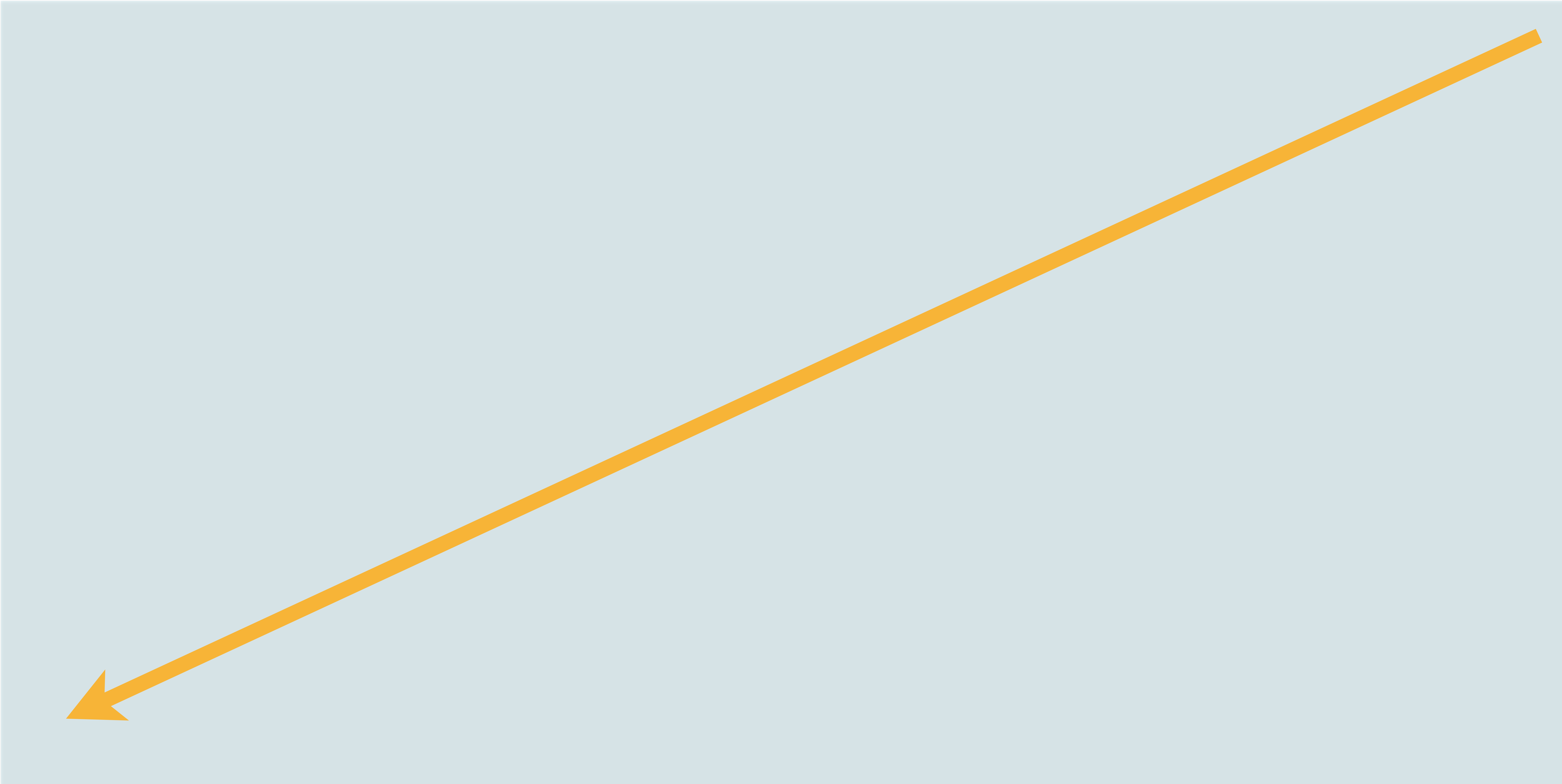
Background: Physics and Math of Shading

Naty Hoffman
2K

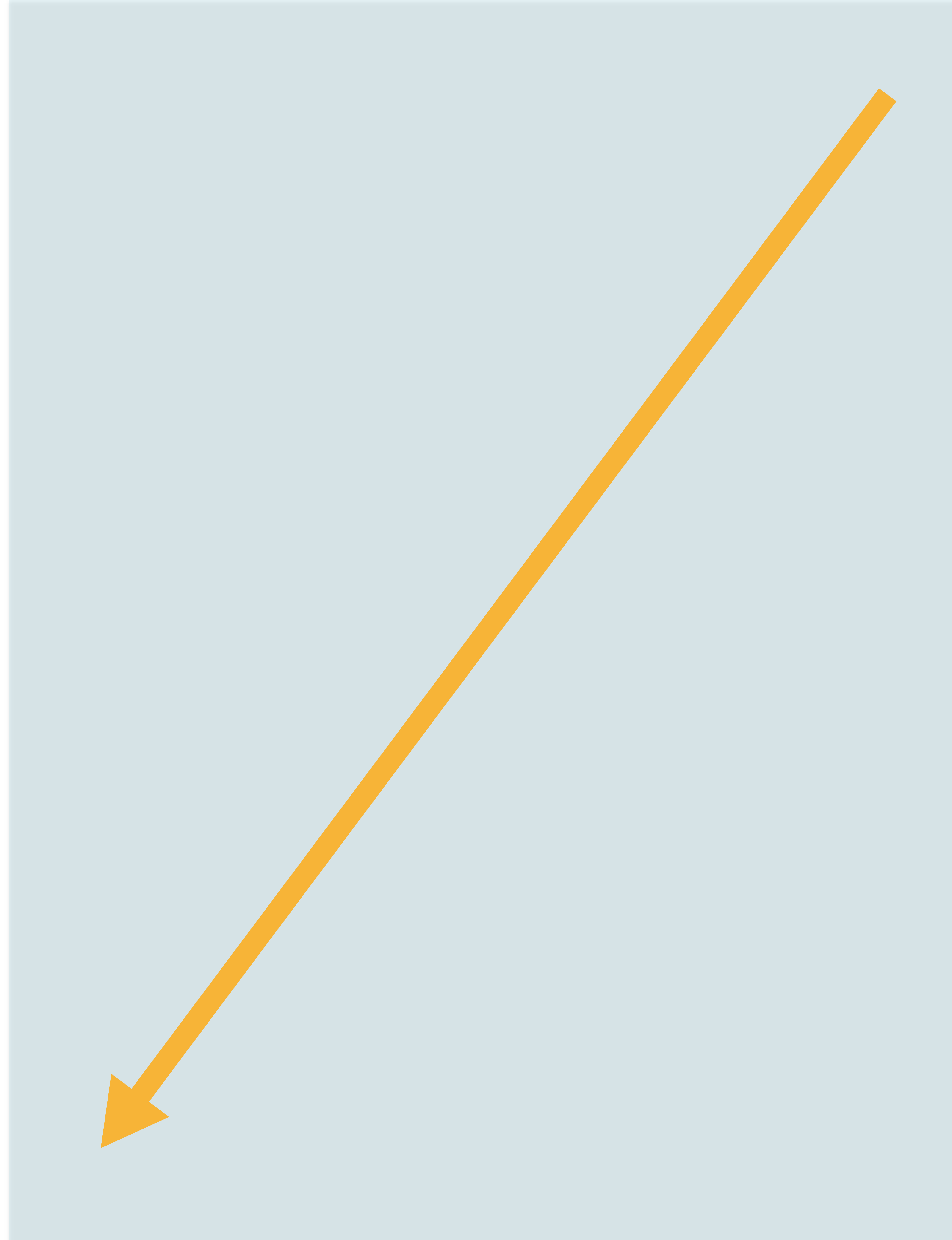
SIGGRAPH2014

Hi. Over the next 15 minutes I'll be going from the physics underlying shading, to the math used to describe it and from there to the kind of rendering implementations we'll see in the rest of the course.

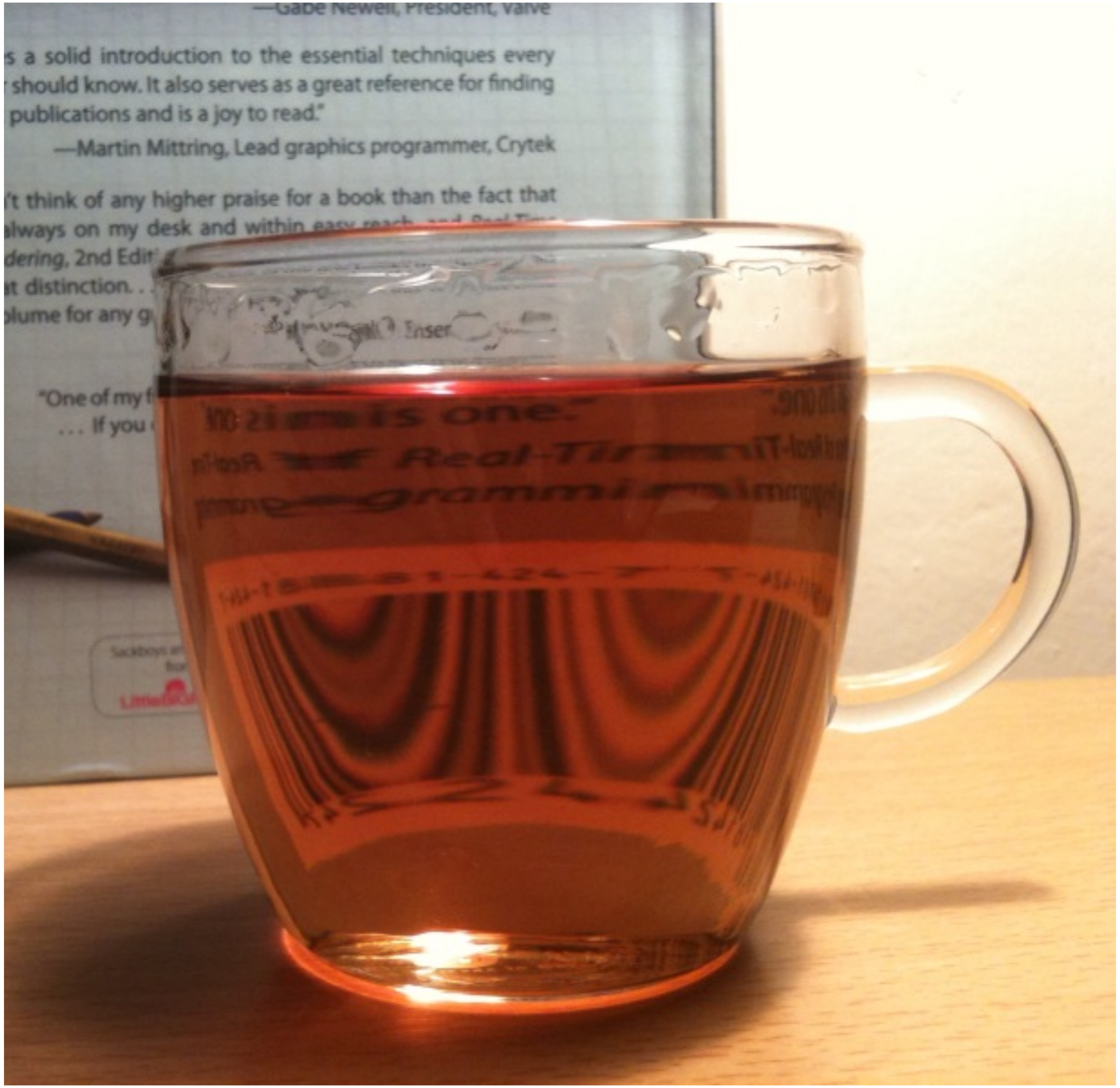
We'll start with the physics that happen when light interacts with matter.

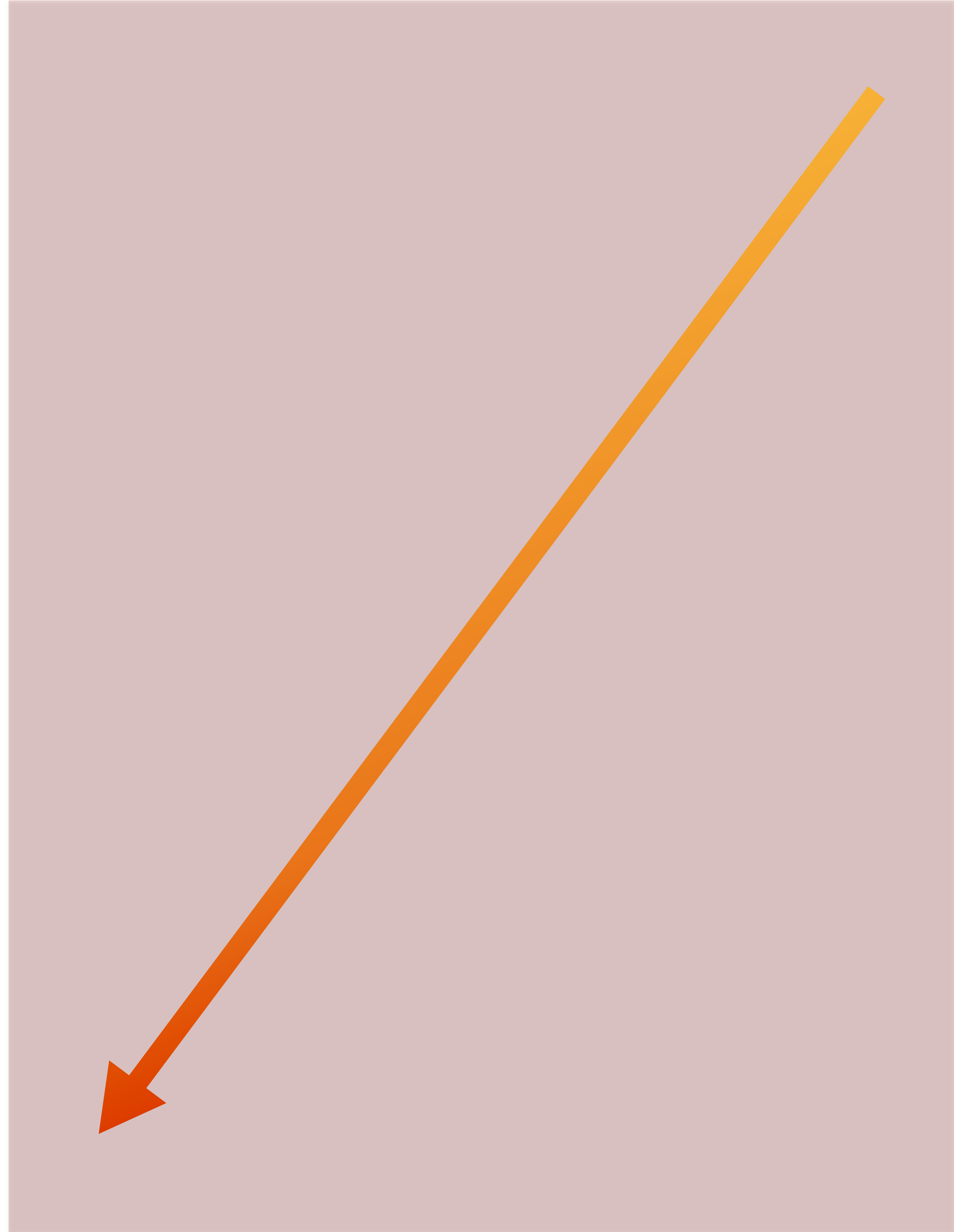


The simplest case is light propagating through a *homogeneous medium* with exactly the same properties everywhere. In this case light moves in a straight line.

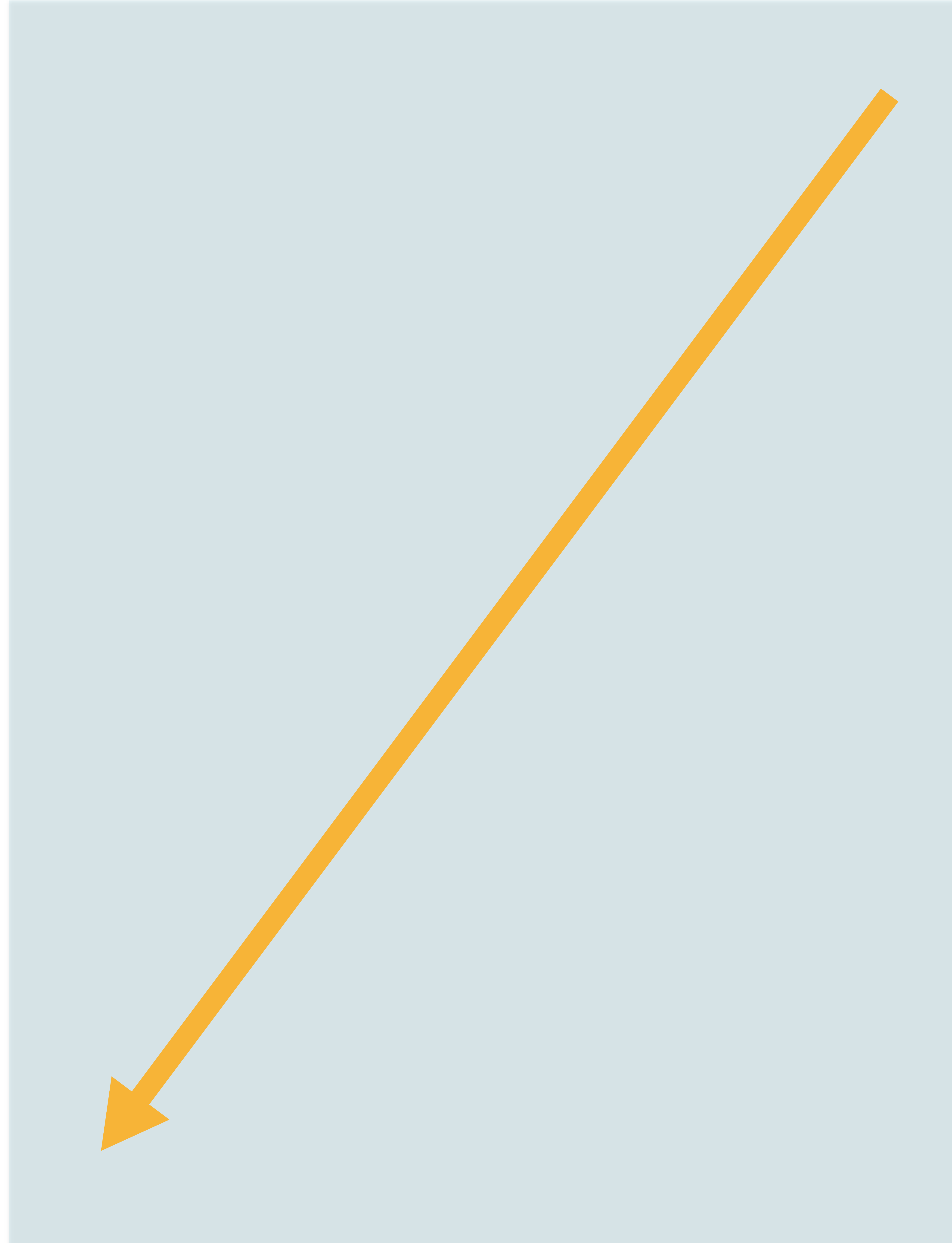


Some homogeneous media don’t significantly change the light’s color or intensity...

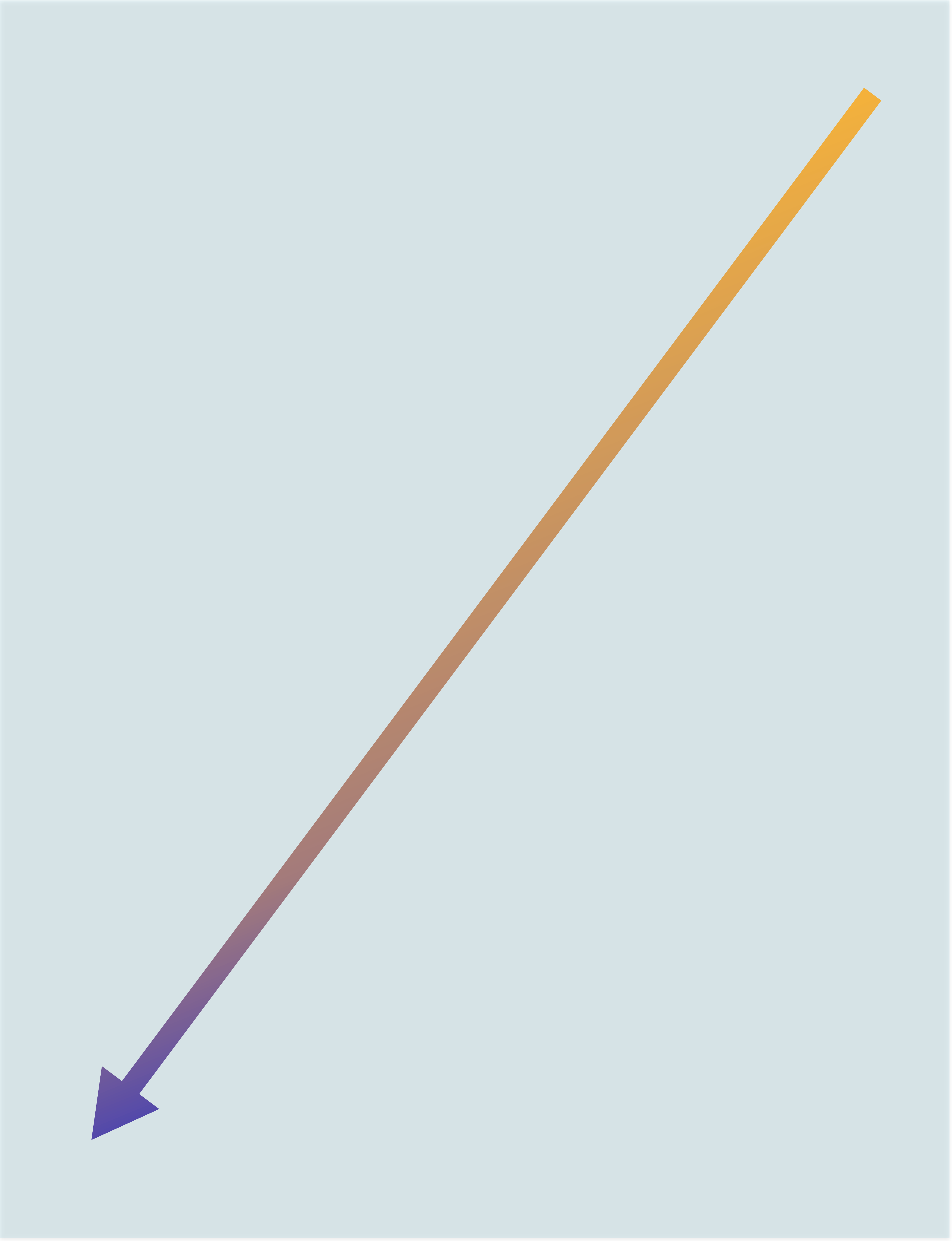




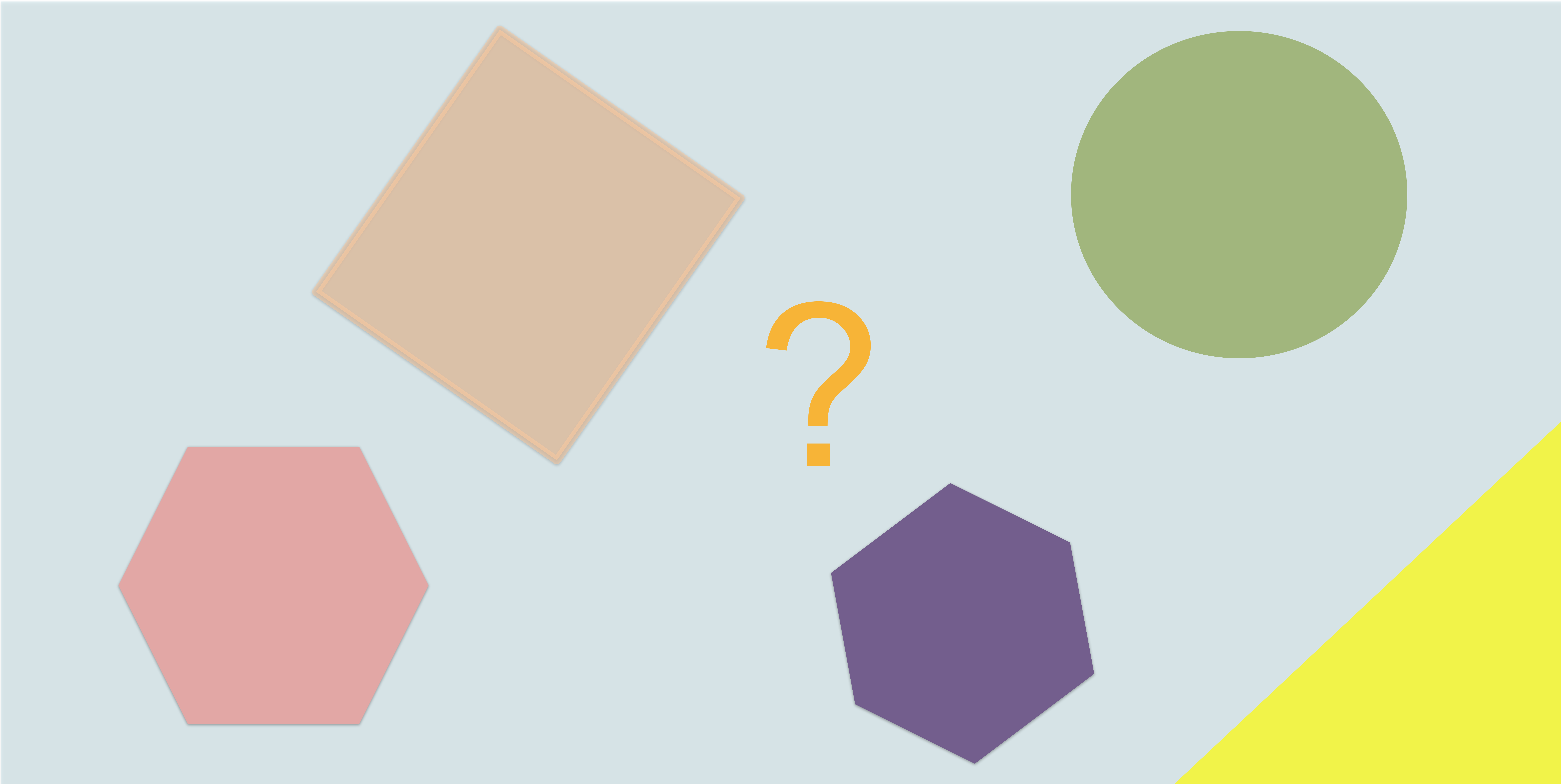
...while others absorb part of the visible light traveling through them, changing its intensity and potentially its color. For example, this medium absorbs more light in the blue part of the spectrum, giving it a red appearance.



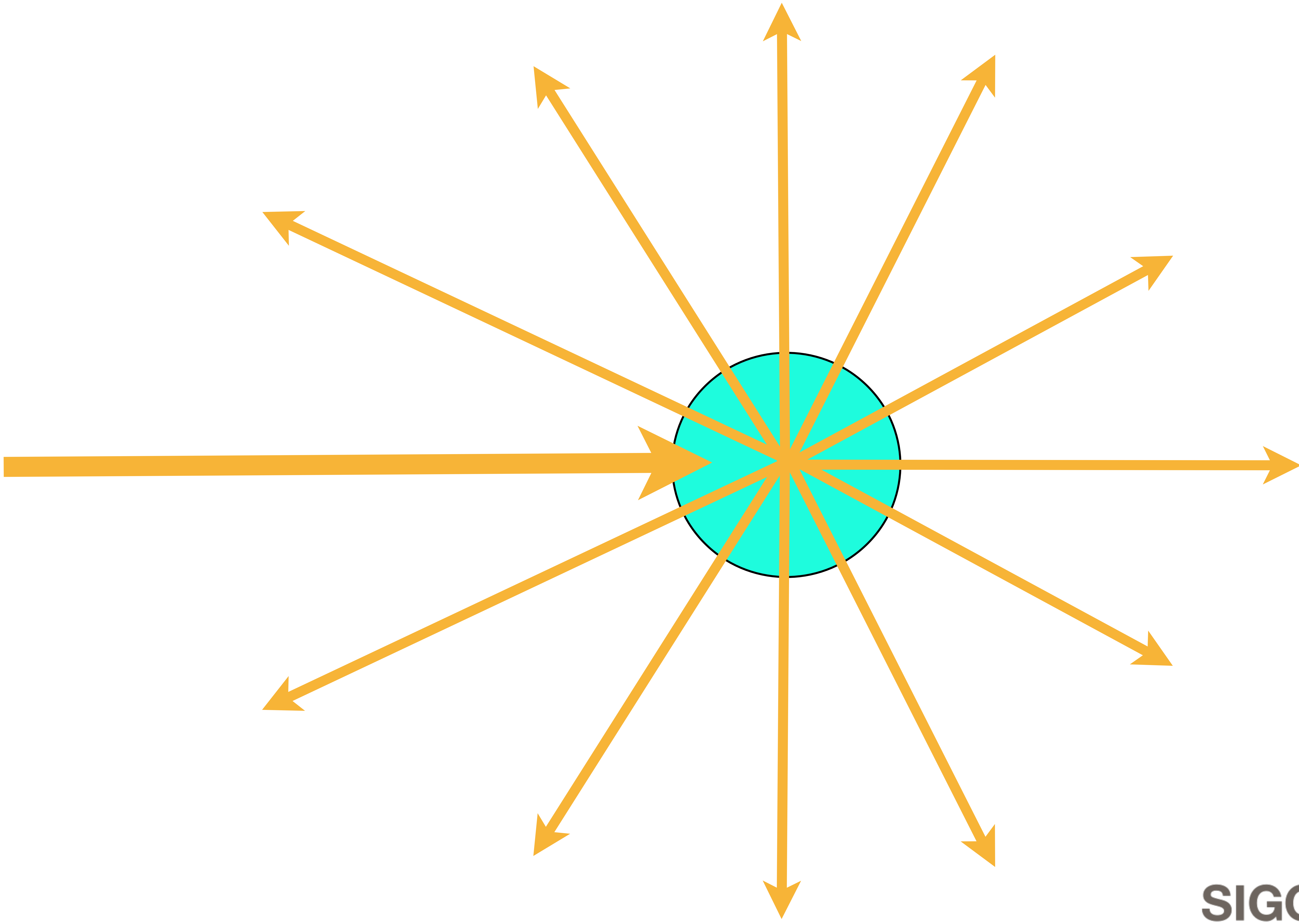
Scale is important. For example, clean water does absorb a little light in the red end of the visible spectrum, but it’s not noticeable over a few inches.



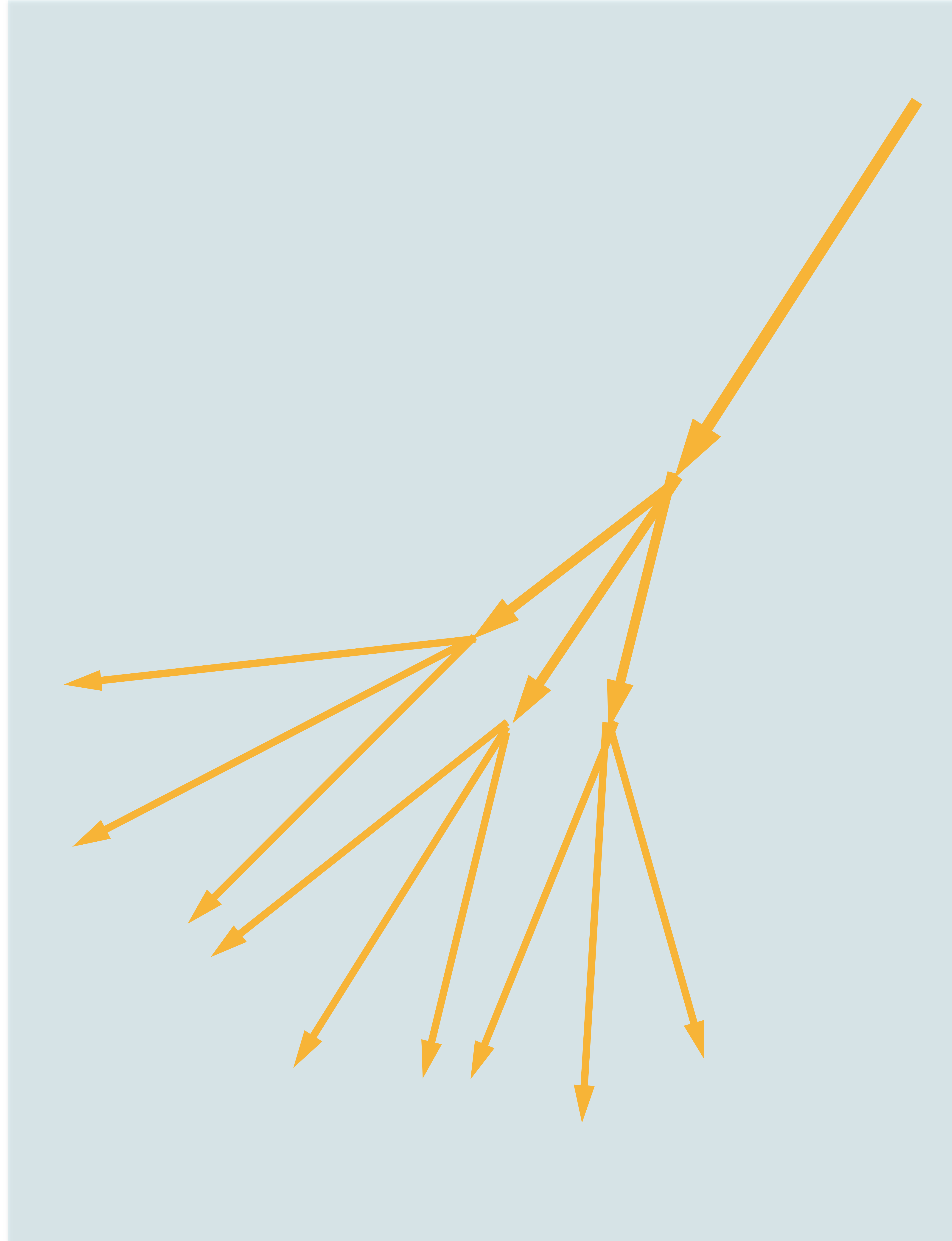
But this absorption is quite significant over distances of dozens of yards.



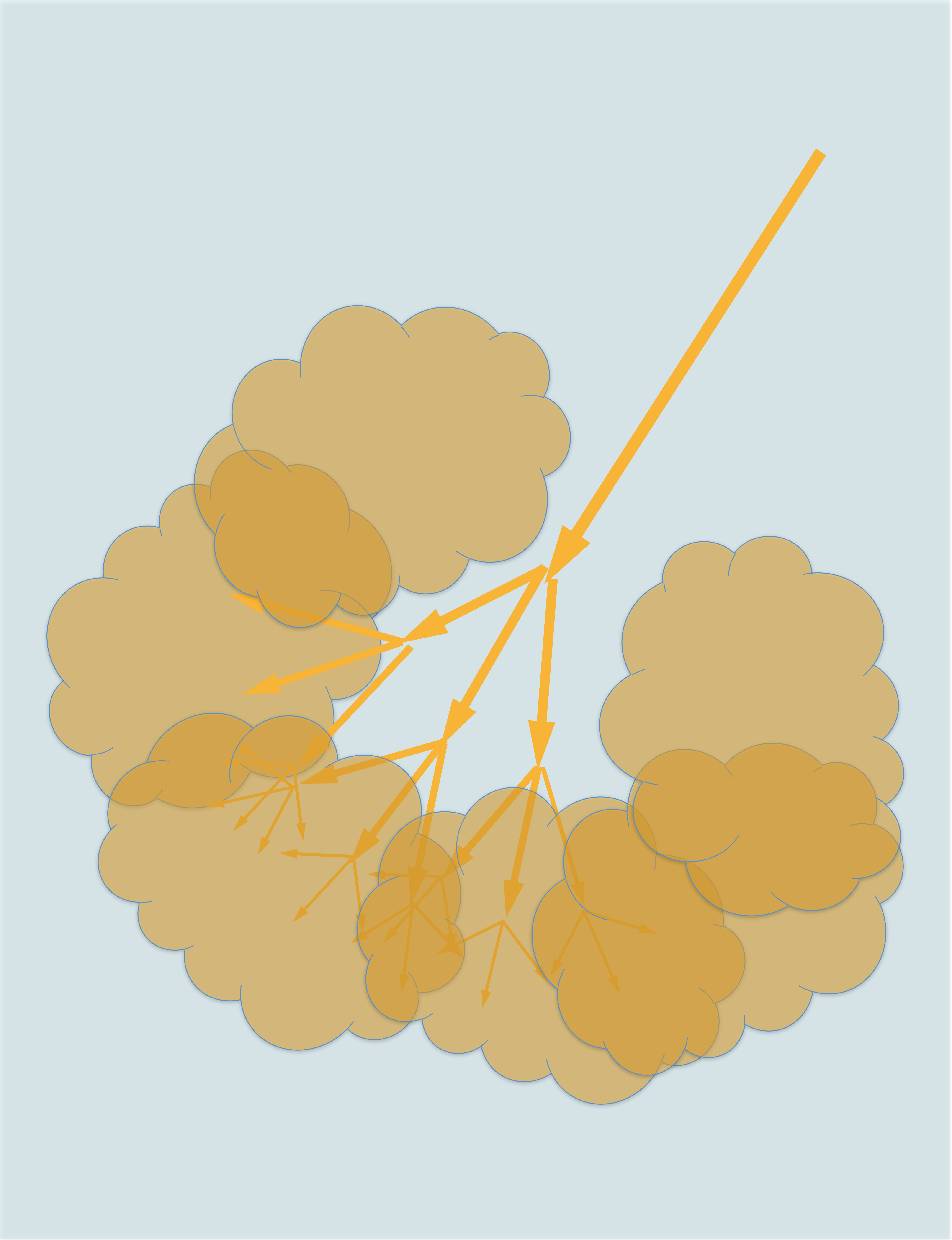
In an inhomogeneous medium, the index of refraction (which is the property of matter that affects light) changes. This causes light to no longer move in a straight line.



Abrupt changes in the index of refraction cause scattering, which changes the direction of light propagation.



An inhomogeneous medium contains numerous scattering particles. These could be dense enough to randomize the light's direction somewhat, giving a cloudy appearance...

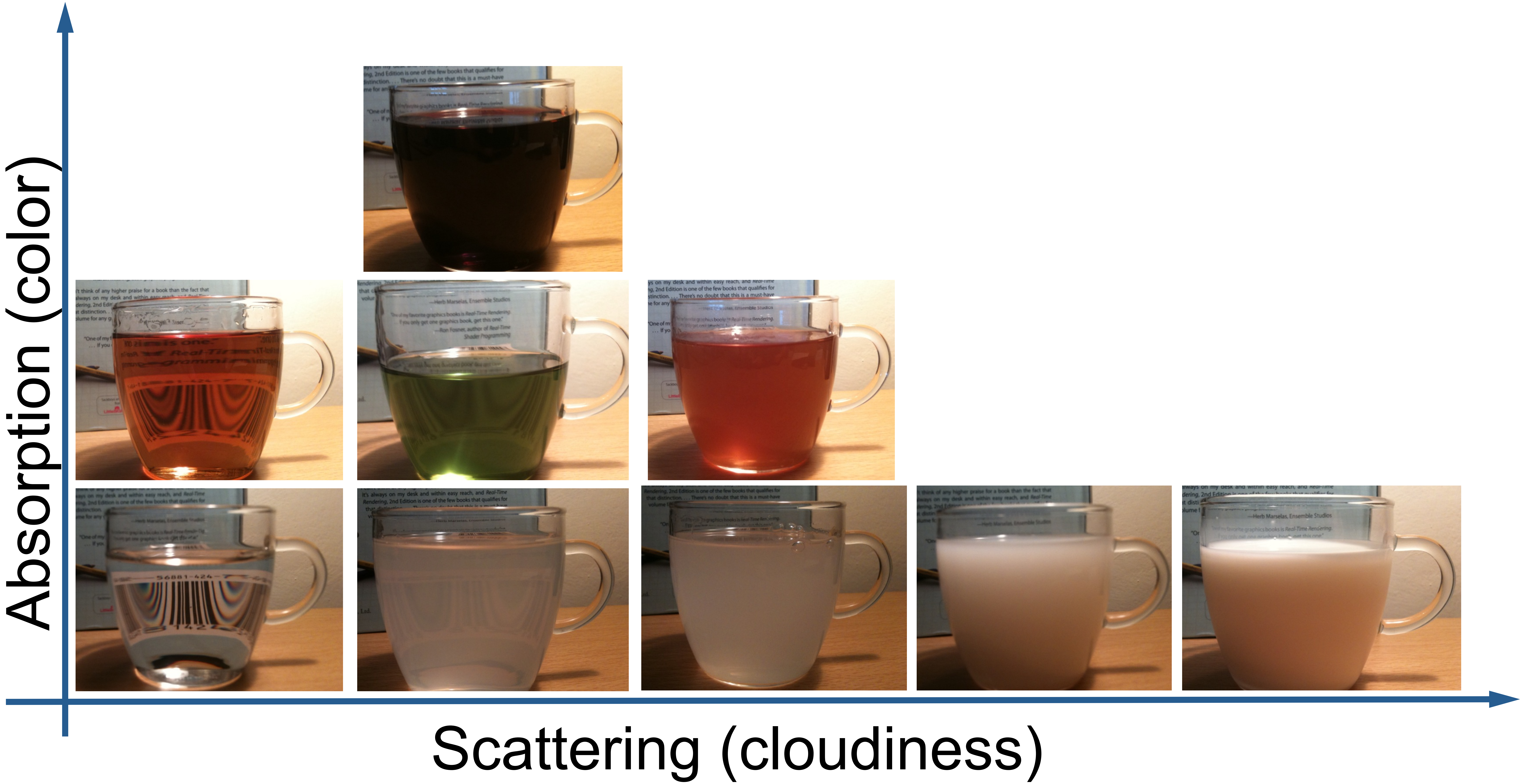


...or to randomize it completely, giving the medium an opaque appearance.

SIGGRAPH2014

Scale also matters for scattering: for example, clean air doesn't noticeably scatter light over a few yards, but it definitely does over a distance of miles.

To summarize, there are three basic modes of light/matter interaction: absorption (which changes light's intensity and/or color), scattering (which changes light's direction), and emission (which creates new light; most materials don't exhibit emission and I won't further discuss it in this talk).



The overall appearance of a medium is determined by the combination of its absorption and scattering properties. For example, a white appearance (like the whole milk in the lower right corner) is caused by high scattering and low absorption.

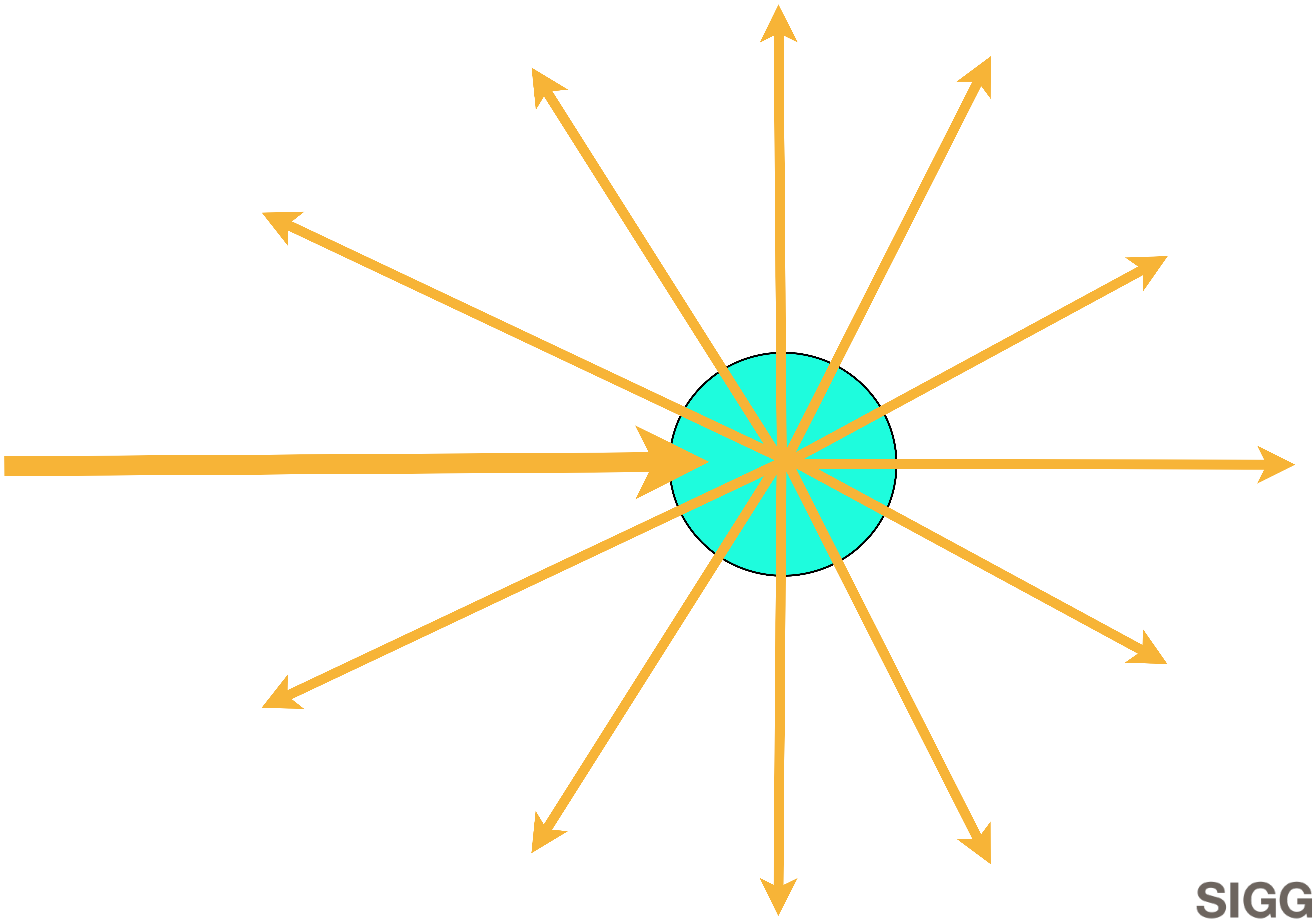







SIGGRAPH2014

While media are easy to understand, most of the time in graphics we are concerned with rendering solid objects, in particular the <u>surfaces</u> of these objects.



SIGGRAPH2014

You may recall that a few slides ago, I said that abrupt changes in index of refraction cause scattering. Small particles (like those found in cloudy liquids) are one special case of this; they scatter light in all directions.

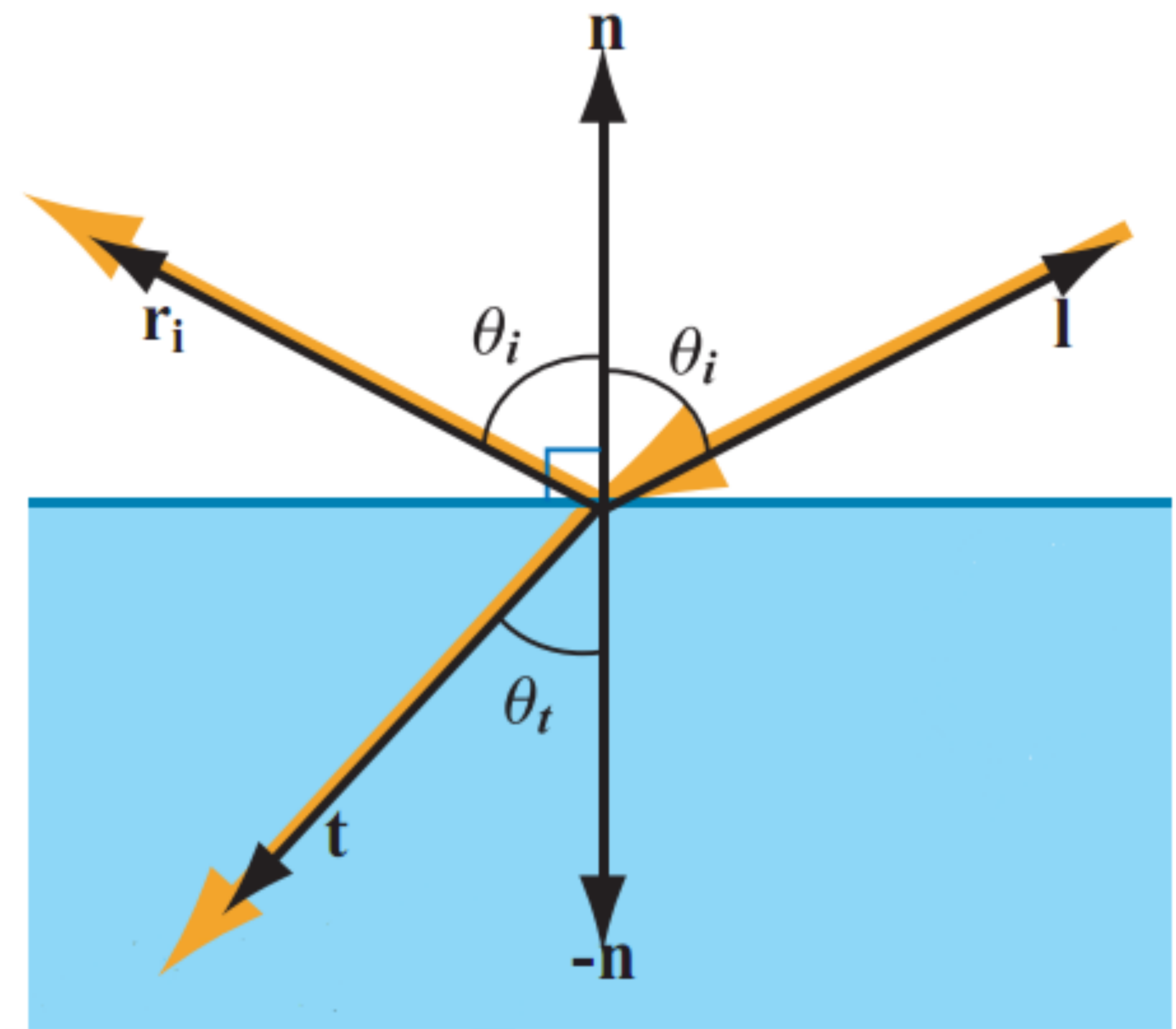


Image from “Real-Time Rendering, 3rd Edition”, A K Peters 2008

A flat surface (defined as a plane separating two volumes with different indices of refraction) is another special case of scattering; such a surface scatters light into exactly two directions: reflection and refraction. In this case “flat” means *optically flat*: any irregularities are smaller than visible light wavelengths and thus do not affect visible light.

Microgeometry

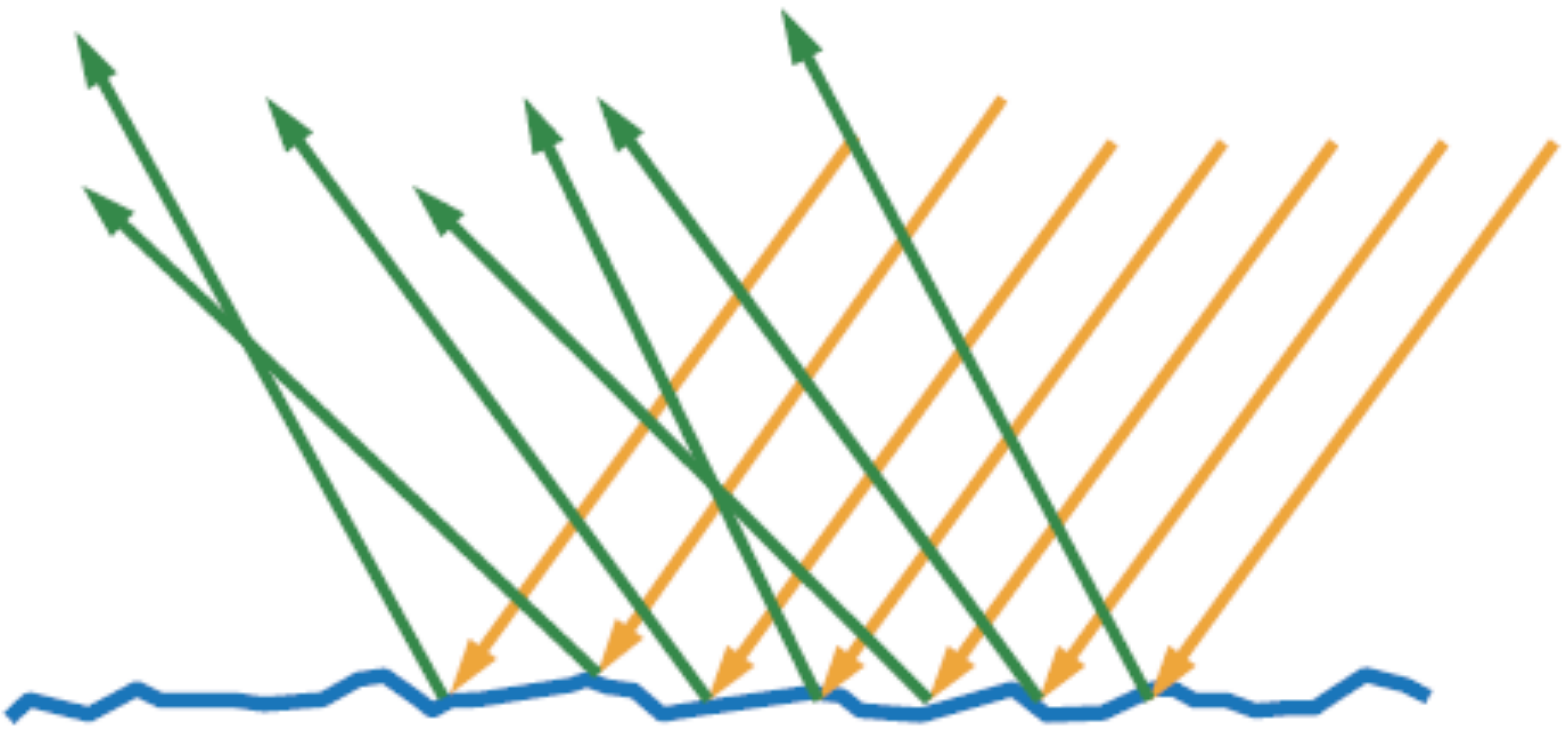
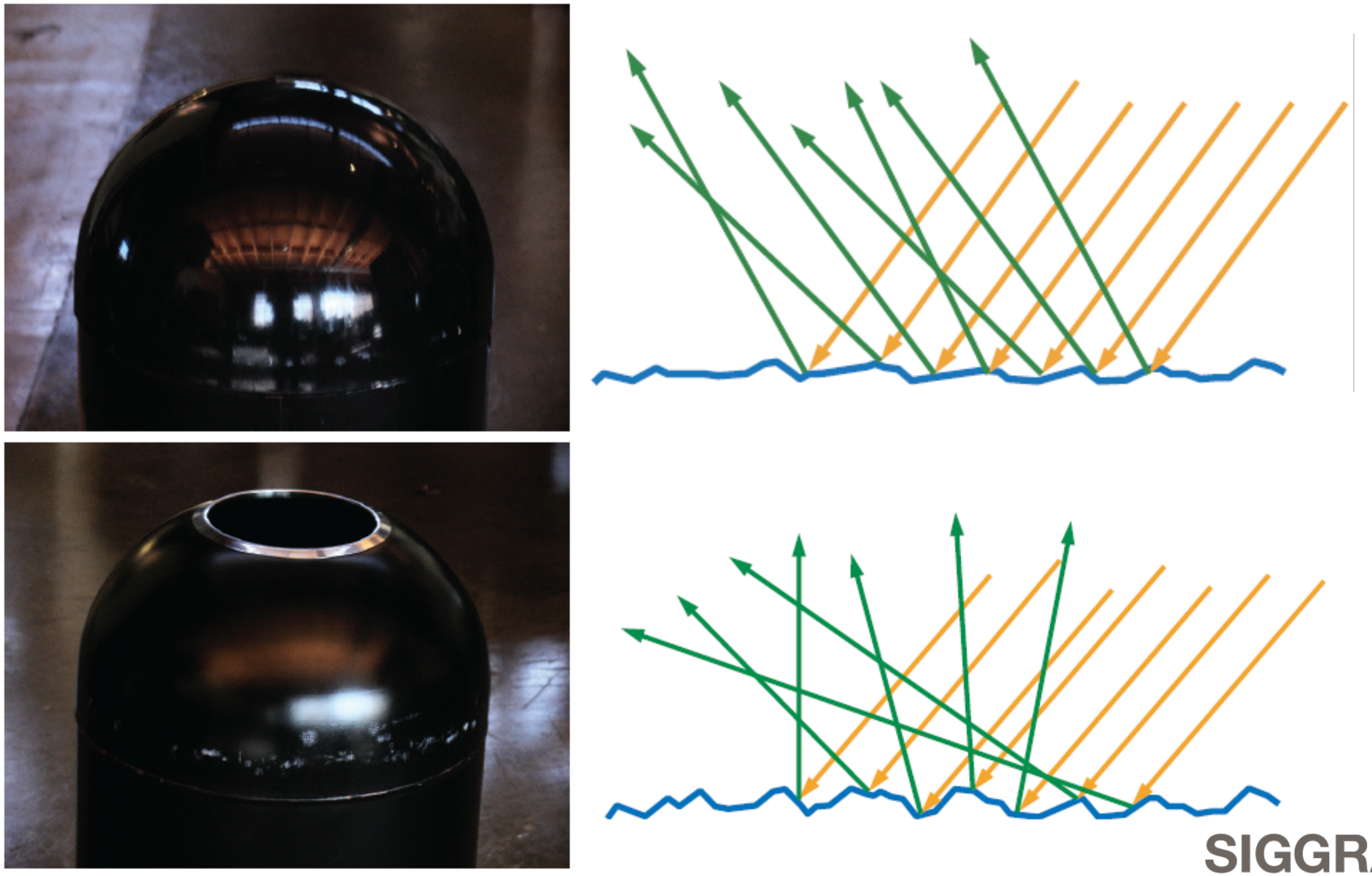


Image from “Real-Time Rendering, 3rd Edition”, A K Peters 2008

SIGGRAPH2014

Some rare real-world surfaces (like high-end telescope optics) are optically flat, but most aren’t. Most have microgeometry: bumps that are bigger than a light wavelength but too small to be individually visible. Each surface point reflects (and refracts) light in a different direction: the surface appearance is the aggregate result of all the different reflection and refraction directions.

Rougher = Blurrier Reflections



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

SIGGRAPH2014

These two surfaces—equally smooth to the naked eye—differ in roughness at the microscopic scale. The surface on the top is only a little rough; incoming light rays hit bits of the surface that are angled slightly differently and get reflected to somewhat different outgoing directions, causing slightly blurred reflections. The surface on the bottom is much rougher, causing much blurrier reflections.

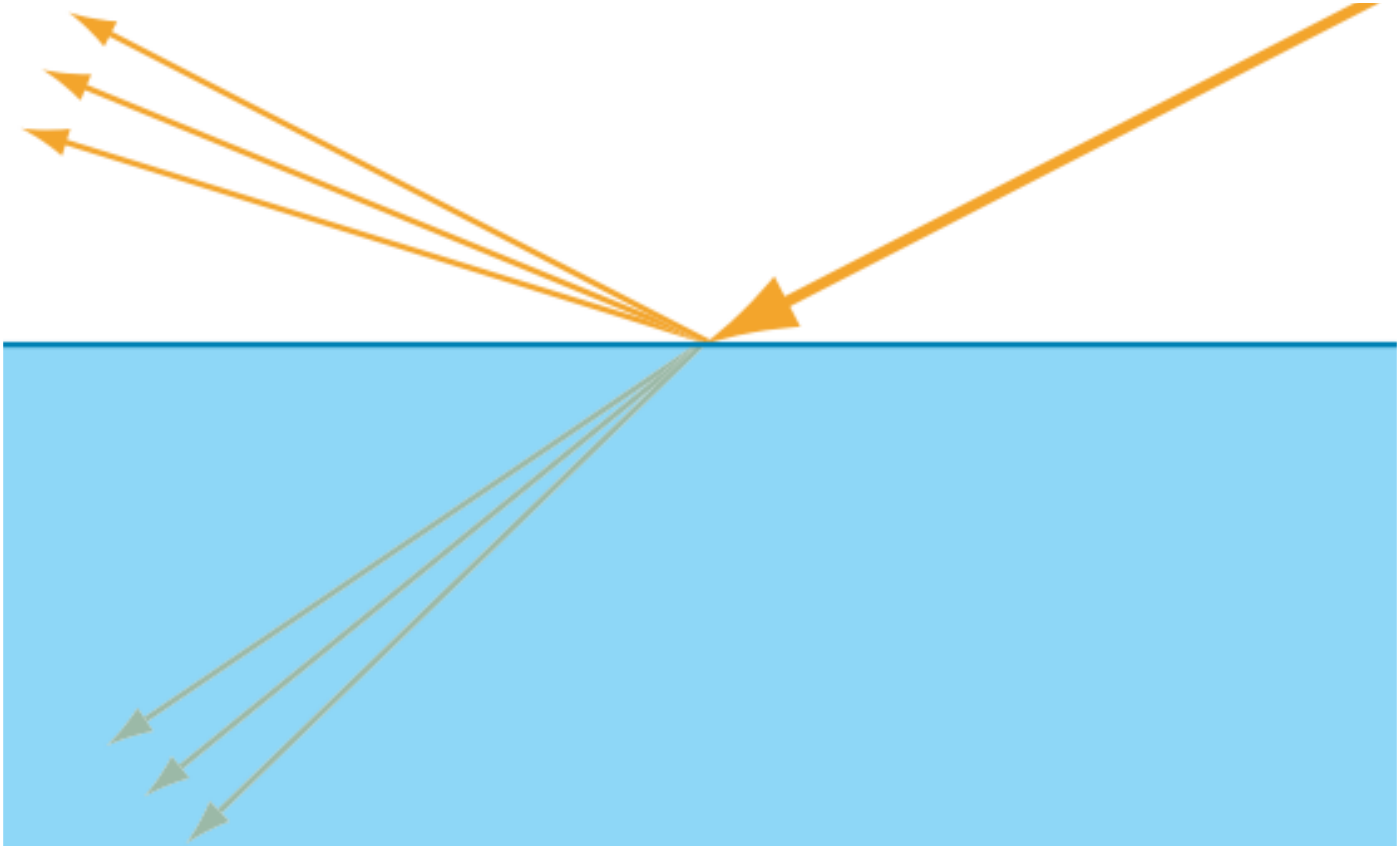
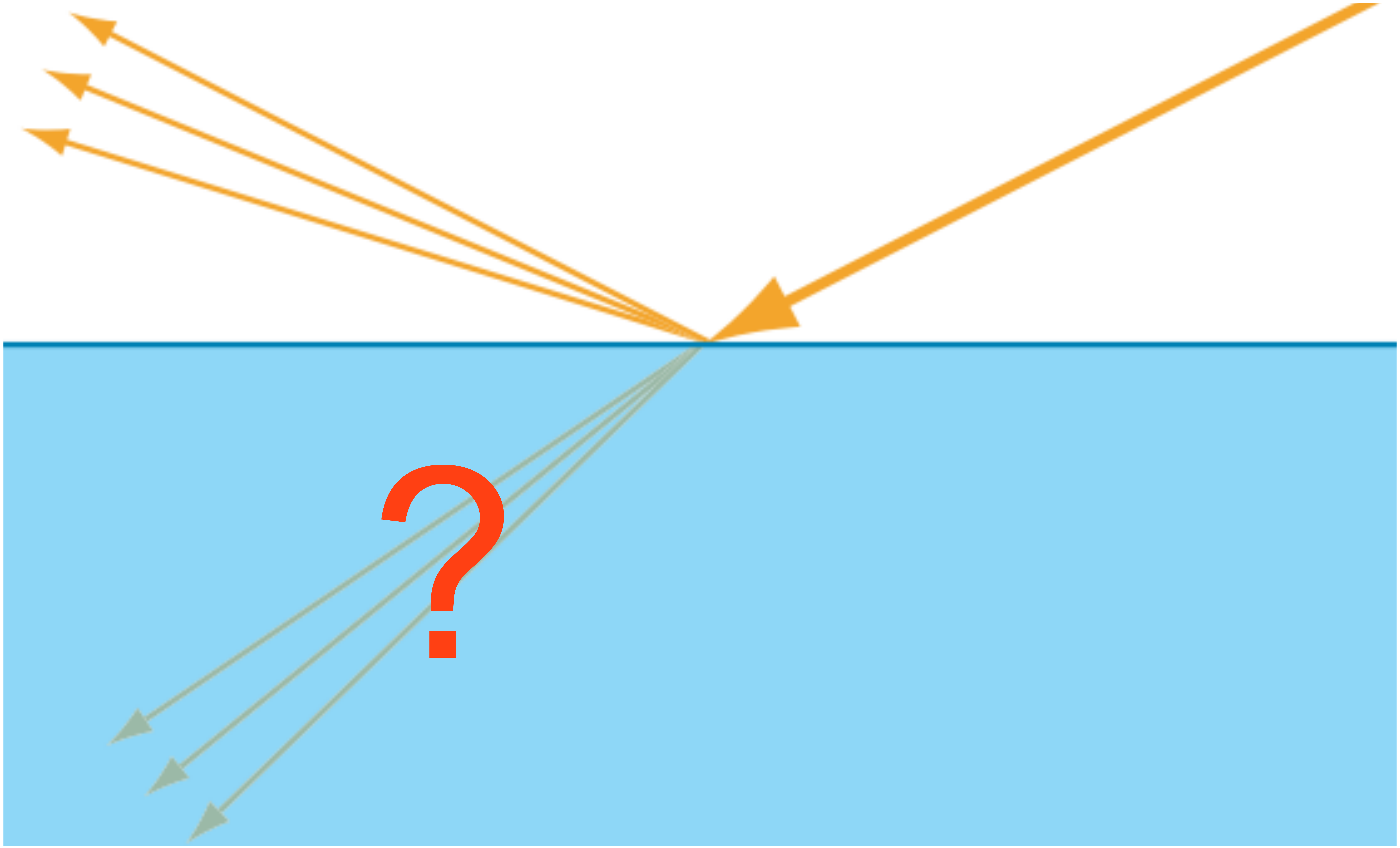


Image from “Real-Time Rendering, 3rd Edition”, A K Peters 2008

SIGGRAPH2014

In the macroscopic view, we treat the microgeometry statistically and view the surface as reflecting (and refracting) light in multiple directions. The rougher the surface, the wider the cones of reflected and refracted directions will be.



SIGGRAPH2014

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

Metals (Conductors)

Dielectrics (Insulators)

Semiconductors

Light is composed of electromagnetic waves. So the optical properties of a substance are closely linked to its electric properties. Materials can be grouped into three main optical categories: metals (or conductors), dielectrics (or insulators), and semiconductors.

Metals

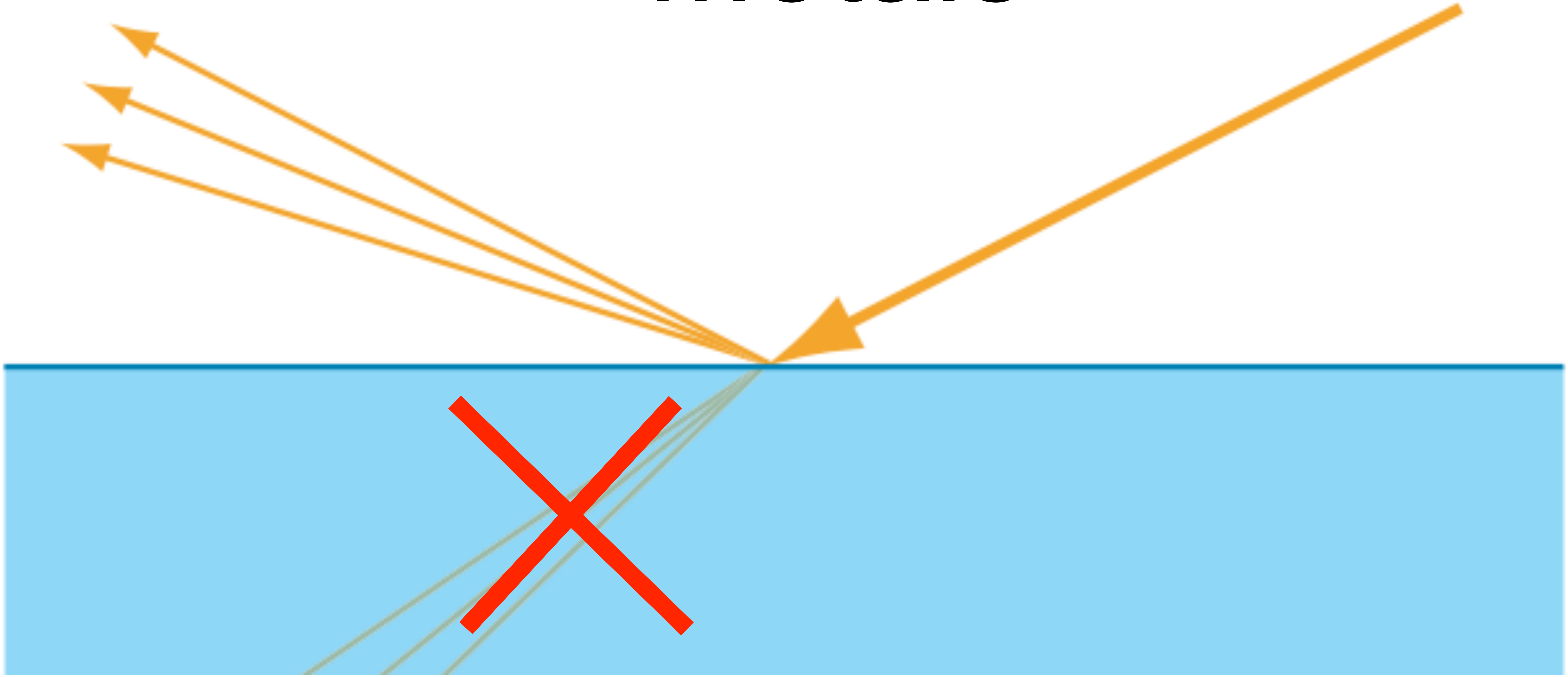
Non-Metals

~~Semiconductors~~

SIGGRAPH2014

Since visible object surfaces are rarely semiconductors, for practical purposes we can do a simpler grouping, into metals and non-metals.

Metals



Metals immediately absorb all refracted light.

Non-Metals

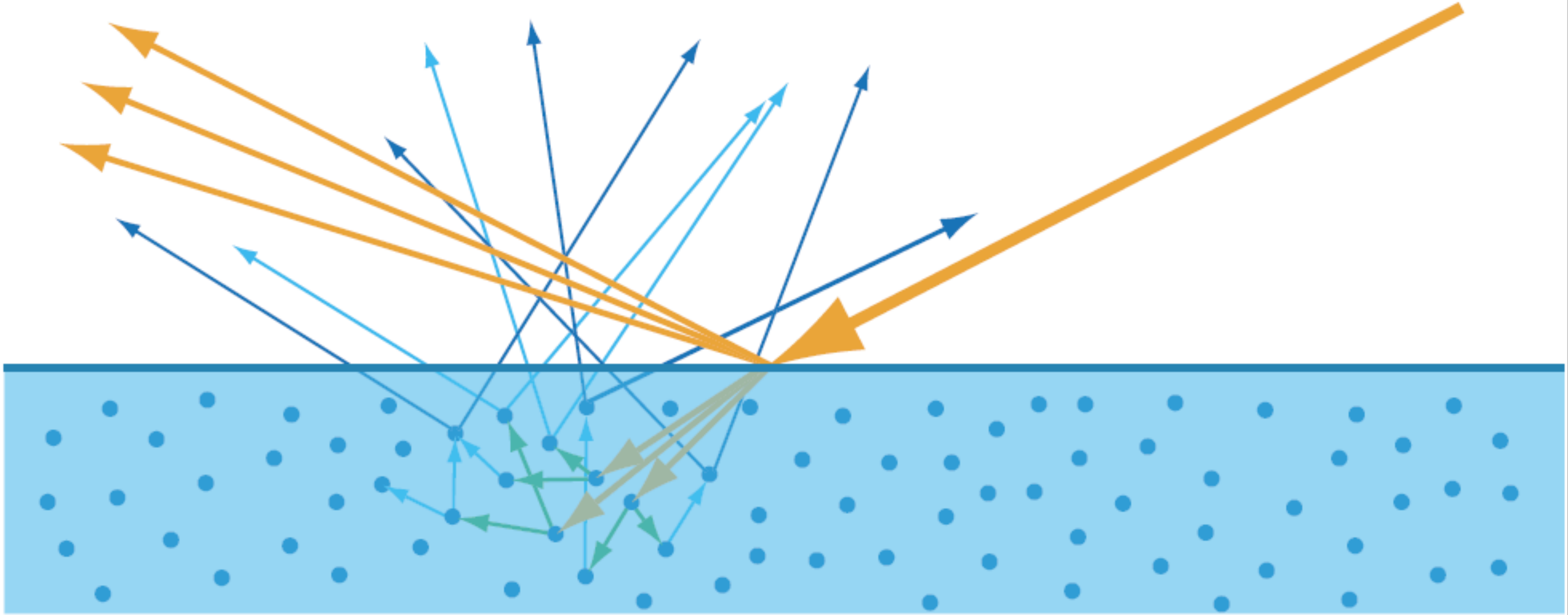


Image from "Real-Time Rendering, 3rd Edition", A K Peters 2008

SIGGRAPH2014

Non-metals behave like those cups of liquid we saw earlier: refracted light is scattered and/or absorbed to some degree. Unless the object is made out of a clear substance like glass or crystal, there will be enough scattering that some of the refracted light is scattered back out of the surface: these are the blue arrows you see coming out of the surface in various directions.

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

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

SIGGRAPH2014

If the pixel size (or shading sample area) is large (like the red-bordered green circle) compared to the entry-exit distances, we can assume that the distances are effectively zero for shading purposes.

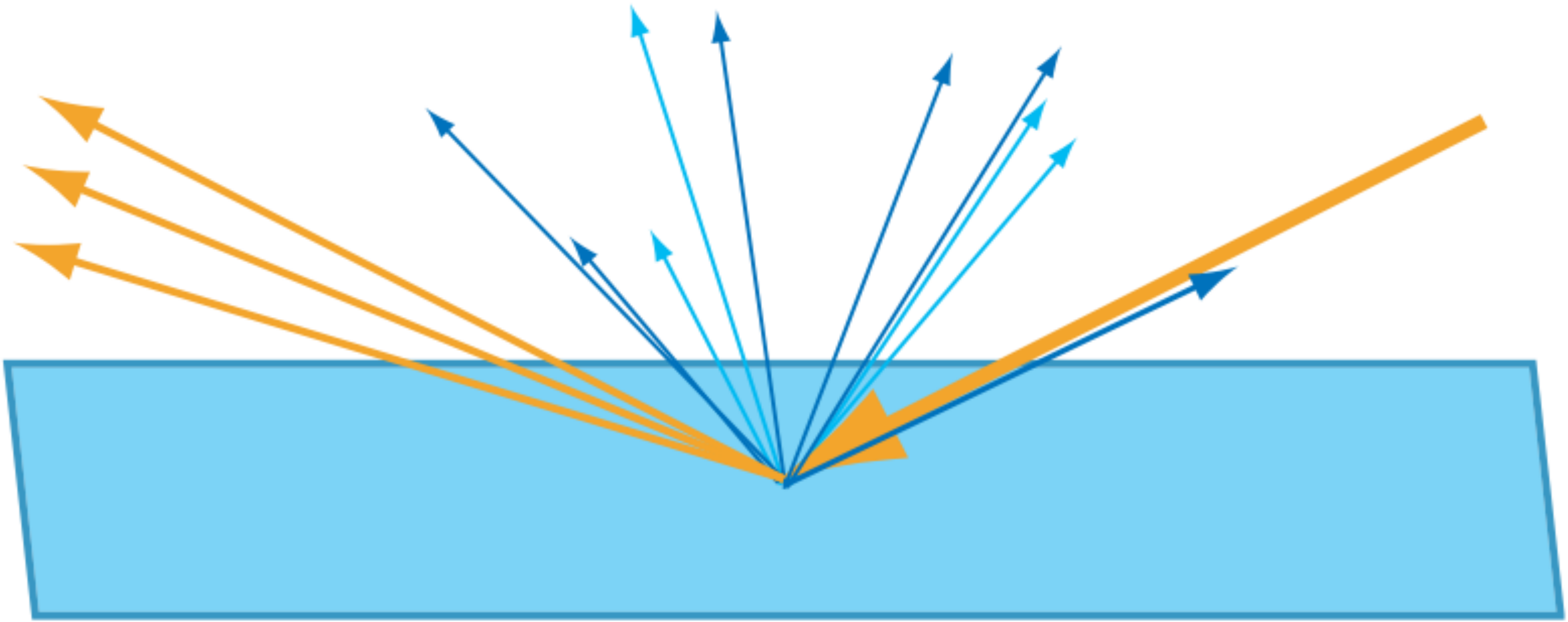


Image from “Real-Time Rendering, 3rd Edition”, A K Peters 2008

SIGGRAPH2014

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.

It is convenient to split these two very different light-material interactions into different shading terms. We call the surface reflection term “specular” and the term resulting from refraction, absorption, scattering, and re-refraction we call “diffuse”.

If the pixel is small compared to the entry-exit distances (like the red-bordered green circle), then special “subsurface scattering” rendering techniques are needed. Even regular diffuse shading is a result of subsurface scattering: the difference is the shading resolution compared to the scattering distance. For example, plastic has visible diffusion close up, as discussed in the Sunday session on “The Lego Movie”.

Physics → Math

SIGGRAPH2014

So far we've discussed the physics of light/matter interactions. To turn these physics into mathematical models that can be used for shading, the first step is to quantify light as a <u>number</u>.

Radiance

SIGGRAPH2014

Radiometry is the measurement of light. Of the various radiometric quantities, we'll use *radiance*...

Radiance

Single Ray

...which measures the intensity of light along a single ray...

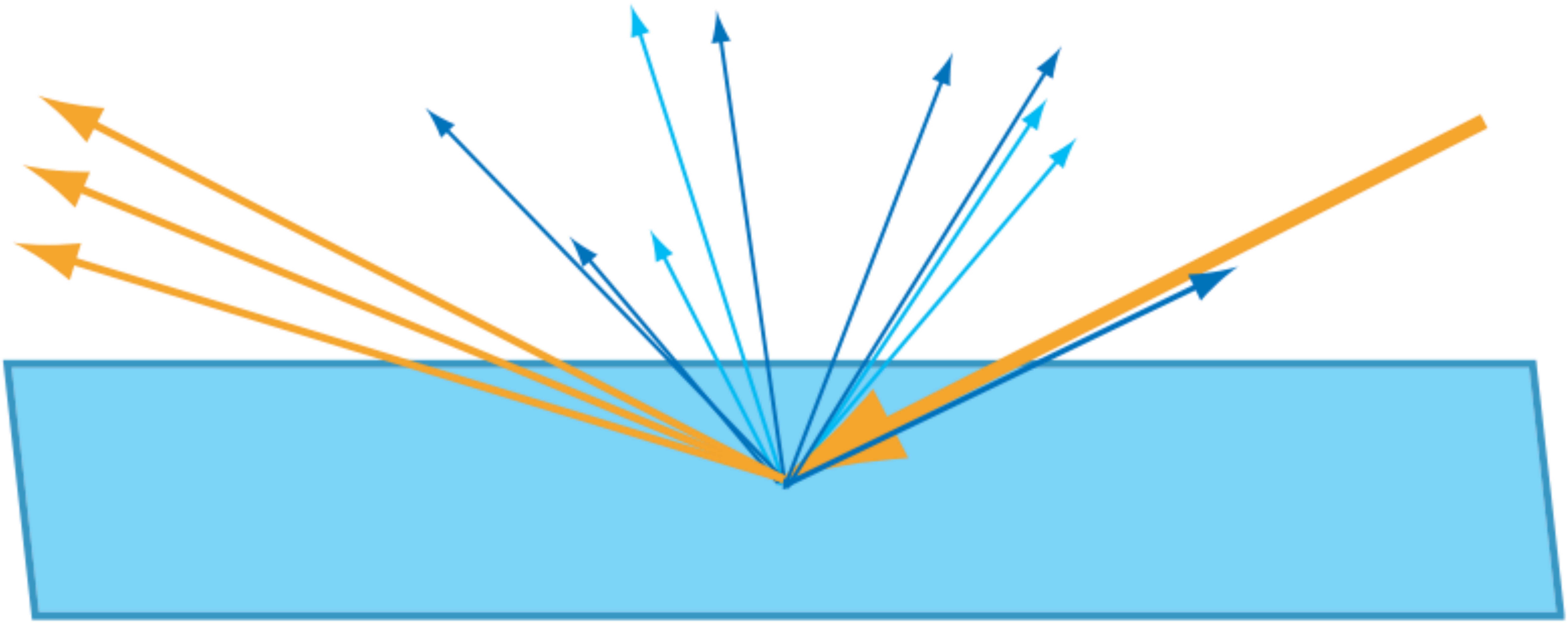
Radiance

Single Ray

Spectral/RGB

SIGGRAPH2014

...Radiance is spectral (it varies with wavelength): it's technically a continuous spectral power distribution, but for production purposes it's represented as an RGB triplet. Its units are Watts per steradian per square meter.



SIGGRAPH2014

Given the assumption that shading can be handled locally, light response at a surface point only depends on the light and view directions.

Bidirectional
Reflectance
Distribution
Function

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

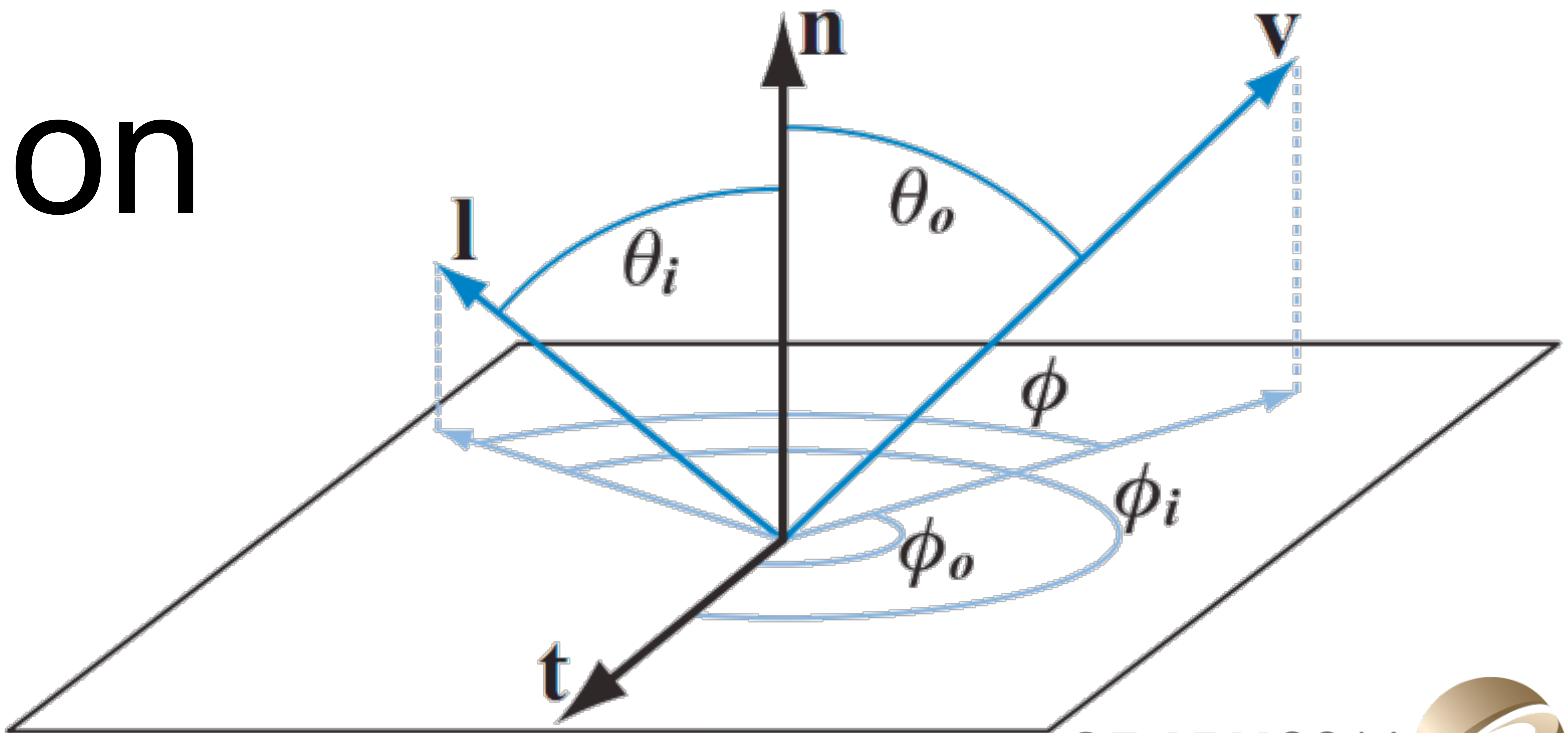


Image from "Real-Time Rendering, 3rd Edition", A K Peters 2008

SIGGRAPH2014

We represent this variation with the *BRDF*, a function of light direction $\mathbf{l}$ and view direction $\mathbf{v}$. In principle, the BRDF is a function of the 3 or 4 angles shown in the figure. In practice, BRDF models use varying numbers of angles. Note that the BRDF is only defined for light and view vectors above the macroscopic surface; see the course notes for some tips on how to handle other cases.

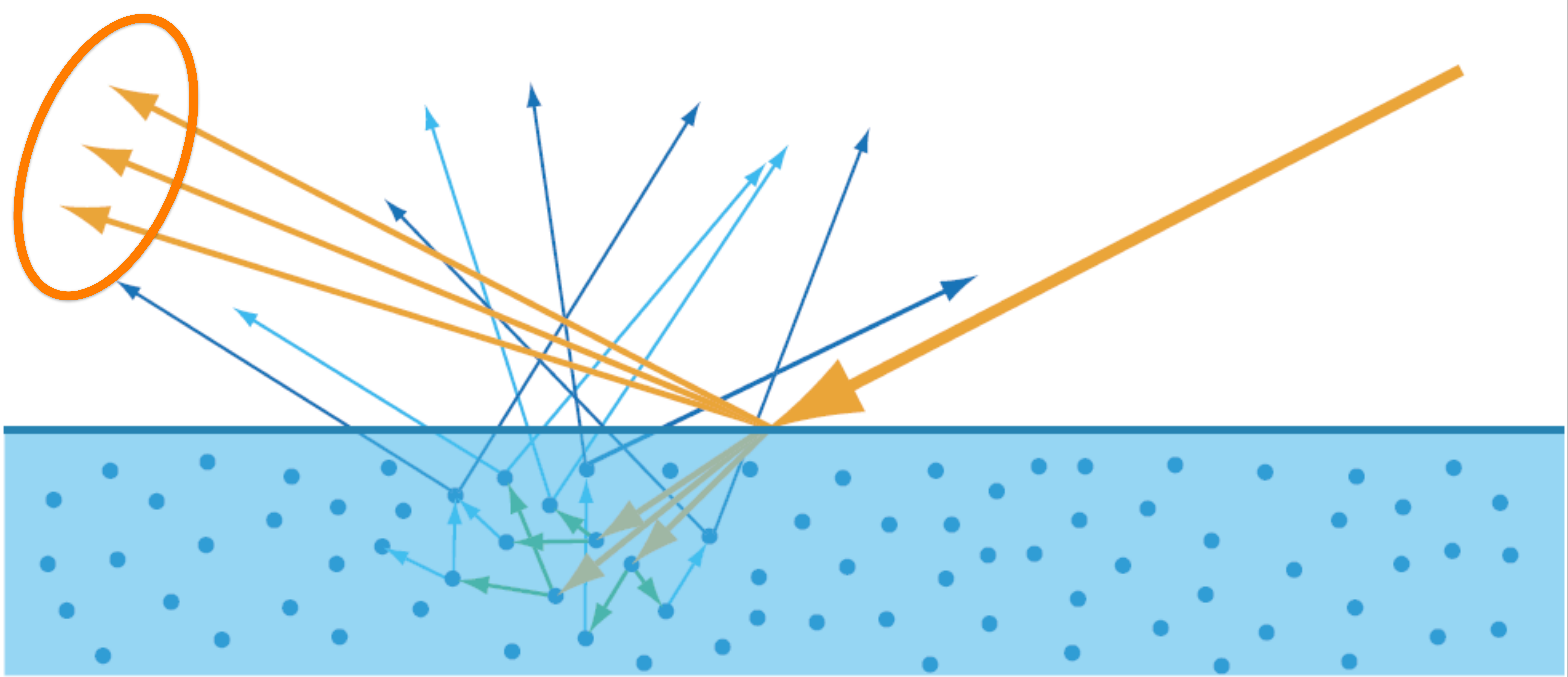
The Reflectance Equation

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

SIGGRAPH2014

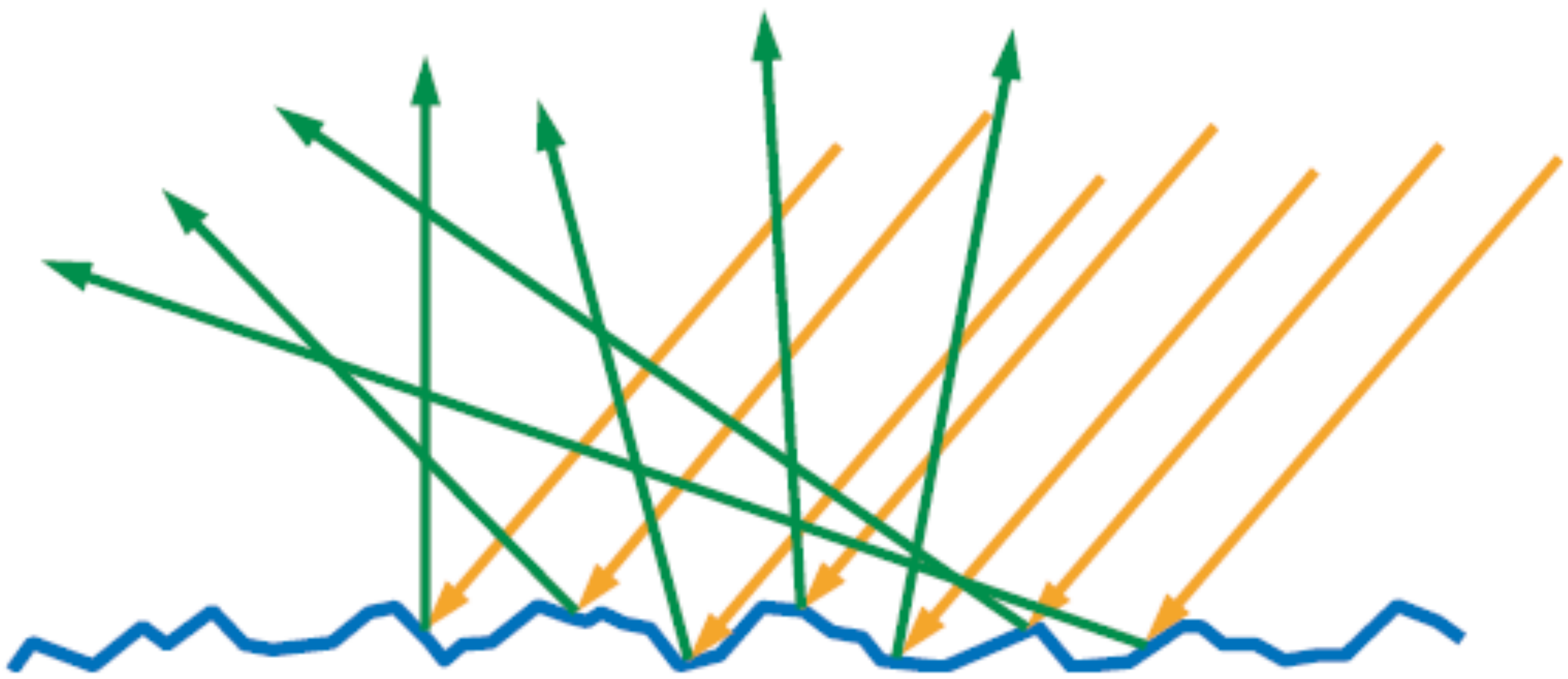
This scary-looking equation just says that outgoing radiance from a point equals the integral of incoming radiance times BRDF times a cosine factor, over the hemisphere of incoming directions. If you're not familiar with integrals you can think of this as a sort of weighted average over all incoming directions. The "X in circle" notation is from the *Real-Time Rendering* book: it means component-wise RGB multiplication.

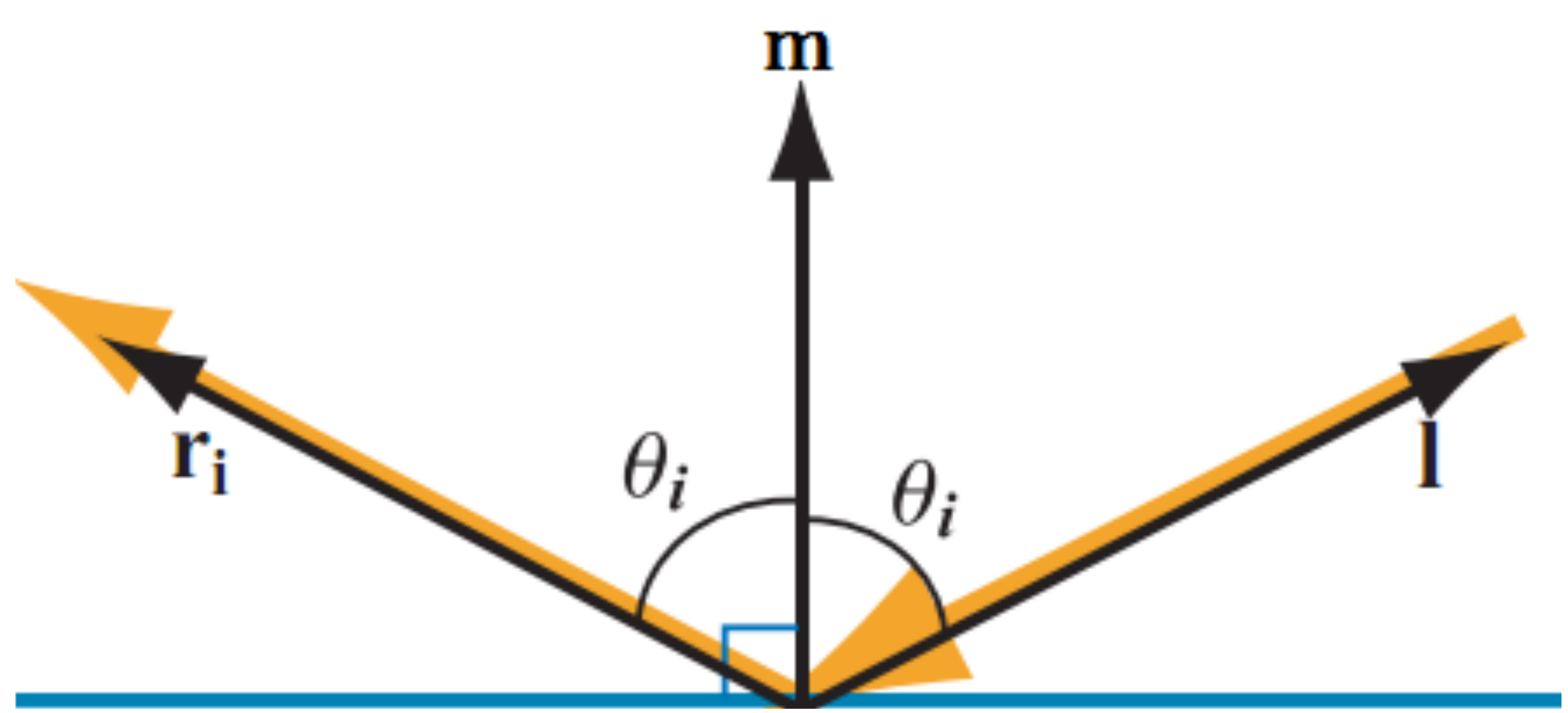
Surface Reflection (Specular Term)



We’ll start by looking at the surface, or specular term. In this figure, it is denoted by the orange arrows reflecting back from the surface.

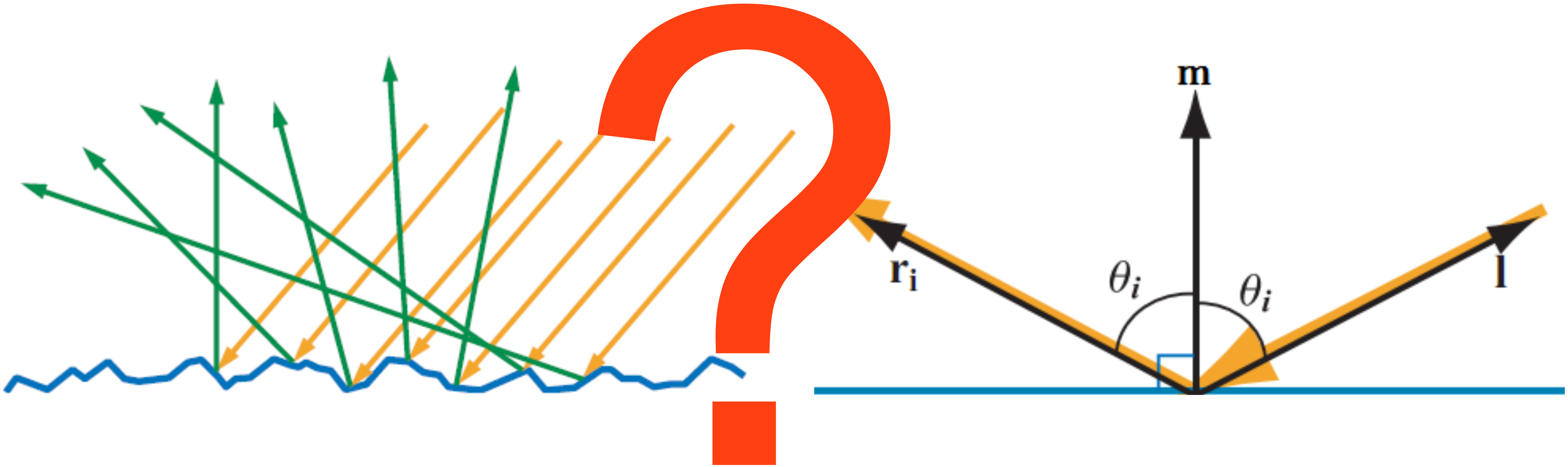
Microfacet Theory





Microfacet theory is a way to derive BRDFs for surface reflection from non-optically flat surfaces. The assumption behind it is a surface with detail that is small compared to the scale of observation but large compared to a light wavelength. Each point is locally a perfect mirror, reflecting each incoming ray of light into one outgoing direction, which depends on the light direction $\mathbf{l}$ and the microfacet normal $\mathbf{m}$.

Microfacet Theory



Microfacet theory has several limitations, which Eric Heitz will touch upon in the following talk. However it is still the best tool that we have for describing surface reflection.

The Half Vector

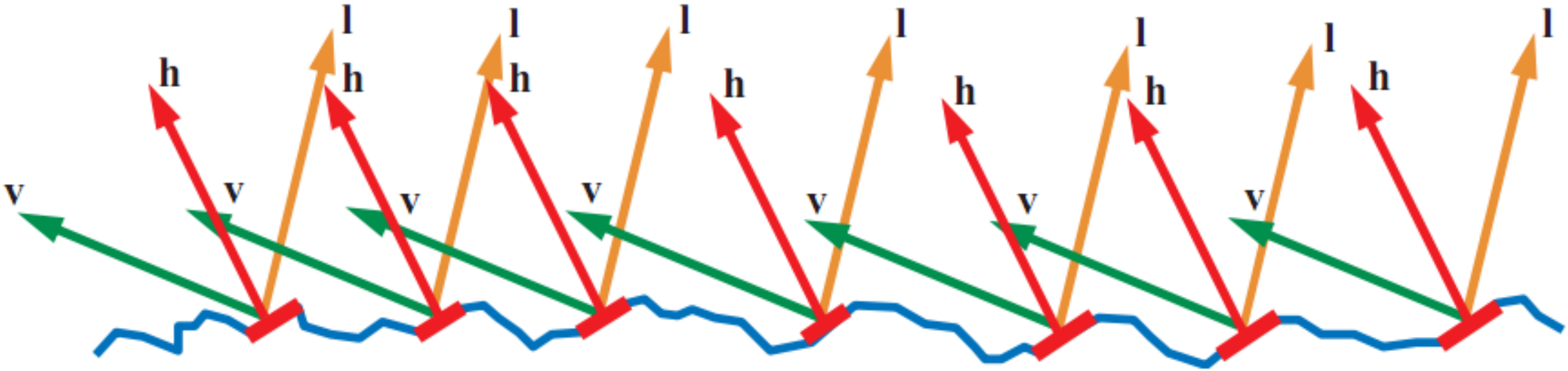
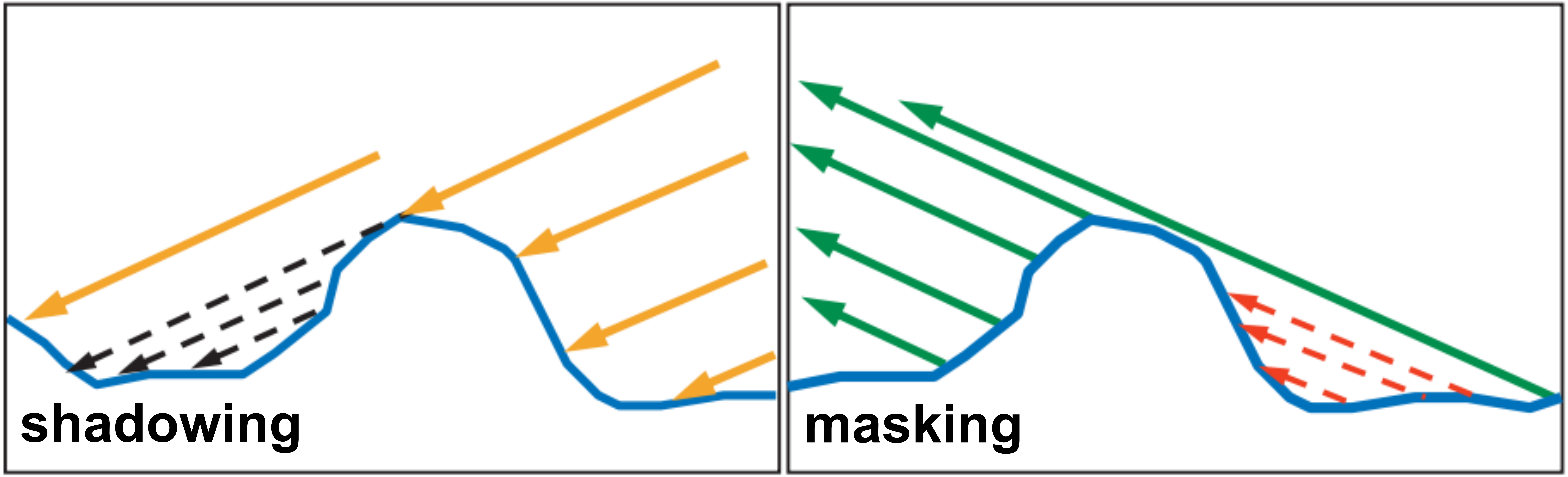


Image from "Real-Time Rendering, 3rd Edition", A K Peters 2008

SIGGRAPH2014

Only those microfacets that happen to have their surface normal **m** oriented exactly halfway between **l** and **v** will reflect any visible light: this direction is the half-vector **h**.

Shadowing and Masking



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

SIGGRAPH2014

Not all microfacets with $\mathbf{m} = \mathbf{h}$ will contribute: some will be blocked by other microfacets from either the light direction (*shadowing*) or the view direction (*masking*).

Multiple Surface Bounces

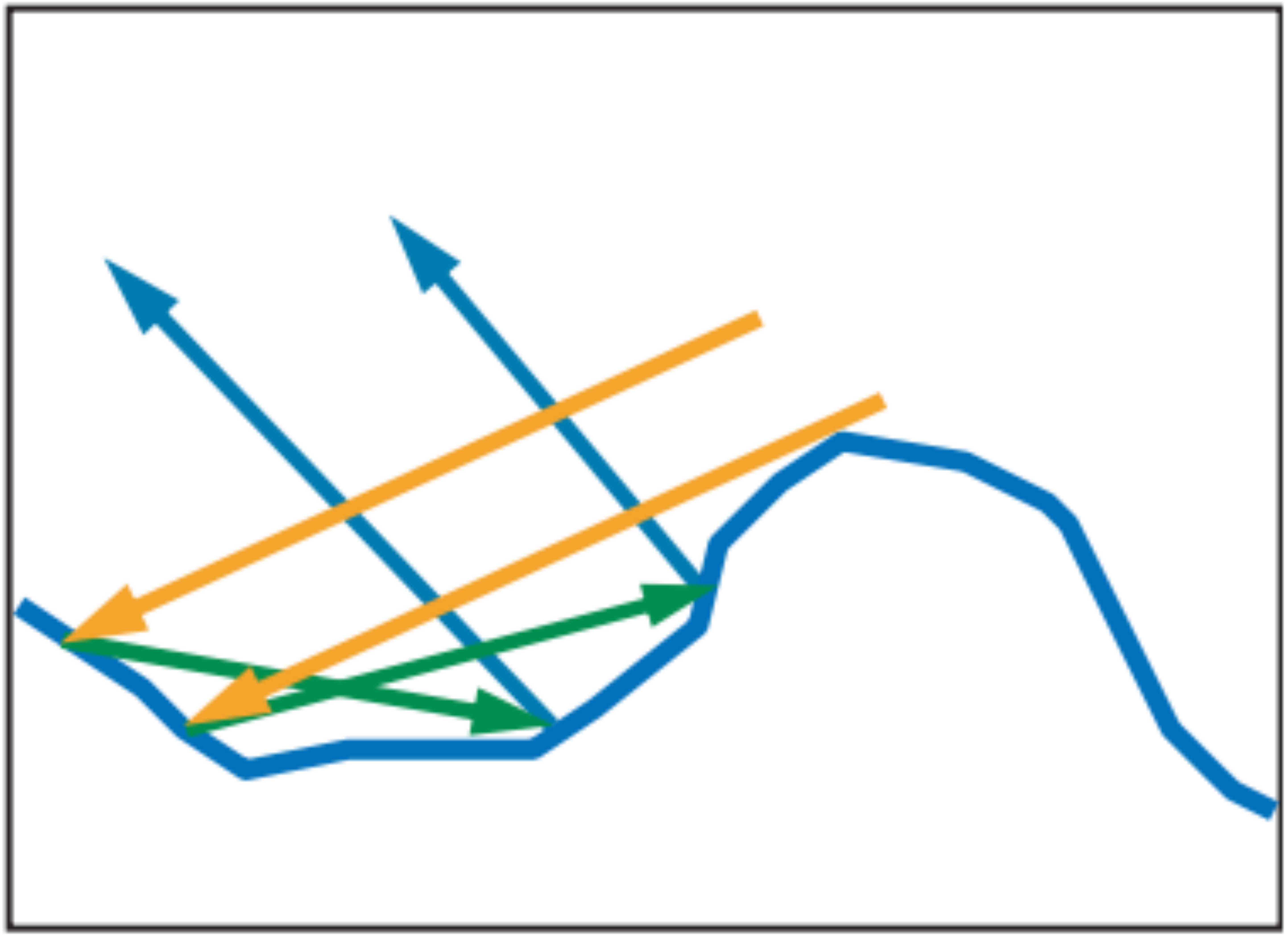


Image from “Real-Time Rendering, 3rd Edition”, A K Peters 2008

In reality, blocked light continues to bounce; some will eventually contribute to the BRDF. Microfacet BRDFs ignore this, so effectively they assume all blocked light is lost. Eric Heitz’ talk will discuss this limitation in more detail.

Microfacet Specular BRDF

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

SIGGRAPH2014

This is a general microfacet specular BRDF. I’ll go over its various parts, explaining each.

Fresnel Reflectance

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

SIGGRAPH2014

The Fresnel reflectance is the fraction of incoming light that is reflected (as opposed to refracted) from an optically flat surface of a given substance. It varies based on the light direction and the surface (in this case microfacet) normal. Fresnel reflectance tells us how much of the light hitting the relevant microfacets (the ones facing in the half-angle direction) is reflected.

Fresnel Reflectance

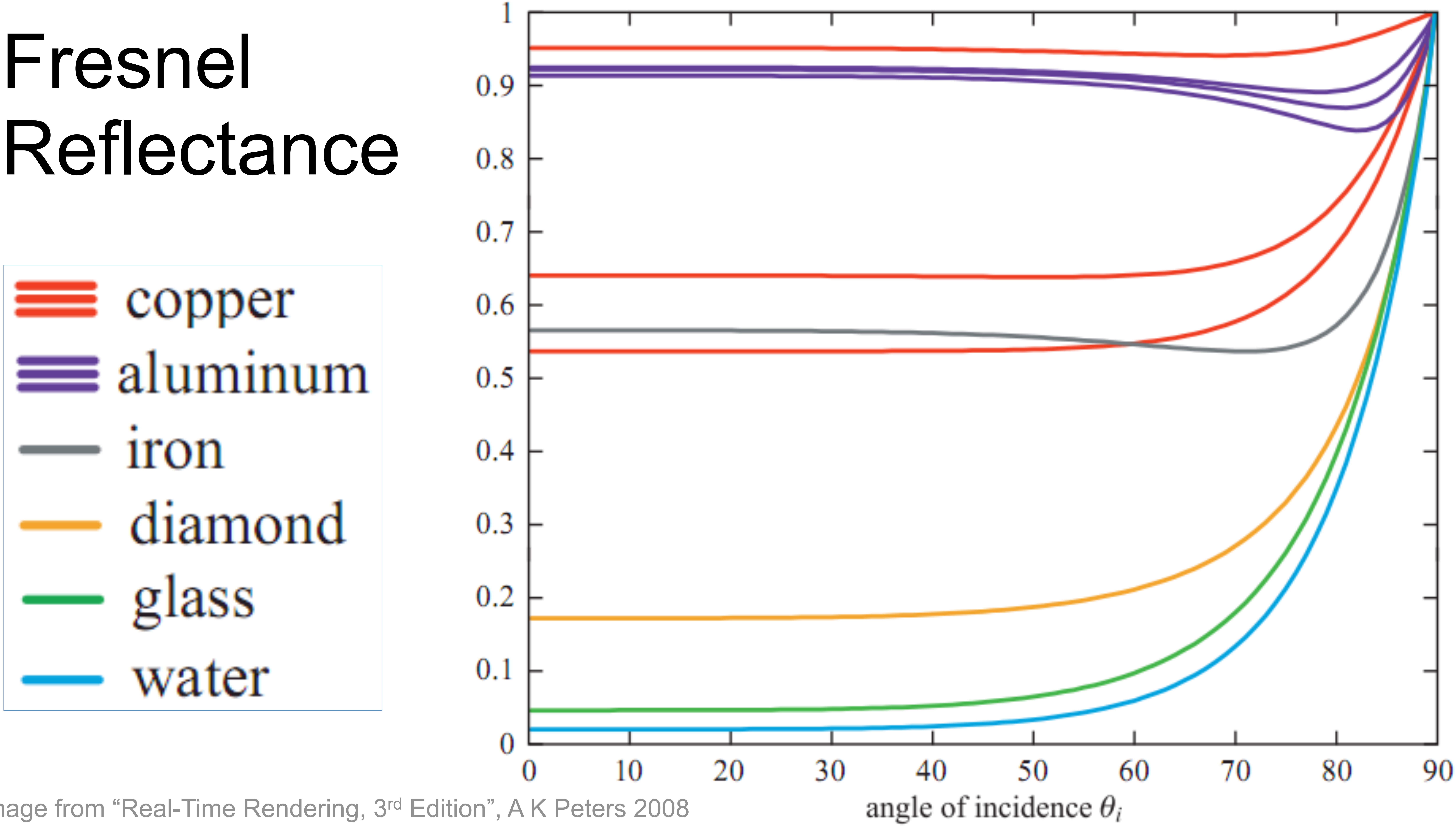


Image from “Real-Time Rendering, 3rd Edition”, A K Peters 2008

Fresnel reflectance (on the y-axis in this graph) depends on refraction index (in other words, what the object’s made of) and the incoming light angle (which is plotted here on the x-axis). In this graph, substances with three lines (copper & aluminum) have colored reflectance, which is plotted separately for the R, G and B channels—the other substances, with one line, have uncolored reflectance.

With an optically flat surface, the relevant angle for Fresnel reflectance is the one between the view and normal vectors. This image shows the Fresnel reflectance of glass (the green curve from the previous slide) over a 3D shape. See how the dark reflectance color in the center brightens to white at the edges.

Fresnel Reflectance

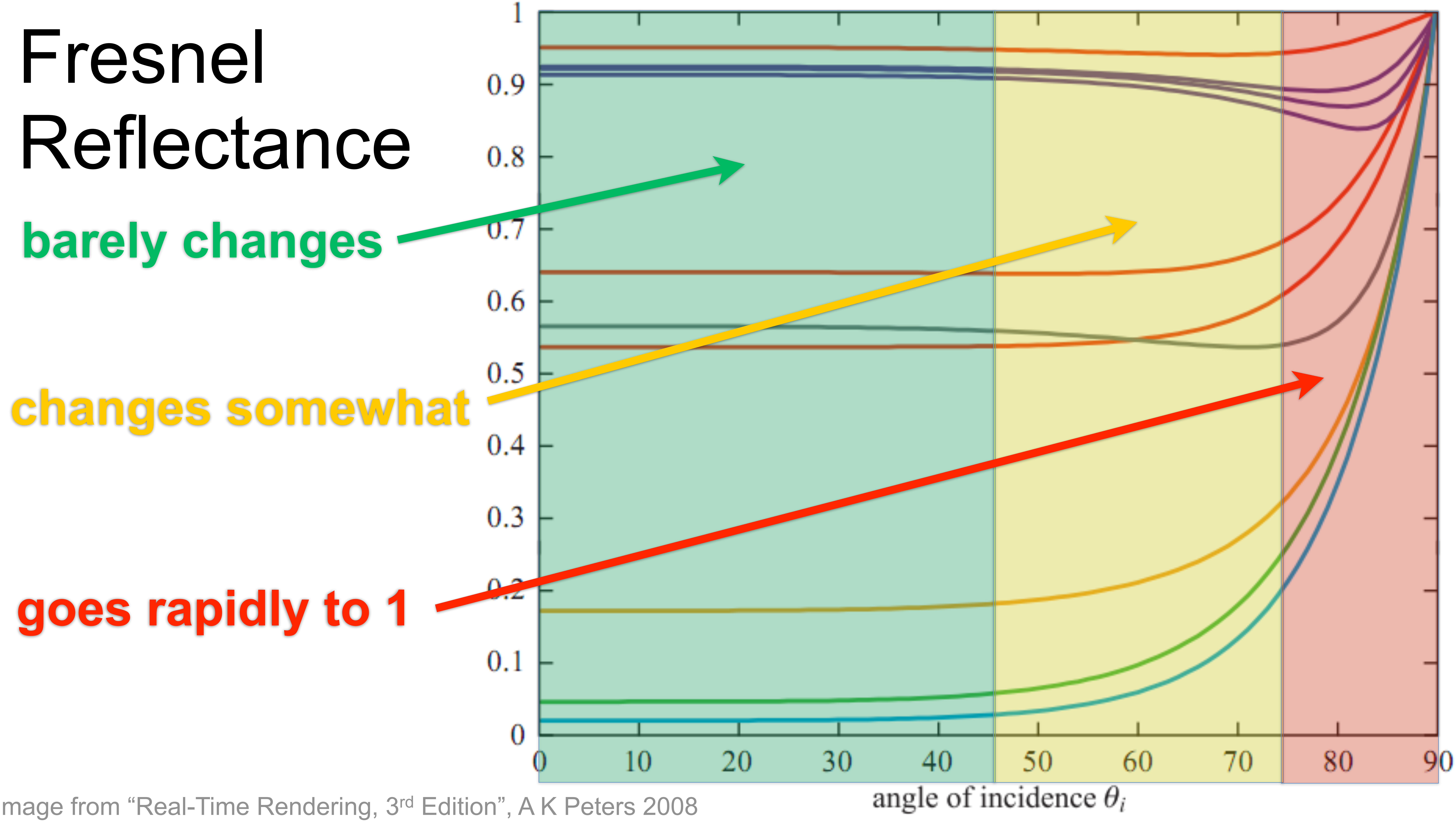


Image from "Real-Time Rendering, 3rd Edition", A K Peters 2008

As the angle increases, the Fresnel reflectance barely changes for the first 45 degrees (the green area on the graph); afterwards it starts changing, first slowly (the yellow area, up to about 75 degrees) and then for very glancing angles (the red zone) it rapidly goes to 100% at all wavelengths.

Here’s a visualization of the same zone colors over a 3D object. We can see that the vast majority of visible pixels are in the areas where the reflectance changes barely at all (green) or only slightly (yellow).

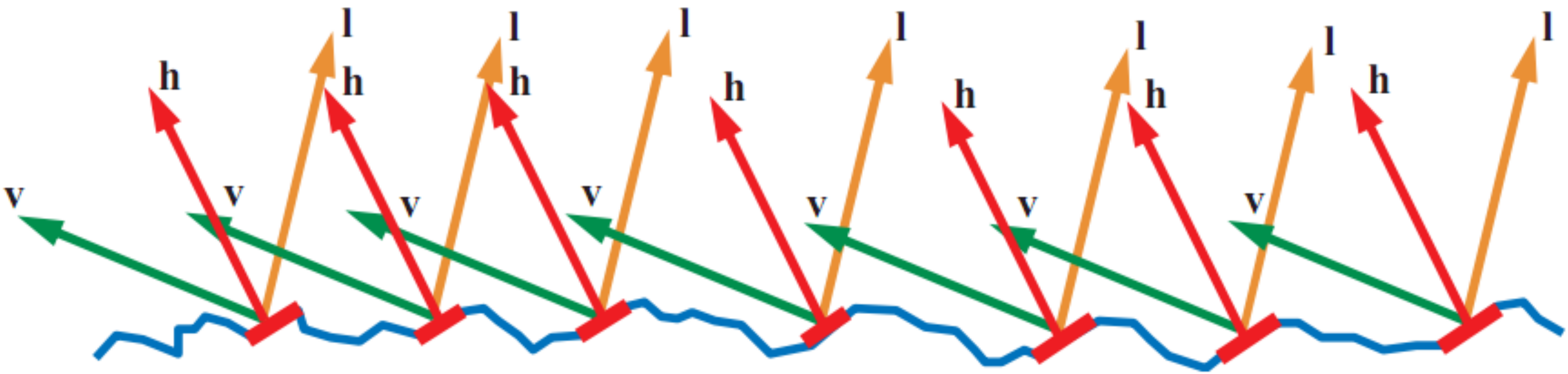


Image from “Real-Time Rendering, 3rd Edition”, A K Peters 2008

SIGGRAPH2014

Recall that in a microfacet BRDF, the relevant normal direction is the **h** vector (only the microfacets with normals aligned to **h** are visible). This means that we need to use the angle between **l** and **h** for Fresnel reflectance (or **v** and **h**; it's the same angle).

Unlike Fresnel for surface normals...

...when visualizing Fresnel zones for the **h** vector it would appear that depending on the light angle, the whole object could be in the yellow, or even the red zone.

But when we combine the Fresnel zone colors with the rest of the BRDF then we can see that green still predominates, and red can only be seen—rarely—at the object edges.

Fresnel Reflectance

$$F_0 = F(0°)$$

Is the surface's characteristic specular color

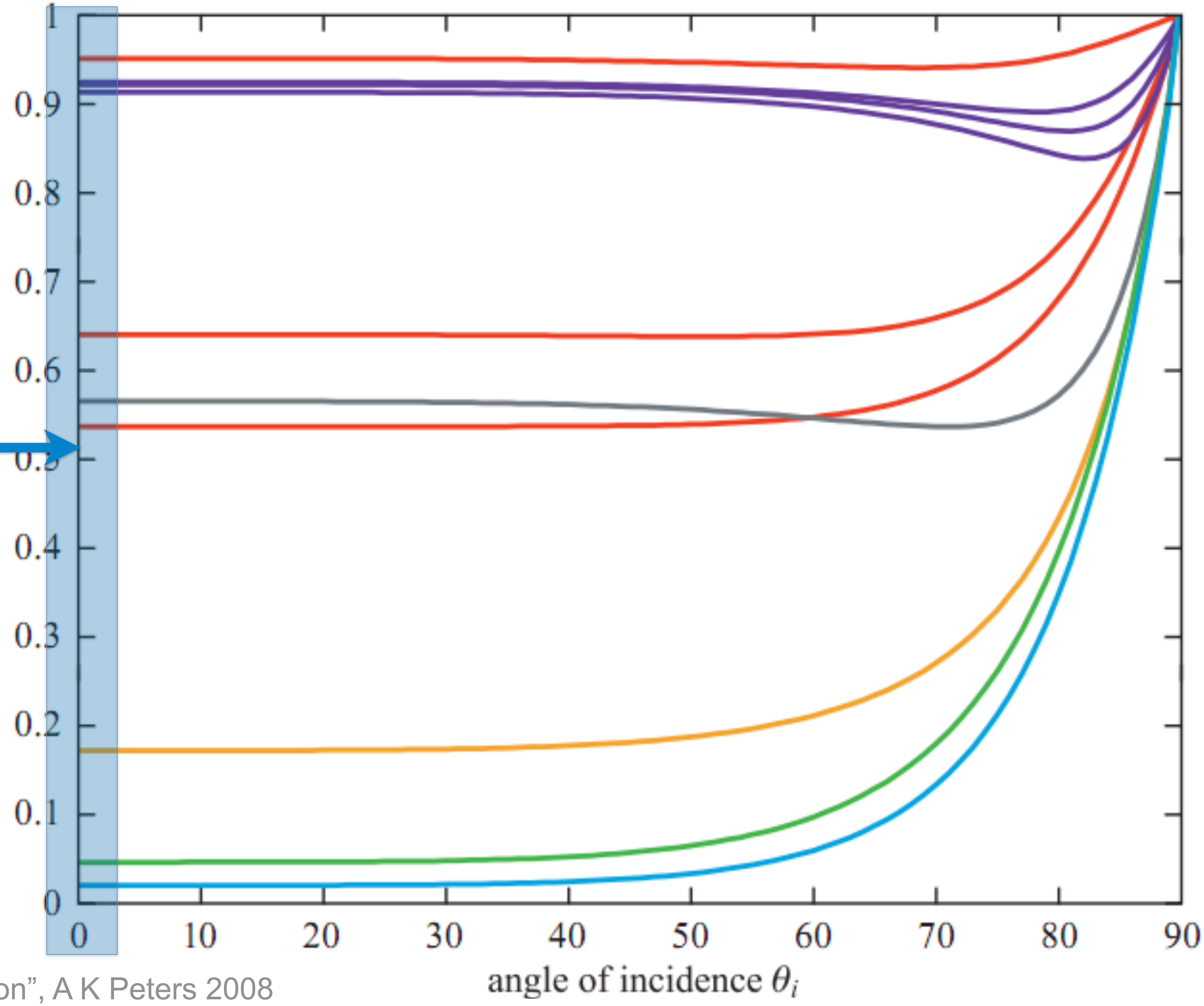


Image from "Real-Time Rendering, 3rd Edition", A K Peters 2008

Since over most of the visible surface the Fresnel reflectance value is similar to the value for 0 degrees, we can treat this value as the surface's characteristic specular color.

Metal	F_0 (Linear, Float)	F_0 (sRGB, U8)	Color
Titanium	0.542,0.497,0.449	194,187,179	
Chromium	0.549,0.556,0.554	196,197,196	
Iron	0.562,0.565,0.578	198,198,200	
Nickel	0.660,0.609,0.526	212,205,192	
Platinum	0.673,0.637,0.585	214,209,201	
Copper	0.955,0.638,0.538	250,209,194	
Palladium	0.733,0.697,0.652	222,217,211	
Zinc	0.664,0.824,0.850	213,234,237	
Gold	1.022,0.782,0.344	255,229,158	
Aluminum	0.913,0.922,0.924	245,246,246	
Silver	0.972,0.960,0.915	252,250,245	

As noted earlier, it's useful to divide substances into metals, dielectrics and semiconductors. Metals have bright specular; with one exception (gold blue channel), the linear values in this table never go far below 0.5 and most are much higher. Besides linear values, we also give 8-bit sRGB values for texture authoring. Since they lack subsurface scattering, metals get their color from surface reflection.

Metal	F_0 (Linear, Float)	F_0 (sRGB, U8)	Color
Titanium	0.542,0.497,0.449	194,187,179	
Chromium	0.549,0.556,0.554	196,197,196	
Iron	0.562,0.565,0.578	198,198,200	
Nickel	0.660,0.609,0.526	212,205,192	
Platinum	0.673,0.637,0.585	214,209,201	
Copper	0.955,0.638,0.538	250,209,194	
Palladium	0.733,0.697,0.652	222,217,211	
Zinc	0.664,0.824,0.850	213,234,237	
Gold	1.022,0.782,0.344	255,229,158	
Aluminum	0.913,0.922,0.924	245,246,246	
Silver	0.972,0.960,0.915	252,250,245	

Some metals are strongly colored, especially gold; besides an unusually low blue channel value, its red channel value is greater than 1 (it's outside sRGB gamut). The fact that gold is so strongly colored probably contributes to its unique cultural and economic significance. Despite its low blue value, gold is also one of the brightest metals—this table is ordered by lightness (CIE Y coordinate) of specular color.

F_0 Values for Dielectrics

Dielectric	F_0 (Linear, Float)	F_0 (sRGB, U8)	Color
Water	0.020	39	
Plastic, Glass	0.040 – 0.045	56 – 60	
Crystalware, Gems	0.050 – 0.080	63 – 80	
Diamond-like	0.100 – 0.200	90 – 124	

SIGGRAPH2014

On the other hand, dielectrics have dark specular colors which are achromatic, which is why this table gives single values instead of RGB triples. They also typically have a diffuse color in addition to the specular color shown in this table, so unlike metals, this is not the only source of surface color.

F_0 Values for Dielectrics

Dielectric	F_0 (Linear, Float)	F_0 (sRGB, U8)	Color
Water	0.020	39	
Plastic, Glass	0.040 – 0.045	56 – 60	
Crystalware, Gems	0.050 – 0.080	63 – 80	
Diamond-like	0.100 – 0.200	90 – 124	

SIGGRAPH2014

Going over the values in the table, in the first row we have water, which at 2% probably has the lowest F_0 value you will encounter, outside of exotic antireflective coatings.

F_0 Values for Dielectrics

Dielectric	F_0 (Linear, Float)	F_0 (sRGB, U8)	Color
Water	0.020	39	
Plastic, Glass	0.040 – 0.045	56 – 60	
Crystalware, Gems	0.050 – 0.080	63 – 80	
Diamond-like	0.100 – 0.200	90 – 124	

SIGGRAPH2014

The 2nd row at 4 to 4 1/2 percent includes values for plastics as well as ordinary or crown glass, in other words the kind of glass you will see in windowpanes, glassware, etc.

F_0 Values for Dielectrics

Dielectric	F_0 (Linear, Float)	F_0 (sRGB, U8)	Color
Water	0.020	39	
Plastic, Glass	0.040 – 0.045	56 – 60	
Crystalware, Gems	0.050 – 0.080	63 – 80	
Diamond-like	0.100 – 0.200	90 – 124	

SIGGRAPH2014

The 3rd row at 5 to 8 percent includes flint glass (also called crystal glass), which is typically used in decorative objects, wine decanters, some wine glasses, etc. It also includes most gems. The brightest types of flint glass and gems are not included in this row because they are distinctive enough to warrant their own category...

F_0 Values for Dielectrics

Dielectric	F_0 (Linear, Float)	F_0 (sRGB, U8)	Color
Water	0.020	39	
Plastic, Glass	0.040 – 0.045	56 – 60	
Crystalware, Gems	0.050 – 0.080	63 – 80	
Diamond-like	0.100 – 0.200	90 – 124	

SIGGRAPH2014

...which includes diamonds and substances used to imitate diamond, such as cubic zirconia. At the bottom end of this range at 10% we have the brightest flint glass, used in Swarovski crystals; the top at 20% is Moissanite, a diamond simulant, which is actually slightly brighter than real diamonds (which are around 17%).

F_0 Values for Semiconductors

Substance	F_0 (Linear, Float)	F_0 (sRGB, U8)	Color
Diamond-like	0.100 – 0.200	90 – 124	
Crystalline Silicon	0.345,0.369,0.426	159,164,174	
Titanium	0.542,0.497,0.449	194,187,179	

SIGGRAPH2014

What about semiconductors? As you would expect, they tend to have values in between the brightest dielectrics and the darkest metals, as we can see here using silicon as an example. Typically you will never see semiconductor surfaces in production scenes, so for practical purposes the range of F_0 values between 20 and 45 percent is a “forbidden zone” which should be avoided for realistic surfaces.

F_0 Values for Dielectrics

Dielectric	F_0 (Linear, Float)	F_0 (sRGB, U8)	Color
Water	0.020	39	
Plastic, Glass	0.040 – 0.045	56 – 60	
Crystalware, Gems	0.050 – 0.080	63 – 80	
Diamond-like	0.100 – 0.200	90 – 124	

SIGGRAPH2014

Going back to dielectrics, it's important to note that the range covered by the top two rows includes not only water, plastic and glass but also the vast majority of non-metals (basically anything that isn't crystals or gems); in other words this range is where you will find wood, leather, fabric, skin, fur, hair, paint, carpeting, tile, rocks, vegetation, earth, etc. etc.

F_0 Values for Dielectrics

Dielectric	F_0 (Linear, Float)	F_0 (sRGB, U8)	Color
Almost Everything	0.040	56	

SIGGRAPH2014

In fact, for these substances, surface gloss and Fresnel edge brightening will cause more visual variation than the relatively small variation in F_0 values. So it's often convenient to just use a constant representative value (I use 4% here, but you could also use 3 1/2 % or something similar) to represent everything that isn't an exposed metal, gem or crystal. Which is most of the surfaces you have in a typical scene.

"Metalness" Texturing

Type of Substance	F_0	Diffuse Color
Metal	Varies	0.0
Non-Metal	0.040	Varies

Metalness Mask

Color Texture

SIGGRAPH2014

There are a few ways a production can exploit the idea of using a fixed F_0 value. One is to have a simplified shader used for most surfaces that uses a hard-coded F_0 value instead of a texture. Another is to use a "metalness" mask to select between the two rows of this table, and a single "color" texture as either F_0 or diffuse color based on the metalness mask value.

Fresnel Reflectance

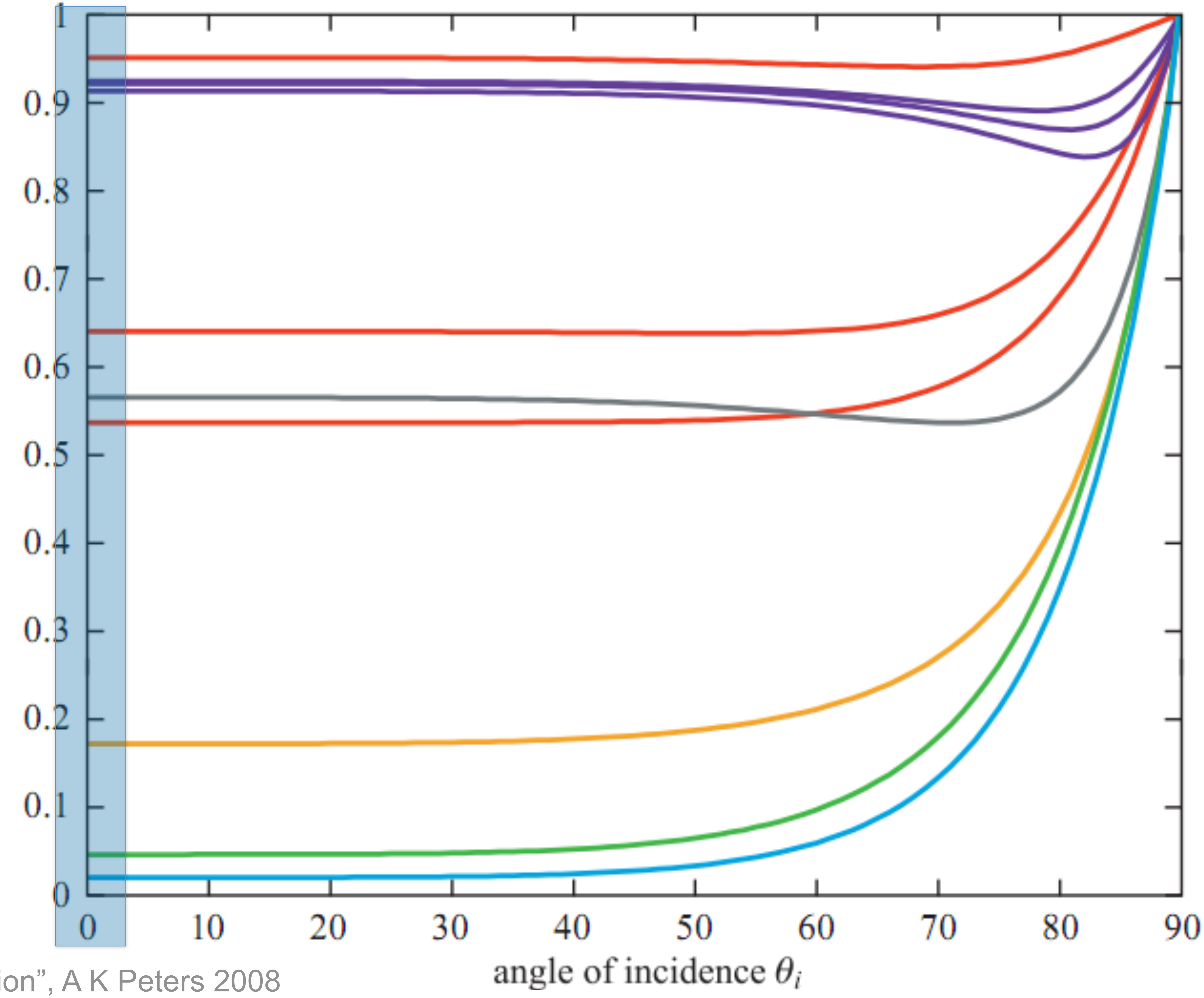


Image from “Real-Time Rendering, 3rd Edition”, A K Peters 2008

We’ve talked about how to get the value for 0 degrees. But what about the angular variation?

The Schlick Approximation to Fresnel

- Fairly accurate, cheap, parameterized by F_0

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

- For microfacet BRDFs ($\mathbf{m} = \mathbf{h}$):

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

SIGGRAPH2014

In production, the Schlick approximation to Fresnel is commonly used. It is cheap and reasonably accurate; more importantly, it is parameterized by specular color. As we saw previously, when using it in microfacet BRDFs, the $\mathbf{h}$ vector is used in place of the normal.

Normal Distribution Function

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

SIGGRAPH2014

The next part of the microfacet BRDF we will discuss is the microfacet normal distribution function, or NDF. The NDF gives the concentration of microfacet normals pointing in a given direction (in this case, the half-angle direction), relative to surface area. The NDF determines the size and shape of the highlight.

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

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

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

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

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

SIGGRAPH2014

The course notes detail various options for NDFs.

Some NDFs are Gaussian with “blobby” highlights...

Others have a more “spiky” shape with long tails, leading to sharp highlights with “halos” around them.

Geometry Function

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

SIGGRAPH2014

The geometry or shadowing-masking function gives the chance that a microfacet with a given orientation (again, the half-angle direction is the relevant one) is lit and visible (in other words, not shadowed and/or masked) from the given light and view directions.

Cheap Option

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

Equivalent to:

$$G_{\text{implicit}}(\mathbf{l},\mathbf{v},\mathbf{m}) = (\mathbf{n}\cdot\mathbf{l})(\mathbf{n}\cdot\mathbf{v})$$

SIGGRAPH2014

A cheap option sometimes used in games is to just omit this entire subexpression, effectively equating it to one. This implies an “implicit” geometry function equal to n-dot-l times n-dot-v. Although it is very cheap and somewhat plausible (going from one when view and light are in the normal direction, to zero when either is at 90 degrees), it isn’t accurate.

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

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

$$G_{\mathrm{s}}(\mathbf{l},\mathbf{v},\mathbf{h}) = G_{\mathrm{s1}}(\mathbf{l},\mathbf{h})G_{\mathrm{s1}}(\mathbf{v},\mathbf{h})$$

SIGGRAPH2014

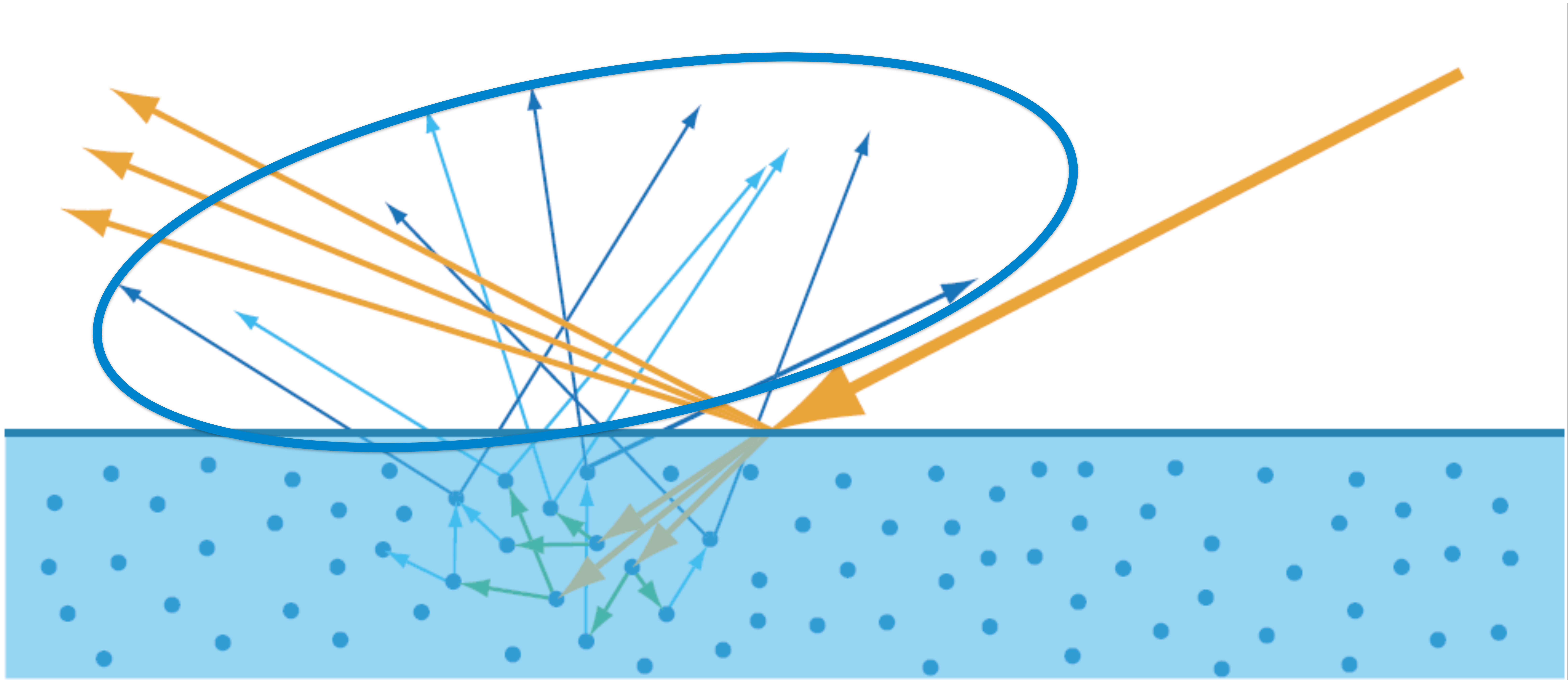
The literature has various other options for the geometry function. The following talk by Eric Heitz will give some great insights into shadowing and masking in microfacet BRDFs, and will show how to determine which of these functions are correct.

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

SIGGRAPH2014

Putting it all together, we see that the BRDF is proportional to the concentration of active microfacets (the ones with normals aligned with h) times their visibility times their Fresnel reflectance. The rest of the BRDF (in the denominator) consists of correction factors relating to the various frames involved (light frame, view frame, local surface frame).

Subsurface Reflection (Diffuse Term)



Until now we've been focusing on the specular—or surface—reflection term. Next, we'll take a quick look at the diffuse (or subsurface) term.

Lambert

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

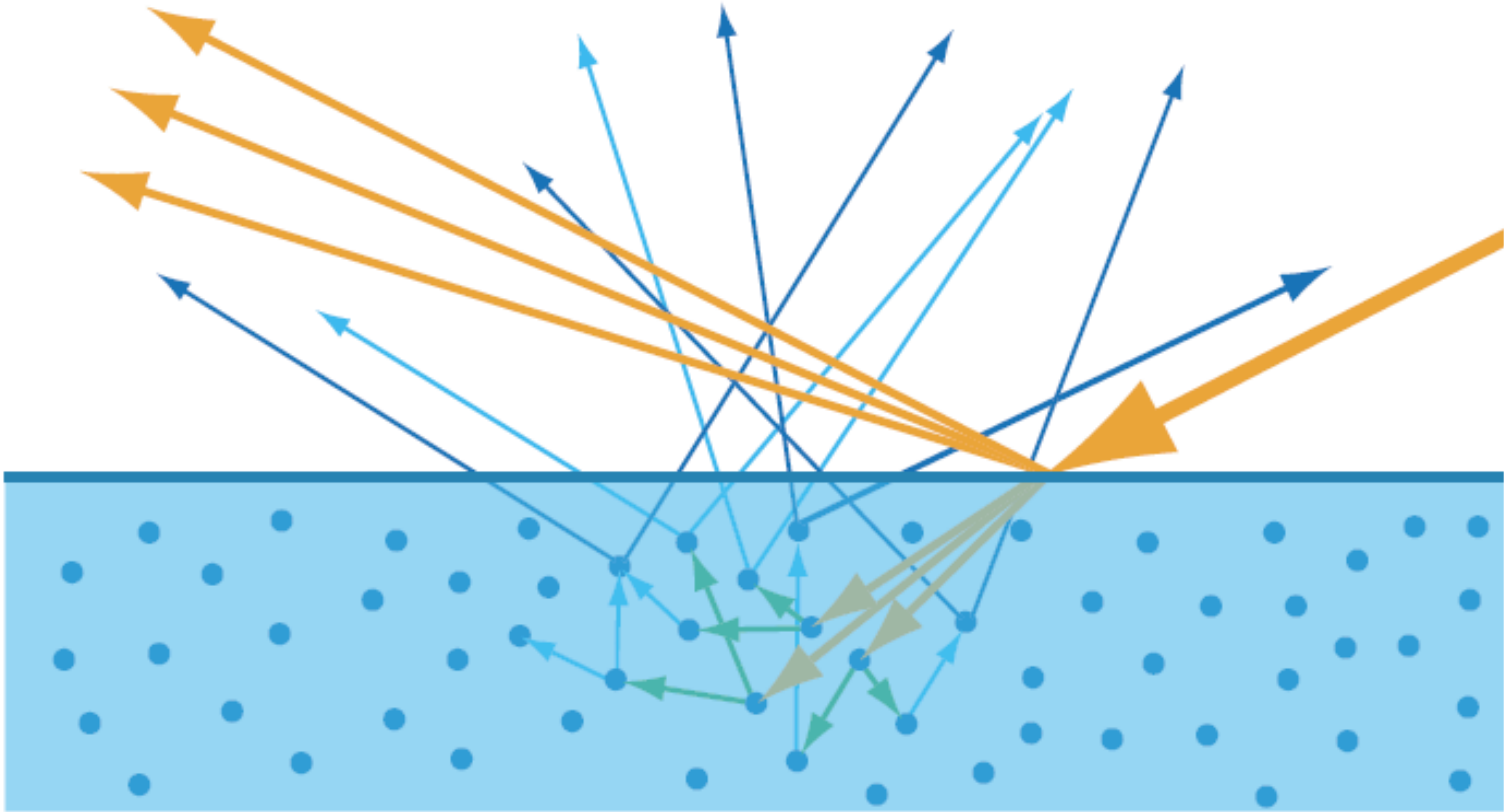
$$f_{\text{Lambert}}(\mathbf{l}, \mathbf{v}) = \frac{\mathbf{c}_{\text{diff}}}{\pi}$$

- $\mathbf{c}_{\text{diff}}$: fraction of light reflected, or diffuse color

SIGGRAPH2014

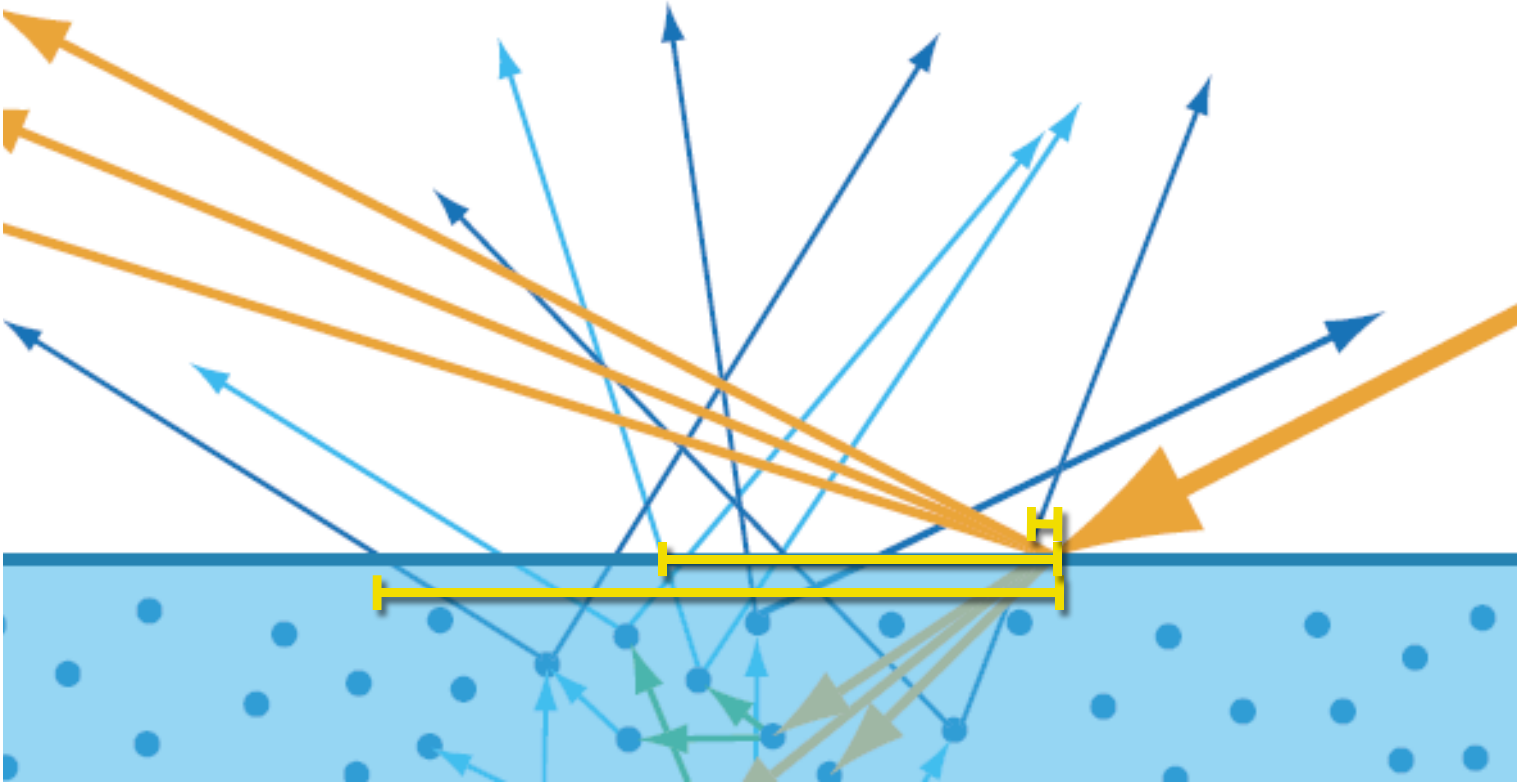
The Lambert model is the most common diffuse term used in game and film production. By itself, it's the simplest possible BRDF: a constant value. The well-known cosine factor is part of the reflectance equation, not the BRDF.

Beyond Lambert: Diffuse-Specular Tradeoff



There are a few important physical phenomena that Lambert doesn't account for. Diffuse comes from refracted light. Since the specular term comes from surface reflection, in a sense it 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%. The course notes discuss a few ways to model this.

Beyond Lambert: Surface Roughness



Lambert also doesn't account for surface roughness. In most cases, microscopic roughness only affects specular; diffuse reflectance at a point comes from incoming light over an area, which tends to average out any microgeometry variations. But some surfaces have microgeometry larger than the scattering distance, and these do affect diffuse reflectance. That's when you need models like Oren-Nayar.

Math $\longrightarrow$ Rendering

We’ve talked about how to represent the physics of light/matter interactions in mathematical shading models. But how do we implement these in a renderer?

shading model + **illumination model**

= rendering implementation

SIGGRAPH2014

To be implemented in a renderer, a shading model needs to be combined with a specific illumination model.

- General Lighting

SIGGRAPH2014

In the most general illumination model, the BRDF is integrated against continuous incoming light from all directions (area light sources, skylight, indirect reflections). Implementing this requires global illumination algorithms such as ray tracing. However, even ray tracers can gain significant performance advantages from using less general illumination models, such as...

- General Lighting
- Image-Based Lighting

SIGGRAPH2014

...image-based lighting, where incoming radiance from various directions is cached into an image, such as an environment map.

- General Lighting
- Image-Based Lighting
- Area Light Sources

SIGGRAPH2014

It is often advantageous to separate light sources such as the sun and lamps into specialized illumination models which take account of their brightness and area. Area light sources are commonly used in film but are difficult to implement efficiently enough to be used in games; Brian Karis' talk from last year's course showed one way to do so. Instead of area light sources, games typically use...

- General Lighting
- Image-Based Lighting
- Area Light Sources
- Punctual Light Sources

SIGGRAPH2014

...punctual light sources, which can be implemented much more efficiently (these are also still used in film to some extent).

- General Lighting
- Image-Based Lighting
- Area Light Sources
- Punctual Light Sources
- Ambient Light

SIGGRAPH2014

Ambient light includes various low-frequency lighting representations, ranging from a single constant light color and intensity over all incoming directions to more complex representations such as spherical harmonics. We will now focus on the most commonly used light representations: punctual lights and image-based lighting.

Punctual Light Sources

- Parameterized by light color $\mathbf{c}_{\text{light}}$ and direction to the light (center) position $\mathbf{l}_c$
- $\mathbf{c}_{\text{light}}$ equals radiance from a white Lambertian surface illuminated by the light at 90 degrees

SIGGRAPH2014

Infinitely small and bright, punctual light sources are not realistic, but are the easiest to implement with arbitrary BRDFs. The resulting shading appears reasonable for rough surfaces, less so for smooth ones. Punctual lights are parameterized by direction and color: the latter typically related to the brightness of a white diffuse surface lit by the light.

Punctual Light Equation

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

SIGGRAPH2014

Implementing a shading model with punctual lights is extremely simple: just evaluate the BRDF in a single light direction and multiply by π (the derivation of this is in the course notes). Games often clamp the dot product between the light and normal vectors as a convenient method to remove the contribution of backfacing lights.

Image-Based Lighting

Optically flat (mirror) surface is easy: just multiply reflection by Fresnel function; same F_0, different angle:

– $F(\mathbf{v}, \mathbf{n})$ instead of $F(\mathbf{l}, \mathbf{h})$ or $F(\mathbf{v}, \mathbf{h})$

SIGGRAPH2014

Image-based lighting (also called environment mapping) is useful for all materials, but especially for smooth or metallic ones. The trivial case is optically flat surfaces, for which there is a single reflection direction and a single surface normal.

Image-Based Lighting

Non-mirror (glossy/diffuse) surfaces require many samples

- Importance sampling helps
- Prefiltering (alone or with importance sampling)

SIGGRAPH2014

Surfaces that aren't simple mirrors require sampling the environment many times. Film uses importance sampling (sometimes with prefiltering) to reduce sample count. Games have an extremely low sample budget (typically a single sample is used per IBL source; several IBL sources may be combined in a shader), so they rely on prefiltering.

Filtering & Antialiasing

- As we have seen, the BRDF is intimately tied to the rendering scale
- Bumps which are individually visible when near turn into microgeometry as distance increases
  – Appearance changes unless modeled in the BRDF
  – Aliasing is also common in these situations

SIGGRAPH2014

We have touched a few times upon the relationship between the BRDF and the scale of observation. Specular especially is tied into sub-visible surface detail, but as the distance to an object changes, detail moves between visible and sub-visible. This causes both changes in appearance and aliasing artifacts.

Filtering & Antialiasing

- Several solutions exist:
  - Toksvig AA
  - Variance Mapping
  - LEAN/CLEAN Mapping
  - LEADR Mapping

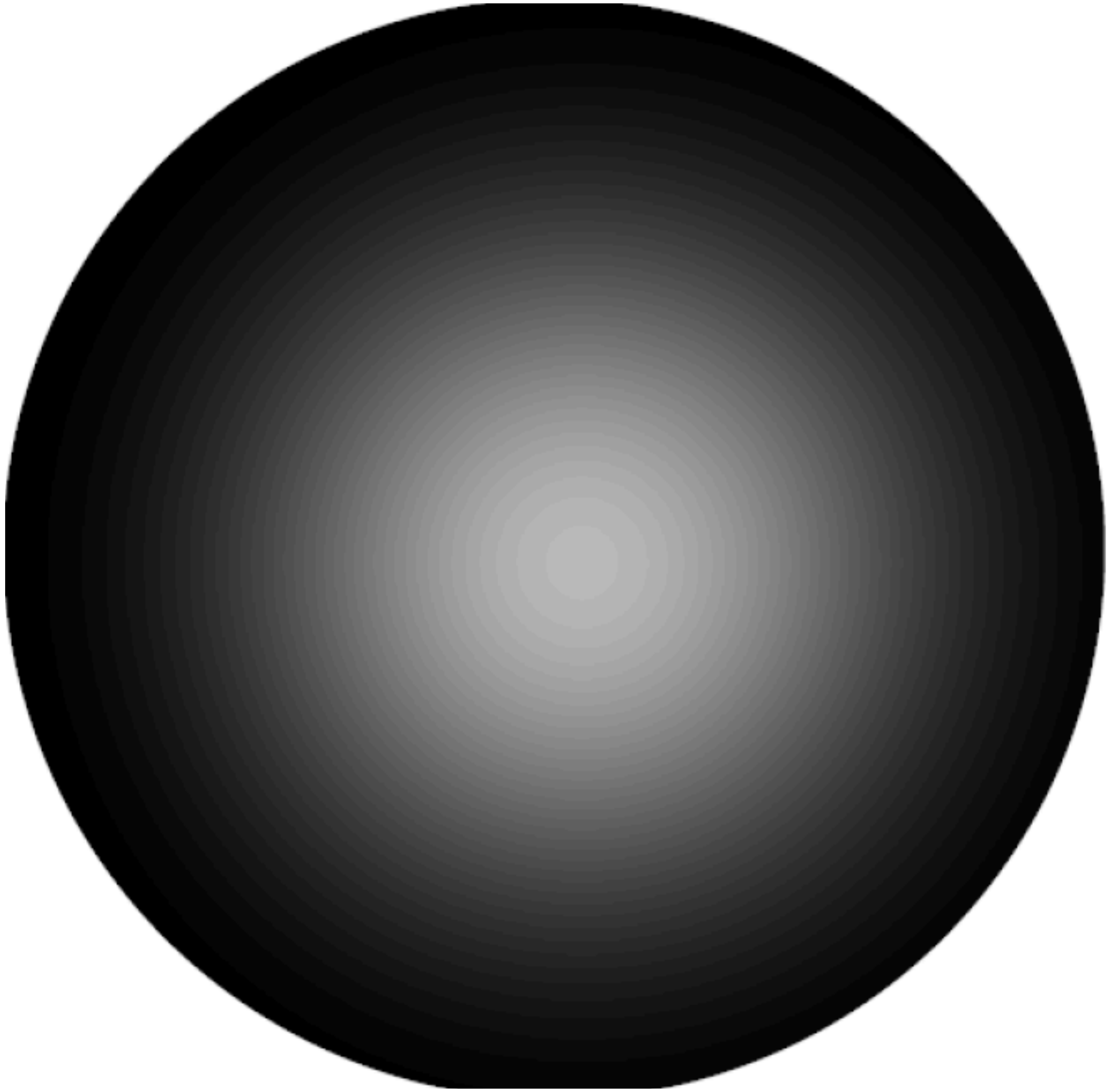


Image from “Rendering Glints on High-Resolution Normal-Mapped Specular Surfaces”, Yan et al., SIGGRAPH 2014

There are several solutions to this problem; one of them—LEADR Mapping— will be discussed later in this course. Some of these solutions approximate the pixel NDF (distribution of normals covered by a pixel) with a single isotropic lobe...

Filtering & Antialiasing

- Several solutions exist:
  - Toksvig AA
  - Variance Mapping
  - LEAN/CLEAN Mapping
  - LEADR Mapping

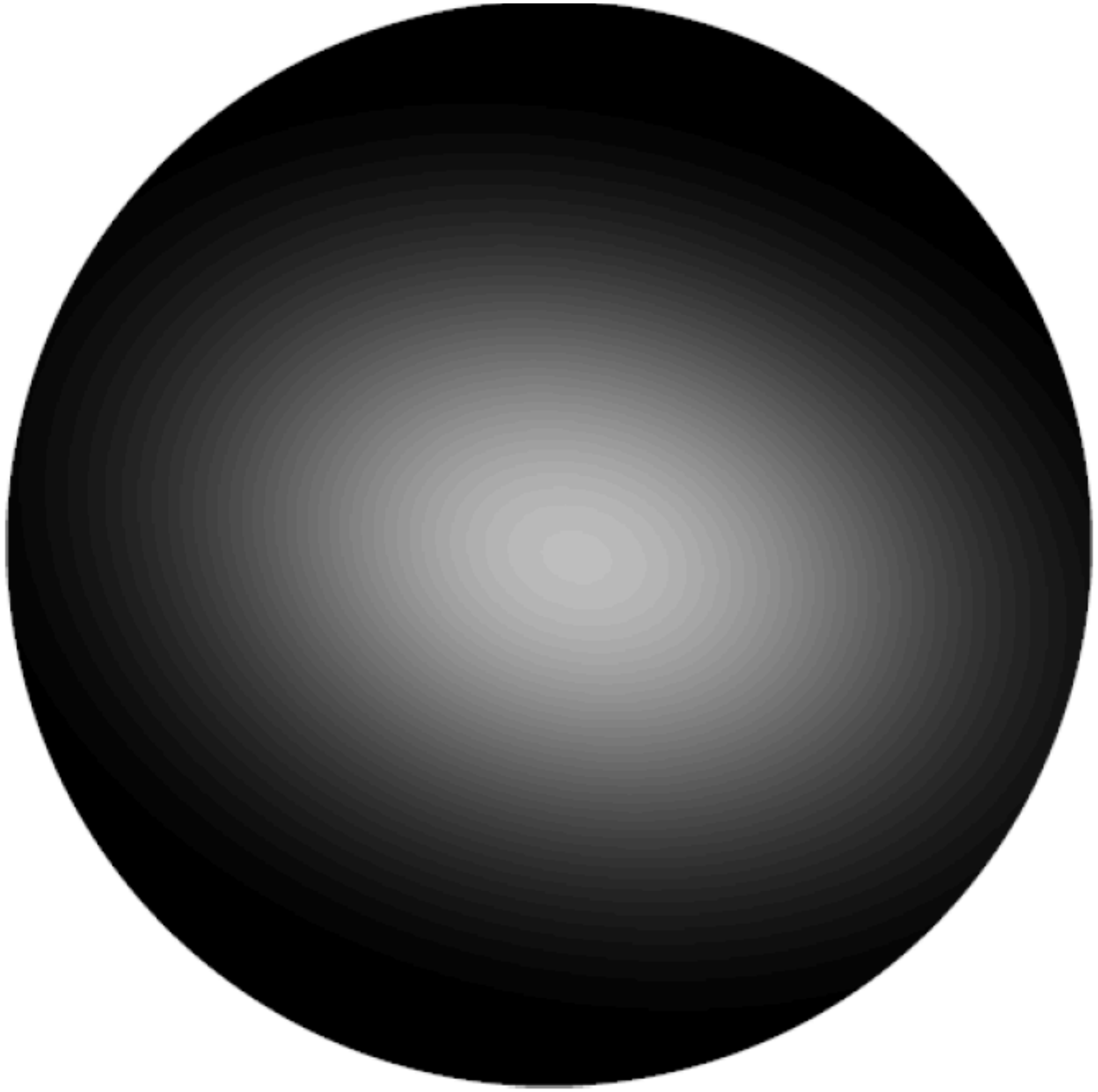


Image from “Rendering Glints on High-Resolution Normal-Mapped Specular Surfaces”, Yan et al., SIGGRAPH 2014

...others with a single anisotropic lobe. However,...

Filtering & Antialiasing

- Several solutions exist:
  - Toksvig AA
  - Variance Mapping
  - LEAN/CLEAN Mapping
  - LEADR Mapping

Image from “Rendering Glints on High-Resolution Normal-Mapped Specular Surfaces”, Yan et al., SIGGRAPH 2014

...for many surfaces, the actual pixel NDF may look something like this. Clearly a lot of detail is being lost by these filtering techniques, but they are still much better than doing nothing. Two papers are presented this SIGGRAPH with more complete solutions; not fast enough for games but well worth looking at for film applications.

Acknowledgements

- Steve Hill: assistance with course notes & slides, WebGL framework used for Fresnel visualization
- Brent Burley, Paul Edelstein, Yoshiharu Gotanda, Eric Heitz, Christophe Hery, Sébastien Lagarde, Dimitar Lazarov, Cedric Perthuis, Brian Smits: inspirational discussions on physically based shading models
- A K Peters: permission to use RTR3 images

SIGGRAPH2014

I'd like to thank some people who helped me with this talk...

...and finally, 2K is hiring: there are positions across many of our studios. In addition, the 2K Core Tech group is looking for top-notch rendering & engine programmers, as well as research collaborators for various real-time rendering topics.