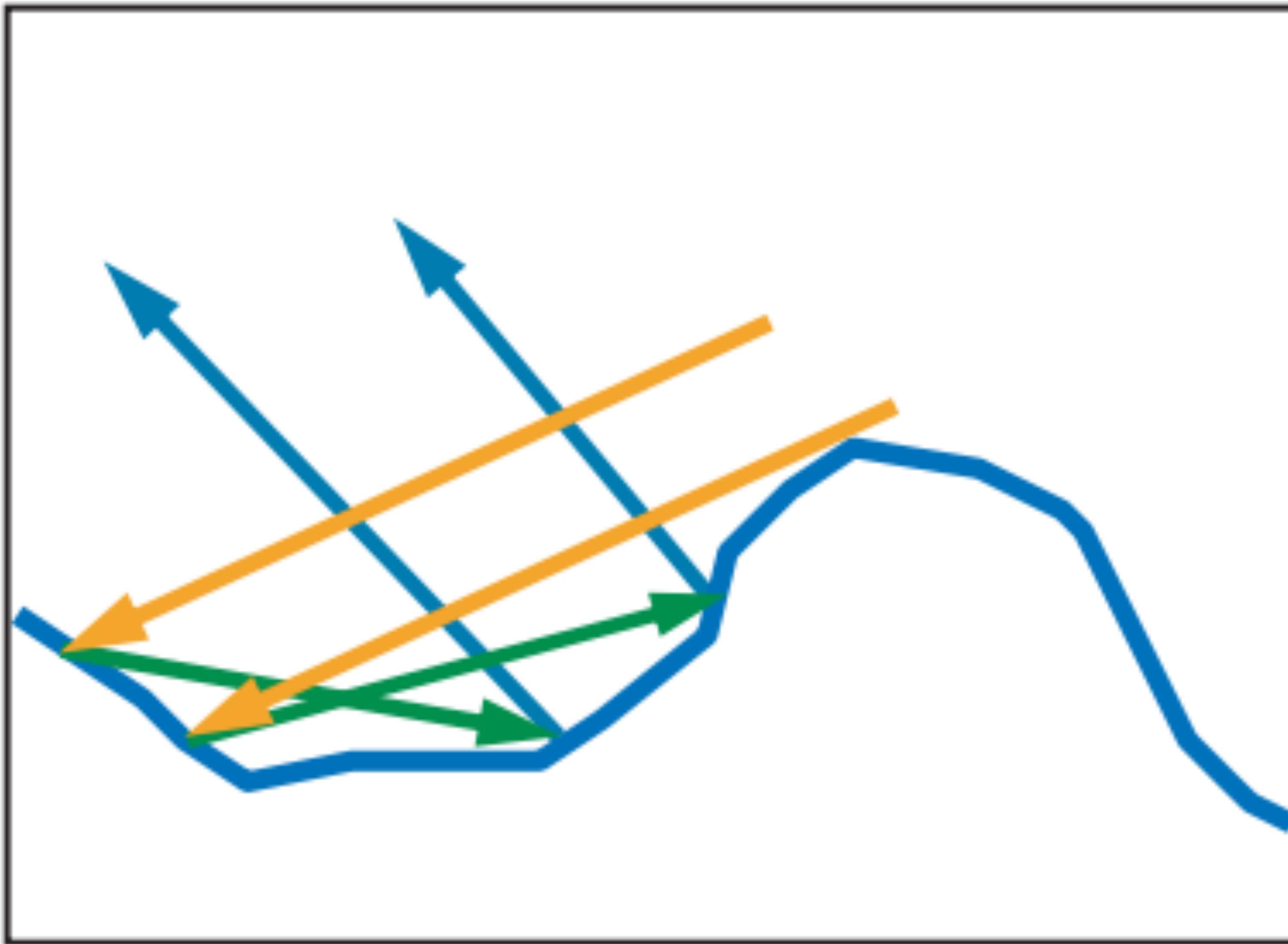
Images from "Real-Time Rendering, 3rd Edition", A K Peters 2008

SIGGRAPH 2012



Not all microfacets with $m = h$ will contribute - some will be blocked by other microfacets from either **I** (*shadowing*) or **v** (*masking*).

Multiple Surface Bounces



SIGGRAPH 2012

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

In reality, blocked light continues to bounce; some will eventually contribute to the BRDF. Microfacet BRDFs ignore this, and assume all blocked light is lost

Microfacet BRDF

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



This is the microfacet BRDF. I'll go over its various parts, explaining each.

Fresnel Reflectance

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



The Fresnel reflectance is the fraction of incoming light reflected (vs. refracted) from an optically flat surface of a given substance given light direction and surface normal. This tells us much how much of the light hitting the relevant microfacets (the ones facing in the half-angle direction) is reflected.

Fresnel Reflectance

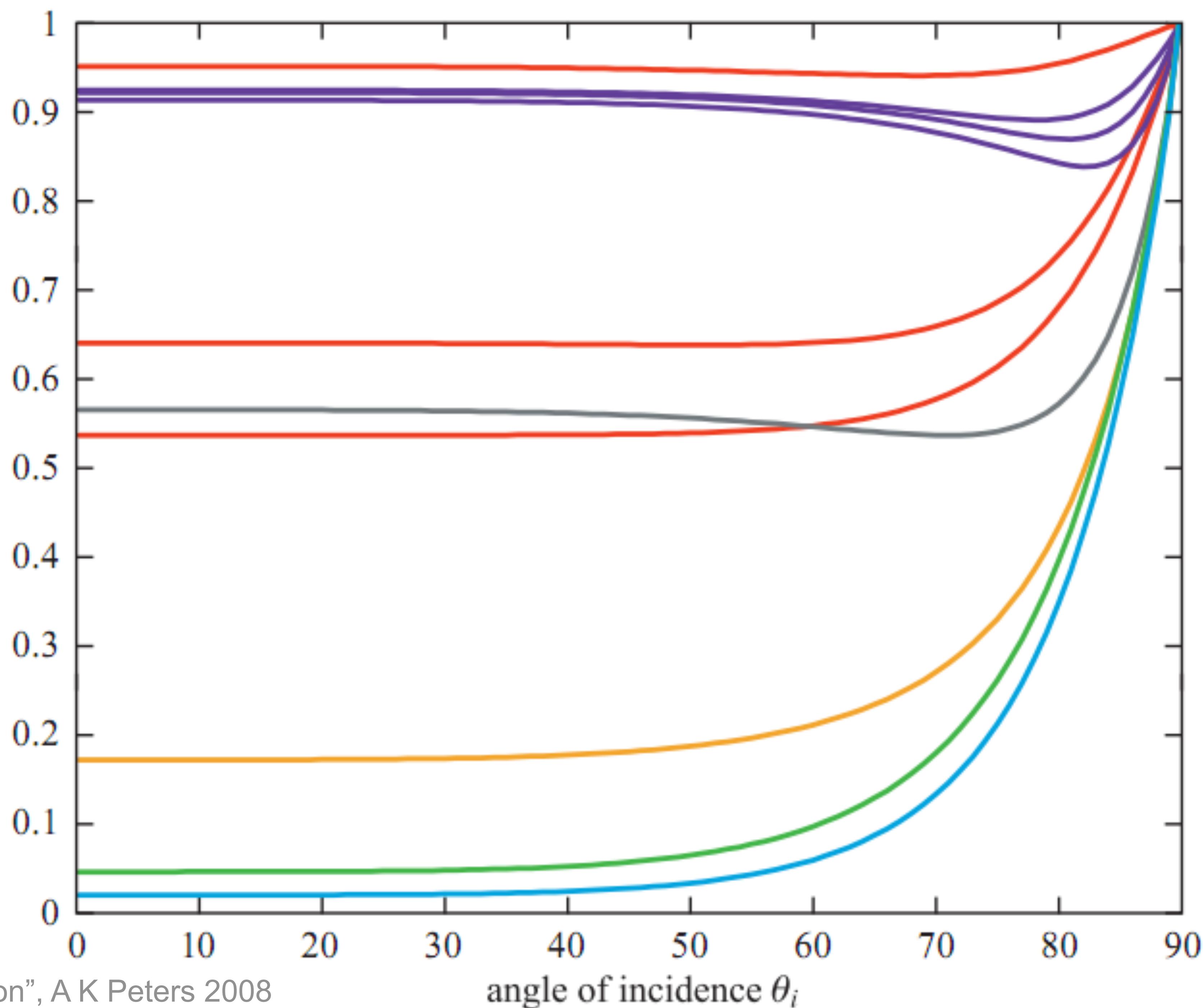


Image from “Real-Time Rendering, 3rd Edition”, A K Peters 2008

angle of incidence θ_i

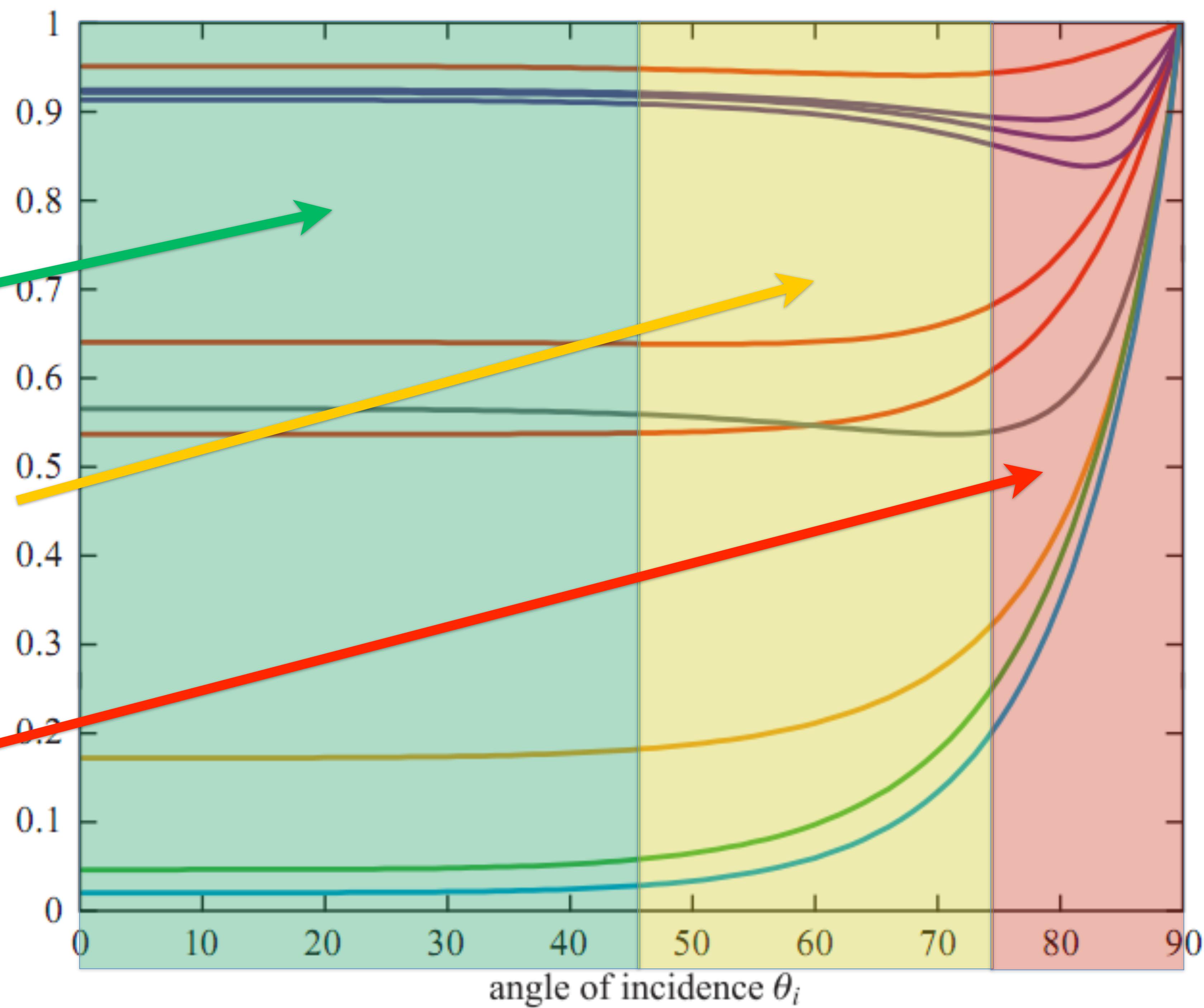
Fresnel reflectance depends on refraction index (in other words, what the object's made of) and light angle. In this graph, substances with three lines are R, G and B – substances with one line have uncolored reflectance.

Fresnel Reflectance

barely changes

changes somewhat

goes rapidly to 1



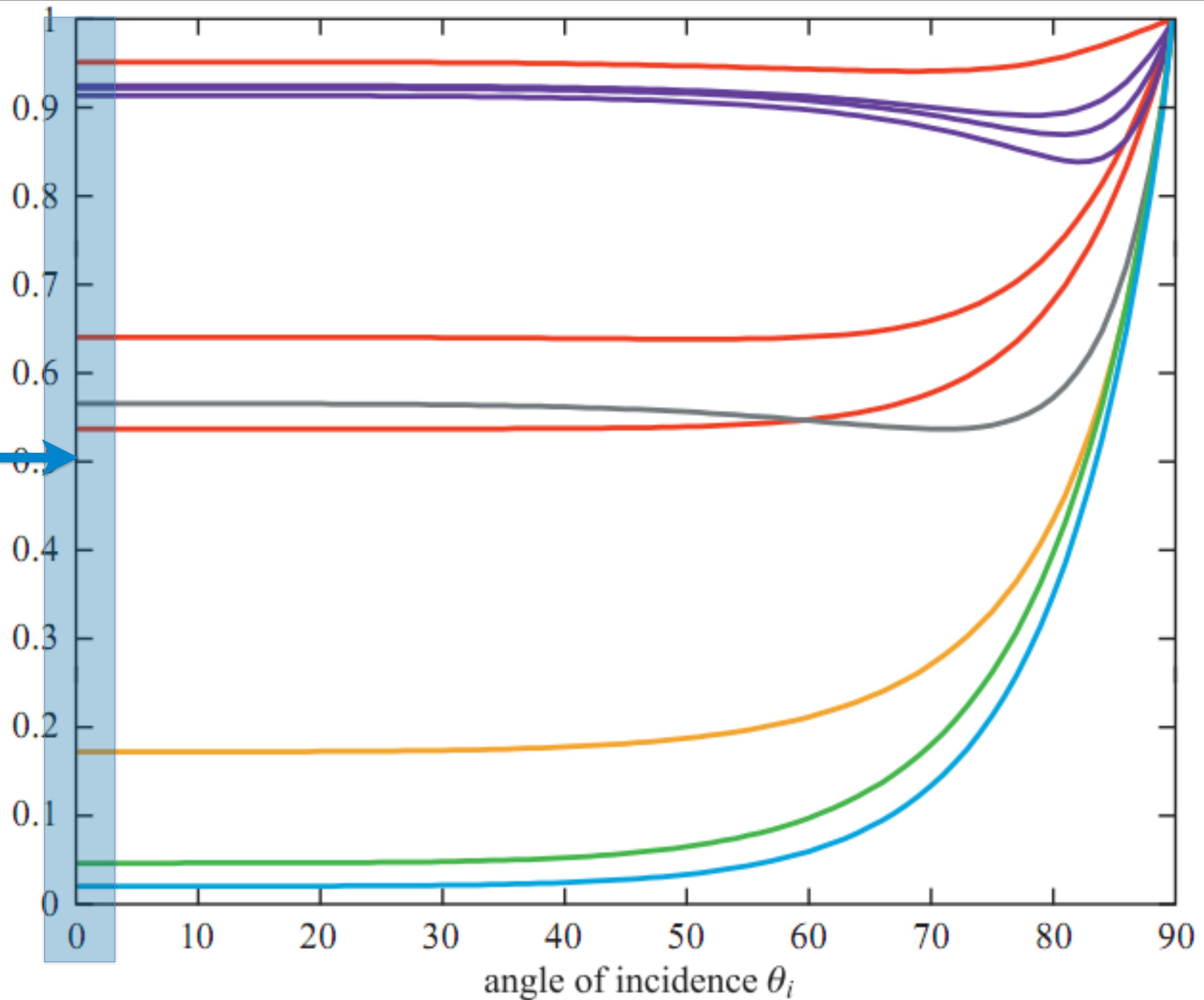
As angle increases, at first the Fresnel reflectance stays almost the same, then for very glancing angles it goes to 100% at all wavelengths.

Fresnel Reflectance

$F(0^\circ)$

Is the surface's characteristic specular color:

c_{spec}



Since the reflectance over most angles is close to that at normal incidence, the normal-incidence reflectance $F(0)$ is the surface's characteristic specular color.

Normal-Incidence Fresnel for Metals

Metal	$F(0^\circ)$ (Linear)	$F(0^\circ)$ (sRGB)	Color
Gold	1.00,0.71,0.29	1.00,0.86,0.57	
Silver	0.95,0.93,0.88	0.98,0.97,0.95	
Copper	0.95,0.64,0.54	0.98,0.82,0.76	
Iron	0.56,0.57,0.58	0.77,0.78,0.78	
Aluminum	0.91,0.92,0.92	0.96,0.96,0.97	



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

Note that metals have no subsurface term, so the surface Fresnel reflectance is the material’s only source of color. The “linear” and “sRGB” columns refer to whether the values are in linear or gamma space. Note that for metals the specular colors are all relatively high.

Normal-Incidence Fresnel for Non-Metals

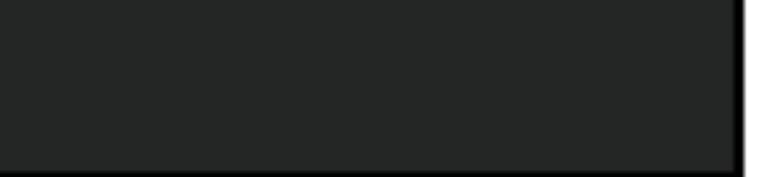
Insulator	$F(0^\circ)$ (Linear)	$F(0^\circ)$ (sRGB)	Color
Water	0.02,0.02,0.02	0.15,0.15,0.15	
Plastic / Glass (Low)	0.03,0.03,0.03	0.21,0.21,0.21	
Plastic High	0.05,0.05,0.05	0.24,0.24,0.24	
Glass (High) / Ruby	0.08,0.08,0.08	0.31,0.31,0.31	
Diamond	0.17,0.17,0.17	0.45,0.45,0.45	



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

Most non-metals also have a subsurface (diffuse) color in addition to this Fresnel (specular) reflectance. Note that for non-metals the specular colors are achromatic (gray) and are relatively low (especially if excluding gems and crystals).

The Schlick Approximation to Fresnel

- Pretty accurate, cheap, parameterized by c_{spec}

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

- For microfacet BRDFs ($m = h$):

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



The Schlick approximation to Fresnel is commonly used. It is cheap and reasonably accurate - more importantly it has a convenient parameter (normal incidence Fresnel reflectance, or specular color).

Microfacet Normal Distribution

$$f(\mathbf{l}, \mathbf{v}) = \frac{F(1, h) G(1, \mathbf{v}, h) D(h)}{4(\mathbf{n} \cdot \mathbf{l})(\mathbf{n} \cdot \mathbf{v})}$$



The microfacet normal distribution tells us what is the concentration of microfacet normals pointing in a given direction (in this case, the half-angle direction). The NDF determines the size and shape of the highlight.

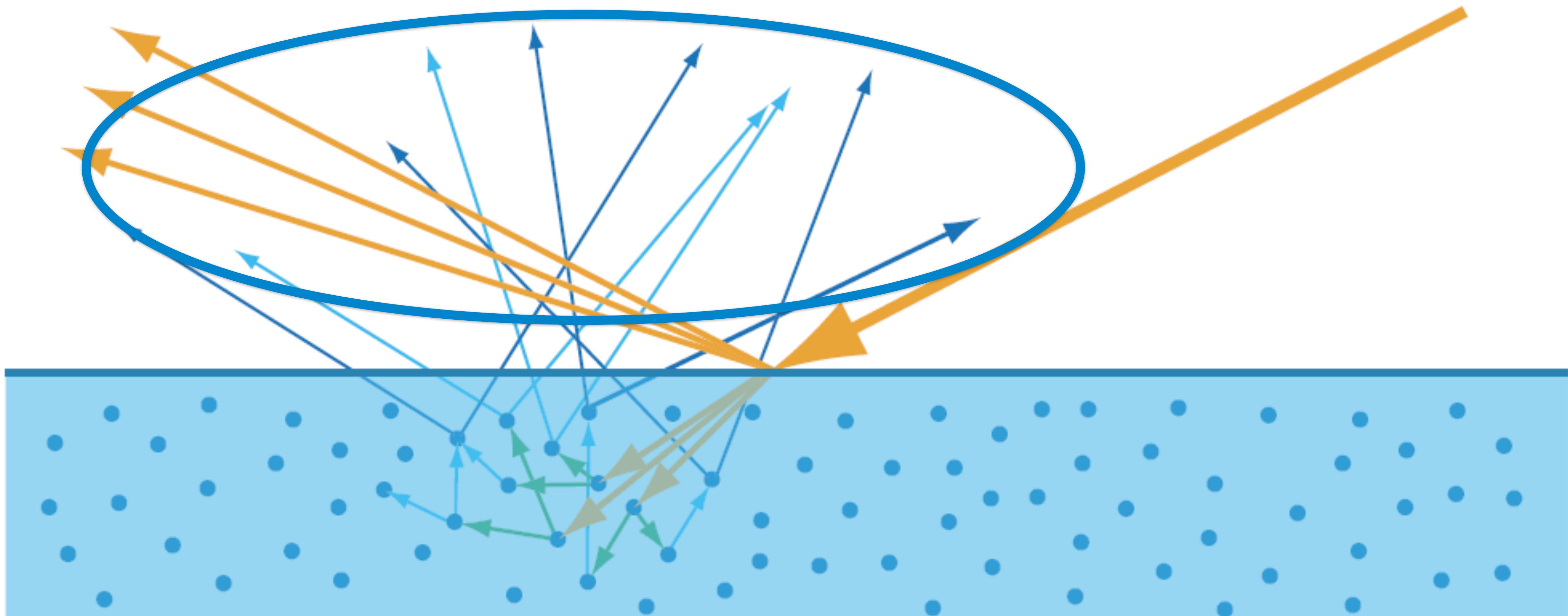
Geometry Factor

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



The geometry factor gives the chance that a microfacet of the given orientation is lit and visible (e.g., not shadowed and/or masked) from the given \mathbf{l} and \mathbf{v} directions.

Subsurface Reflection (Diffuse Term)



SIGGRAPH 2012



Next, we'll take a quick look at the diffuse or subsurface term.

Lambert

- Constant value ($\mathbf{n} \cdot \mathbf{l}$ is part of reflection equation):

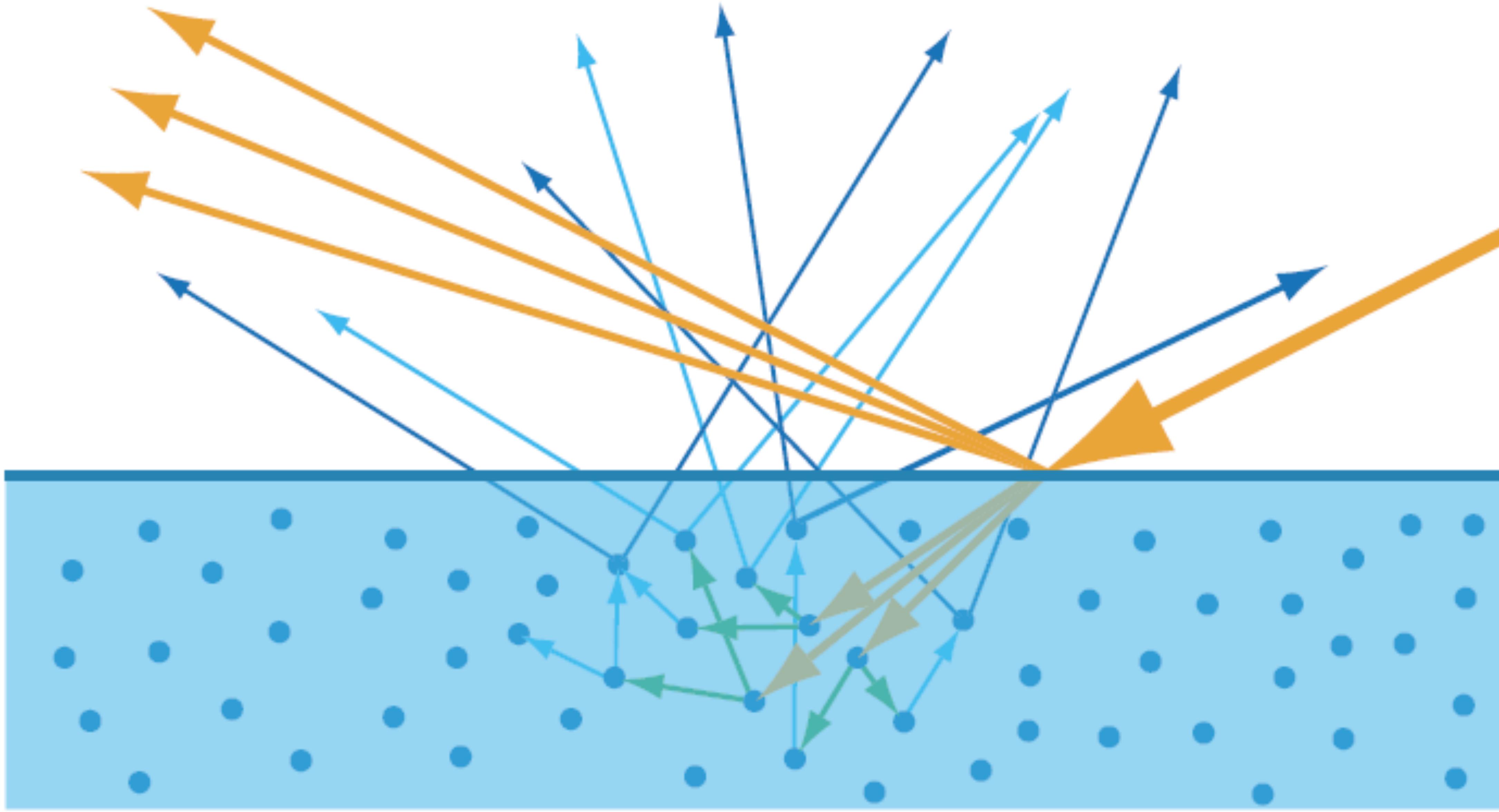
$$f_{\text{Lambert}}(\mathbf{l}, \mathbf{v}) = \frac{c_{\text{diff}}}{\pi}$$

- c_{diff} : fraction of light reflected, or diffuse color



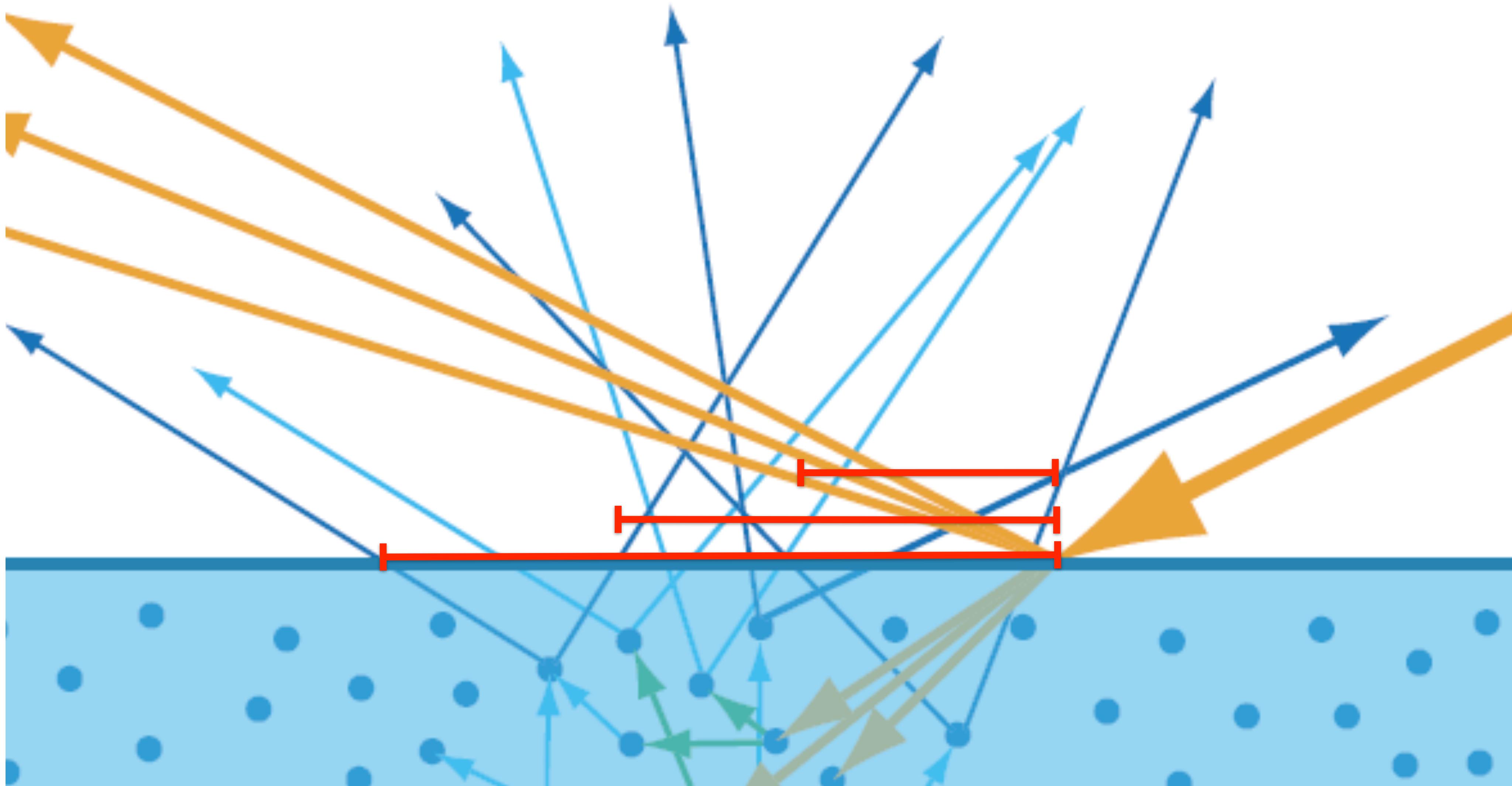
The Lambert model is the most common diffuse term used in game and film production. By itself, it's the simplest possible BRDF - a constant value (the well-known cosine factor is part of the reflection equation).

Beyond Lambert: Diffuse-Specular Tradeoff



There are a few important physical phenomena that Lambert doesn't account for. Diffuse comes from refracted light - in a sense the specular term gets "dibs" on the incoming light and diffuse gets the leftovers. Since surface reflection goes to 100% at glancing angles, it follows that diffuse should go to 0%. There are various ways to account for this - perhaps the simplest is to multiply diffuse by $(1-F)$ (this turns Fresnel into a "lerp factor" between specular and diffuse).

Beyond Lambert - Surface Roughness



SIGGRAPH 2012



Another phenomenon not modeled by Lambert is the effect of surface roughness. Roughness typically only affects specular; since diffuse reflectance at a point comes from incoming light over an area, the microgeometry variations tend to be averaged out for diffuse reflectance. But some surfaces have microgeometry larger than the scattering distance, and these do affect diffuse. These are the surfaces that should be modeled with specialized diffuse models like Oren-Nayar or Minnaert.

Implementing Physical Shading Models

- To implement a shading equation in a renderer, combine it with various illumination models:
 - General Lighting
 - Image-Based Lighting
 - Area Light Sources
 - Punctual Light Sources
 - Ambient Light



General lighting: integrate BRDF against continuous incoming light from all directions (area light sources, skylight, indirect reflections)
- requires GI algorithm like ray-tracing. Image-based lighting refers to environment maps and the like. Area light sources are starting to be commonly used in film but not in games, which typically use punctual light sources (still used in film as well). Ambient is the simplest lighting model - just constant light from all directions. We will talk more about punctual lights and image-based lighting due to their common use in game and film production rendering.

Punctual Light Sources

- Parameterized by light color c_{light} and direction to the light (center) position l_c
- c_{light} equals radiance from a white Lambertian surface illuminated by the light at 90 degrees



These are commonly used in games and film - infinitely small and infinitely bright, they are not realistic, but are the easiest type of light to implement with arbitrary BRDFs. They look OK for rough surfaces, not so much for smooth ones. Punctual lights are parameterized by color and direction.

Punctual Light Reflection Equation

$$L_o(\mathbf{v}) = \pi f(\mathbf{l}_c, \mathbf{v}) \otimes \mathbf{c}_{\text{light}}(\mathbf{n} \cdot \mathbf{l}_c)$$



The course notes have the derivation. Games often clamp the dot product as a convenient method of skipping backfacing lights.

Image Based Lighting (Environment Maps)

- Perfectly smooth surface is easy - just multiply by Fresnel function; same c_{spec} , different angle:
 - $F(\mathbf{v}, \mathbf{n})$ instead of $F(\mathbf{l}, \mathbf{h})$ (or $F(\mathbf{v}, \mathbf{h})$; same)
- Accurate glossy/diffuse requires many samples
 - Importance sampling helps
 - Prefiltering (alone or with importance sampling)

Building a Microfacet Model

- The choice of $D()$ and $G()$ are independent; you can mix and match from different models
- Most papers proposing a new BRDF model are really introducing a new $D()$ or a new $G()$



We will now look at some options for building a microfacet model. Don't worry about copying the equations down - they will all be in the course notes.

Normalizing a Normal Distribution Function

$$1 = \int_{\Theta} D(\mathbf{m}) (\mathbf{n} \cdot \mathbf{m}) d\omega_m$$



To be used in a microfacet BRDF, an NDF should be normalized based on the equation above. If you have a non-normalized term, just divide it by its cosine-weighted integral. In the general case the integral is over the sphere but we will only deal with heightfield surfaces where you want to integrate over the hemisphere. Heightfield NDFs should always evaluate to 0 for backfacing directions; if you may evaluate it for these directions (e.g. normal mapping can sometimes cause backfacing half-vectors) then you should ensure it evaluates to 0 in these cases. For NDFs that go to 0 at 90 degrees this can simply be done by clamping the n-dot-m dot product to 0; other NDFs may require a conditional expression.

Phong NDF

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



Correctly normalized, the Phong (cosine power lobe) becomes a valid NDF.

Beckmann NDF

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



The Beckmann NDF is similar to Phong in some important ways and different in others - see the course notes for more details.

Trowbridge-Reitz (GGX) NDF

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



The Trowbridge-Reitz NDF has a fundamentally different shape than Phong & Beckmann, with a narrower “peak” and broader “tails”. More details can be found in the course notes.

ABC NDF

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

$$k_{abc} = \frac{\alpha_{abc1}^2 (1 + \alpha_{abc1})^{\alpha_{abc2}} (-2 + \alpha_{abc2})(-1 + \alpha_{abc2})}{2\pi ((1 + \alpha_{abc1})^2 + (1 + \alpha_{abc1})^{\alpha_{abc2}} (-1 + \alpha_{abc1}(-2 + \alpha_{abc2})))}$$

$$D_{abc}(\mathbf{m}) = k_{abc} D_{uabc}(\mathbf{m})$$



The ABC function has been around for a while in the optical engineering community, not as an NDF but as a spectral power distribution for the heightfield of the surface. Recent work has proposed using this function as an NDF. I got the normalization factor from Mathematica and didn't spend any time simplifying or optimizing it; note that it has singularities at $\alpha_{abc2}=1.0$ and $\alpha_{abc2}=2.0$. I suspect a much cheaper curve could be fit to it. More details can be found in the course notes.

Shifted-Gamma NDF

$$p22[x] = \frac{\alpha_{sgd1}^{\alpha_{sgd2}-1}}{\Gamma(1 - \alpha_{sgd2}, \alpha_{sgd1})} \frac{e^{-\frac{\alpha_{sgd1}^2 + x}{\alpha_{sgd1}}}}{\left(\alpha_{sgd1}^2 + x\right)^{\alpha_{sgd2}}}$$

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



The shifted-Gamma NDF is a relatively new one. Like ABC it has two parameters, rather than the single roughness parameter most isotropic NDFs have. Like ABC, its normalization factor looks expensive, but a cheap approximation could likely be found. Again similarly to ABC, the authors who proposed this NDF suggest that it is a closer fit to measured BRDFs (such as the Matusik database) than previous models.

The Visibility Term

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

$$V(l, v) = \frac{G(l, v, h)}{(n \cdot l)(n \cdot v)}$$

SIGGRAPH 2012



In some cases, expressions are found for the geometry factor divided by the n-dot-l-times-n-dot-v “foreshortening term”. We’ll call this combined term the “visibility term”. The following slides will describe some choices for representing either just the geometry factor or the entire visibility term.

The Simplest Possible Visibility Term

$$\frac{G(l, v, h)}{(n \cdot l)(n \cdot v)} = 1$$

- Equivalent to:

$$G_{\text{implicit}}(l_c, v, h) = (n \cdot l_c)(n \cdot v)$$

Many BRDFs (especially game ones) have no visibility term at all. This means the visibility term is effectively equal to one, which implies a geometry factor equal to $n \cdot l \cdot n \cdot v$. This “implicit” geometry factor is plausible - it equals one when view and light are in the normal direction, and zero when either one of them is at 90 degrees. And it is “cheaper than free” since the entire visibility term is missing. But it goes to 0 too fast at glancing angles (compared to real-world materials) and it isn’t affected by roughness, which is implausible.

The Cook-Torrance Geometry Factor

$$G_{ct}(l, v, h) = \min \left(1, \frac{2(n \cdot h)(n \cdot v)}{(v \cdot h)}, \frac{2(n \cdot h)(n \cdot l)}{(v \cdot h)} \right)$$



This was actually invented by Blinn in 1975 as a reformulation of the Torrance-Sparrow geometry factor. It's been used a lot over the years (especially in film), but it has some problems: it's based on an unrealistic microgeometry model (isotropic surface composed of infinity long grooves) and it's not affected by roughness, which is implausible.

Kelemen-Szirmay-Kalos Geometry Factor

$$\frac{G_{ct}(l, v, h)}{(n \cdot l)(n \cdot v)} \approx \frac{1}{(l \cdot h)^2}$$



In addition to its other issues, Cook-Torrance is a little on the expensive side for games. There is a very cheap and effective approximation for it however; almost as cheap as the implicit geometry factor - you just need to divide by the square of a dot product that you need to compute anyway for Fresnel.

Smith Geometry Factor



This is probably the best one to use today. It is affected by the roughness, and it has been generalized to a process for generating a matching geometry factor for any NDF. You can think of it as a “black box” where you plug in an NDF and get a matching geometry factor. The process may not result in an analytical expression but that’s fine since you most likely want to fit a simpler curve to the result in any case. Note that the Schlick approximation to this function (which has been referenced in many places - including my own book!) is not correct since it approximates the wrong version of Smith. More details on the Smith shadowing factor can be found in the course notes.

Acknowledgements

- A K Peters for permission to use RTR3 images
- Brent Burley, Paul Edelstein, Yoshiharu Gotanda, Christophe Hery, Dimitar Lazarov, and Brian Smits for thought-provoking discussions



I'd like to close by thanking some people who helped me with this talk.