Some Thoughts on the Fresnel Term

Naty Hoffman
Lucasfilm

Hi. I’m Naty Hoffman, Principal Engineer and Architect at Lucasfilm’s Advanced Development Group. In this talk, I’ll try to shed some new light on a familiar topic: the Fresnel term.

I'll start with an overview of its history.

(Image credit: NASA, cropped from Hubble 30 year anniversary image, public domain.)

1823: Fresnel's Equations

$$F(\theta) = \frac{F_s(\theta) + F_p(\theta)}{2}$$

$$F_s(\theta) = \frac{a^2 + b^2 - 2a\cos\theta + \cos^2\theta}{a^2 + b^2 + 2a\cos\theta + \cos^2\theta}$$

$$F_p(\theta) = F_s(\theta)\frac{a^2 + b^2 - 2a\sin\theta\tan\theta + \sin^2\theta\tan^2\theta}{a^2 + b^2 + 2a\sin\theta\tan\theta + \sin^2\theta\tan^2\theta}$$

$$2a^2 = \sqrt{(\eta^2 - \kappa^2 - \sin^2\theta)^2 + 4\eta^2\kappa^2} + (\eta^2 - \kappa^2 - \sin^2\theta)$$

$$2b^2 = \sqrt{(\eta^2 - \kappa^2 - \sin^2\theta)^2 + 4\eta^2\kappa^2} - (\eta^2 - \kappa^2 - \sin^2\theta)$$

Fresnel's equations were initially published in 1823*. They hold for light reflecting from a perfectly planar mirror surface.

* A.-J. Fresnel: Mémoire sur la loi des modifications que la réflexion imprime à la lumière polarisée (memoir on the law of the modifications that reflection impresses on polarized light.) Mémoires, French Academy of Sciences, 1823.

1823: Fresnel's Equations

$$F(\theta) = \frac{F_s(\theta) + F_p(\theta)}{2}$$

$$F_s(\theta) = \frac{a^2 + b^2 - 2a\cos\theta + \cos^2\theta}{a^2 + b^2 + 2a\cos\theta + \cos^2\theta}$$

$$F_p(\theta) = F_s(\theta)\frac{a^2 + b^2 - 2a\sin\theta\tan\theta + \sin^2\theta\tan^2\theta}{a^2 + b^2 + 2a\sin\theta\tan\theta + \sin^2\theta\tan^2\theta}$$

$$2a^2 = \sqrt{(\eta^2 - \kappa^2 - \sin^2\theta)^2 + 4\eta^2\kappa^2} + (\eta^2 - \kappa^2 - \sin^2\theta)$$

$$2b^2 = \sqrt{(\eta^2 - \kappa^2 - \sin^2\theta)^2 + 4\eta^2\kappa^2} - (\eta^2 - \kappa^2 - \sin^2\theta)$$

The equations depend on the angle of incidence, theta…

1823: Fresnel’s Equations

$$F(\theta) = \frac{F_s(\theta) + F_p(\theta)}{2}$$

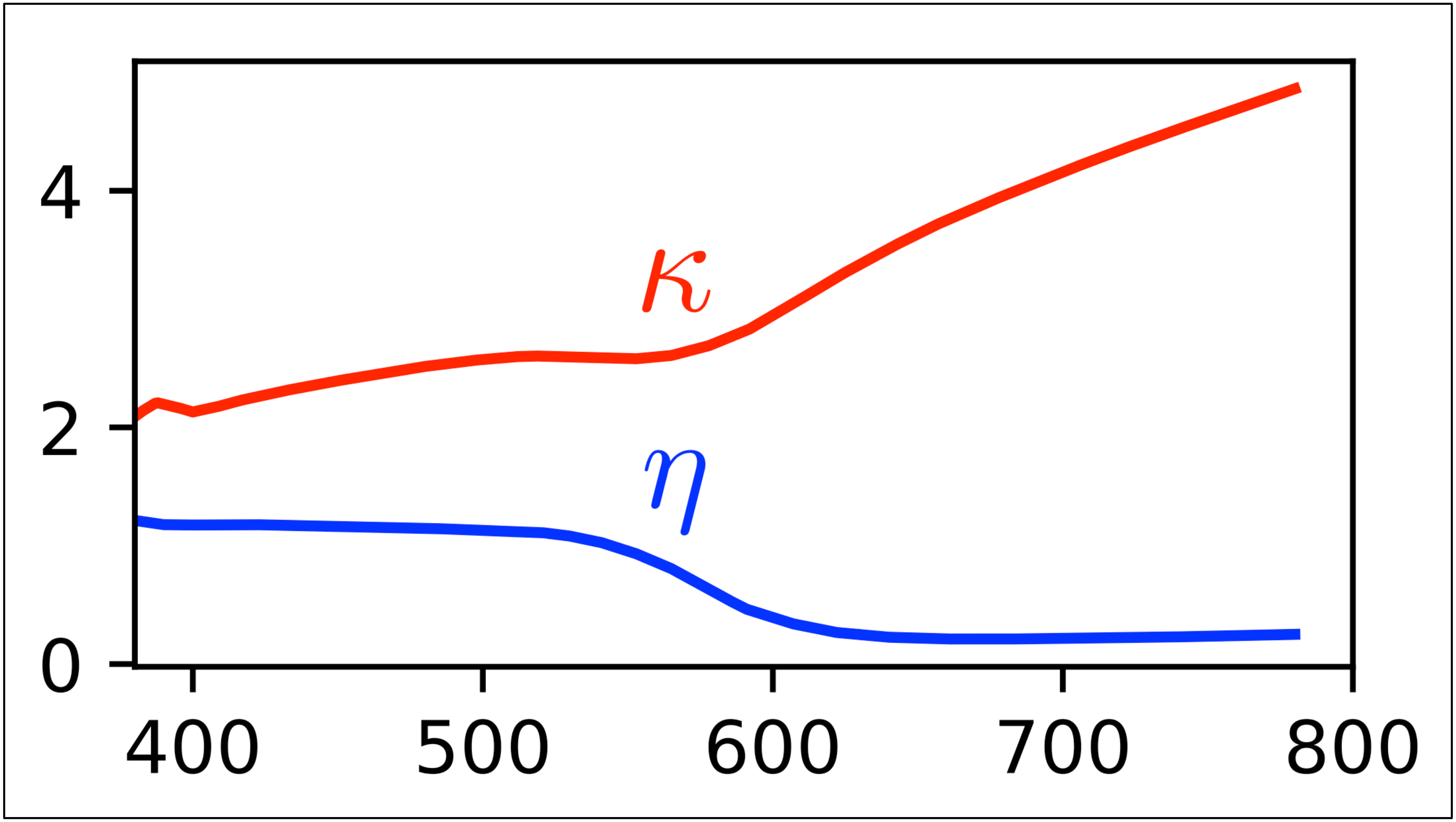
$$F_s(\theta) = \frac{a^2 + b^2 - 2a\cos\theta + \cos^2\theta}{a^2 + b^2 + 2a\cos\theta + \cos^2\theta}$$

$$F_p(\theta) = F_s(\theta)\frac{a^2 + b^2 - 2a\sin\theta\tan\theta + \sin^2\theta\tan^2\theta}{a^2 + b^2 + 2a\sin\theta\tan\theta + \sin^2\theta\tan^2\theta}$$

$$2a^2 = \sqrt{(\eta^2 - \kappa^2 - \sin^2\theta)^2 + 4\eta^2\kappa^2} + (\eta^2 - \kappa^2 - \sin^2\theta)$$

$$2b^2 = \sqrt{(\eta^2 - \kappa^2 - \sin^2\theta)^2 + 4\eta^2\kappa^2} - (\eta^2 - \kappa^2 - \sin^2\theta)$$

…and are parameterized by eta and kappa, which together comprise the complex index of refraction, or IOR.



Eta and kappa values may vary for different wavelengths over the visible spectrum, especially for colored metals such as copper, shown here.

Fresnel's Equations (dielectrics)

$$F(\theta) = \frac{F_s(\theta) + F_p(\theta)}{2}$$

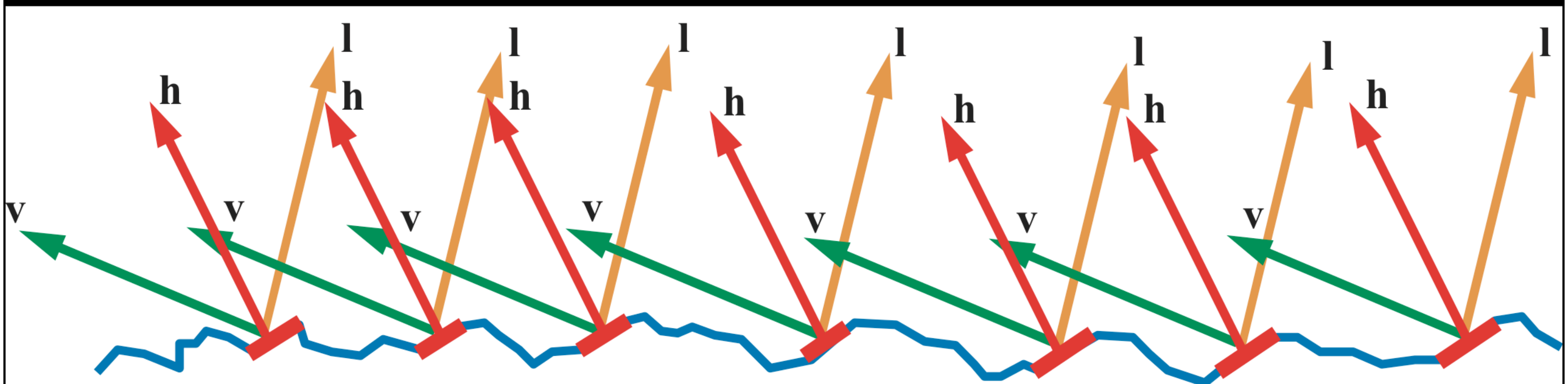
$$F_s(\theta) = \left(\frac{a - \cos\theta}{a + \cos\theta}\right)^2$$

$$F_p(\theta) = F_s(\theta)\left(\frac{a - \sin\theta\tan\theta}{a + \sin\theta\tan\theta}\right)^2$$

$$a^2 = \eta^2 - \sin^2\theta$$

For dielectrics — or non-metals — kappa is zero, which simplifies the equations considerably.

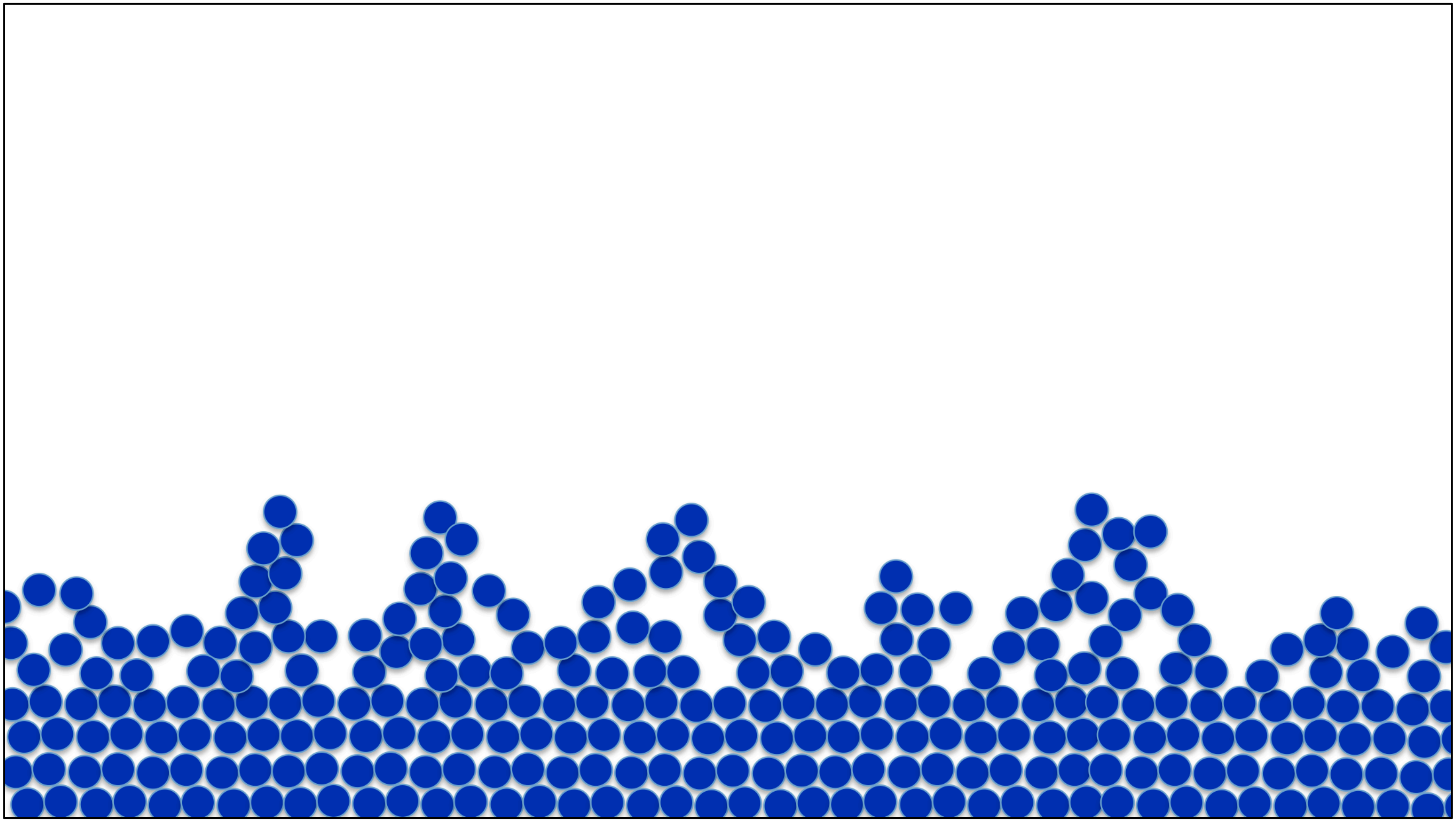
1967: Fresnel with Microfacets



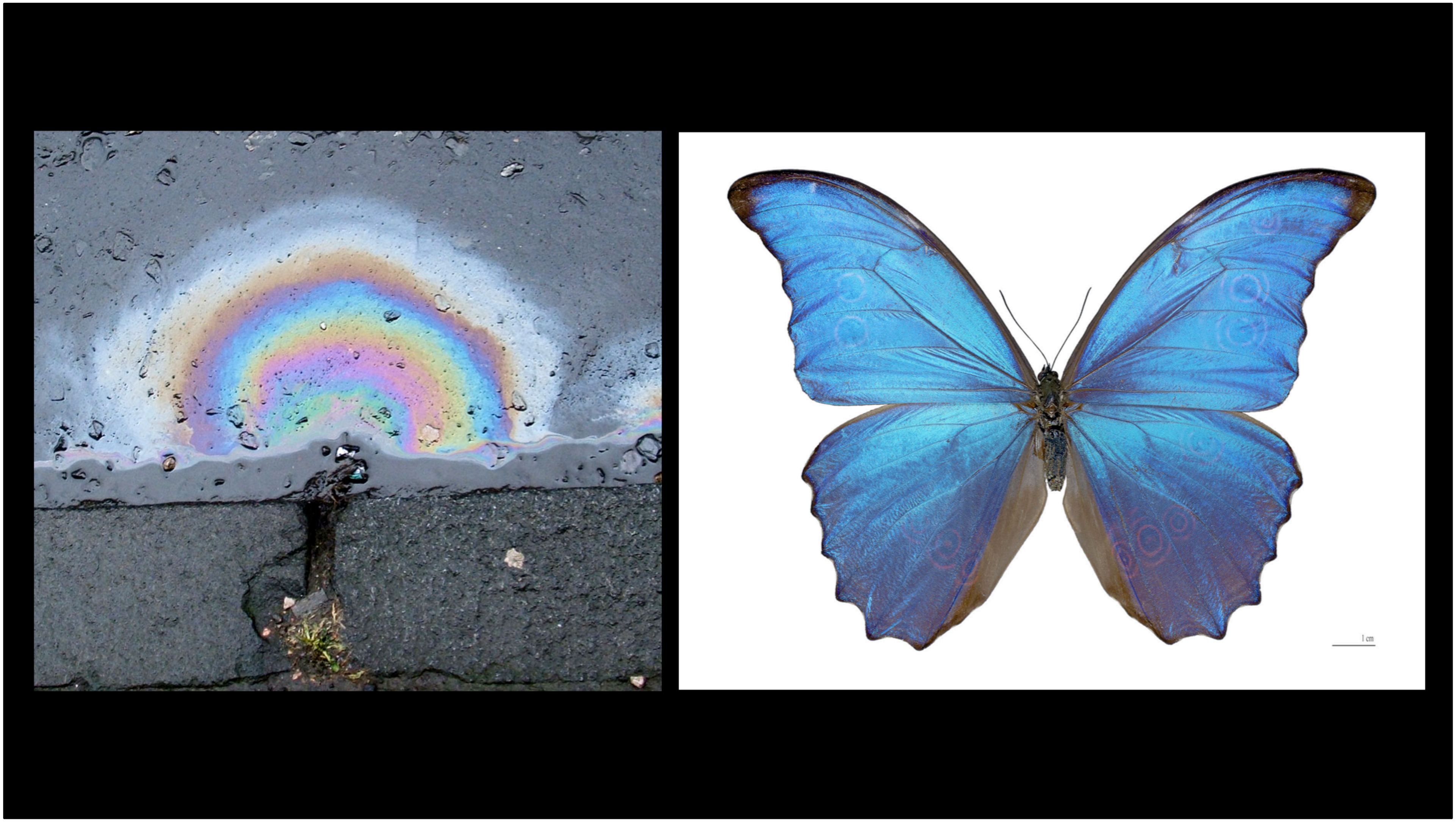
Microfacet theory, introduced by Torrance and Sparrow in 1967*, expanded the use of the Fresnel equations to more general surfaces by treating them as statistical assemblies of perfect mirrors.

* K. E. Torrance and E. M. Sparrow, Theory for Off-Specular Reflection From Roughened Surfaces. J. Opt. Soc. Am., 1967.

(Image credit: Real-Time Rendering, 4th Edition, CRC Press.)



However, there are surfaces that cannot be modeled as a simple interface between an exterior and a uniform interior. The Fresnel equations do not apply in such cases, for example loosely packed fine dust with individual grains smaller than a visible light wavelength. In this case the transition from air to dense dust is a gradual one, passing through a continuous variation of refractive index values.



Other examples include surfaces exhibiting thin-film interference (where the light interacts with multiple layers) and diffractive surfaces (which have repeating structure at a scale smaller than a light wavelength).

(image credits:
"Diesel Spill on a Road" by John, licensed CC BY-SA 2.5 via Wikimedia Commons.
Morpho butterfly image by Didier Descouens, licensed CC BY-SA 4.0 via Wikimedia Commons.)

1977: Fresnel in Computer Graphics

$$\frac{D(\theta_h)G(\theta_i, \theta_o, \theta_h)F(\theta_h)}{\cos\theta_o}$$

Blinn's 1977 paper* was the first notable use of Fresnel in computer graphics. He proposed a microfacet BRDF, in which the Fresnel term uses theta_h, the incidence angle of the individual microfacets. Other parts of the BRDF use theta_i and theta_o, which are the incident and outgoing angles with respect to the macroscopic surface normal. The paper only mentioned Fresnel briefly and somewhat inaccurately (applying the dielectric form to both dielectrics and metals).

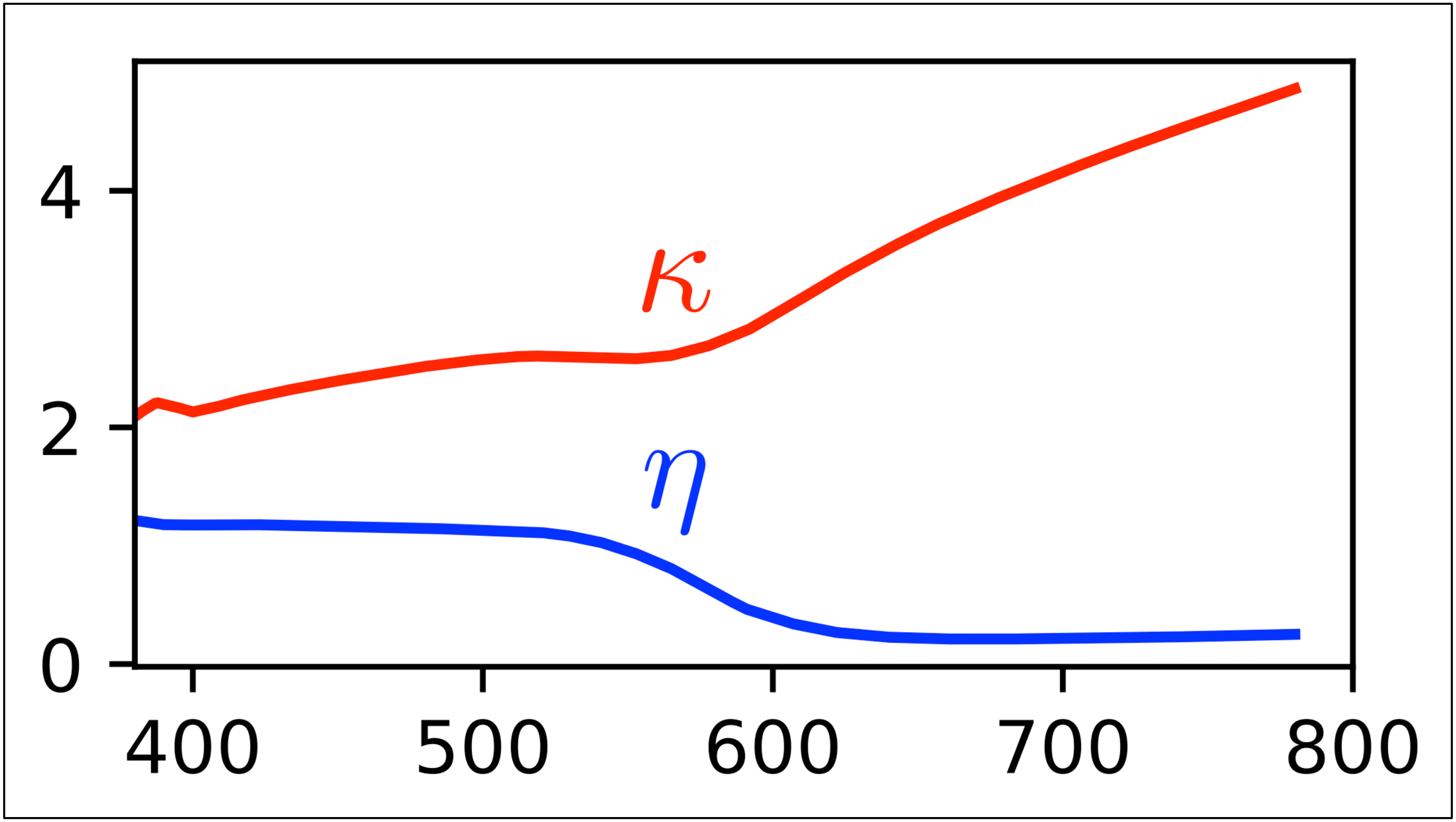
- J. F. Blinn, Models of Light Reflection for Computer Synthesized Pictures. Computer Graphics, 1977.

1981: Cook-Torrance

$$\frac{F(\theta_h)D(\theta_h)G(\theta_i, \theta_o, \theta_h)}{4\cos\theta_i\cos\theta_o}$$

Like Blinn's paper, the 1981 paper by Cook and Torrance* was based on the Torrance-Sparrow microfacet model.

* R. L. Cook and K. E. Torrance, A Reflectance Model for Computer Graphics. Computer Graphics, 1981.



The Cook-Torrance paper also contained an in-depth discussion of spectral Fresnel computations and their relation to final RGB pixel values. This discussion had three main parts…

1. Full procedure: spectral computation of reflected light spectral power distribution (SPD), then conversion to RGB pixel color.

A full procedure, involving a spectral computation of the reflected light power distribution, or SPD, which is then converted to RGB pixel colors.

1. Full procedure: spectral computation of reflected light spectral power distribution (SPD), then conversion to RGB pixel color.

2. An approximation of the above, based on color interpolation.

An approximation of the above procedure, based on color interpolation.

1. Full procedure: spectral computation of reflected light spectral power distribution (SPD), then conversion to RGB pixel color.

2. An approximation of the above, based on color interpolation.

3. A procedure for approximating spectral η and κ from normal-incidence reflectance.

A method to approximate spectral eta and kappa from the reflectance at zero degrees (normal incidence).

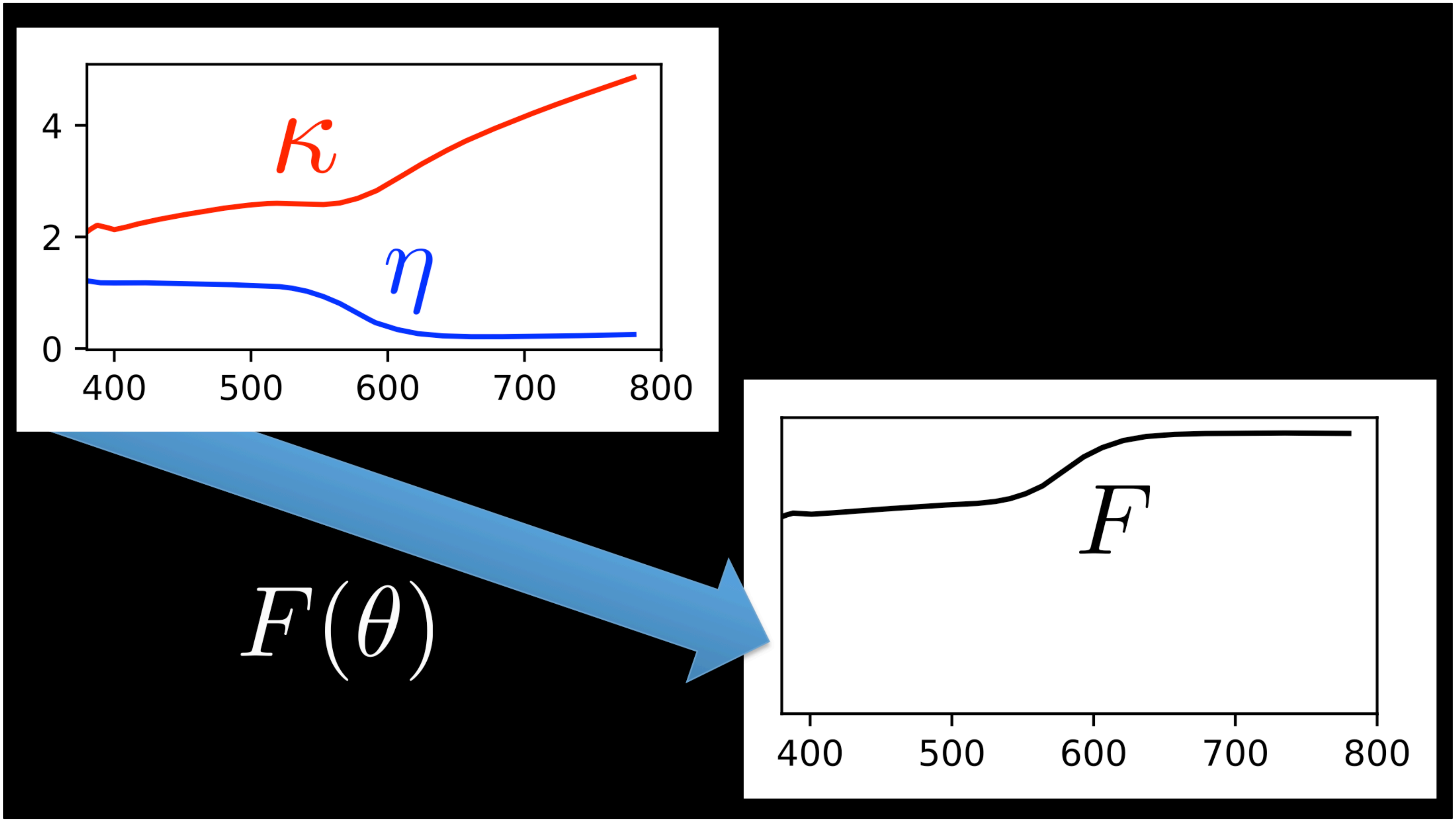
1. Full procedure: spectral computation of reflected light spectral power distribution (SPD), then conversion to RGB pixel color. ✓
2. An approximation of the above, based on color interpolation. ✓
3. A procedure for approximating spectral η and κ from normal-incidence reflectance. ✗

It turns out that the third procedure yields gross errors and should not be used. Further details on this are available in the course notes. Fortunately, that part is not key to the paper…

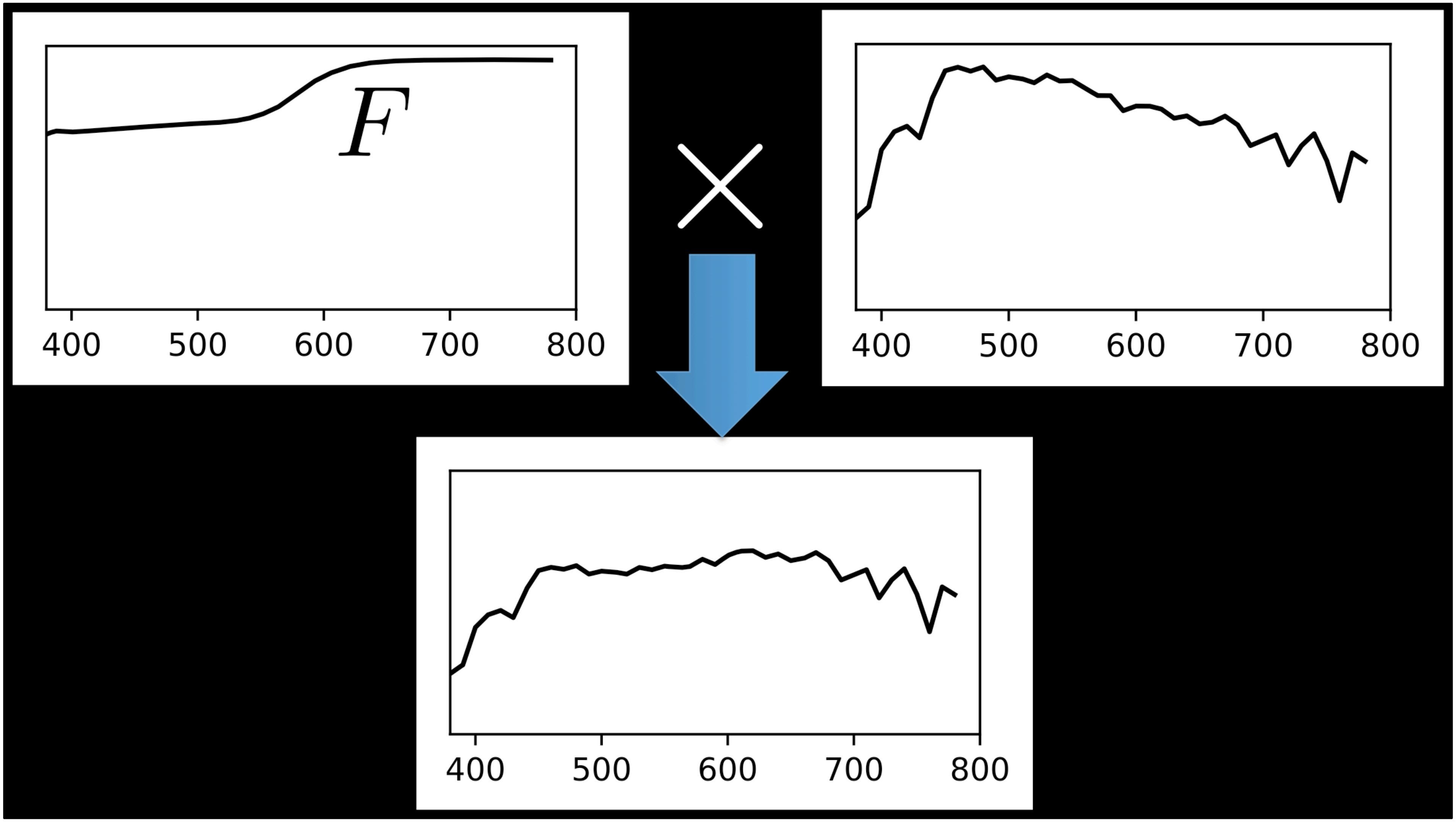
1. Full procedure: spectral computation of reflected light spectral power distribution (SPD), then conversion to RGB pixel color. ✓

2. An approximation of the above, based on color interpolation. ✓

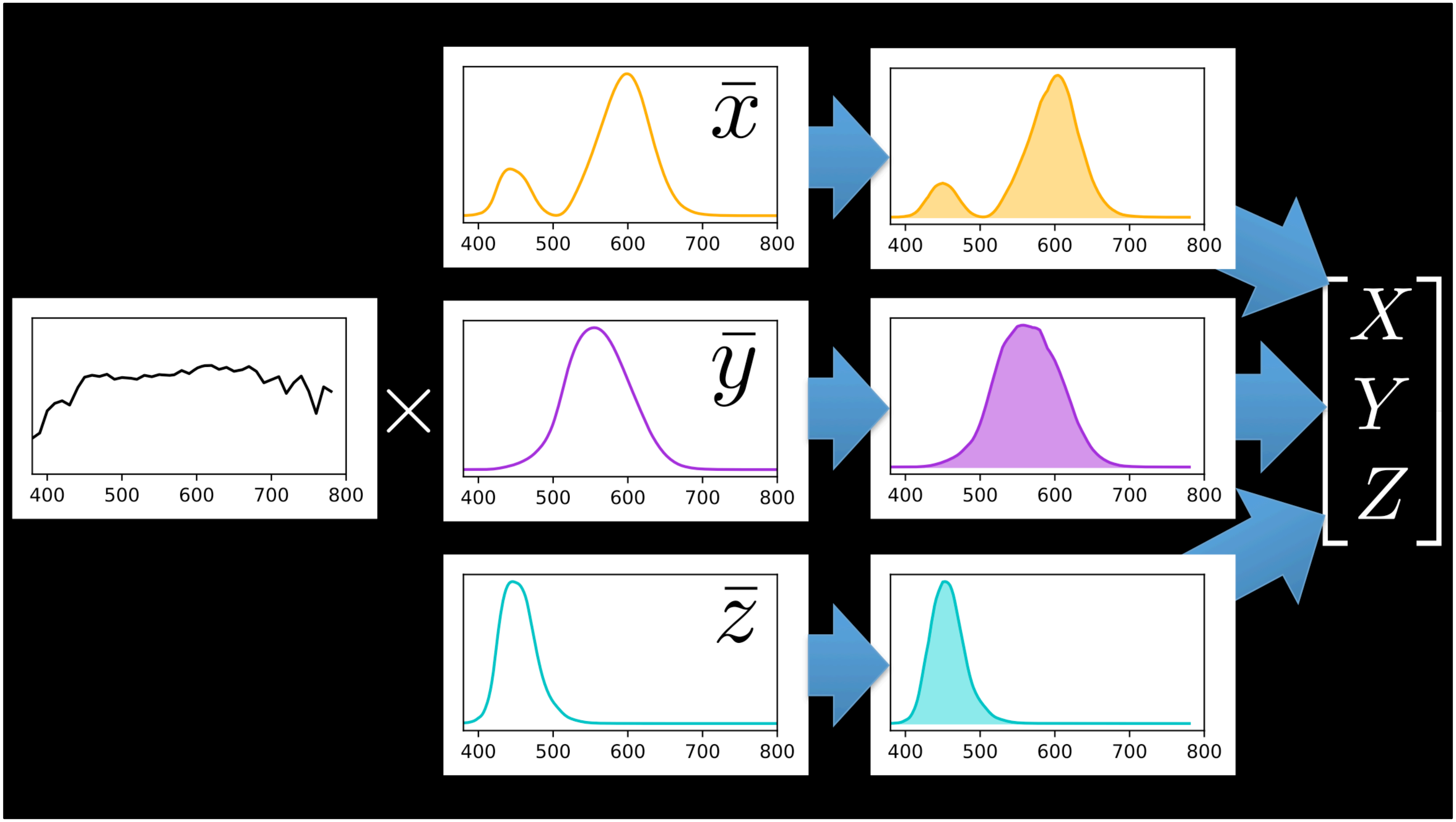
…so we will focus on the other two parts.



In the full procedure, the Fresnel equations are evaluated for a given incidence angle, theta, and for a dense spectral sampling of IOR data (this example shows measured data for copper). This results in a spectral reflectance curve.



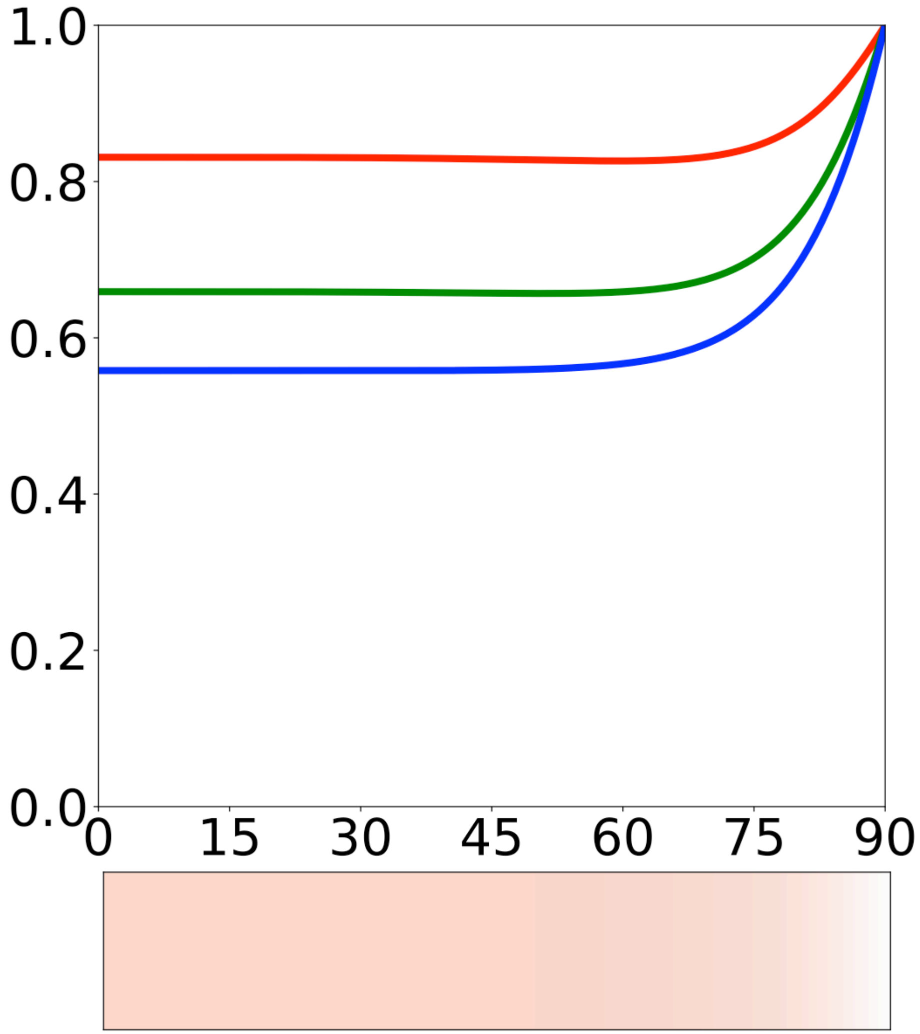
The spectral reflectance for the current angle is then multiplied by the SPD of the incoming light to yield the SPD of the reflected light.



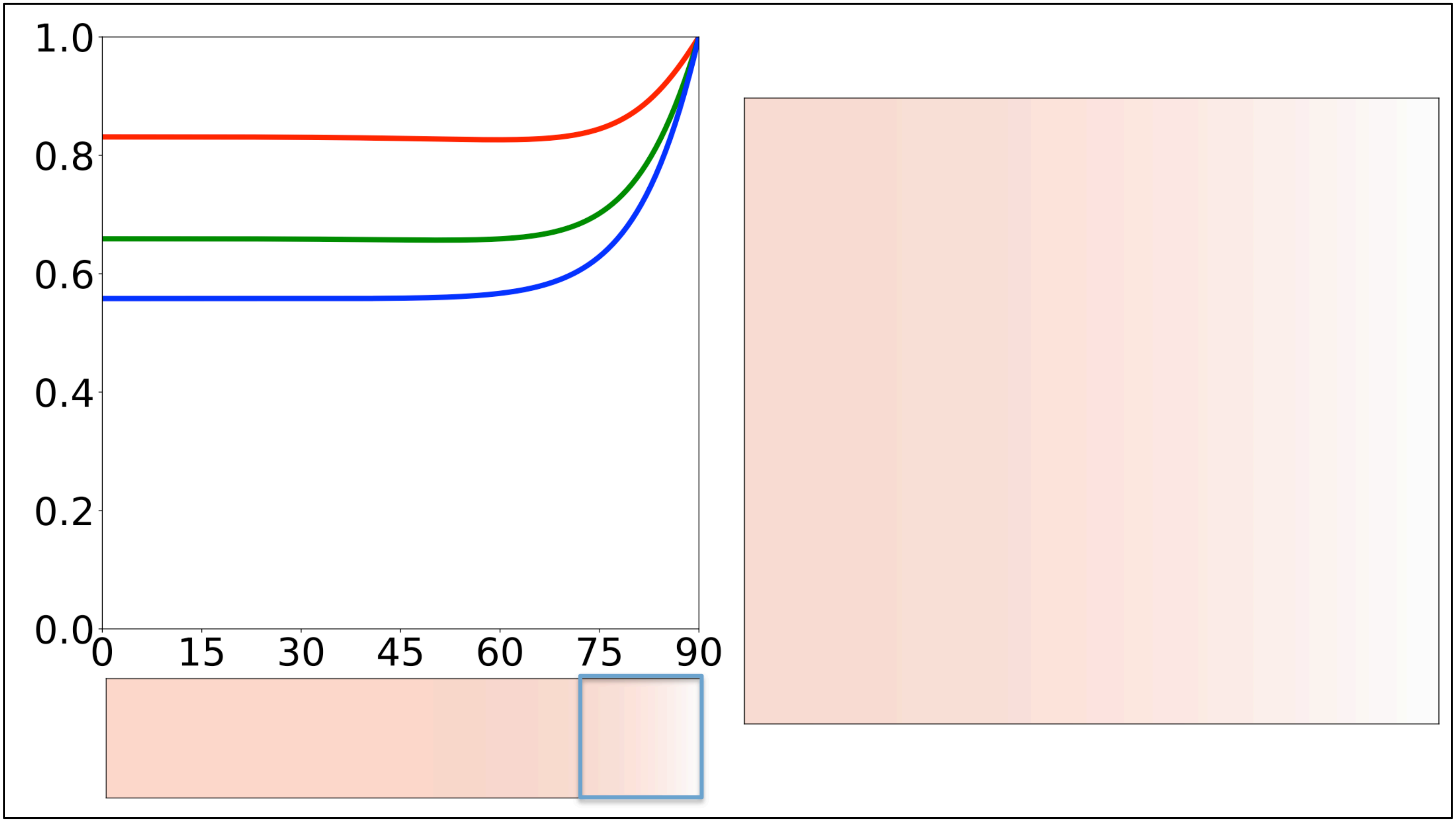
The reflected SPD is converted to perceptual RGB values in the standard way: multiplication by color-matching functions, integration of the results…

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \times \begin{bmatrix} M \end{bmatrix} = [\mathrm{R\ G\ B}]$$

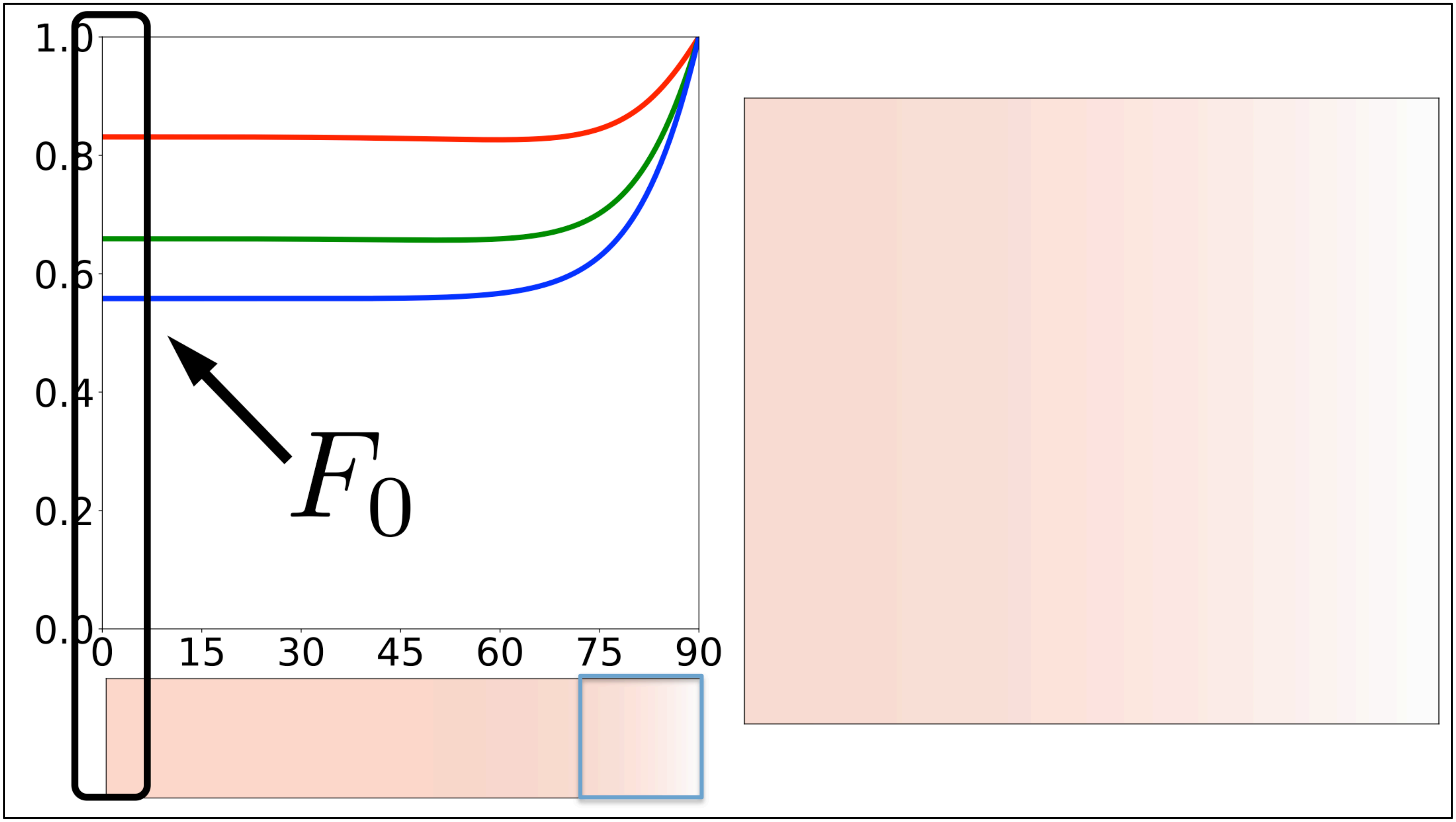
…and matrix multiplication, resulting in the final RGB display value.



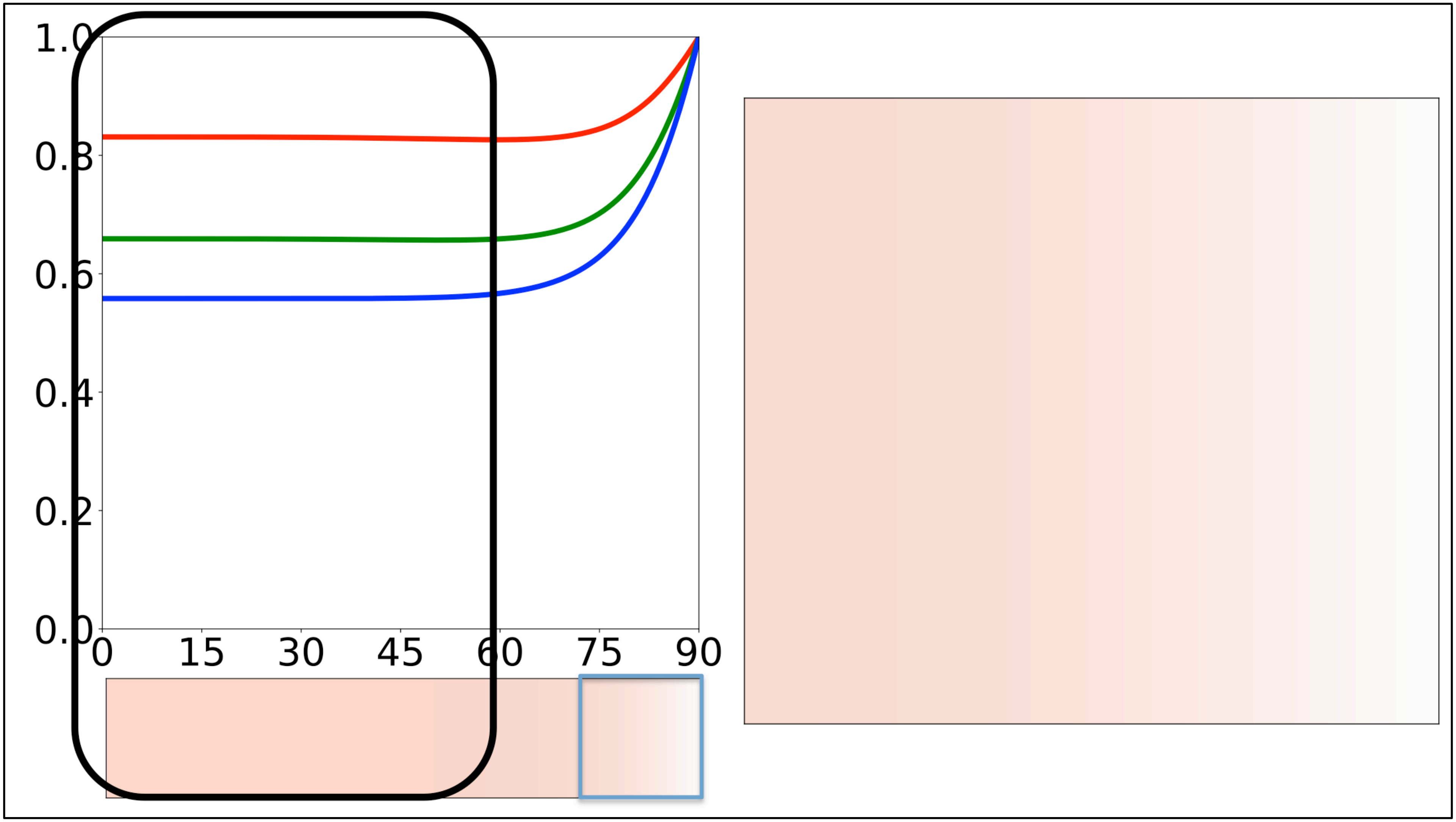
For now, we will visualize the results for the 0 to 90 degree range of incidence angles in two ways: as plots for the values of the RGB channels, and as a color strip.



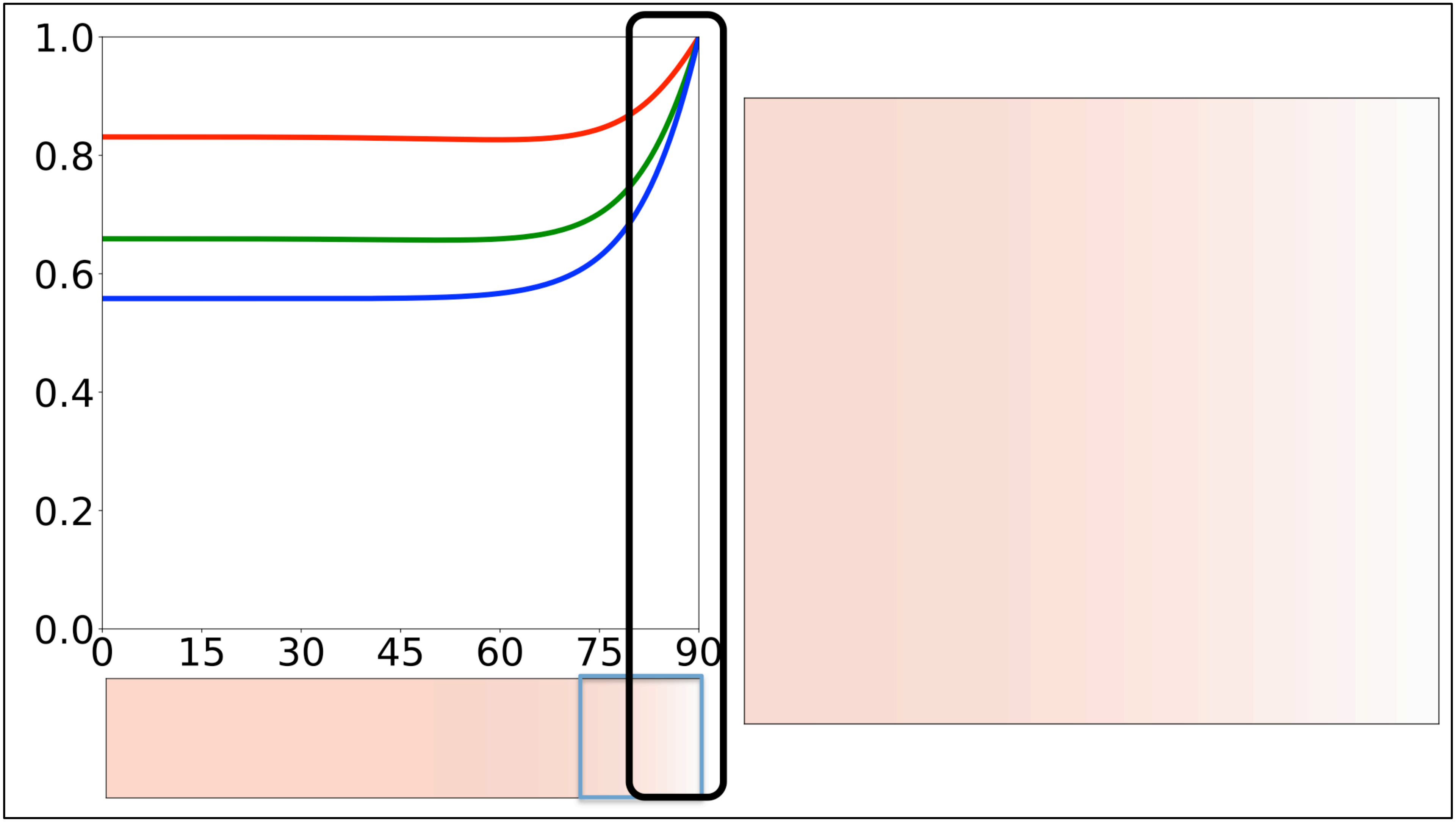
Since most of the variation appears at glancing angles, we'll also zoom in on that part of the angle range. As expected…



…the characteristic copper color of the material appears at an angle of 0 degrees. The value at this angle, F0, will appear a lot during this talk.



…the same color extends almost unchanged over most of the range. For this reason, F0 is a key parameter for describing the surface appearance. Finally,…



…there is a rapid transition to white close to 90 degrees.

1. Full procedure: spectral computation of reflected light spectral power distribution (SPD), then conversion to RGB pixel color.

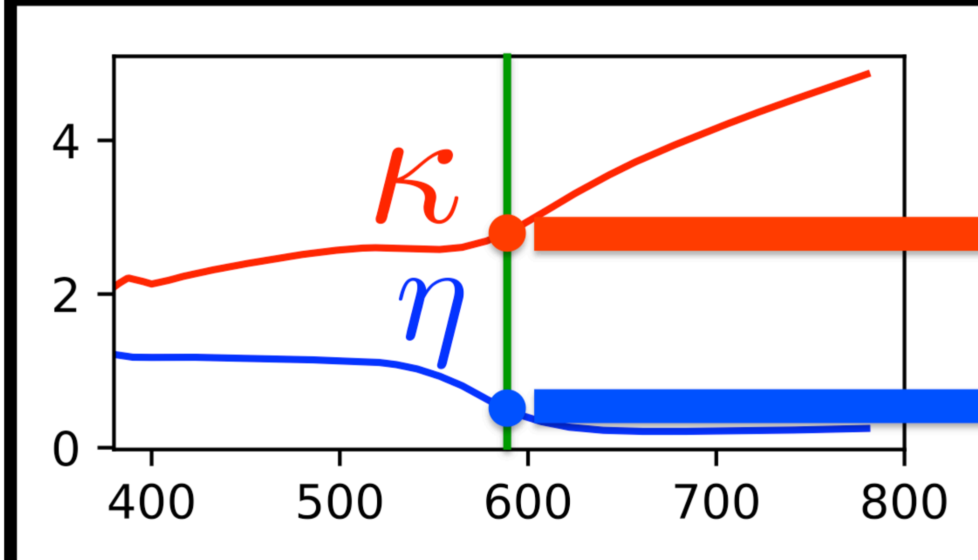
2. An approximation of the above, based on color interpolation.

That was the full procedure. We will now describe the approximation.

$$(\eta', \kappa') \Rightarrow F'(\theta)$$

$$F(\theta) \approx F_0 + (1 - F_0)\frac{\max(0, F'(\theta) - F'(0^\circ))}{1 - F'(0^\circ)}$$

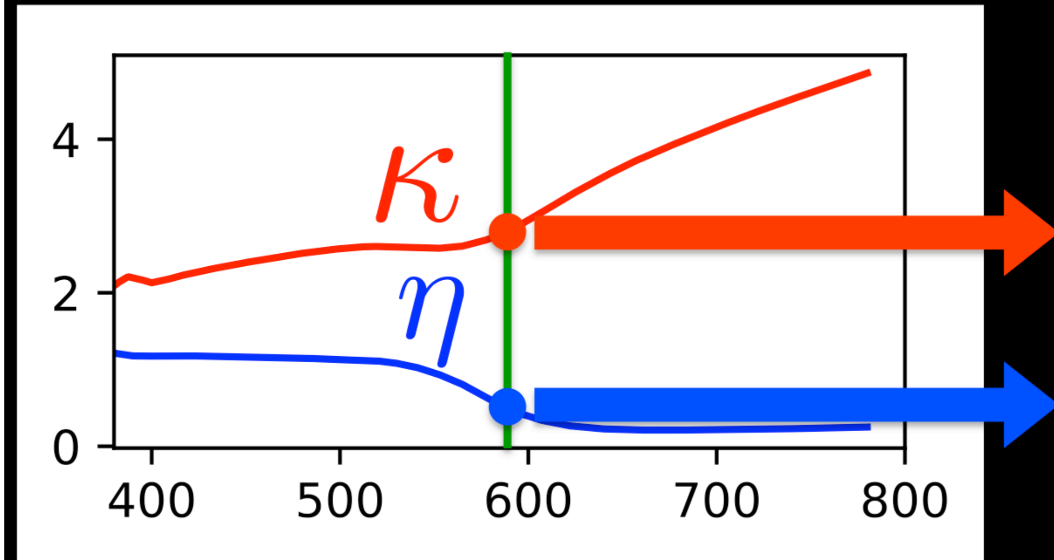
The Fresnel equations are used with a single representative IOR value, resulting in a scalar function we will call F-prime. The value of F-prime is used to interpolate between F0 and white.



$$(\eta', \kappa') \Rightarrow F'(\theta)$$

$$F(\theta) \approx F_0 + (1 - F_0)\frac{\max(0, F'(\theta) - F'(0^\circ))}{1 - F'(0^\circ)}$$

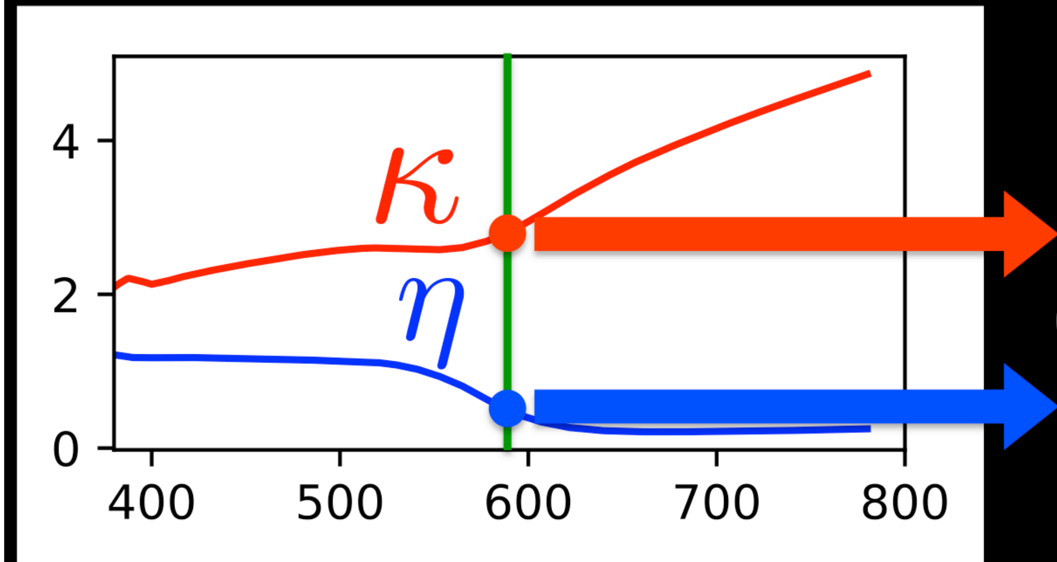
We'll present a version of the approximation that is slightly modified from Cook and Torrance's original version. It produces much better results than the original while preserving its spirit (more details in the course notes). We use a complex IOR value sampled at a single representative wavelength.



$$(\eta', \kappa') \rightarrow F'(\theta)$$

$$F(\theta) \approx F_0 + (1 - F_0)\frac{\max(0, F'(\theta) - F'(0°))}{1 - F'(0°)}$$

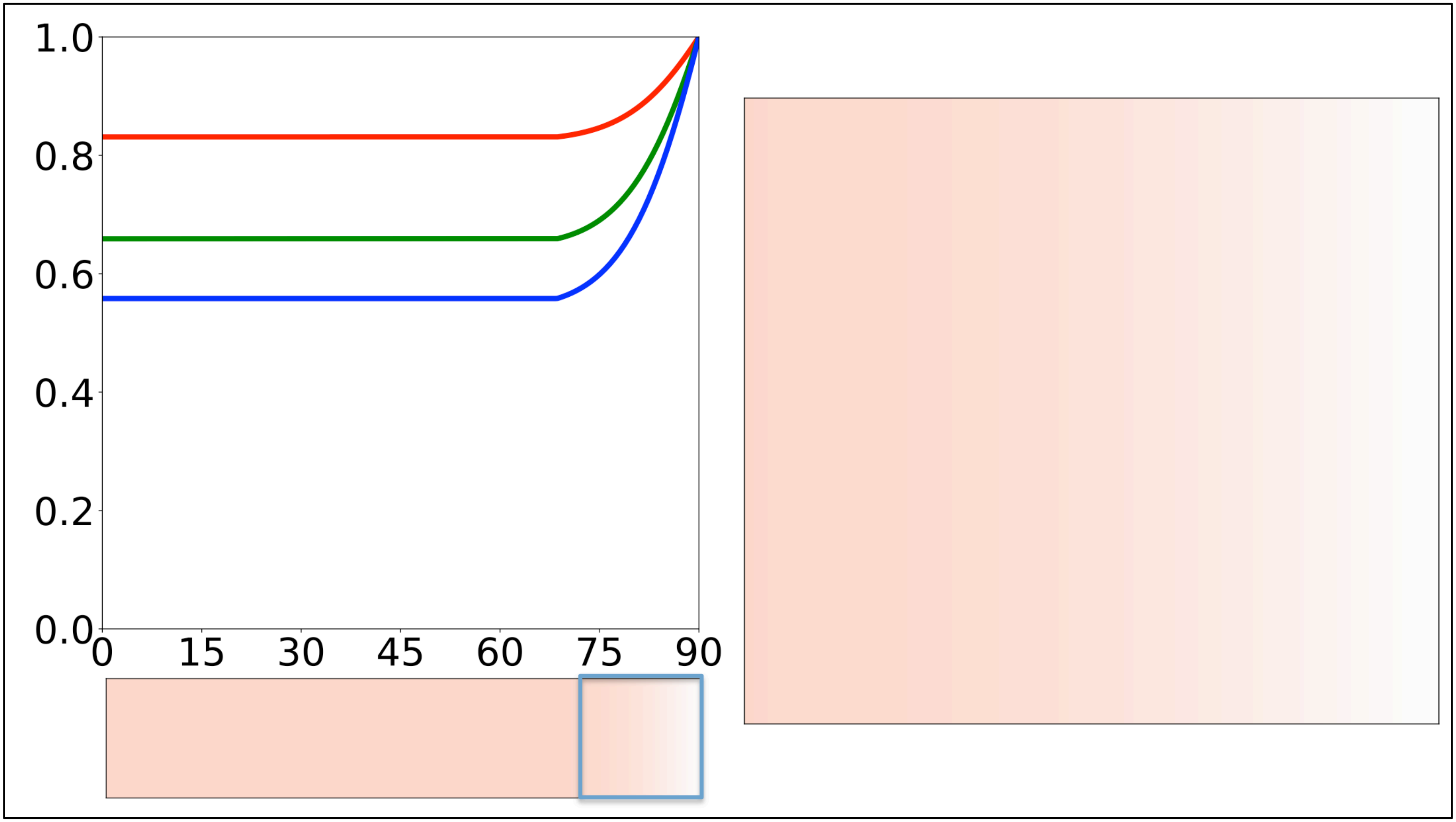
Note that the F-prime values are all scalar…



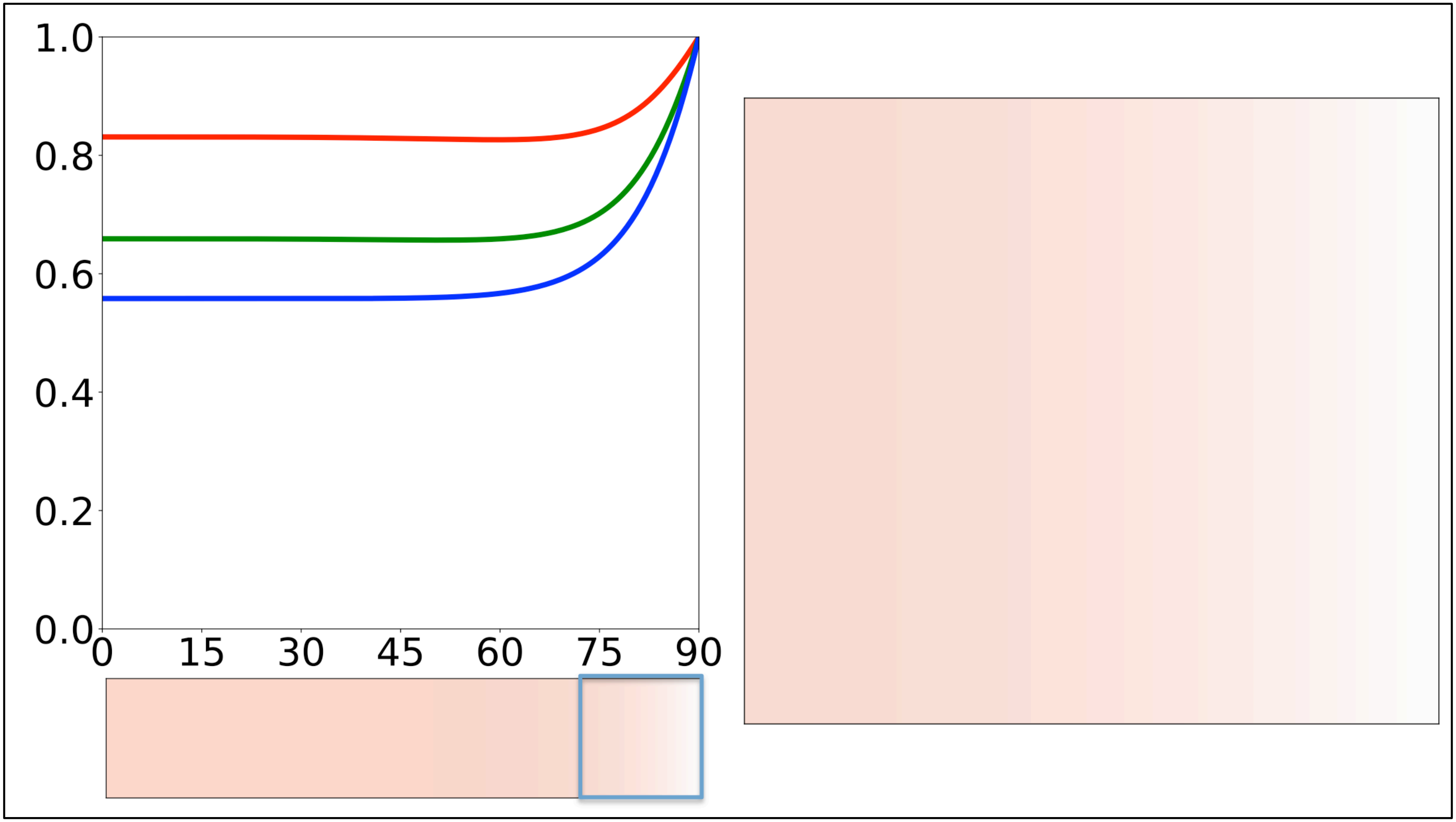
$$(\eta', \kappa') \Rightarrow F'(\theta)$$

$$F(\theta) \approx F_0 + (1 - F_0)\frac{\max(0, F'(\theta) - F'(0^\circ))}{1 - F'(0^\circ)}$$

…and the “F(theta)” and “F0” values are wavelength-dependent, most often RGB.



This is a reasonably good approximation, as we will see when comparing its results…



…to the ground-truth results. When **<u>flipping between them</u>***, we see that the approximate curves have a slight "corner" to them, but are otherwise very close to ground truth. The color strips are indistinguishable.

* The bold underlined text is used as a reminder to the presenter that they need to flip between slides now.

1. Full procedure: spectral computation of reflected light spectral power distribution (SPD), then conversion to RGB pixel color.

---

2. An approximation of the above, based on color interpolation.

So how should we decide which procedure to use?

> “The … approximation must always be used if the spectral energy distribution of the reflected light is not known, in which case all of the RGB values are estimates.”

Cook and Torrance saw their approximation as much more than a mere optimization. They stated that it MUST ALWAYS be used if the renderer does not have accurate spectral data. But this condition almost always applies to production rendering, which typically does not use accurate spectral data. This is especially true for RGB renderers, but even spectral production renderers such as Manuka (Weta Digital) use spectra that are upsampled from RGB values*.

* L. Fascione, J. Hanika, M. Leone, M Droske, J. Schwarzhaupt, T. Davidovič, A. Weidlich, and J. Meng, Manuka: A Batch-Shading Architecture

1. Full procedure: spectral computation of reflected light spectral power distribution (SPD), then conversion to RGB pixel color.

2. An approximation of the above, based on color interpolation.

It’s also interesting to note which option Cook and Torrance did NOT include…

1. Full procedure: spectral computation of reflected light spectral power distribution (SPD), then conversion to RGB pixel color.

"1 ½". RGB computation of Fresnel's equations.

2. An approximation of the above, based on color interpolation.

…in particular, running the Fresnel equations on RGB values directly. This was not an oversight: this option was not mentioned for very good reasons.

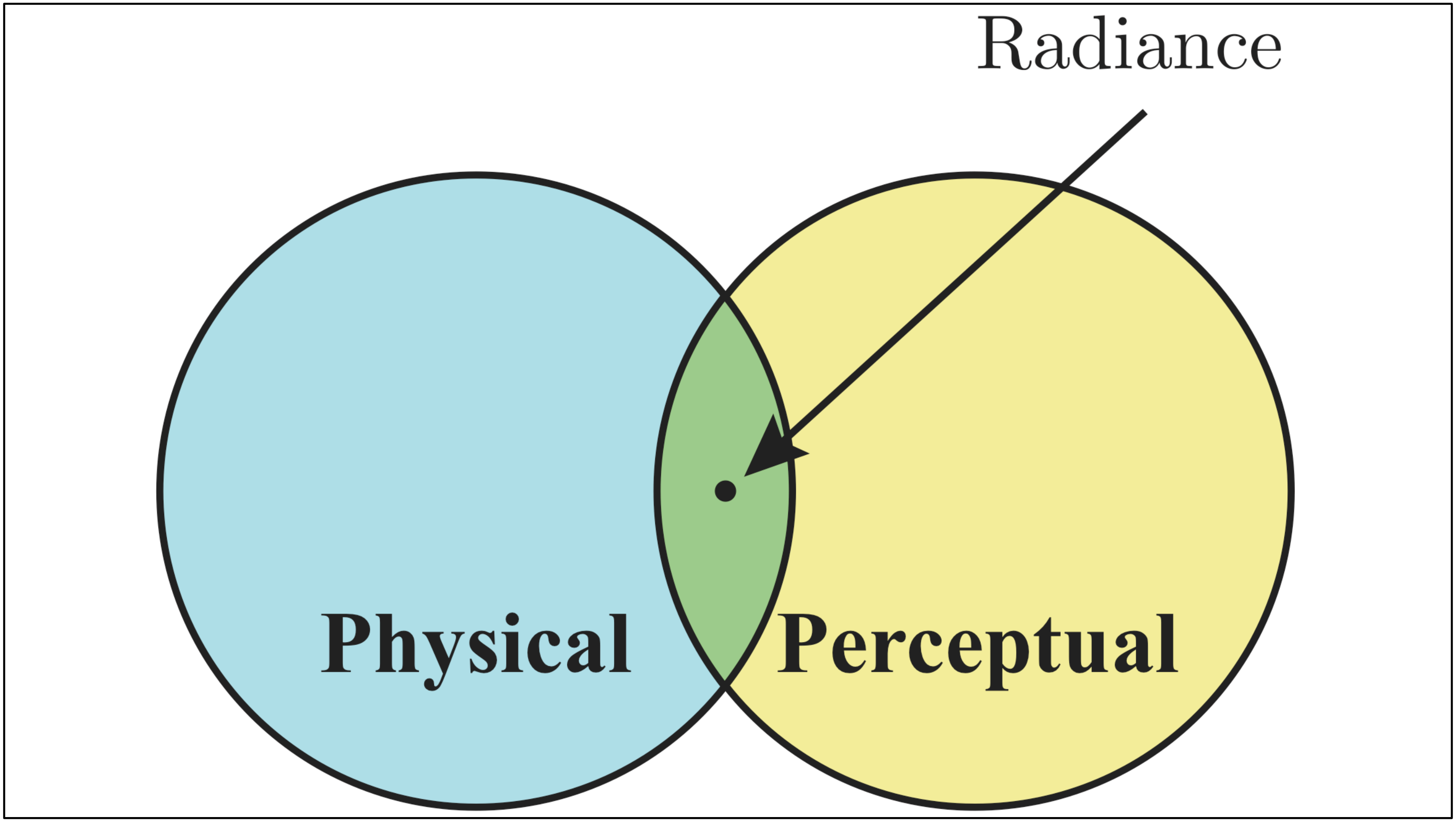
Physics

- Spectral power distributions (radiance, irradiance, etc.)
- Spectral reflectance
- Spectral IOR

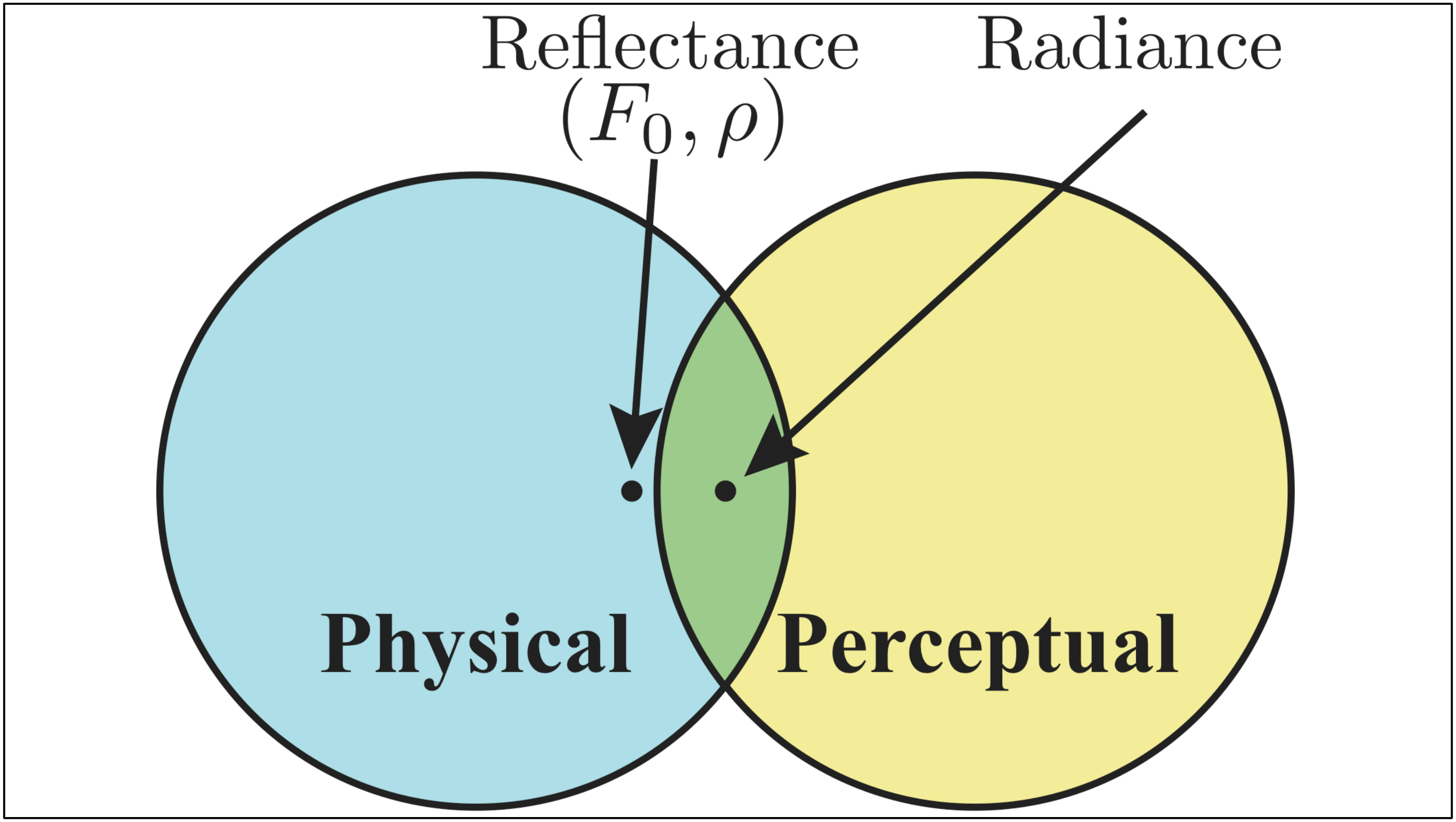
Perception

- Tristimulus (e.g. RGB, XYZ, $L^*a^*b^*$) colors

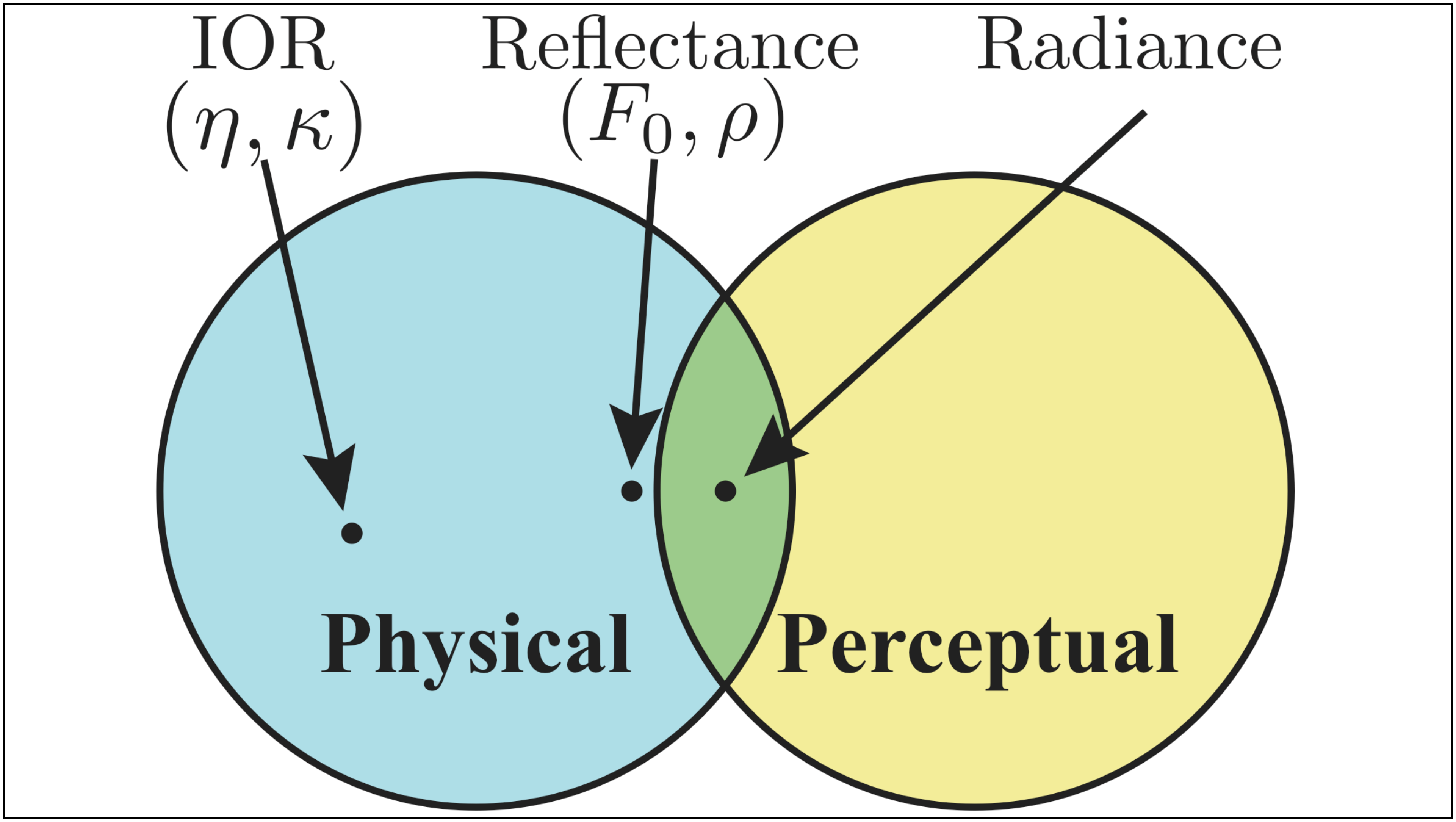
To understand why, let's discuss the nature of RGB rendering. It's a common misconception to think of RGB values as a less accurate version of spectral values. However, they are actually quite different. RGB color spaces are only meaningful for expressing perceptual quantities. But physical rendering math (such as the Fresnel equations) work on physical quantities, which need to be expressed spectrally.



The only physical rendering quantity that is also a perceptual one is radiance, which is directly processed by the human visual system and thus can be meaningfully expressed as an RGB color.



RGB is also used for reflectance colors, which are defined indirectly as reflected radiance from the white reference illuminant. In a sense, reflectance values are "perceptual-adjacent".



In contrast, quantities like eta and kappa have no perceptual analog. Their relationship to final rendered colors (perceptual stimuli) is very indirect and highly nonlinear.

$$F(\theta) \approx F_0 + (1 - F_0)\frac{\max(0, F'(\theta) - F'(0^\circ))}{1 - F'(0^\circ)}$$

Technically, RGB rendering is a category error: performing physical math on perceptual quantities. In practice, it works well — if we are careful. Simple linear operations on RGB colors are fine, but complex nonlinear operations will cause problems. As we saw earlier, the Cook-Torrance Fresnel approximation divides pretty cleanly into a…

$$F(\theta) \approx F_0 + (1 - F_0)\boxed{\frac{\max(0, F'(\theta) - F'(0^\circ))}{1 - F'(0^\circ)}}$$

…nonlinear, scalar section, and a…

$$F(\theta) \approx \boxed{F_0 + (1 - F_0)} \boxed{\frac{\max(0, F'(\theta) - F'(0^\circ))}{1 - F'(0^\circ)}}$$

…linear, RGB section. The nonlinear scalar term is only used to interpolate between two colors.

Fresnel Equations

$$F(\theta) = \frac{F_s(\theta) + F_p(\theta)}{2}$$

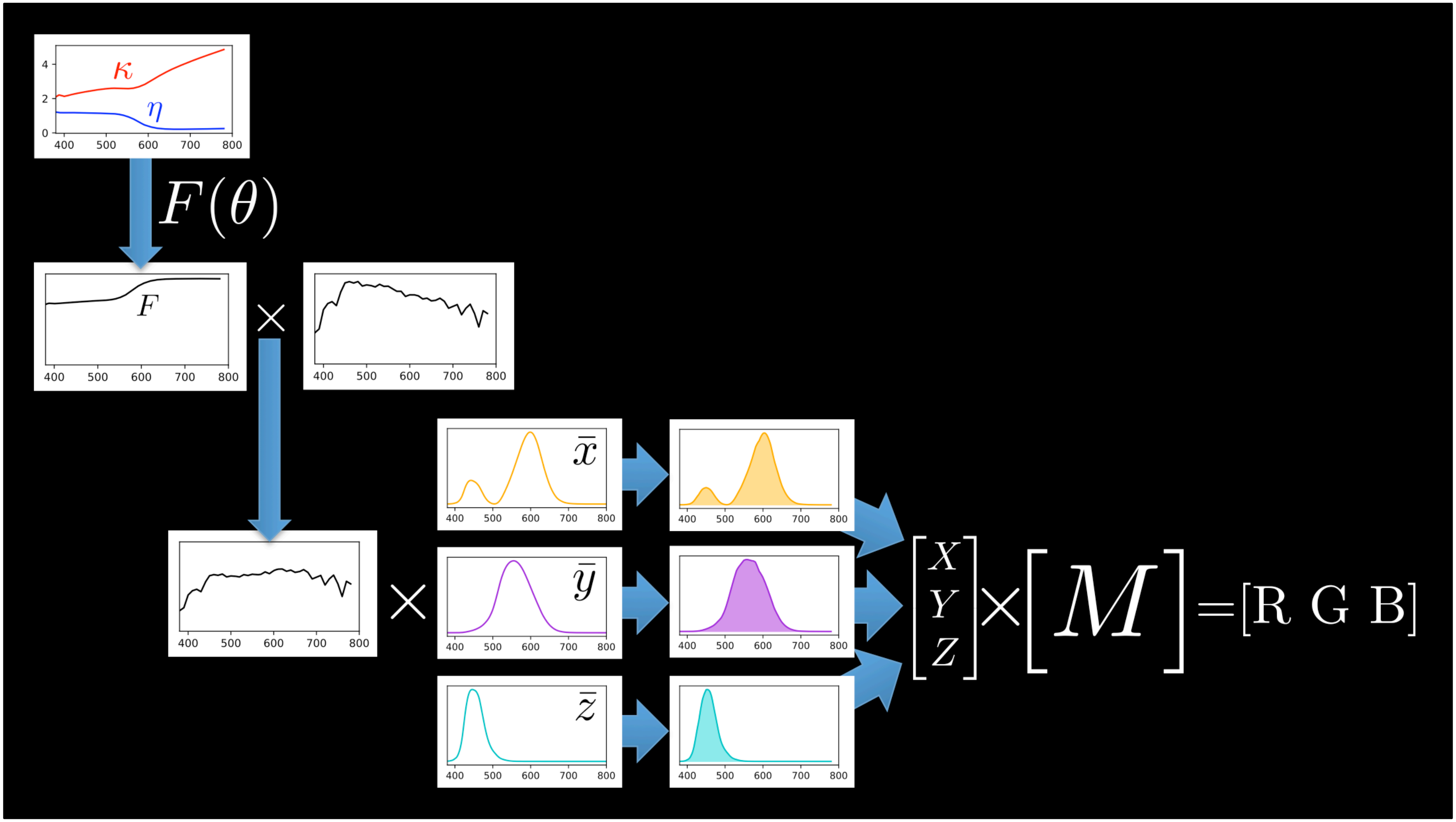
$$F_s(\theta) = \frac{a^2 + b^2 - 2a\cos\theta + \cos^2\theta}{a^2 + b^2 + 2a\cos\theta + \cos^2\theta}$$

$$F_p(\theta) = F_s(\theta)\frac{a^2 + b^2 - 2a\sin\theta\tan\theta + \sin^2\theta\tan^2\theta}{a^2 + b^2 + 2a\sin\theta\tan\theta + \sin^2\theta\tan^2\theta}$$

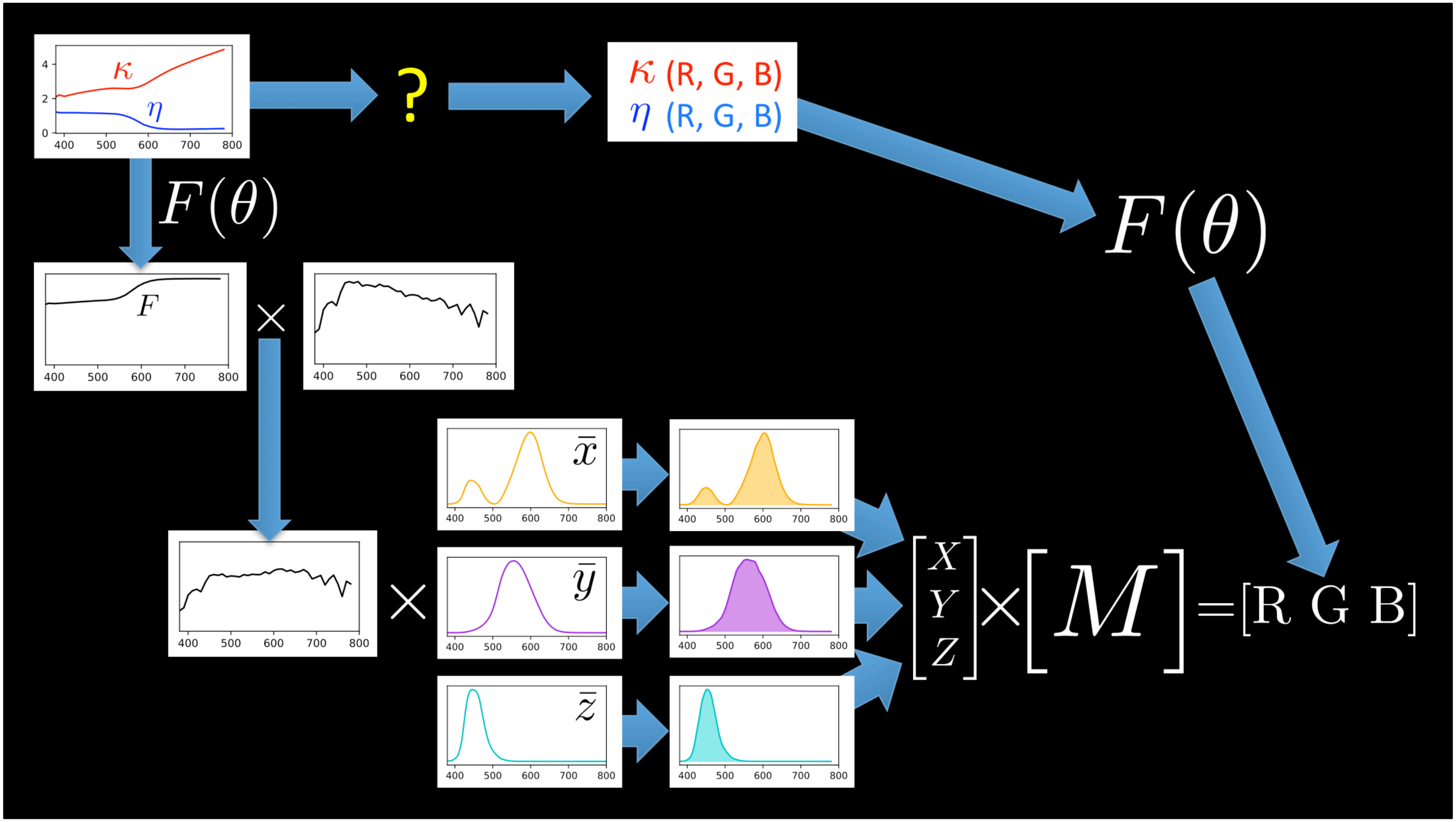
$$2a^2 = \sqrt{(\eta^2 - \kappa^2 - \sin^2\theta)^2 + 4\eta^2\kappa^2} + (\eta^2 - \kappa^2 - \sin^2\theta)$$

$$2b^2 = \sqrt{(\eta^2 - \kappa^2 - \sin^2\theta)^2 + 4\eta^2\kappa^2} - (\eta^2 - \kappa^2 - \sin^2\theta)$$

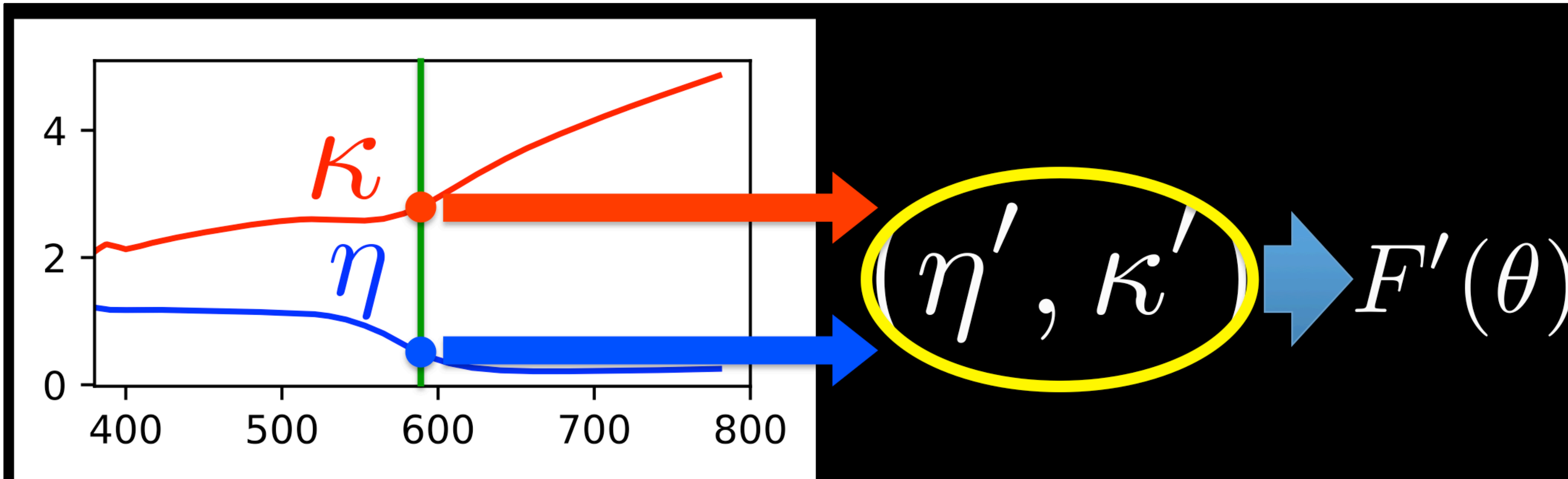
However, RGB math with the full Fresnel equations involves nonlinear operations on perceptual RGB quantities. This can introduce various issues, as we will see later.



To further illustrate the problem, recall the full procedure outlined earlier. To compute an RGB color correctly for a given angle, you have to go through this rigmarole: apply Fresnel to spectral data, and then carefully convert the result to perceptual colors. But if we were to apply the Fresnel equations directly to RGB values, it would look…



…like this. RGB eta and kappa values are (somehow) derived from the spectral ones, and then plugged into the Fresnel equations to go directly to the final RGB values. This doesn't add any correctness over using a principled Fresnel approximation like the Cook-Torrance one, rather the opposite.



$$F(\theta) \approx F_0 + (1 - F_0)\frac{\max(0, F'(\theta) - F'(0^\circ))}{1 - F'(0^\circ)}$$

While Cook and Torrance were definitely onto something with their Fresnel approximation, it also requires — in addition to F0 — a single complex IOR value. This is much less convenient than only using F0, and may be why this approximation did not find wide use*.

* It has been used in a few places — for example, Maxwell Render by Next Limit has had this as part of their core material model since at least 2009.

1994: Schlick Approximation

$$F(\theta) \approx F_0 + (1 - F_0)\frac{\max(0, F'(\theta) - F'(0^\circ))}{1 - F'(0^\circ)}$$

$$F(\theta) \approx F_0 + (1 - F_0)(1 - \cos\theta)^5$$

One tweak was needed to make it a truly production-friendly solution, as shown in Schlick's 1994 paper*. The explicit goal was to create a simplified version of the Cook-Torrance Fresnel approximation that would not require any IOR values…

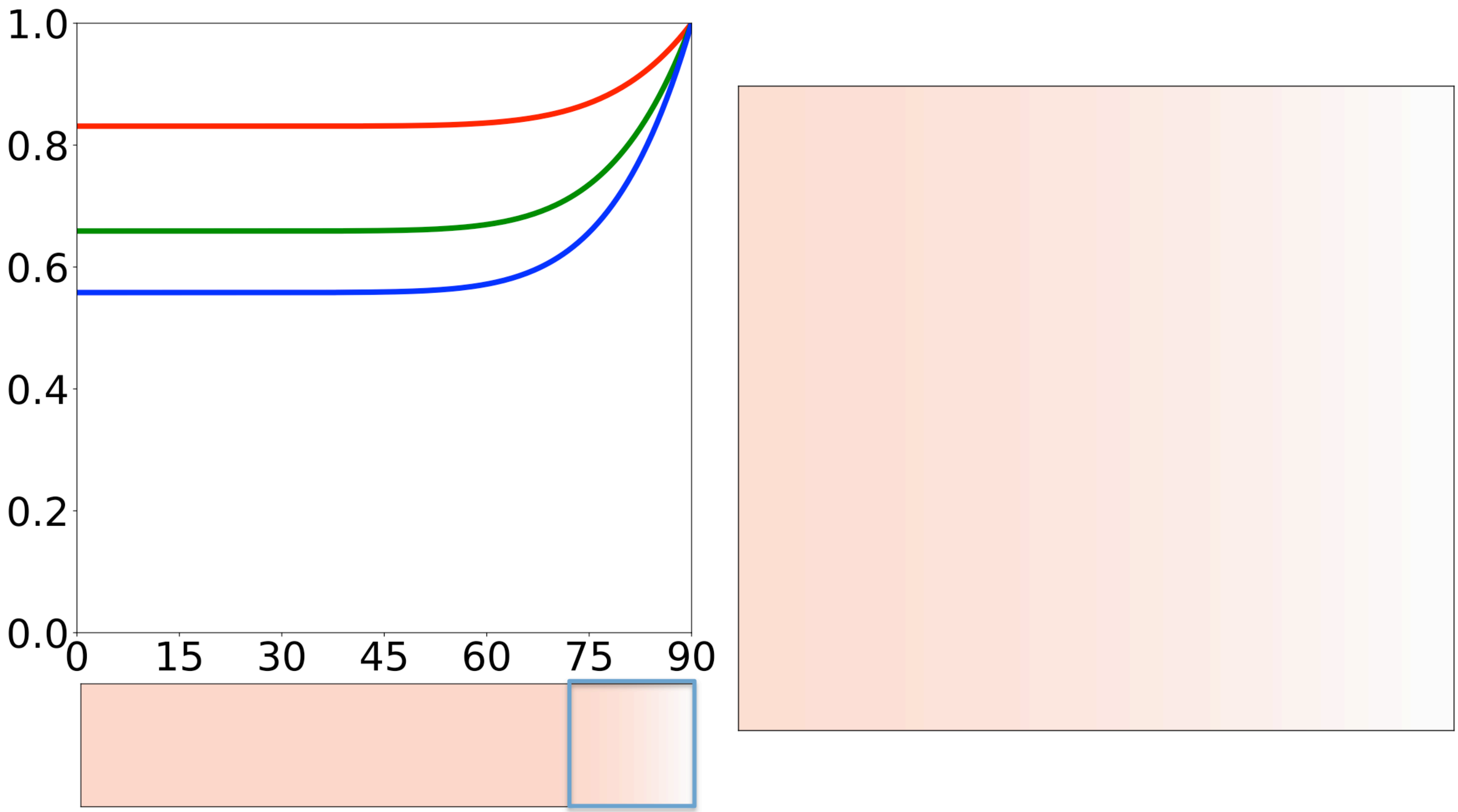
* C. Schlick, An Inexpensive BRDF Model for Physically-Based Rendering. Computer Graphics Forum, 1994.

1994: Schlick Approximation

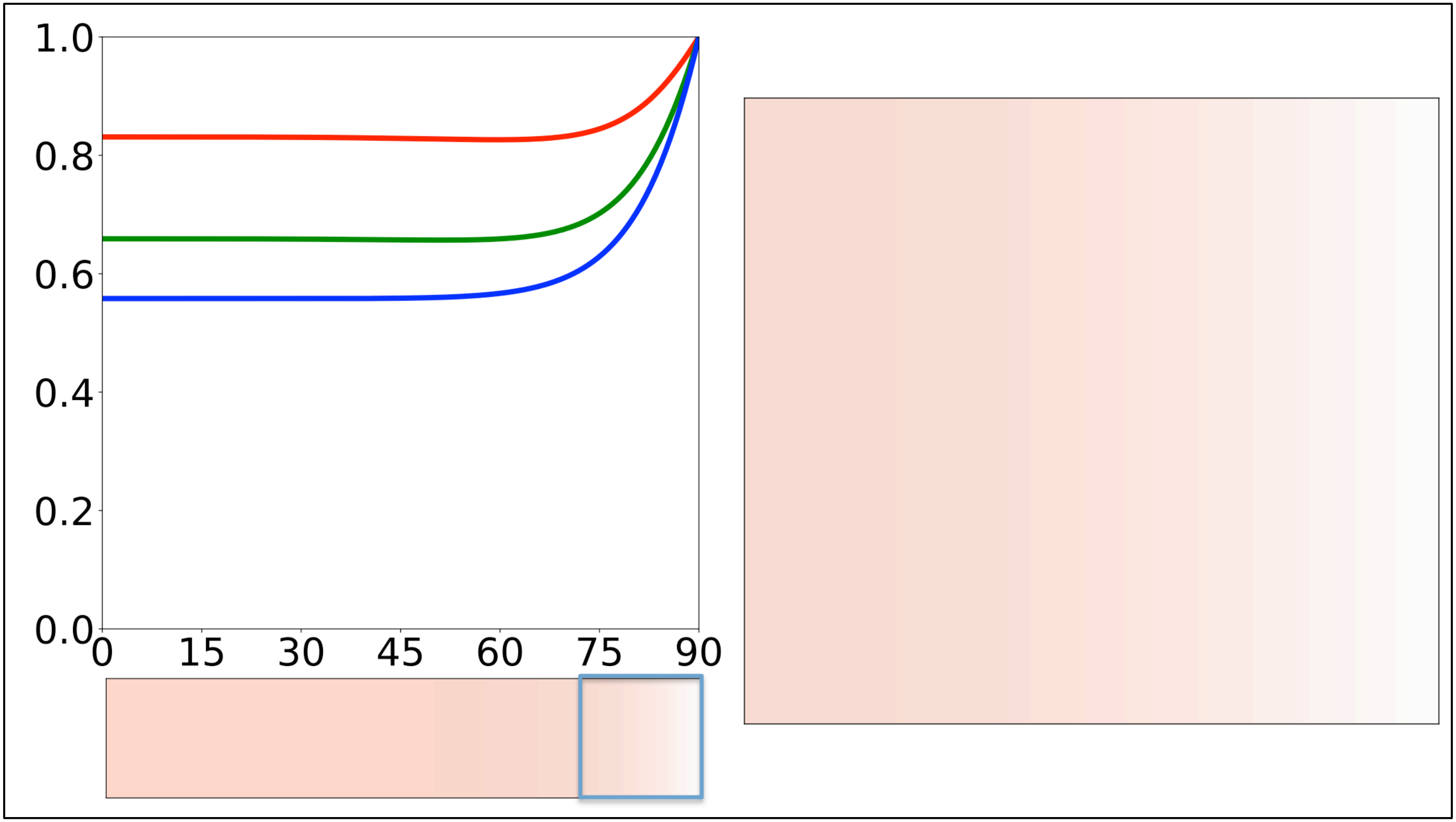
$$F(\theta) \approx F_0 + (1 - F_0)\frac{\max(0, F'(\theta) - F'(0°))}{1 - F'(0°)}$$

$$F(\theta) \approx F_0 + (1 - F_0)(1 - \cos\theta)^5$$

…only F0. The two approximations have a similar structure, though Schlick's is indeed simpler. The resulting Fresnel term is highly expressive, able to cover both dielectrics and metals by simply changing the value of F0. It's important to note that this approximation is not simply an optimization. When it was published, it was the <u>most correct</u> Fresnel term for the majority of practical rendering applications, both real time and offline.



How much accuracy is lost by removing the IOR parameter? Let's compare it to…



…the ground truth. When we **<u>flip between them</u>**, we see that the error is slightly larger than the original Cook-Torrance approximation. But this is just copper. How do other substances fare?

Metal	Schlick RMS dE2000
Silver	0.27
Copper	0.69
Gold	0.69
Bronze	0.78
Brass	1.04
Aluminum	1.12
Palladium	1.70
Nickel	1.82
Titanium	1.87
Mercury	1.89
Platinum	2.08
Iron	2.42
Zinc	2.44
Chromium	2.90

Dielectric	Schlick RMS dE2000
Diamond	1.18547
Cubic Zirconia	1.20586
Moissanite	1.40412
Water	1.46765
Glass SF66	1.49880
Acrylic	1.56734
Glass BK7	1.60457
Glass SF11	1.64151
Quartz	1.64646
Ruby	1.66026
Glass F5	1.67466
Polycarbonate	1.67845
Tourmaline	1.69488
PET	1.70027

Here is the Schlick delta-E 2000 error (averaged over all incidence angles) for a variety of materials, with metals and dielectrics each sorted in order of increasing error. The errors in metals vary quite widely; in contrast, all the dielectrics fall in between Aluminum and Palladium.

Metal	Schlick RMS dE2000
Silver	0.27
Copper	0.69
Gold	0.69
Bronze	0.78
Brass	1.04
Aluminum	1.12
Palladium	1.70
Nickel	1.82
Titanium	1.87
Mercury	1.89
Platinum	2.08
Iron	2.42
Zinc	2.44
Chromium	2.90

Dielectric	Schlick RMS dE2000
Diamond	1.18547
Cubic Zirconia	1.20586
Moissanite	1.40412
Water	1.46765
Glass SF66	1.49880
Acrylic	1.56734
Glass BK7	1.60457
Glass SF11	1.64151
Quartz	1.64646
Ruby	1.66026
Glass F5	1.67466
Polycarbonate	1.67845
Tourmaline	1.69488
PET	1.70027

We see that copper is a material for which Schlick performs unusually well, so let's look at some more representative materials.

Metal	Schlick RMS dE2000
Silver	0.27
Copper	0.69
Gold	0.69
Bronze	0.78
Brass	1.04
Aluminum	1.12
Palladium	1.70
Nickel	1.82
Titanium	1.87
Mercury	1.89
Platinum	2.08
Iron	2.42
Zinc	2.44
Chromium	2.90

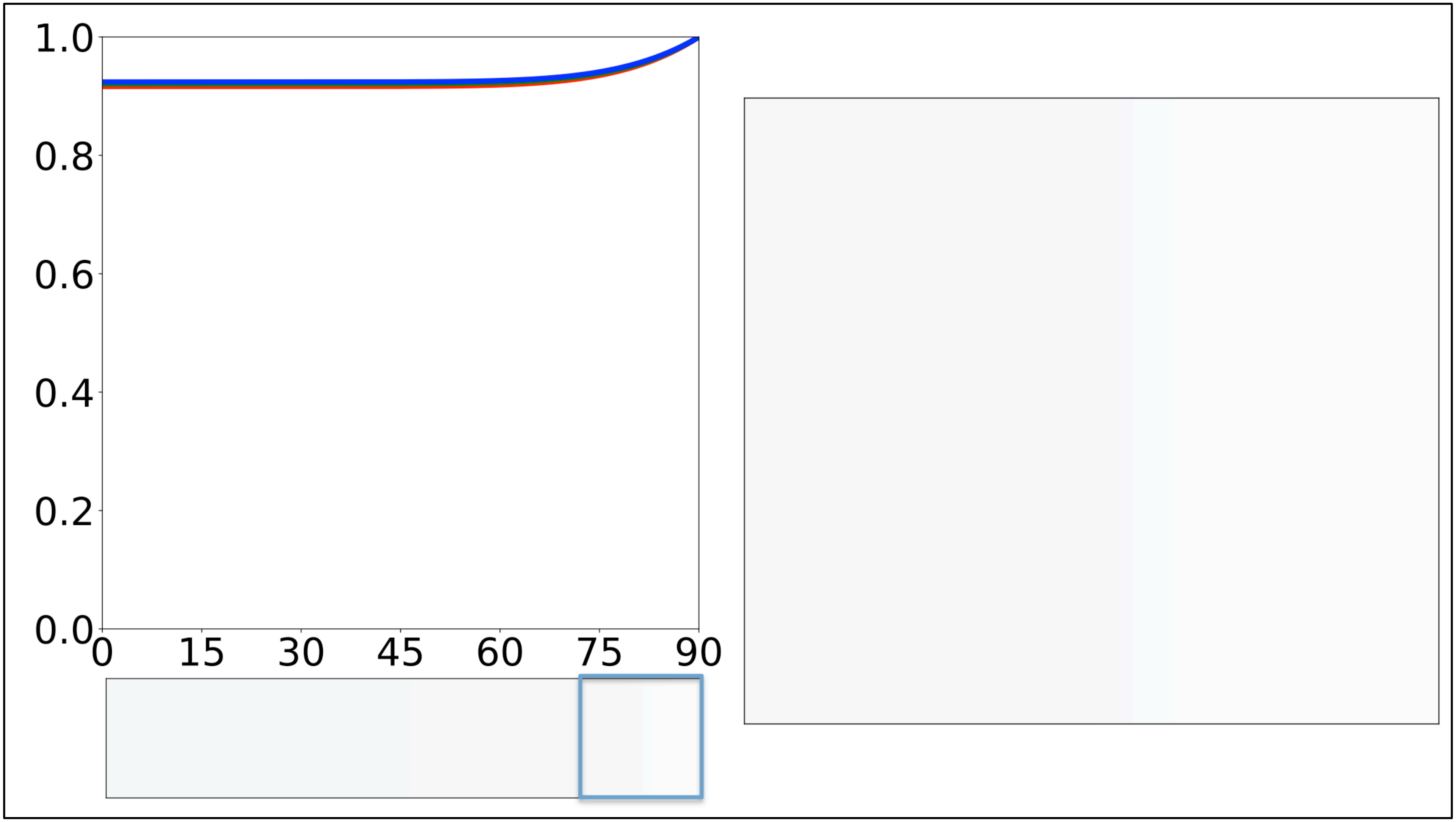
Dielectric	Schlick RMS dE2000
Diamond	1.18547
Cubic Zirconia	1.20586
Moissanite	1.40412
Water	1.46765
Glass SF66	1.49880
Acrylic	1.56734
Glass BK7	1.60457
Glass SF11	1.64151
Quartz	1.64646
Ruby	1.66026
Glass F5	1.67466
Polycarbonate	1.67845
Tourmaline	1.69488
PET	1.70027

Aluminum and SF11 Glass are each reasonably representative of their class.

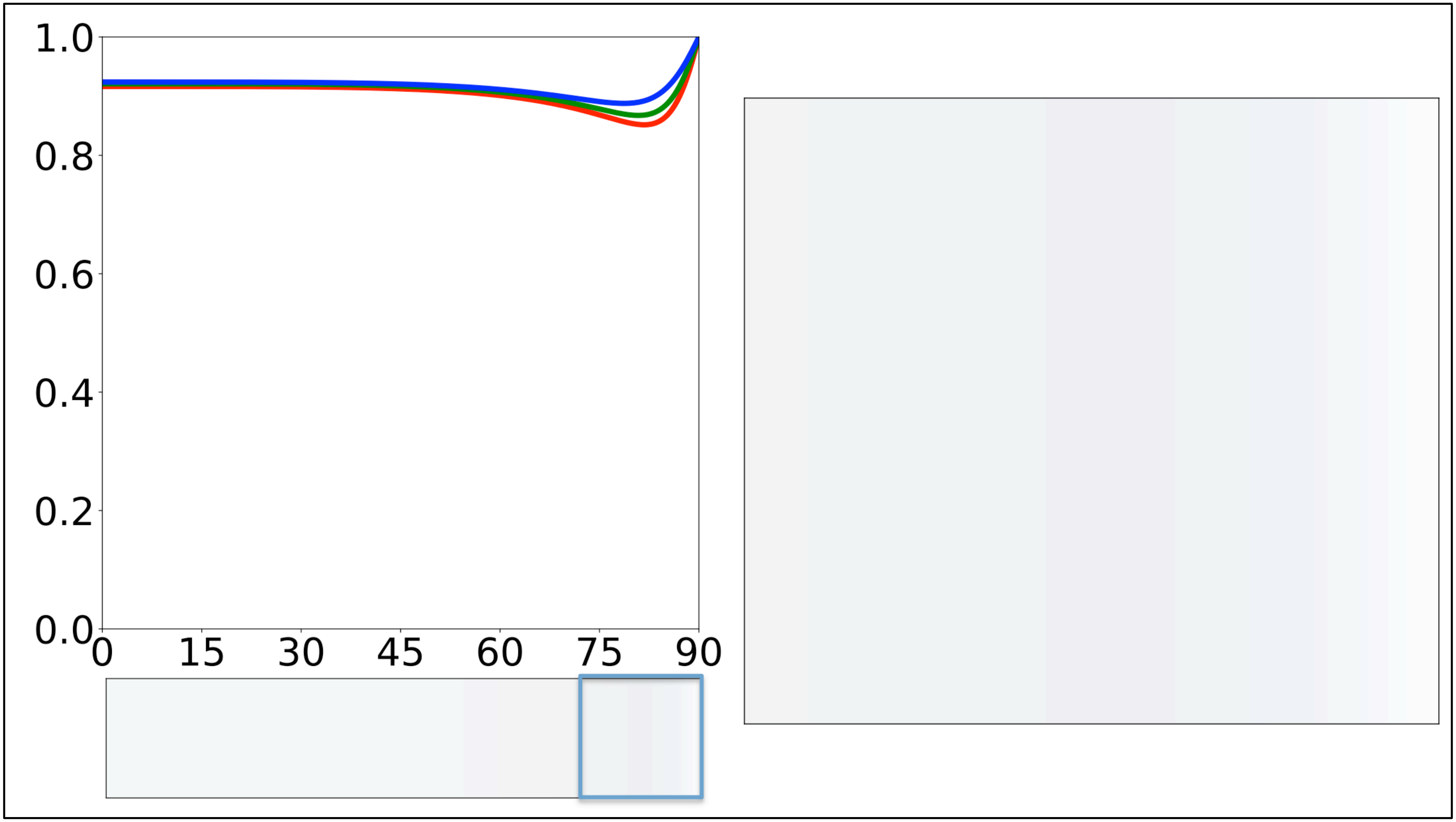
Metal	Schlick RMS dE2000
Silver	0.27
Copper	0.69
Gold	0.69
Bronze	0.78
Brass	1.04
Aluminum	1.12
Palladium	1.70
Nickel	1.82
Titanium	1.87
Mercury	1.89
Platinum	2.08
Iron	2.42
Zinc	2.44
Chromium	2.90

Dielectric	Schlick RMS dE2000
Diamond	1.18547
Cubic Zirconia	1.20586
Moissanite	1.40412
Water	1.46765
Glass SF66	1.49880
Acrylic	1.56734
Glass BK7	1.60457
Glass SF11	1.64151
Quartz	1.64646
Ruby	1.66026
Glass F5	1.67466
Polycarbonate	1.67845
Tourmaline	1.69488
PET	1.70027

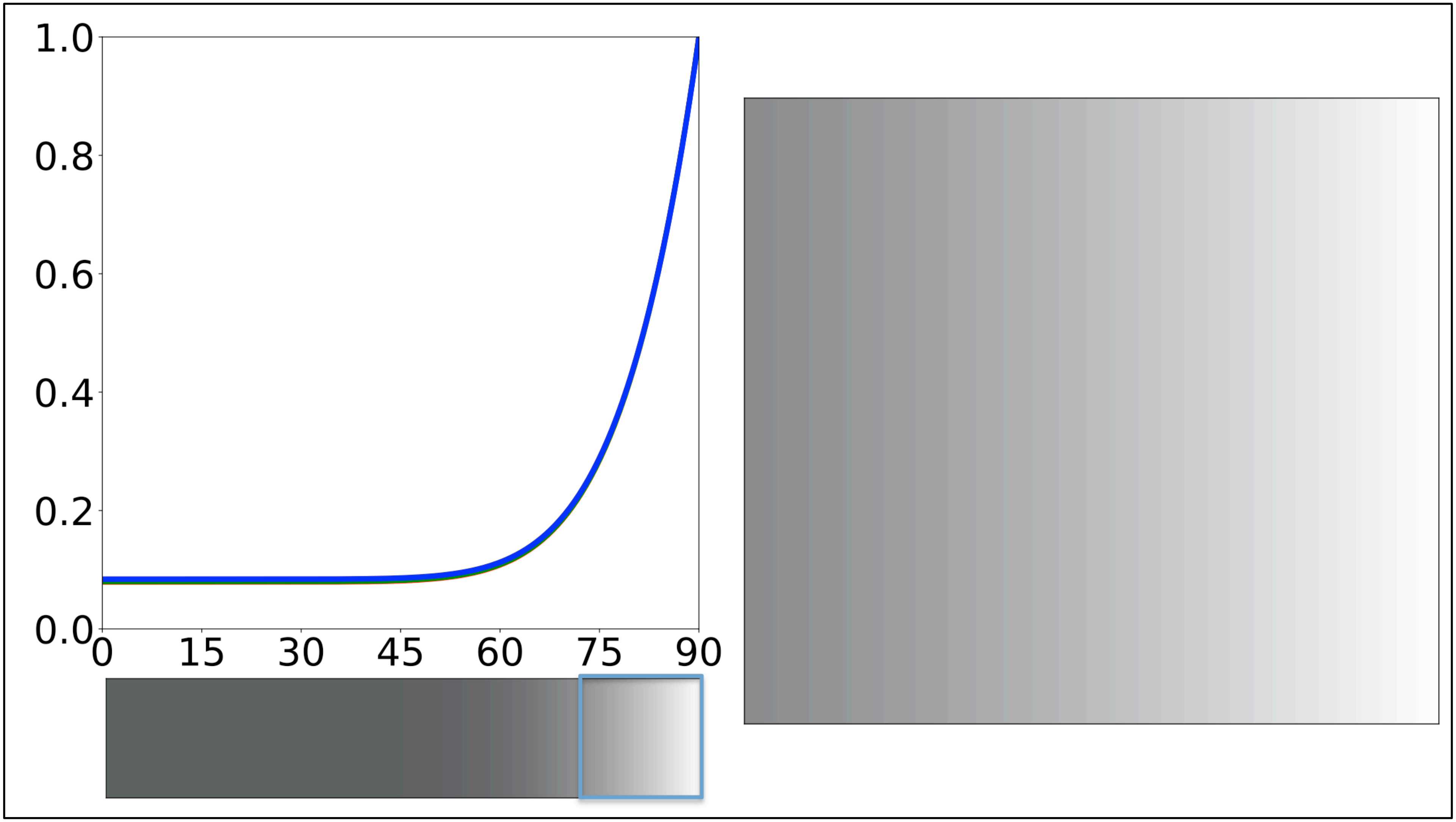
We will also look at the worst case material, which is Chromium.



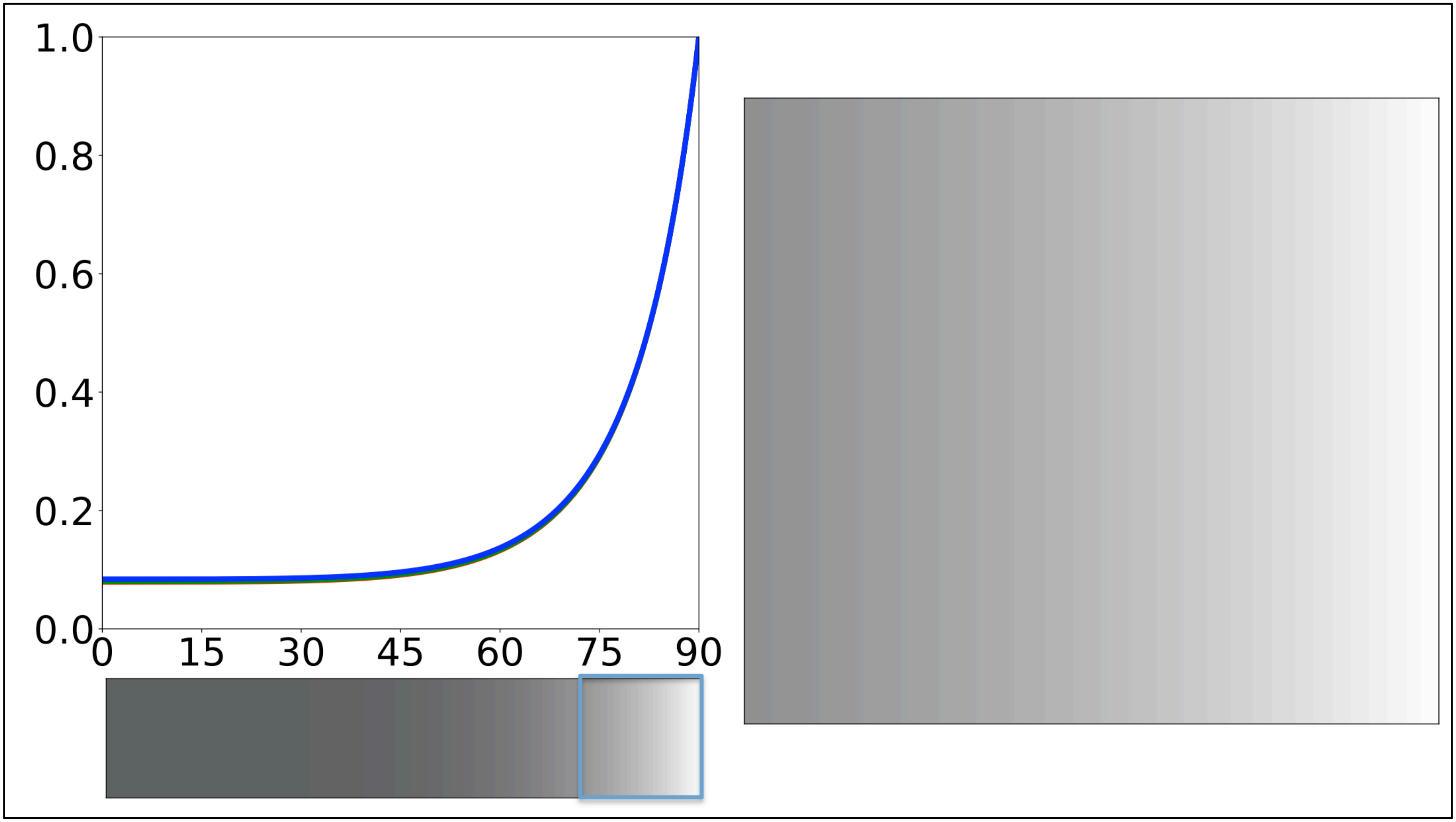
Here are plots and color strips for Aluminum, first for the Schlick approximation…



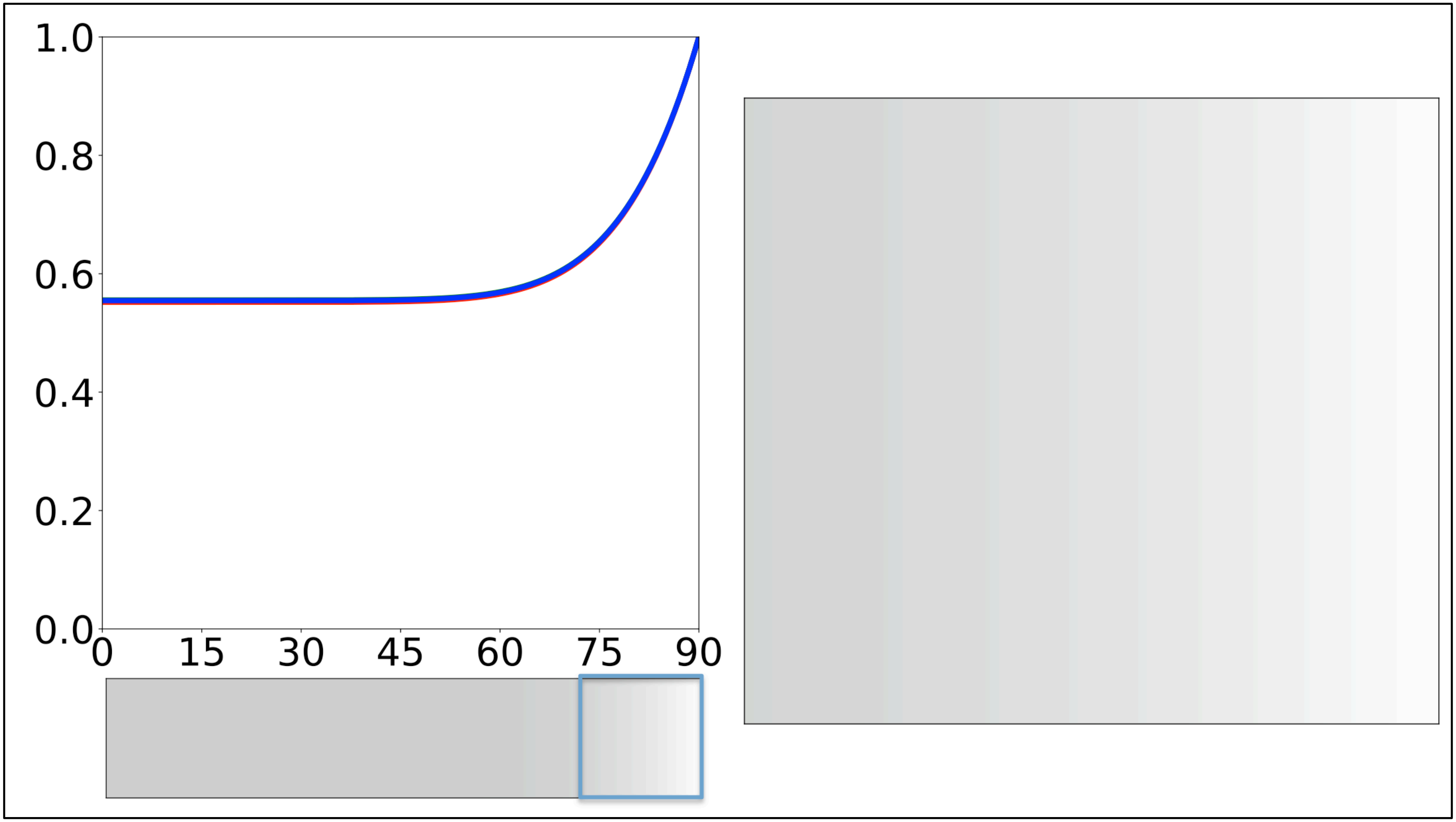
…then the ground truth. When we **<u>flip between them</u>**, we see that Aluminum has a “dip” near 90 degrees that the Schlick approximation does not reproduce.



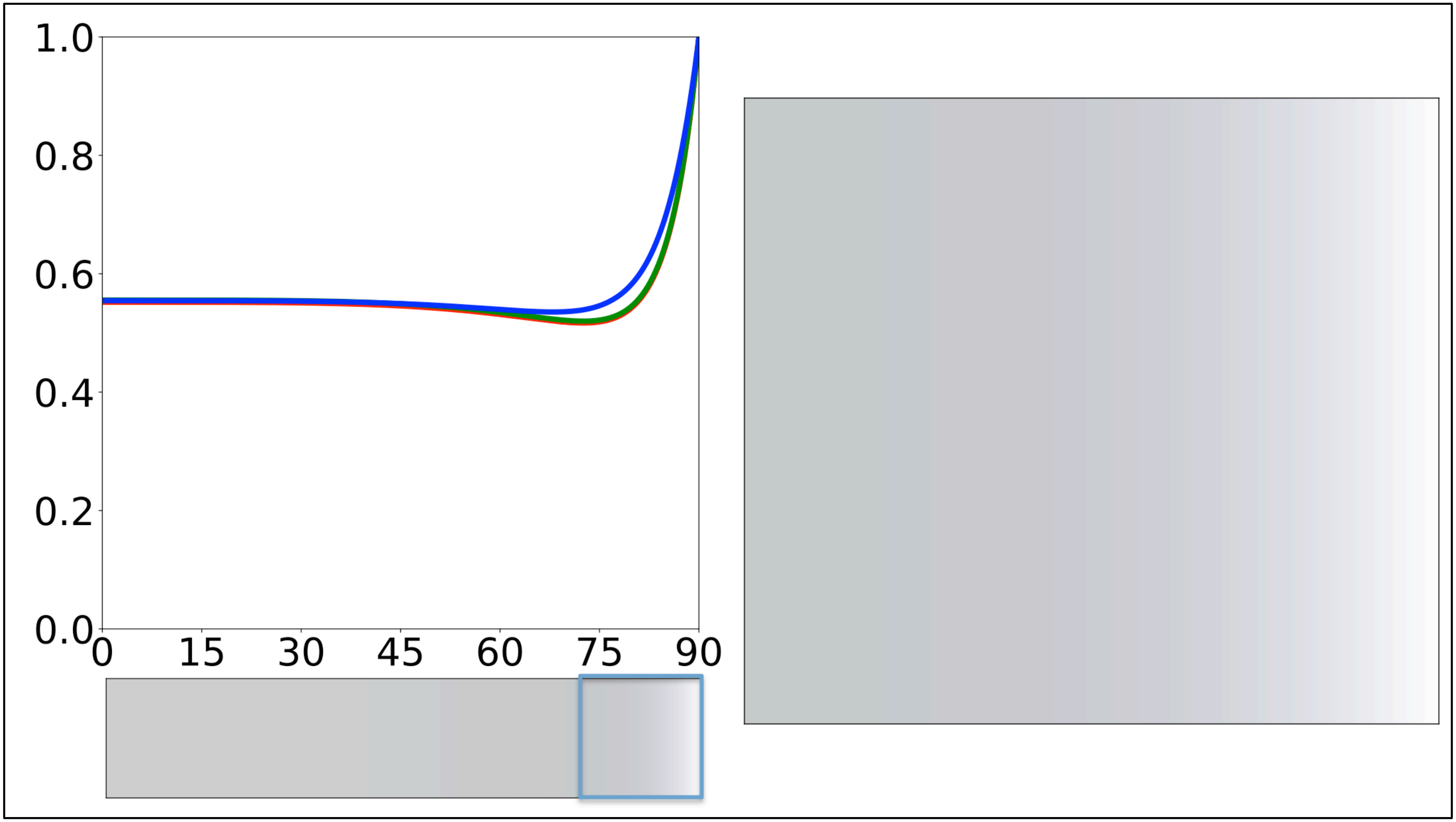
Now glass, first the Schlick approximation…



…and the ground truth. **Flipping between them** shows that, as is typical for dielectrics, there isn't a "dip" but rather a slight mismatch in the curve shape around 60 degrees.



And finally the Schlick approximation for Chromium…



…and the ground truth. When we **flip between them**, we see a more prominent version of the "missing dip" we saw earlier.

2005: Lazányi Approximation

$$F(\theta) \approx F_0 + (1 - F_0)(1 - \cos\theta)^5 - a\cos\theta(1 - \cos\theta)^\alpha$$

The 2005 paper* by Lazányi and Szimay-Kalos attempted to improve the accuracy of the Schlick approximation for metals by …

* I. Lazányi and L. Szirmay-Kalos, Fresnel term approximations for metals. WSCG, 2005.

2005: Lazányi Approximation

$$F(\theta) \approx F_0 + (1 - F_0)(1 - \cos\theta)^5 \boxed{- a\cos\theta(1 - \cos\theta)^\alpha}$$

…adding an additional term aimed specifically at reproducing the “missing dip” in metals.

2005: Lazányi Approximation

$$F(\theta) \approx F_0 + (1 - F_0)(1 - \cos\theta)^5 - a\cos\theta(1 - \cos\theta)^{\alpha}$$

The approximation also added two additional parameters.

2005: Lazányi Approximation

$$F(\theta) \approx F_0 + (1 - F_0)(1 - \cos\theta)^5 - a\cos\theta(1 - \cos\theta)^\alpha$$

$$a = 2\eta$$

The authors propose setting “a” to two times eta but then we need an IOR sample of the material*. And as pointed out by Sébastien Lagarde**, the value of alpha depends on the extremum of the error term which has no analytical solution. Probably due to these difficulties, this approximation did not see significant production use.

* We can’t infer eta from F0 since this substitution explicitly depends on the value of kappa being relatively high.
** S. Lagarde, Memo on Fresnel equations. Blog post, 2013.

2007-2012: Broad Adoption of Physically Motivated Shading Models

- 2008: ILM (*Iron Man, Terminator: Salvation)*
- 2008: Pixar (*Wall-E, Up*)
- 2008: Bungie (*Halo 3*)
- 2009: Imageworks (*2012, Alice in Wonderland*)
- 2010: Treyarch (*Call of Duty: Black Ops*)
- 2012: Disney Animation (*Wreck-It Ralph*)

2007 was the beginning of the “first wave” of significant adoption of physically motivated models in film and game production, with the first titles coming out about a year later. The SIGGRAPH 2012 publication* of the Disney Principled Shading Model (first used in *Wreck-It Ralph*) was hugely influential and to my mind marked the end of the “first wave”. Note that all of these shows and games used the Schlick approximation.

* B. Burley, Physically Based Shading at Disney. Practical Physically Based Shading in Film and Game Production, SIGGRAPH Course, 2012.

Generalized Schlick Approximation

$$F(\theta) \approx F_0 + (F_{90} - F_0)(1 - \cos\theta)^p$$

To be more precise, most used it as originally published, but some used a generalized form of the Schlick approximation…

Generalized Schlick Approximation

$$F(\theta) \approx F_0 + (F_{90} - F_0)(1 - \cos\theta)^p$$

…that added two parameters: a 90-degree color (F90) and a Fresnel power (p). Why are these extra parameters good for production use while those in (for example) the Lazányi approximation are not? They are easy for artists to control and they have default values that revert to original Schlick. Note: these parameters were added to increase expressivity, not accuracy. But they could increase accuracy in some cases that aren't modeled well by the Fresnel equations, like loosely packed fine dust.

$$F(\theta) \approx F_0 + (1 - F_0)\frac{\max(0, F'(\theta) - F'(0^\circ))}{1 - F'(0^\circ)}$$

$$F(\theta) \approx F_0 + (1 - F_0)(1 - \cos\theta)^5$$

$$F(\theta) \approx F_0 + (1 - F_0)(1 - \cos\theta)^5 - a\cos\theta(1 - \cos\theta)^\alpha$$

$$F(\theta) \approx F_0 + (F_{90} - F_0)(1 - \cos\theta)^p$$

The models we’ve discussed so far follow the same approach as Cook and Torrance’s color interpolation approximation, which is really the correct approach for anything other than a spectral renderer using accurate eta and kappa values.

2014: Artist-Friendly Metallic Fresnel

$$F(\theta) = \frac{F_s(\theta) + F_p(\theta)}{2}$$

$$F_s(\theta) = \frac{a^2 + b^2 - 2a\cos\theta + \cos^2\theta}{a^2 + b^2 + 2a\cos\theta + \cos^2\theta}$$

$$F_p(\theta) = F_s(\theta)\frac{a^2 + b^2 - 2a\sin\theta\tan\theta + \sin^2\theta\tan^2\theta}{a^2 + b^2 + 2a\sin\theta\tan\theta + \sin^2\theta\tan^2\theta}$$

$$2a^2 = \sqrt{(\eta^2 - \kappa^2 - \sin^2\theta)^2 + 4\eta^2\kappa^2} + (\eta^2 - \kappa^2 - \sin^2\theta)$$

$$2b^2 = \sqrt{(\eta^2 - \kappa^2 - \sin^2\theta)^2 + 4\eta^2\kappa^2} - (\eta^2 - \kappa^2 - \sin^2\theta)$$

That is not the case for this next approximation. In 2014, Gulbrandsen proposed* a new parameterization for the Fresnel equations. Unfortunately, it is based on the approach of computing the Fresnel equations for RGB values, which we showed earlier to be unsound.

* O. Gulbrandsen, Artist Friendly Metallic Fresnel. Journal of Computer Graphics Techniques (JCGT), 2014.

2014: Artist-Friendly Metallic Fresnel

$$\eta(F_0, g) = g\frac{1 - F_0}{1 + F_0} + (1 - g)\frac{1 + \sqrt{F_0}}{1 + \sqrt{F_0}}$$

$$\kappa(F_0, \eta) = \sqrt{\frac{1}{1 - F_0}\Big(F_0(\eta + 1)^2 - (\eta - 1)^2\Big)}$$

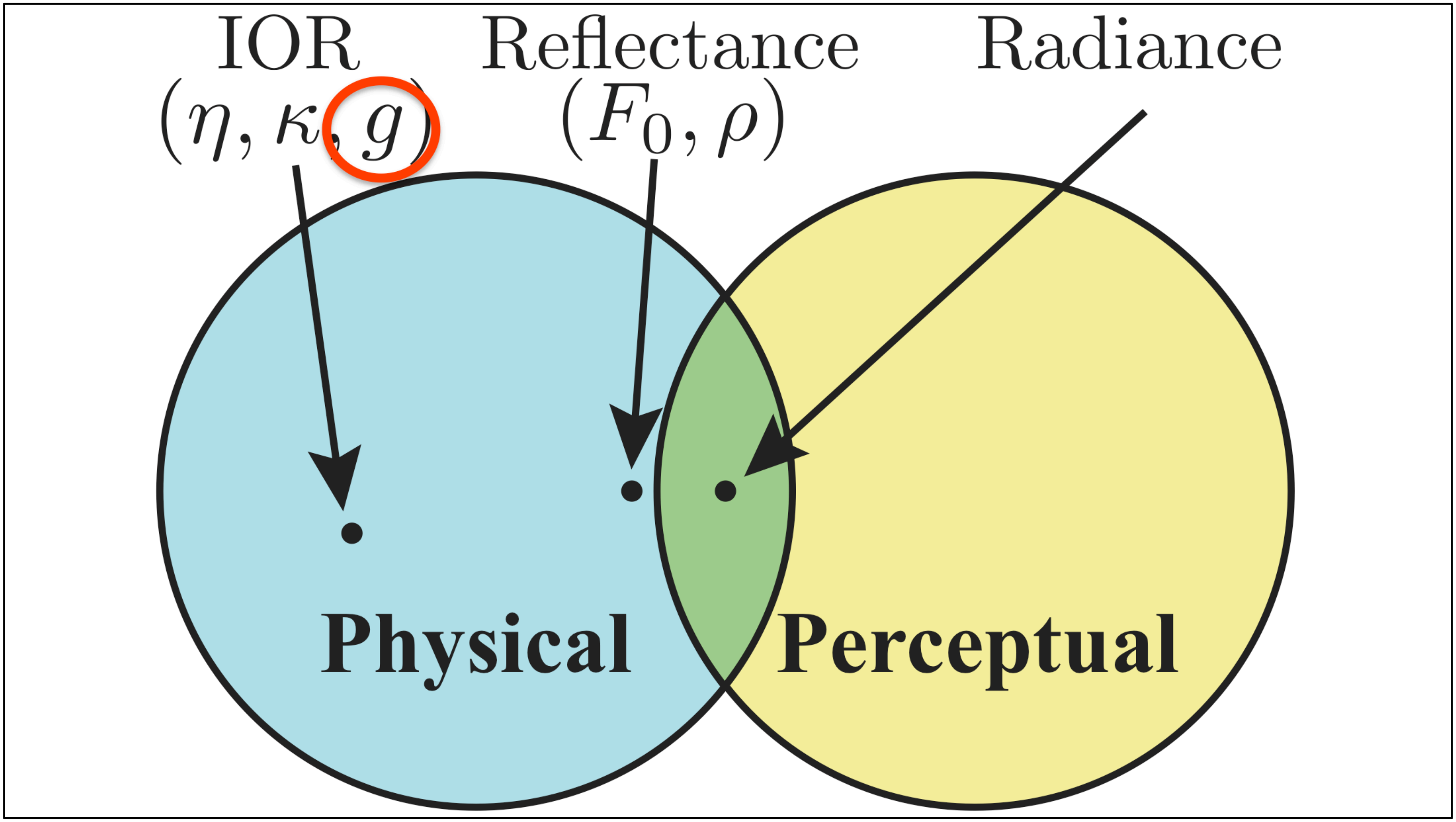
In this paper, the RGB Fresnel equations were reparameterized to use…

2014: Artist-Friendly Metallic Fresnel

$$\eta(F_0, g) = g\frac{1 - F_0}{1 + F_0} + (1 - g)\frac{1 + \sqrt{F_0}}{1 - \sqrt{F_0}}$$

$$\kappa(F_0, \eta) = \sqrt{\frac{1}{1 - F_0}\Big(F_0(\eta + 1)^2 - (\eta - 1)^2\Big)}$$

…F0 and a new edge tint parameter g, instead of eta and kappa. Effectively these parameters are a "user interface" on top of eta and kappa. Like F0, g is an RGB parameter where each channel is in the range of 0 to 1.



However, fundamentally the g edge tint parameter is unlike F0; it is far more akin to eta and kappa. The g parameter has no perceptual analog, and only an indirect and nonlinear relationship to final rendered colors (perceptual stimuli). It superficially resembles a color, but it really isn't one. This has some implications that we will cover later.

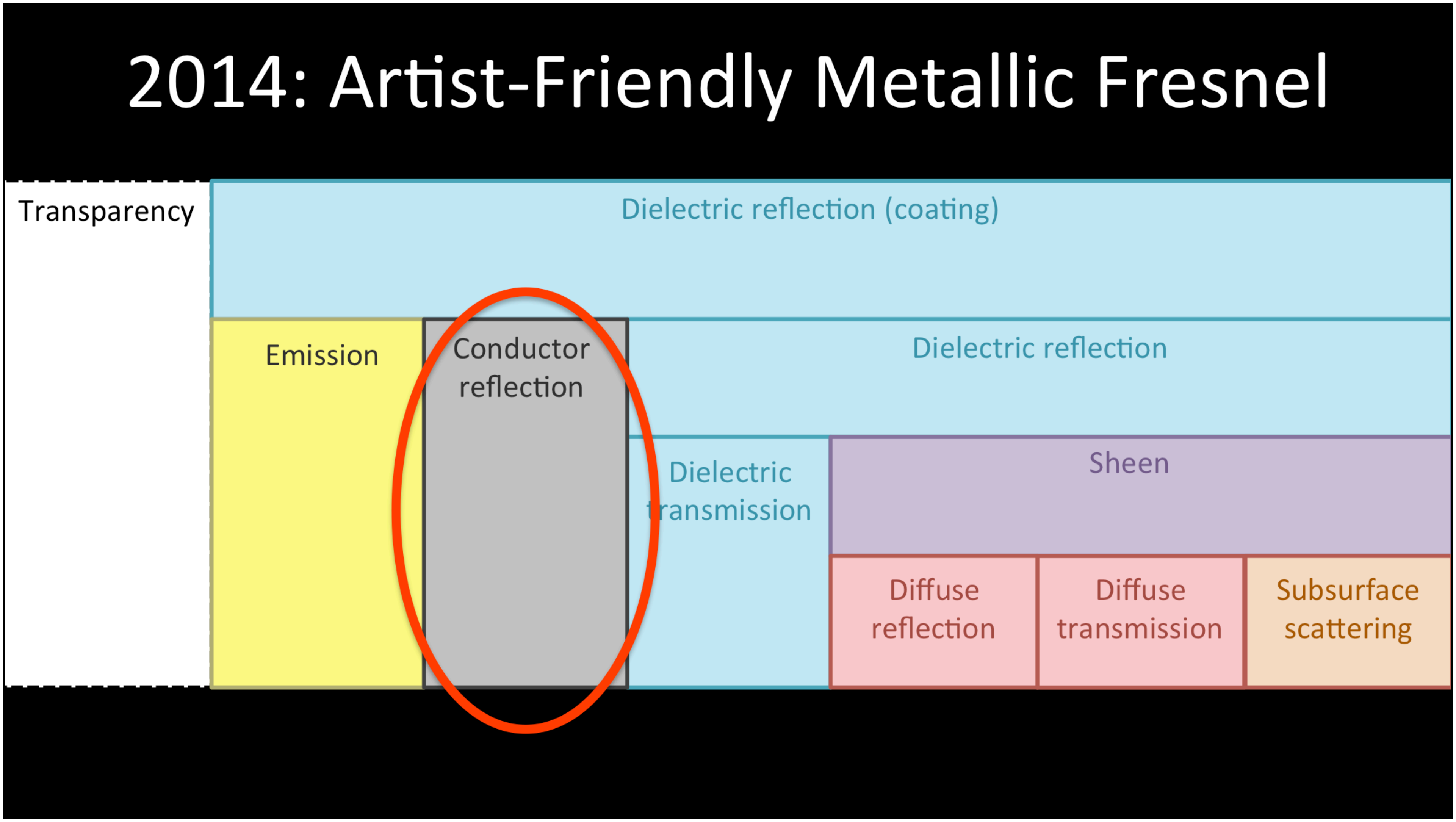
2014: Artist-Friendly Metallic Fresnel

$$\eta(F_0, g) = g\frac{1 - F_0}{1 + F_0} + (1 - g)\frac{1 + \sqrt{F_0}}{1 + \sqrt{F_0}}$$

$$\kappa(F_0, \eta) = \sqrt{\frac{1}{1 - F_0}\Big(F_0(\eta + 1)^2 - (\eta - 1)^2\Big)}$$

Perhaps due to some misplaced perception of improved correctness, this model is seeing increasing use in film rendering. Besides Framestore (where it originated), it is used by Weta Digital* and others…

* Weta Digital perform spectral upsampling on F0 and g textures for use in their Manuka spectral renderer.



…often in a dedicated conductor node next to Fresnel dielectric nodes, as in Autodesk Standard Surface, shown here. At Sony Imageworks, this Fresnel model is an artist-selectable option alongside generalized Schlick. This growing use of a fundamentally unsound model was my primary motivation for working on this talk.

(Image inspired by Figure 1 in the "Autodesk Standard Surface" white paper, I. Georgiev, J. Portsmouth, Z. Andersson, A. Herubel, A. King, S. Ogaki, and F. Servant.)

2019: Reparameterized Lazányi

$$a = \frac{823543}{46656}(F_0 - F_{82}) + \frac{49}{6}(1 - F_0)$$

$$F(\theta) \approx F_0 + (1 - F_0)(1 - \cos\theta)^5 - a\cos\theta(1 - \cos\theta)^6$$

Last year I presented* a model that attempted to include the "metallic dip" but without the shortcomings of other models that do so. It was a reparameterized version of the Lazányi approximation. The alpha parameter was…

* N. Hoffman, Fresnel Equations Considered Harmful. Eurographics Workshop on Material Appearance Modeling (MAM), 2019.

2019: Reparameterized Lazányi

$$a = \frac{823543}{46656}(F_0 - F_{82}) + \frac{49}{6}(1 - F_0)$$

$$F(\theta) \approx F_0 + (1 - F_0)(1 - \cos\theta)^5 - a\cos\theta(1 - \cos\theta)^6$$

…fixed at 6, and a new parameter…

2019: Reparameterized Lazányi

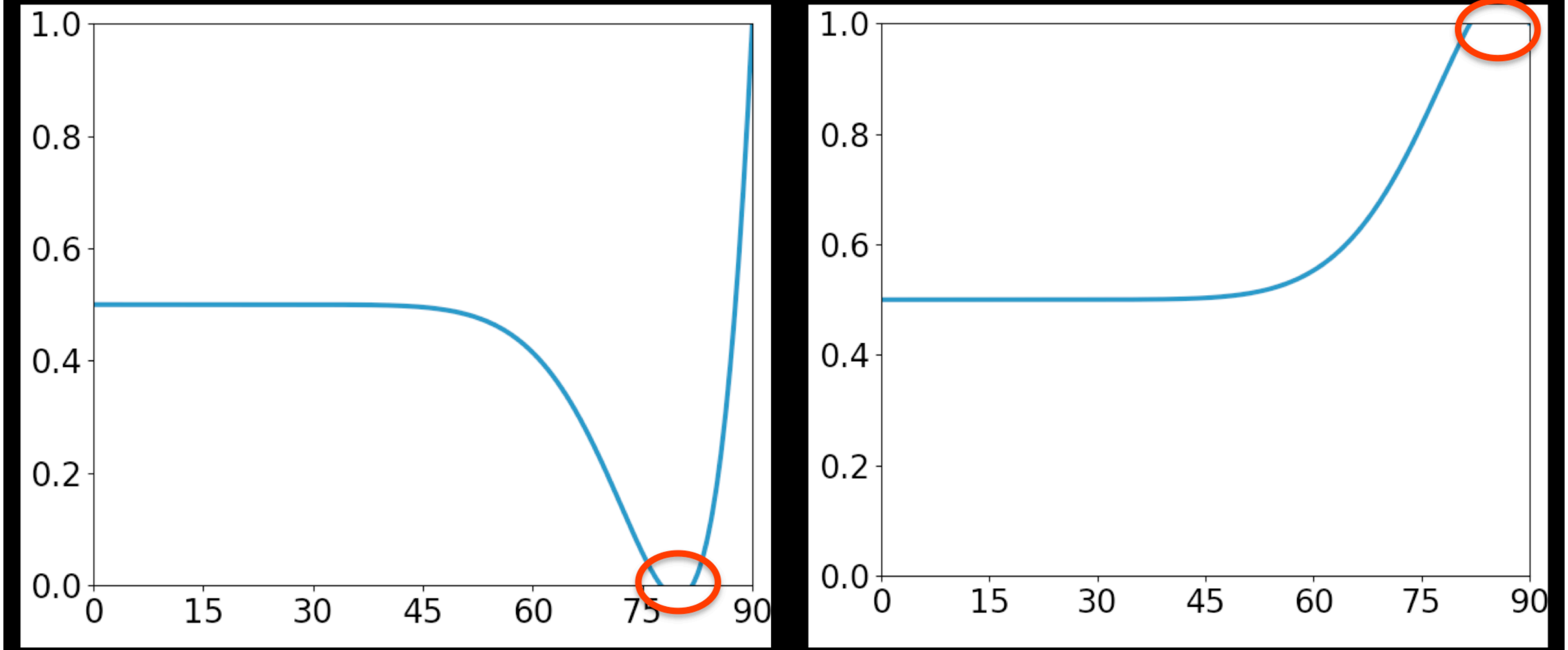
$$a = \frac{823543}{46656}(F_0 - F_{82}) + \frac{49}{6}(1 - F_0)$$

$$F(\theta) \approx F_0 + (1 - F_0)(1 - \cos\theta)^5 - a\cos\theta(1 - \cos\theta)^6$$

…F82 was introduced, from which the value of the other Lazányi parameter, a, is calculated. Like F0, F82 is defined as the reflected color at a particular angle*. Although this approximation does overcome many of the shortcomings of the g edge tint model, it still requires another parameter in addition to F0. It is unclear whether the additional expressiveness and potential for increased accuracy is worth this cost.

* Approximately 82 degrees: to be precise, arccos(1/7).

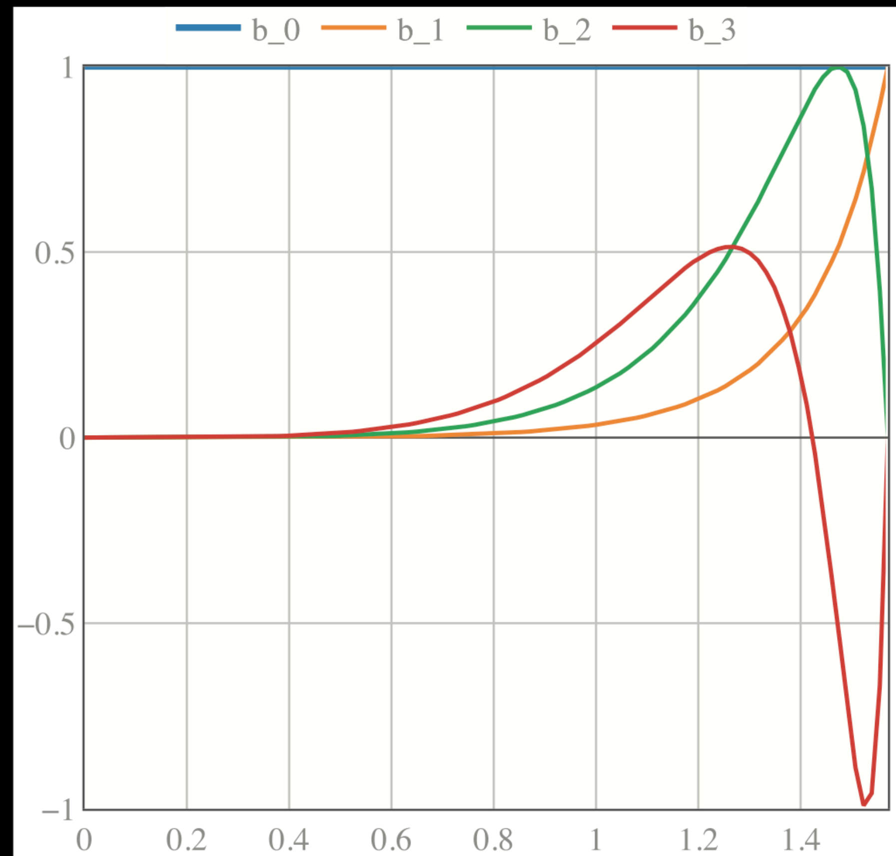
2019: Reparameterized Lazányi



One limitation of the model is that certain extreme combinations of F0 and F82 values can cause the resulting curves to go below 0 or above 1, in which case they need to be clamped, causing C1 discontinuities.

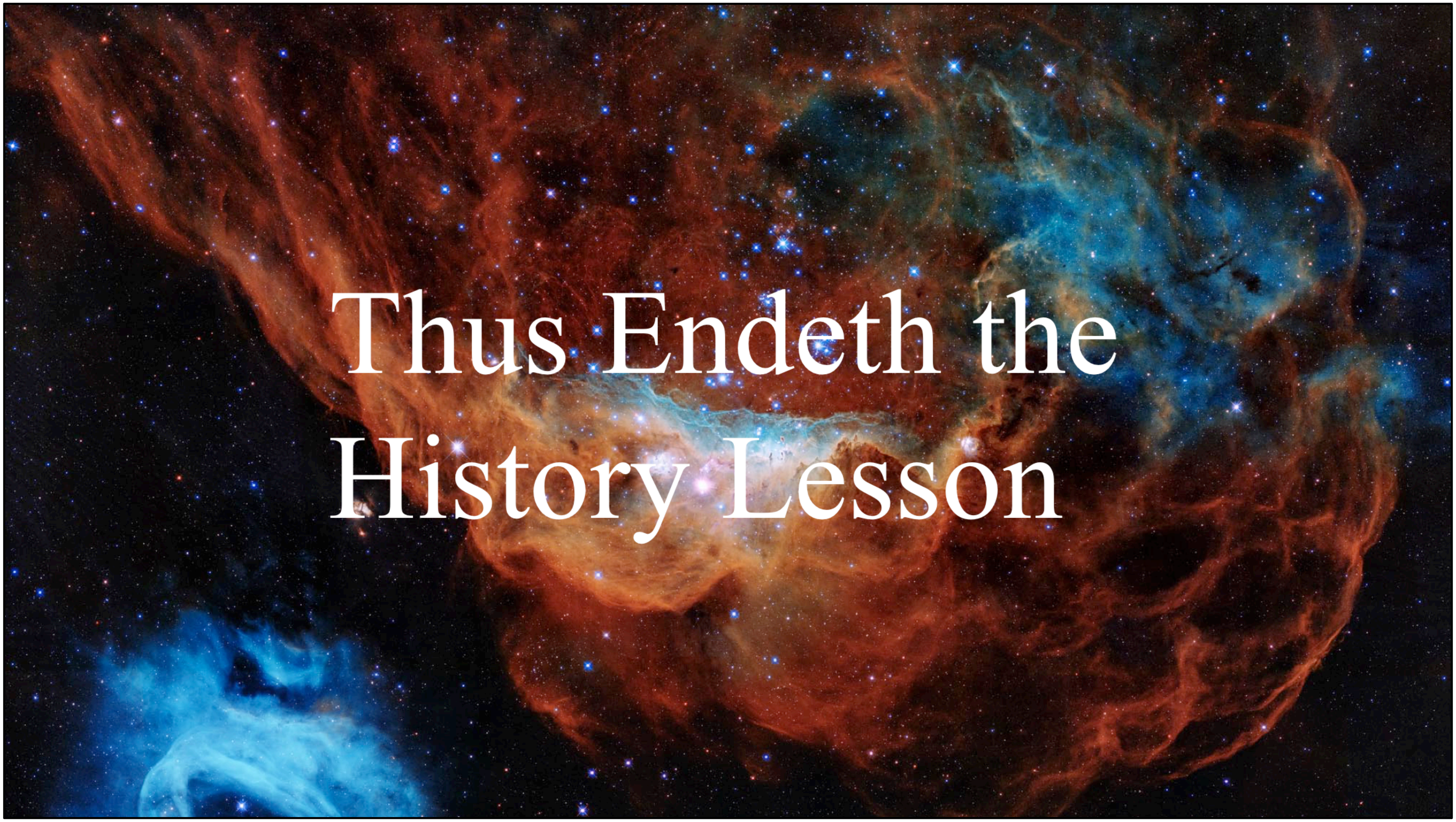
2020: Empirical Fresnel Basis

$$F(\theta) = \sum_i w_i b_i(\theta)$$



Bending the concept of a "history" lesson by extending it 15 minutes into the future, I'll briefly mention the very next talk* in this course, which will be presented by Laurent Belcour. This work extends the "interpolation" approach of Cook, Torrance, Schlick, and Lazányi to interpolating entire curves. It preserves the fundamental soundness of the interpolation approach while opening up a new class of Fresnel models.

* L. Belcour, M. Bati, and P. Barla, Bringing an accurate Fresnel to Real-time Rendering: A Preintegrable Decomposition. Physically Based



Which brings us to here and now.

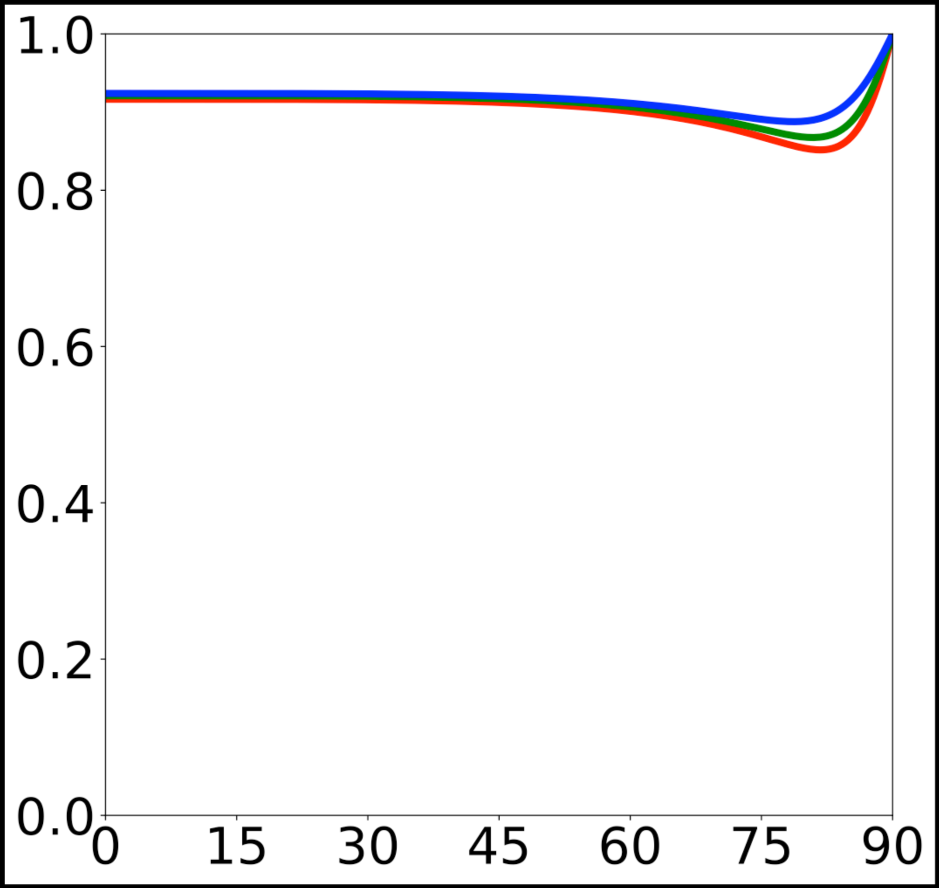
Some Myths about Schlick

I will next discuss some common misconceptions about the Schlick model.

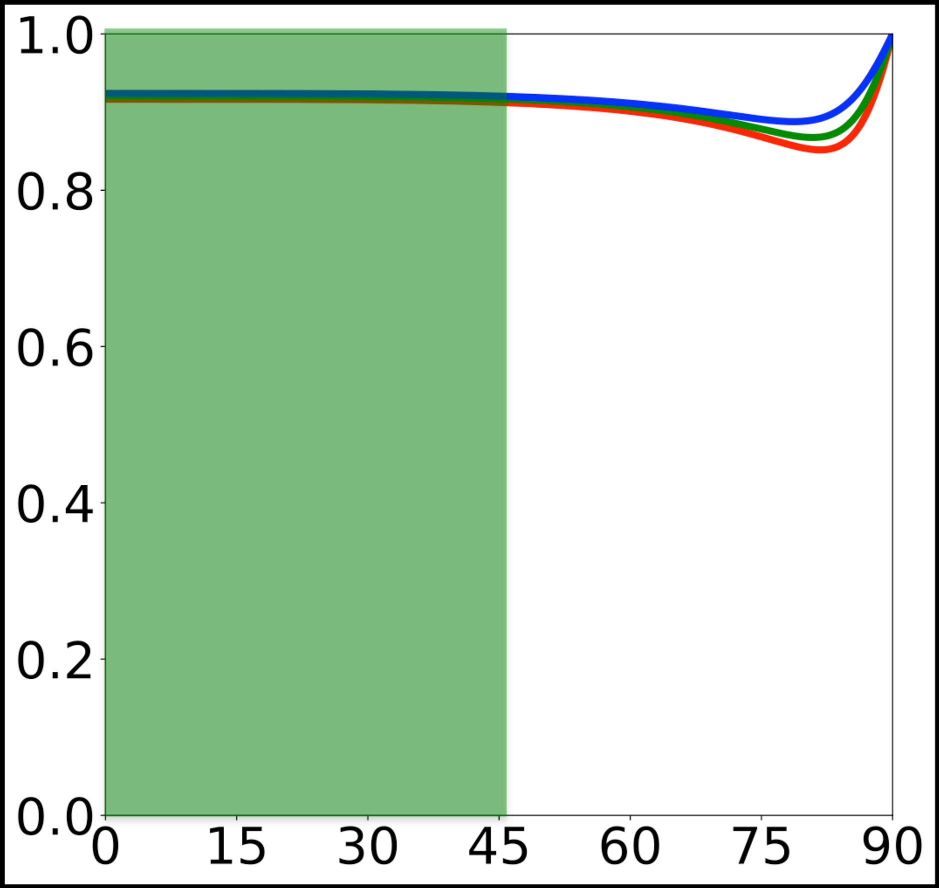
Schlick Myth #1

“It isn’t accurate enough”

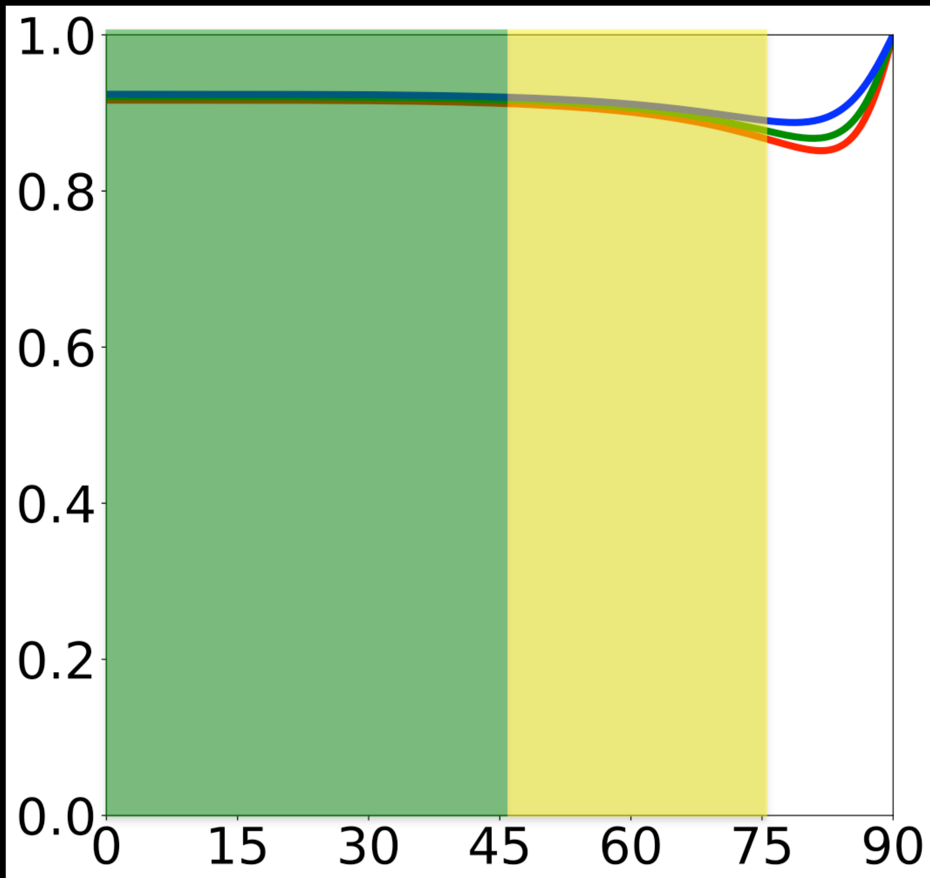
Perhaps the the biggest is that there is an accuracy problem with Schlick that needs to be solved.



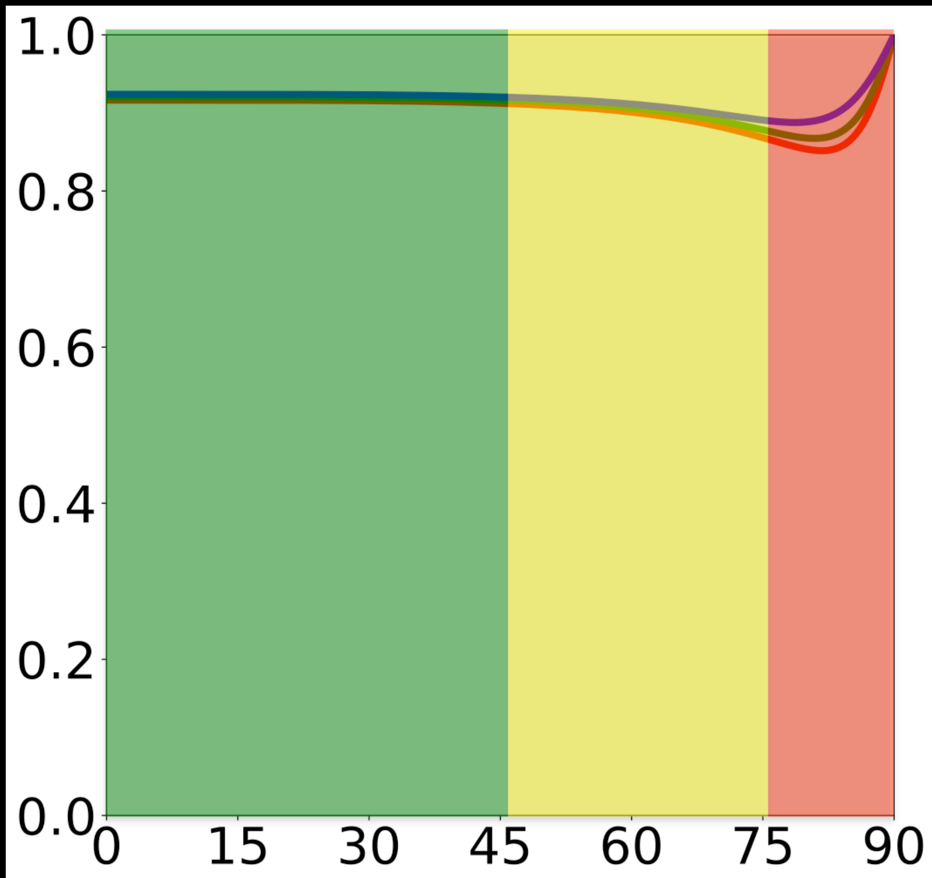
The type of plots I showed earlier can be misleading.



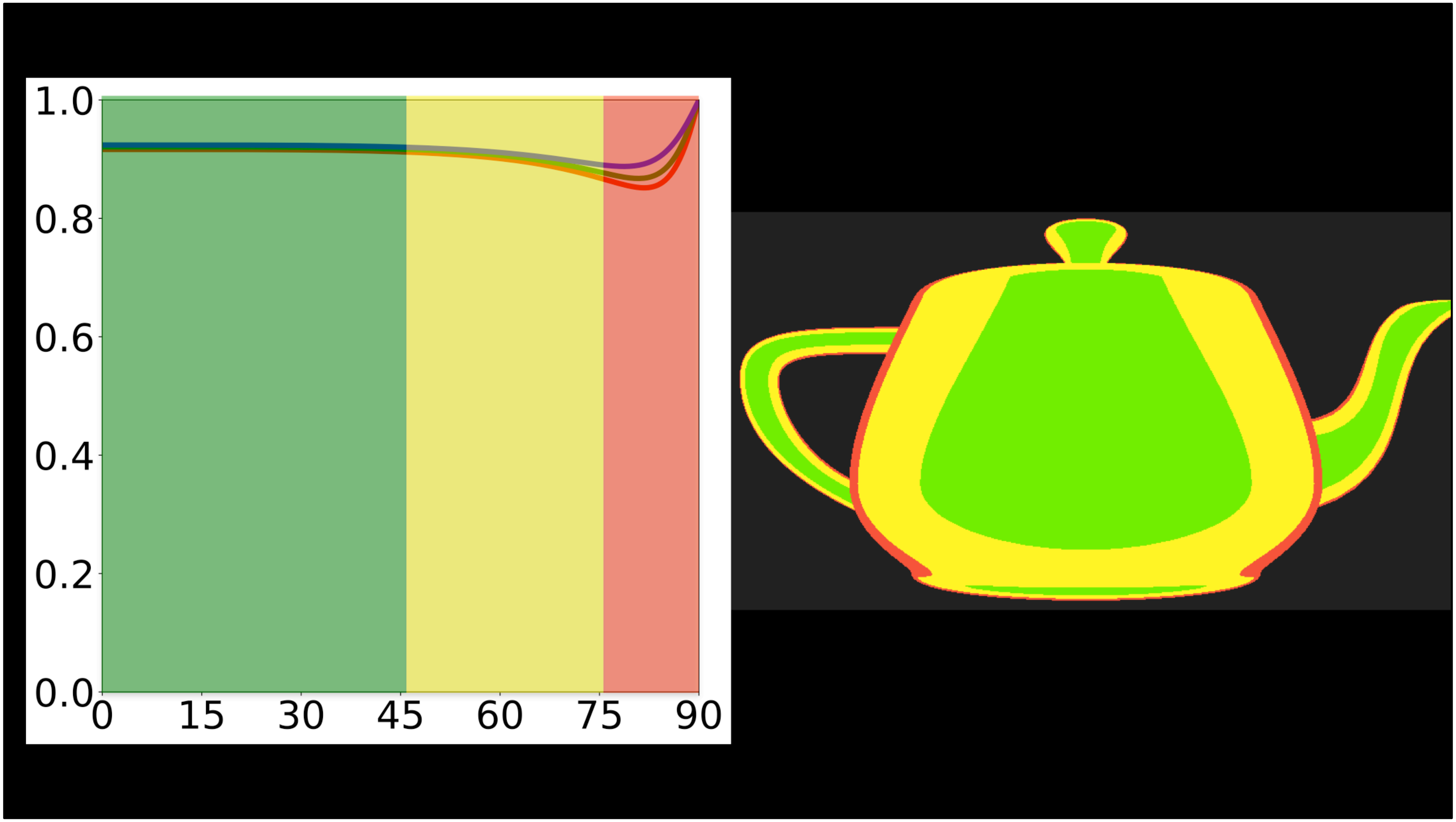
Metal Fresnel curves typically have a region where the reflectance is pretty much constant…



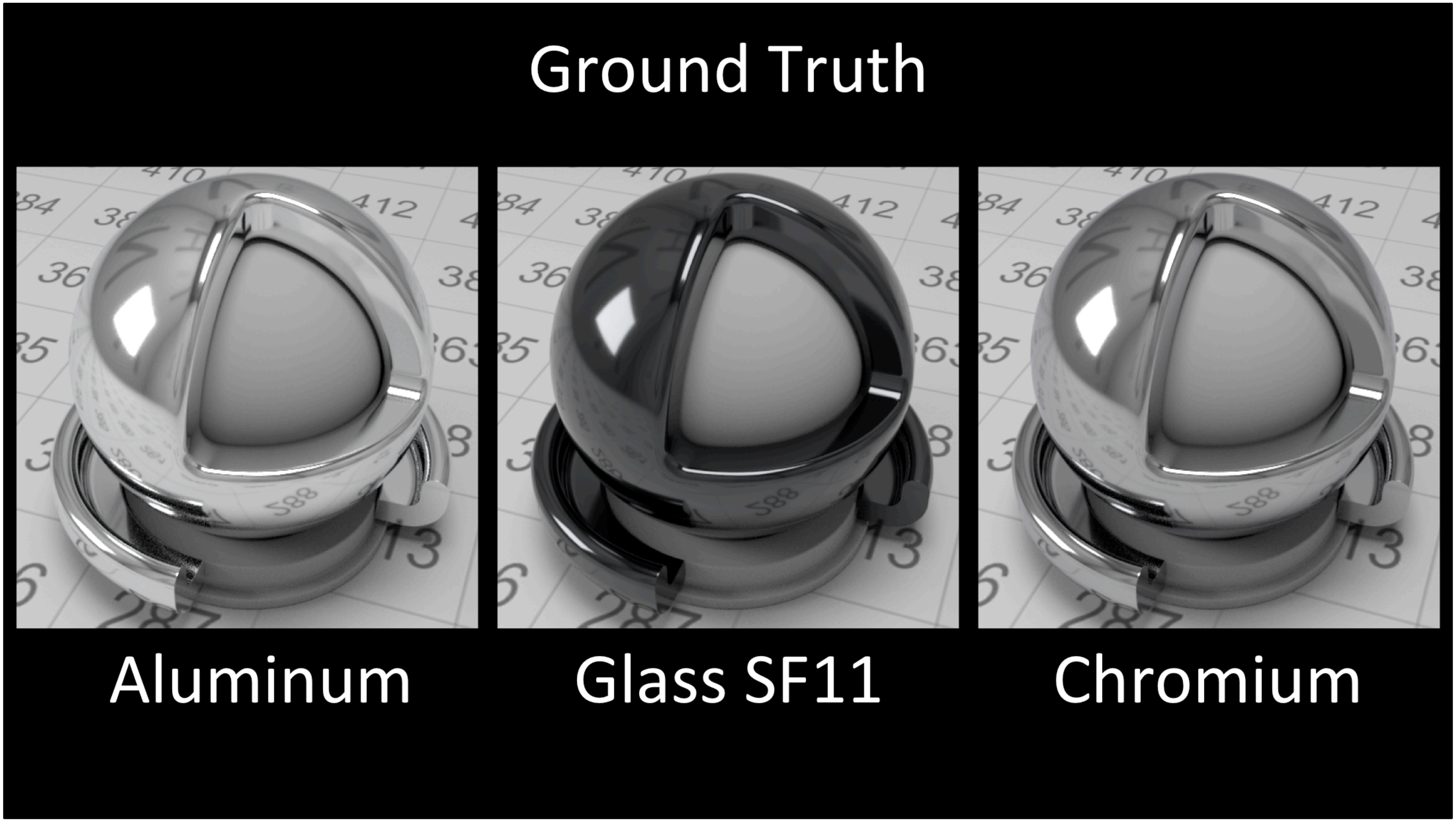
…a region where the curve changes slowly…



…and finally, a region where the interesting edge behavior (like “dips”) tends to occur.

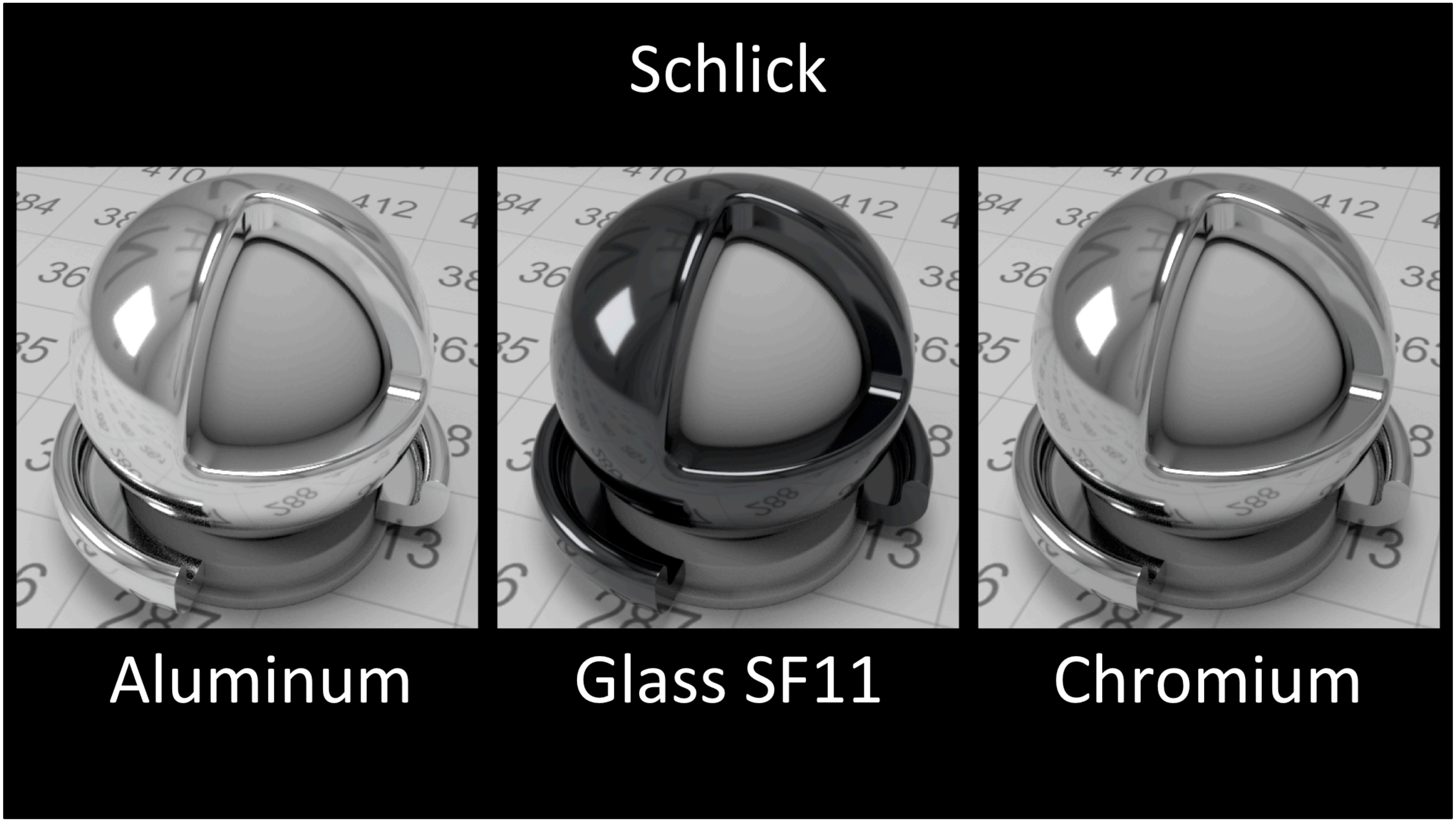


But foreshortening tends to flatten the edge regions, making them cover fewer pixels on screen. So this edge behavior is less prominent than it would seem from the plots and color strips.



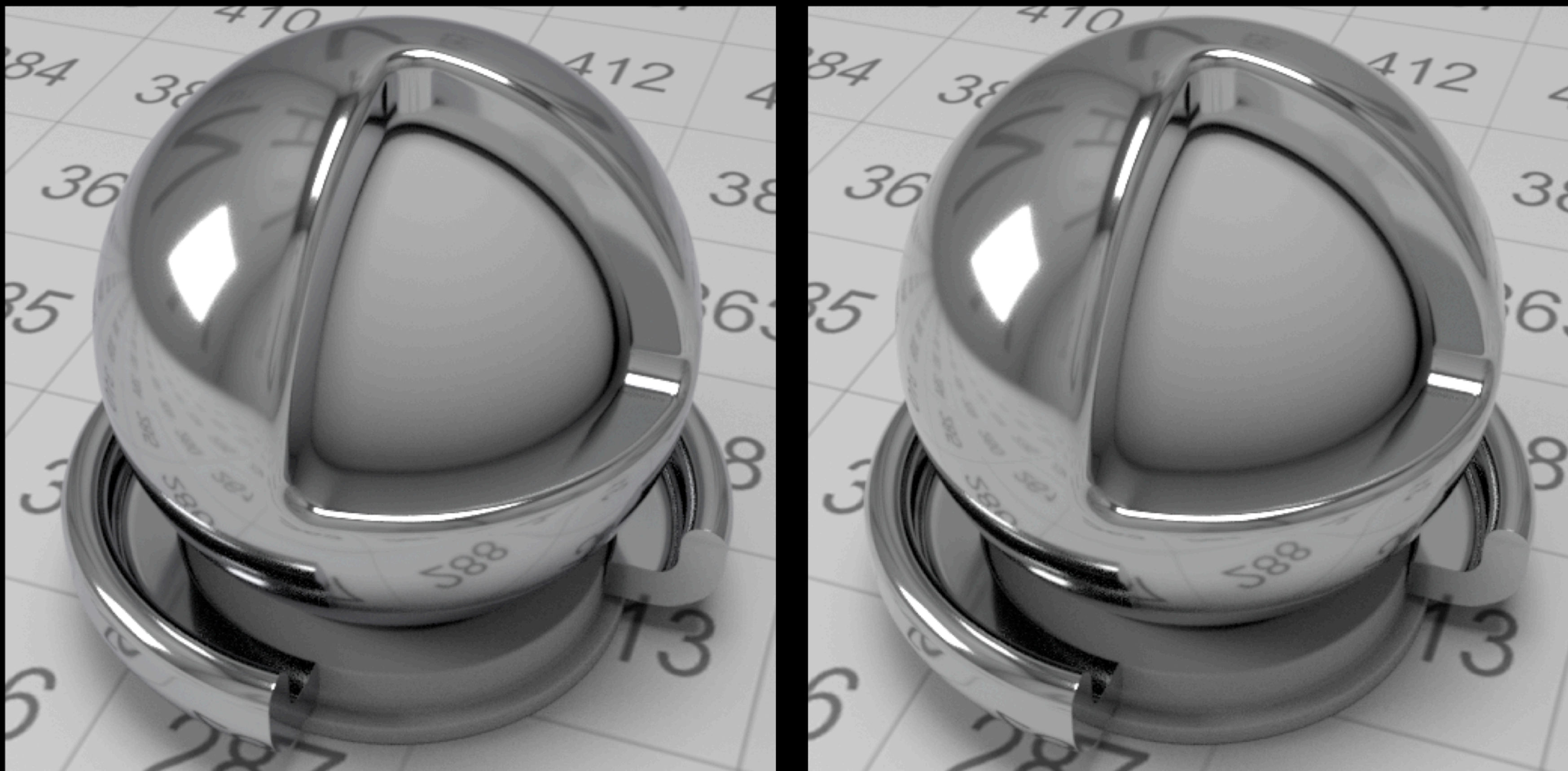
You may recall that these three materials* ranged from medium to high error for the Schlick approximation. Let's compare renders using ground-truth curves…

* The glass is opaque since we're focusing on reflection — we'll cover transmission in later slides.



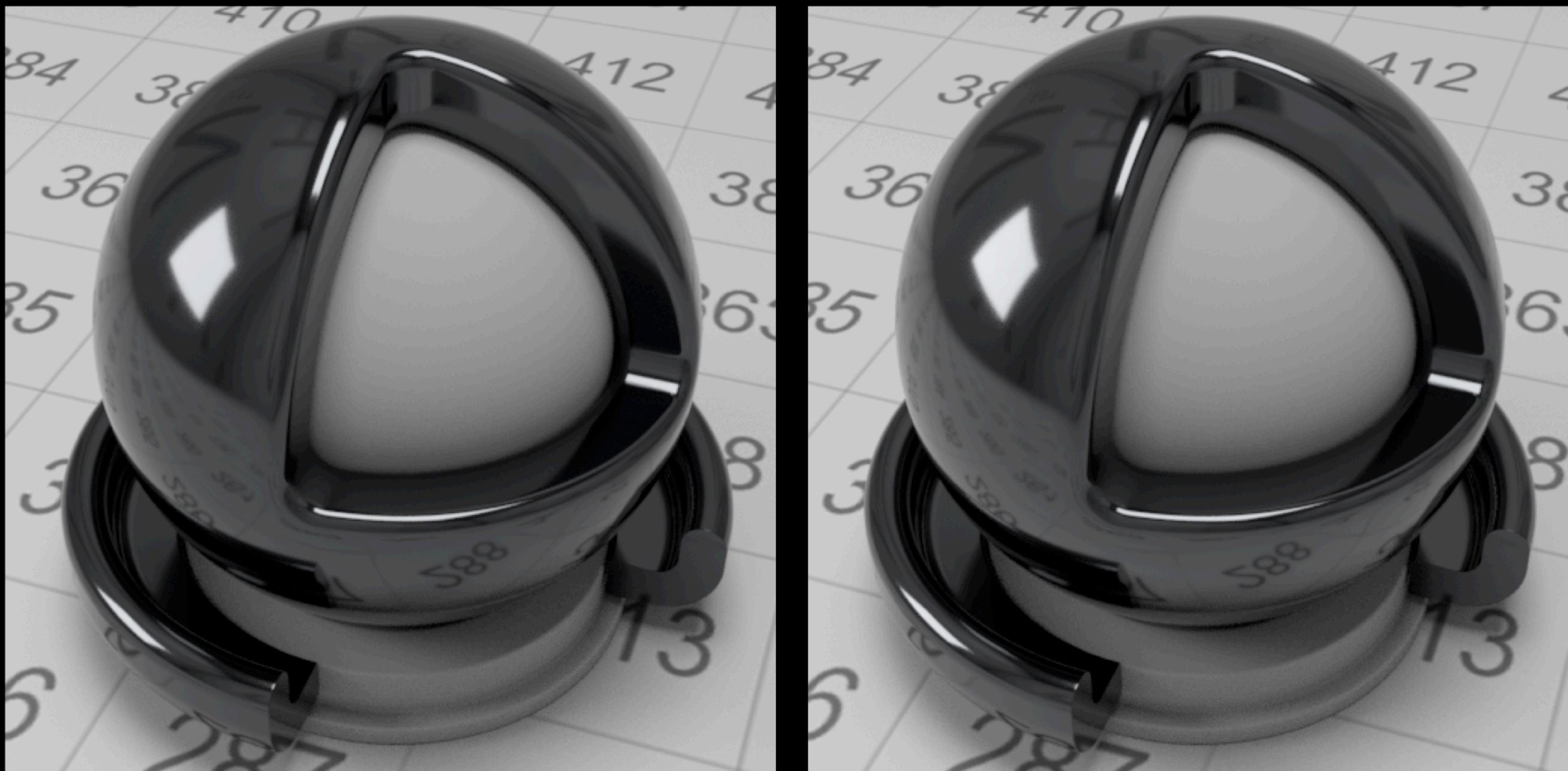
…with renders using Schlick. **<u>Flipping between them</u>** shows differences that are less blatant than we would expect from looking at the plots. But I would argue that even flipping isn't the most relevant test. Flipping between images is useful to zero in on small differences when authoring a look. But if we are worried that a model isn't accurate enough, it's worth asking if we can even tell the difference…

Chromium

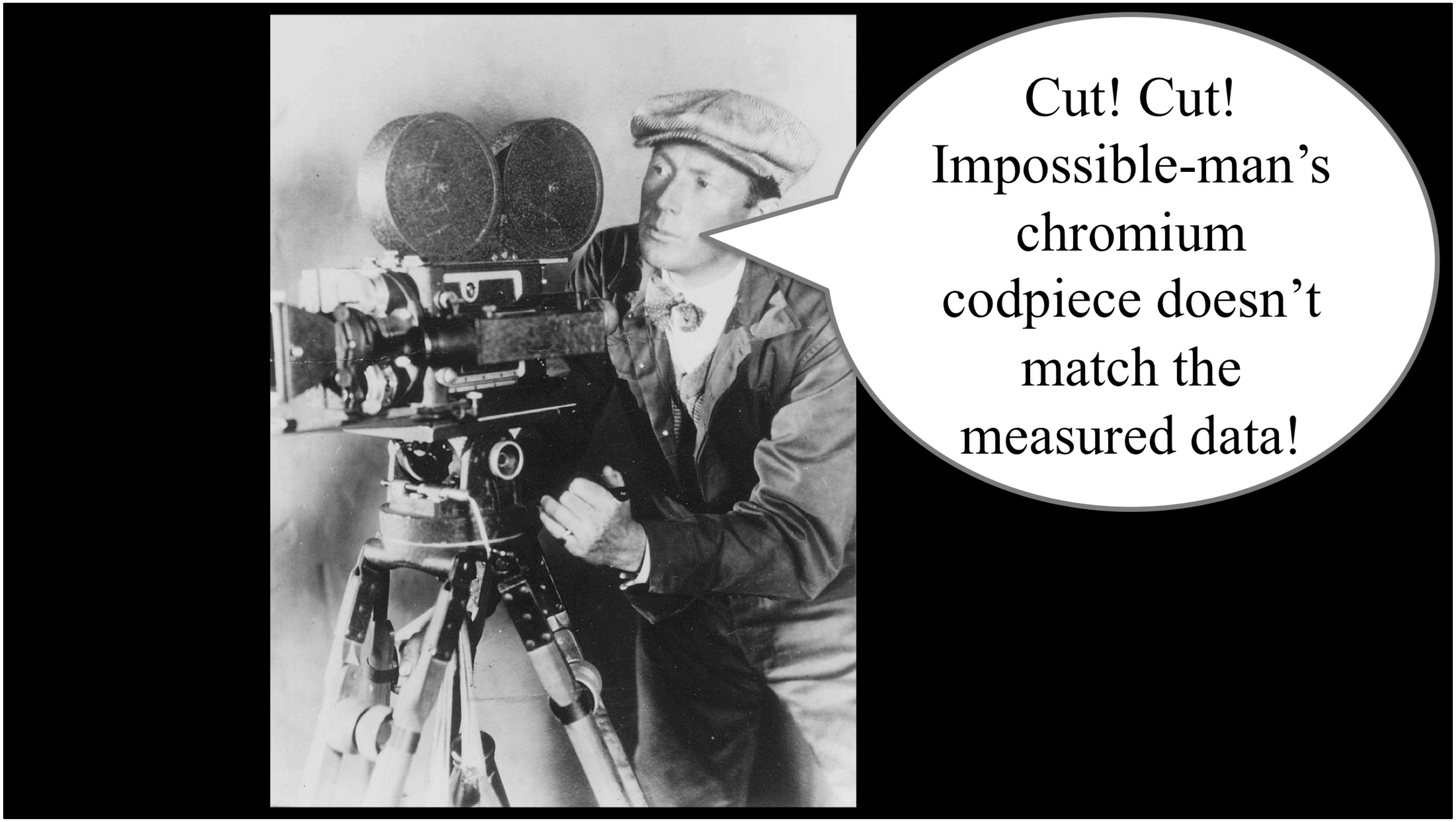


…when looking at them side by side. Can you honestly say that one of these looks more realistic than the other? Remember that Chromium is the material with the worst error for Schlick. If you are wondering, the ground-truth image is the one on the left.

Glass SF11



Here is a similar comparison for glass. Differences that were noticeable when flipping back and forth are hard to see side by side. Definitely, neither one is clearly more accurate. By the way, here the ground-truth image is on the right. Such errors might be worrisome for predictive rendering applications, but not for movies or games.

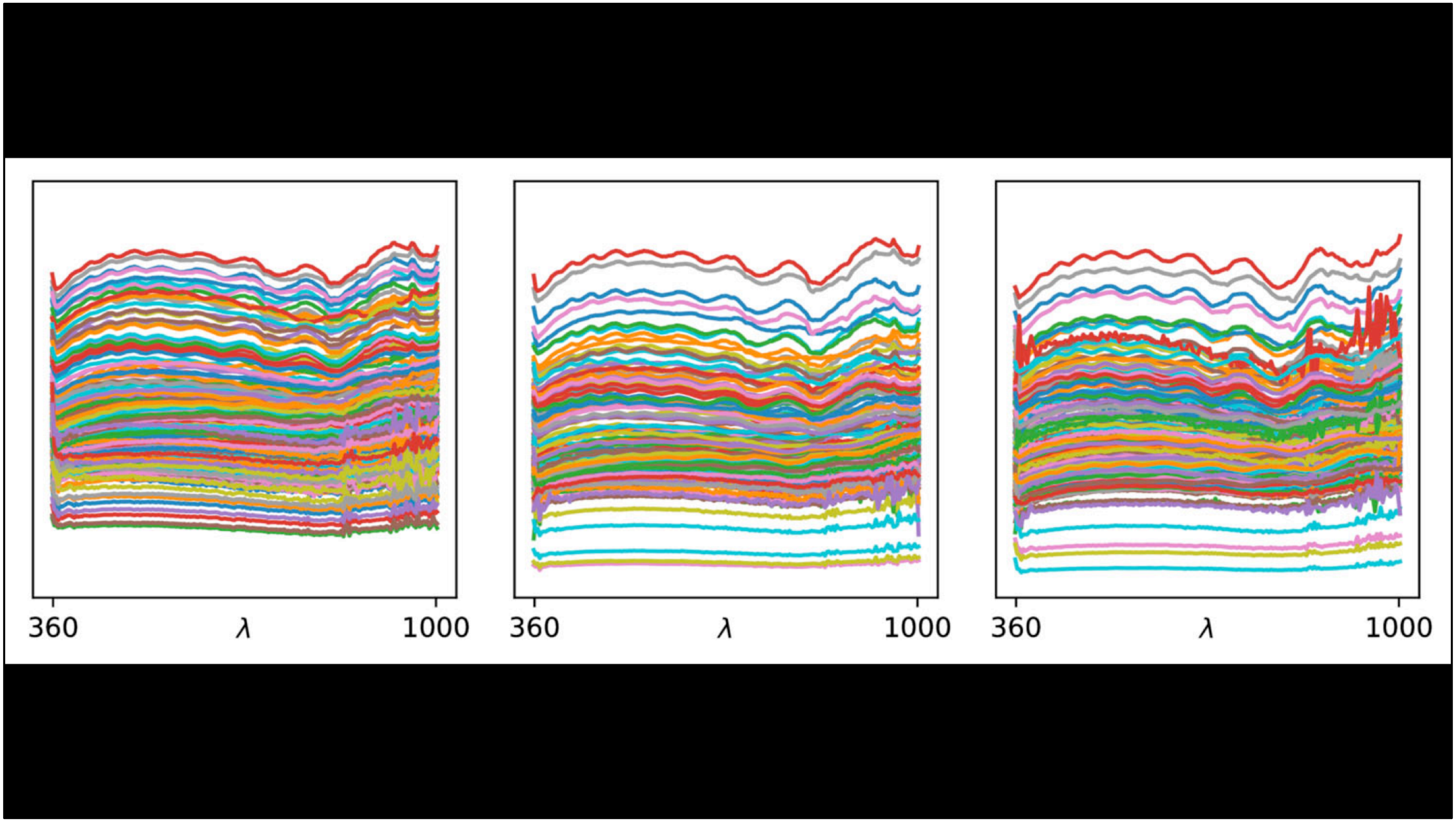


Most of us do entertainment rendering, not predictive simulations. Yes, photorealism is important, but it's easy to become overly enamored with the data on refractiveindex.info — I'm as guilty as anyone. Remember that the measured data is for 100% clean and pure laboratory-grade materials, in contrast to production CGI materials, which are messy, dirty, oxidized, and heavily artistically tweaked. Obsessing over a few percent closer match to curves calculated from such data is counterproductive.

(Image credit: "F. W. Murnau shooting a film in 1920", unknown photographer, public domain.)

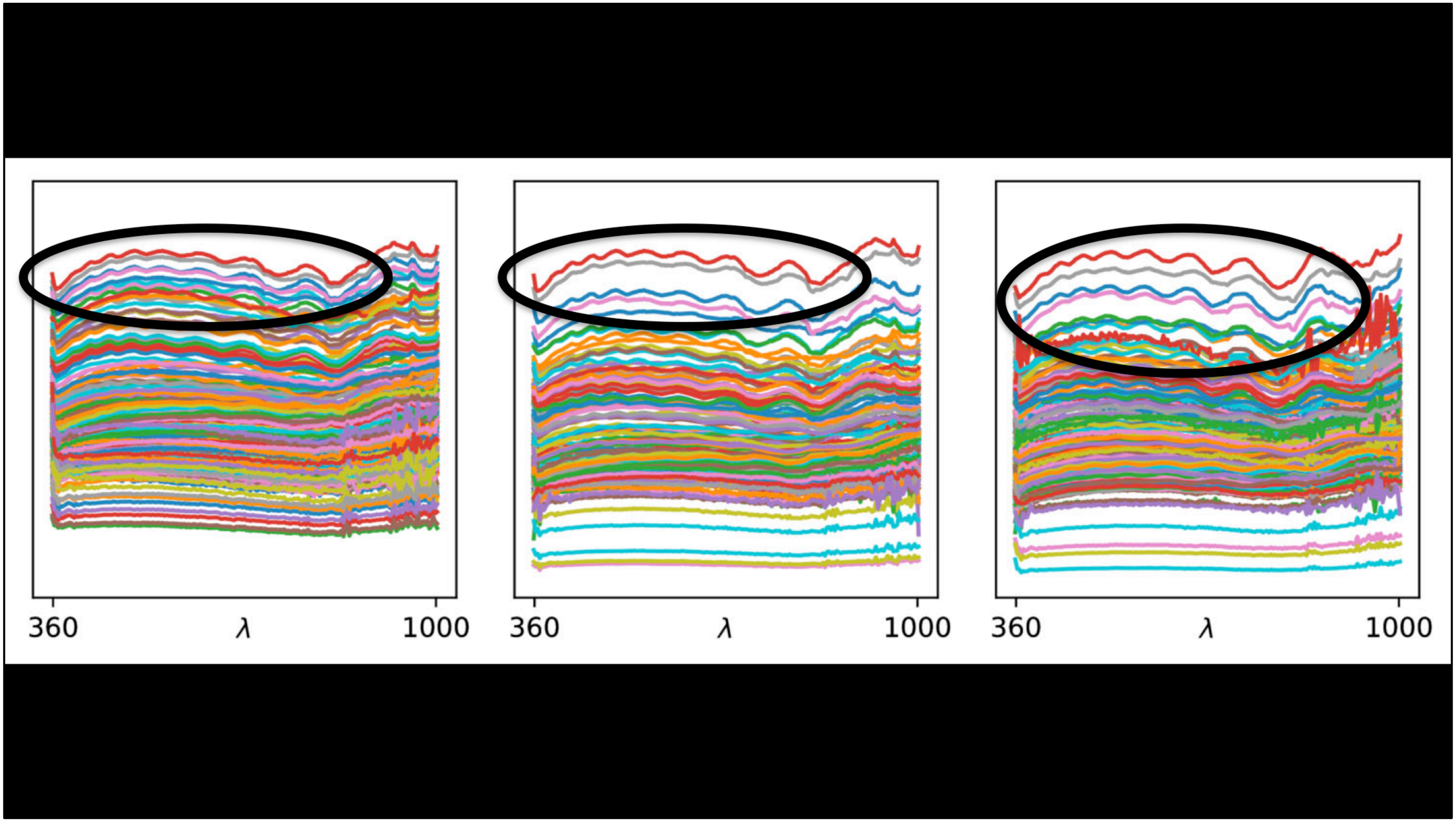
Heck, even when we need to match on-set materials, those materials are often not what they would seem. The L3-37 droid in *Solo: A Star Wars Story* was played by an actress in a practical costume, which the CG render needed to closely match.

(Image credit: “ILM Spotlight: Creating L3-37 for Solo: A Star Wars Story”, used by permission of Lucasfilm Ltd.)



But when the reference material swatches for L3's costume were scanned by Wenzel Jakob for the EPFL material database, the results were surprising. These reflectance spectra are from the metallic material swatch, which in the film appears to be an ordinary unpainted metal, perhaps steel.

(Image credit: from the supplementary material of "An Adaptive Parameterization for Efficient Material Acquisition and Rendering" by Jonathan Dupuy and Wenzel Jakob, SIGGRAPH Asia 2018, used under CC0 license.)



But these wiggles aren't like any metal that ever existed. It turns out that the metallic areas of the costume were actually multiple thin layers of metallic paint, producing an interference effect, which in turn produced these anomalous wiggles.

We must remember that the goal of production rendering is artistic expression, to which physical accuracy must always be subservient. For example it would be valid to complain that Schlick isn't expressive enough for artists who need to control edge color. And then we could discuss if generalized Schlick F90, or reparameterized Lazányi F82 or g edge tint are the best way to address that. But that's a very different conversation than worrying about the "missing dip" in measured metals.

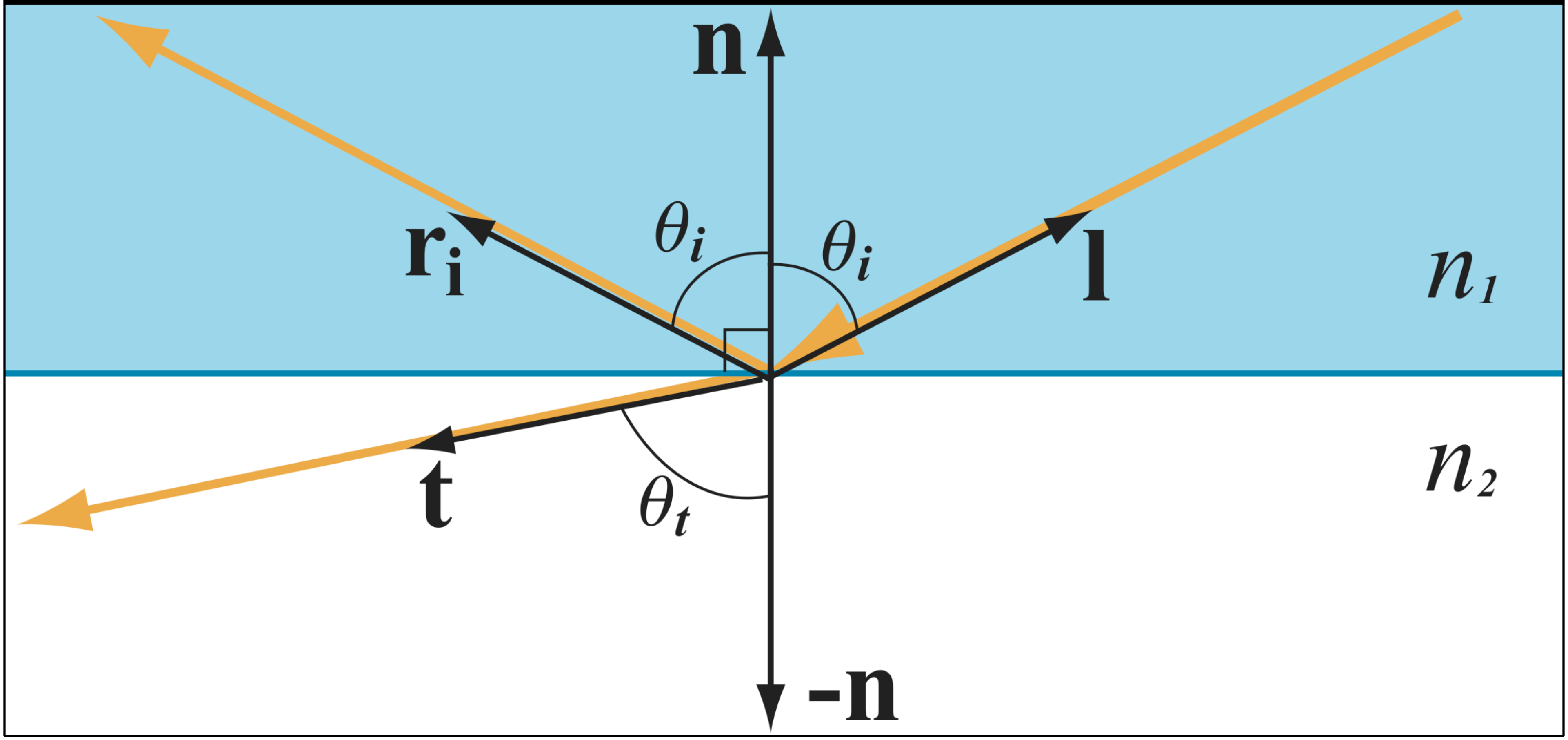
(Image credit: British Library digitized image from "The Merry Ballads of the Olden Time, illustrated in pictures & rhyme", public domain.)

Schlick Myth #2

"It can't be used for transparent dielectrics"

Another misconception is that Schlick can't handle transparent materials. Specifically, internal reflection.

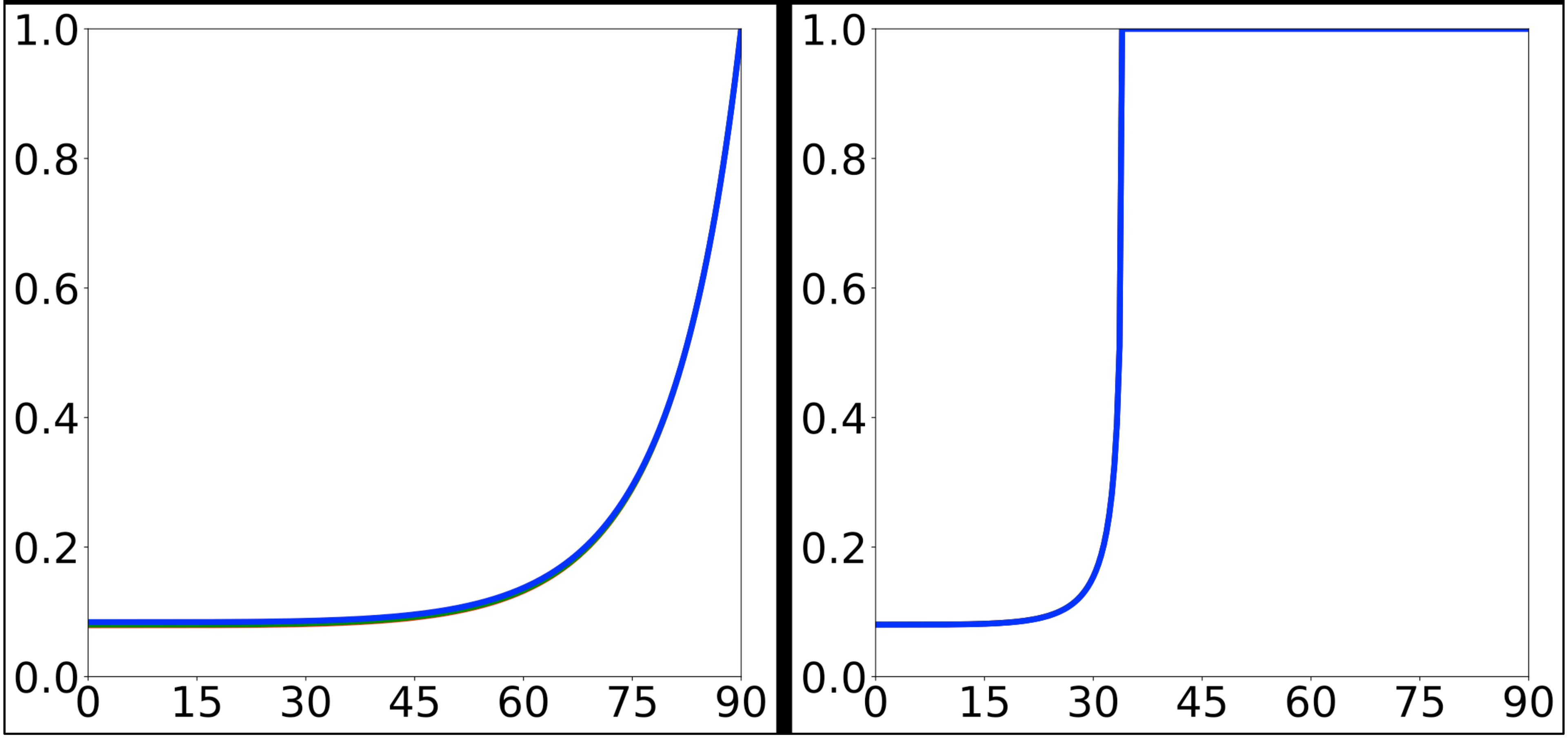
Internal Reflection



Internal reflection is when light reflects from “the inside” of an object’s surface. The Fresnel equations are symmetrical, with the same result if incoming and transmission vectors are swapped. And Snell’s law tells us that here the transmission angle is greater than the incidence angle. This implies that the Fresnel curve for internal reflection will resemble a “squeezed” version of the curve for external reflection, with the same F0.

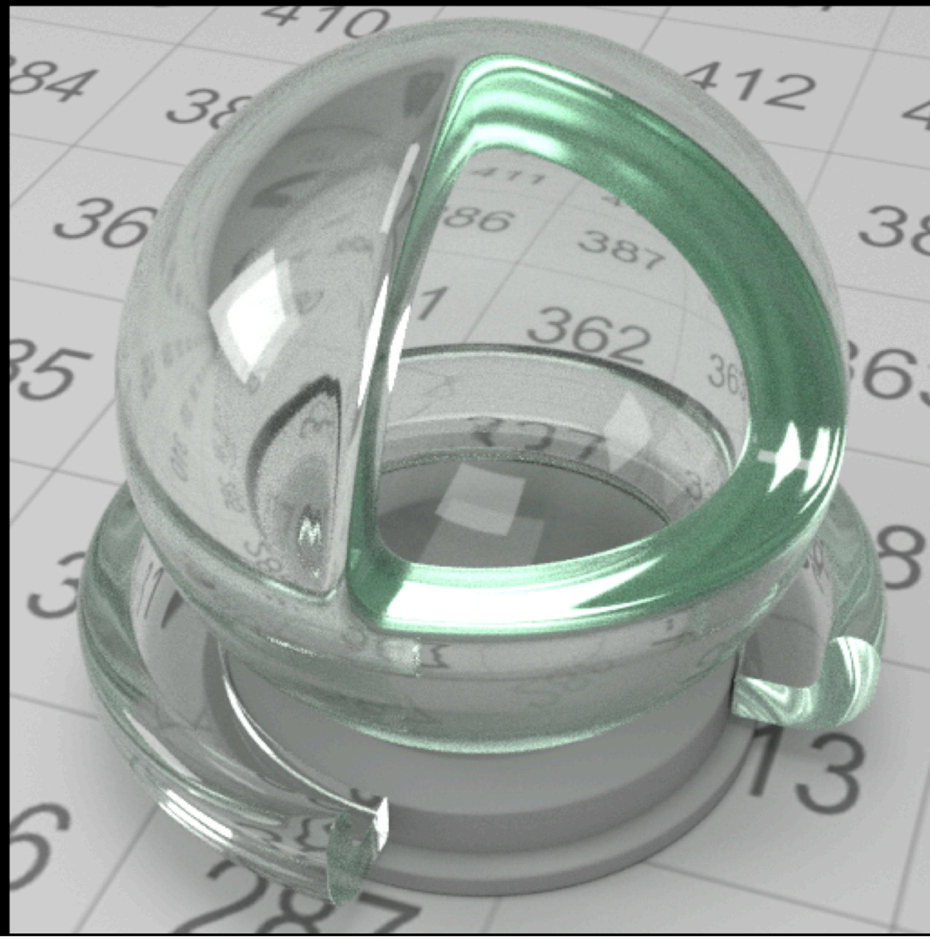
(Image credit: Real-Time Rendering, 4th Edition, CRC Press.)

Internal Reflection



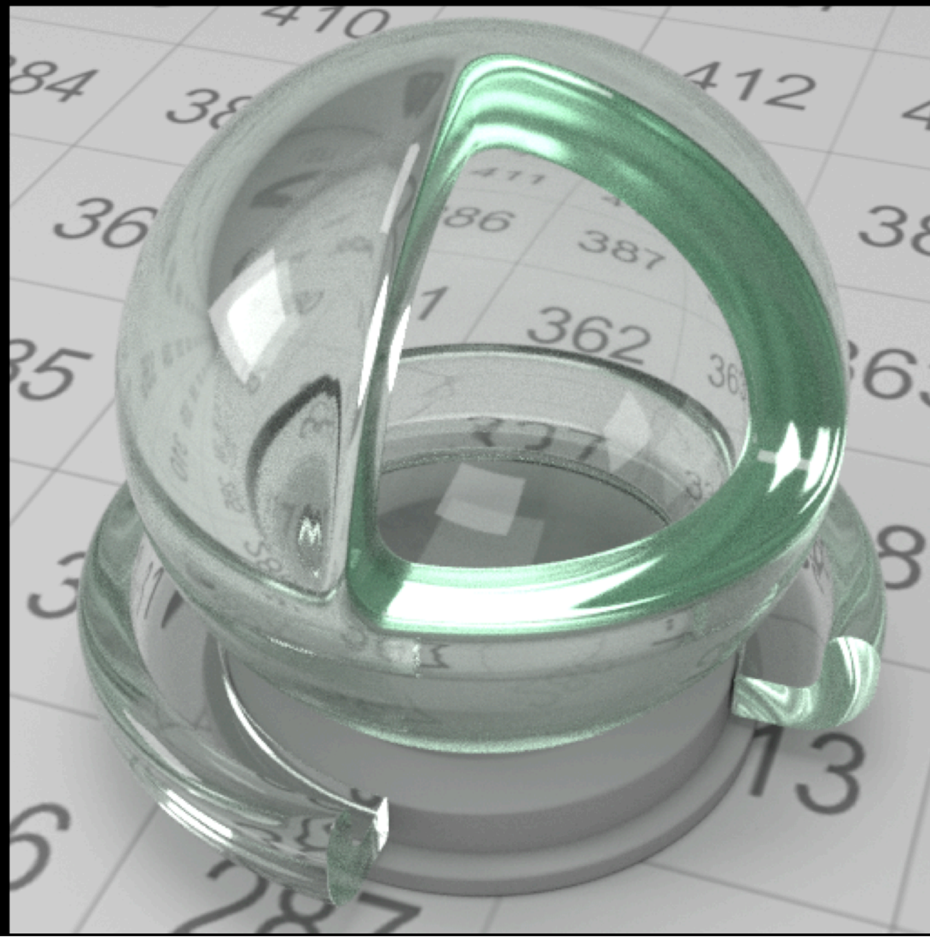
We see that the the internal reflection Fresnel plot for Glass SF11 on the right, indeed resembles a “squeezed” version of the external reflection plot on the left. This suggests that the Schlick approximation can be applied to internal reflection by simply using the transmission angle instead of the incidence angle. When computing the refraction vector this angle is easily available, otherwise it can be computed from eta (which can itself be inferred from F0).

Ground Truth

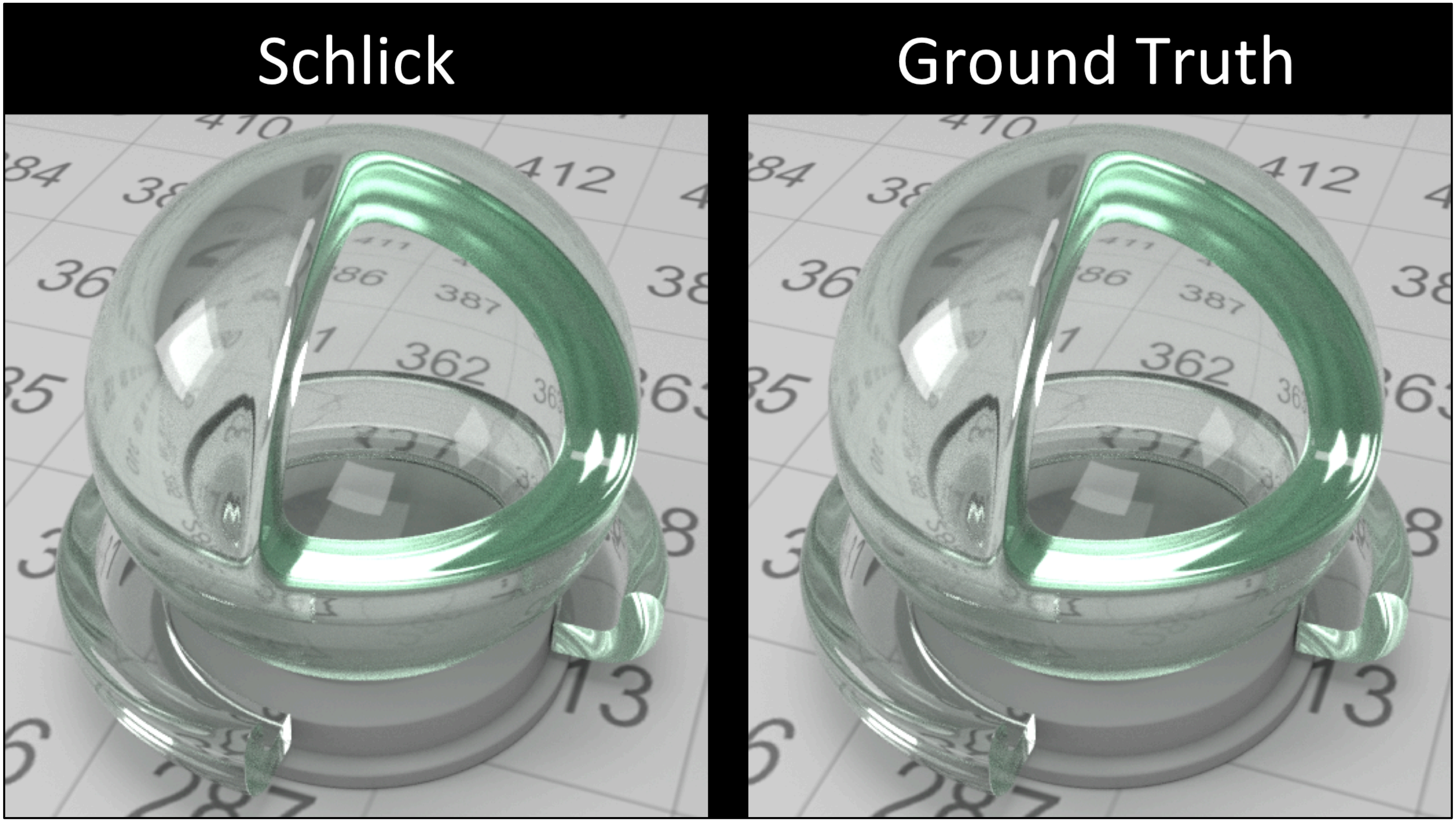


Let's compare a transparent glass render using the Fresnel equations for internal and external reflection…

Schlick



…with one using Schlick for both. When we **flip between them**, we see that the difference is rather subtle…



…and is hard to see at all side by side. Neither render is more realistic than the other.

Schlick Myth #3

"It can't handle the effect of external media"

Another case that Schlick is often assumed not to handle is the effect of external media. Effective IOR — and thus the Fresnel curve — changes when a material is immersed in a medium like water. This case is also important for layered materials. When using the Fresnel equations, adjusting for external media is simply a matter of dividing eta and kappa by the eta of the external medium. It's not obvious how to perform this adjustment with a model like Schlick that has no concept of eta and kappa, so it is often assumed to be impossible.

External Media Adjustment

$$F_0' = \left(\frac{\sqrt{F_0} - \sqrt{F_{0\mathrm{E}}}}{1 - \sqrt{F_0}\sqrt{F_{0\mathrm{E}}}} \right)^2$$

The adjustment is straightforward for dielectrics. Here F0 is the original reflectance, F0E is the reflectance of the external medium, and F0-prime is the adjusted reflectance. Once F0 is adjusted, the overall Schlick curve compares reasonably well to the ground truth.

External Media Adjustment

$$F_0' = \left(\frac{\sqrt{F_0} - \sqrt{F_{0\mathrm{E}}}}{1 - \sqrt{F_0}\sqrt{F_{0\mathrm{E}}}} \right)^2$$

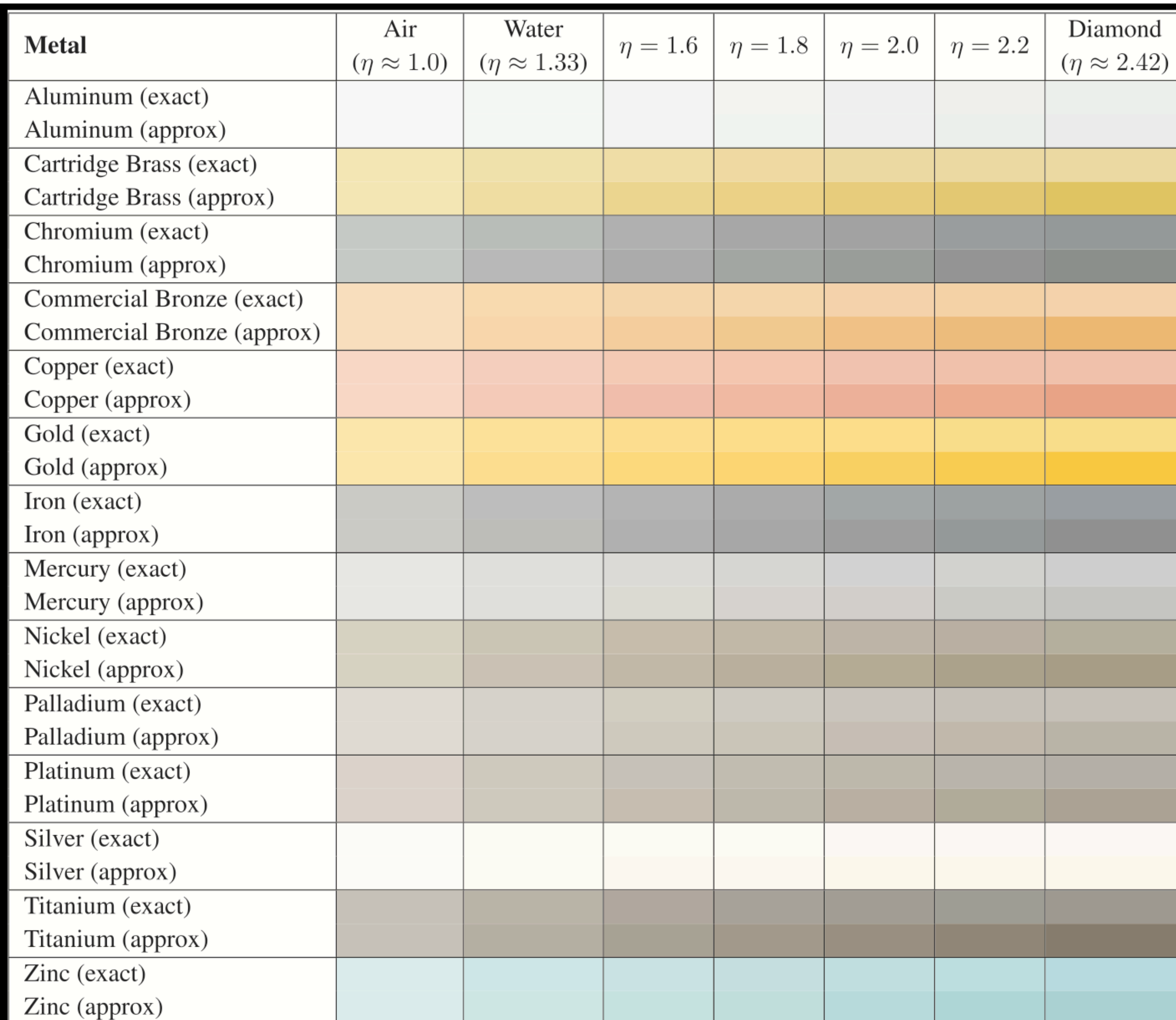
$$\sqrt{F_{0\mathrm{E}}} = \frac{\eta_E - 1}{\eta_E + 1}$$

Often the IOR of the external medium is specified instead of its reflectance, but it's easy to compute one from the other (and you even save a square root!)

External Media Adjustment

?

But what about metals? The external media adjustment depends on both eta and kappa, so it cannot be inferred just from F0.



It turns out that just applying the dielectric adjustment to metals gives pretty good results, especially for low-IOR external media which are the most common case. With high IOR media like diamond then some metals do get overly dark colors.

External Media Adjustment

$$F_0' = \begin{cases} \left(\sqrt{F_0} - \sqrt{F_{0\mathrm{E}}}\right)^2, & \text{if } F_0 < F_{0\mathrm{E}} \\ \left(\frac{\sqrt{F_0} - \sqrt{F_{0\mathrm{E}}}}{1 - \sqrt{F_{0\mathrm{E}}}}\right)^2, & \text{otherwise} \end{cases}$$

If you need accurate renders for metals embedded in diamond, then accuracy for the high-IOR case can be improved by tweaking the formula slightly, at the cost of of making it asymmetrical (switching internal and external IOR no longer gives the same result).

Metal	Air ($\eta \approx 1.0$)	Water ($\eta \approx 1.33$)	$\eta = 1.6$	$\eta = 1.8$	$\eta = 2.0$	$\eta = 2.2$	Diamond ($\eta \approx 2.42$)
Aluminum (exact)							
Aluminum (approx)							
Cartridge Brass (exact)							
Cartridge Brass (approx)							
Chromium (exact)							
Chromium (approx)							
Commercial Bronze (exact)							
Commercial Bronze (approx)							
Copper (exact)							
Copper (approx)							
Gold (exact)							
Gold (approx)							
Iron (exact)							
Iron (approx)							
Mercury (exact)							
Mercury (approx)							
Nickel (exact)							
Nickel (approx)							
Palladium (exact)							
Palladium (approx)							
Platinum (exact)							
Platinum (approx)							
Silver (exact)							
Silver (approx)							
Titanium (exact)							
Titanium (approx)							
Zinc (exact)							
Zinc (approx)							

Here are the results for the tweaked formula.

g edge tint

Now let's turn to discussing the g edge tint parameter, introduced in the paper "Artist Friendly Metallic Fresnel". Each parameter that an artist needs to paint is an added burden, so there should be a commensurate benefit. If the parameter can be left at a default value like the generalized Schlick F90 parameter, then it's harmless. But as we shall see, this isn't the case for g. Furthermore…

g edge tint
is not artist friendly

…despite the title of the paper, g is actually not artist friendly.

g edge tint
is not artist friendly
at all

Really, not at all. Let’s discuss why.

What makes a parameter artist friendly?

First, let's define what makes a parameter artist friendly.

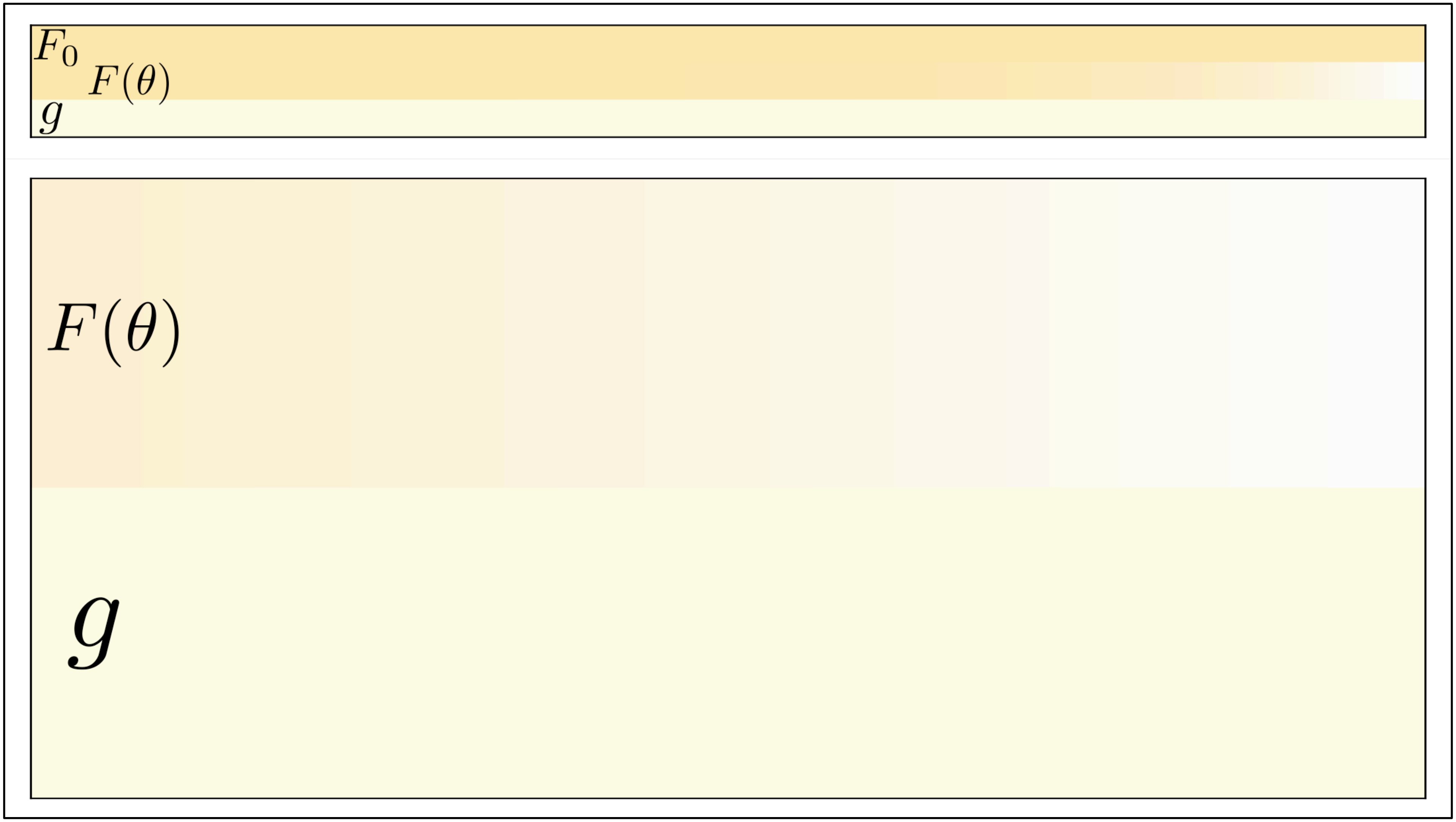
What makes a parameter artist friendly?

- Intuitive relationship between parameter value and surface appearance
  - If a color, that color should be visible on the surface

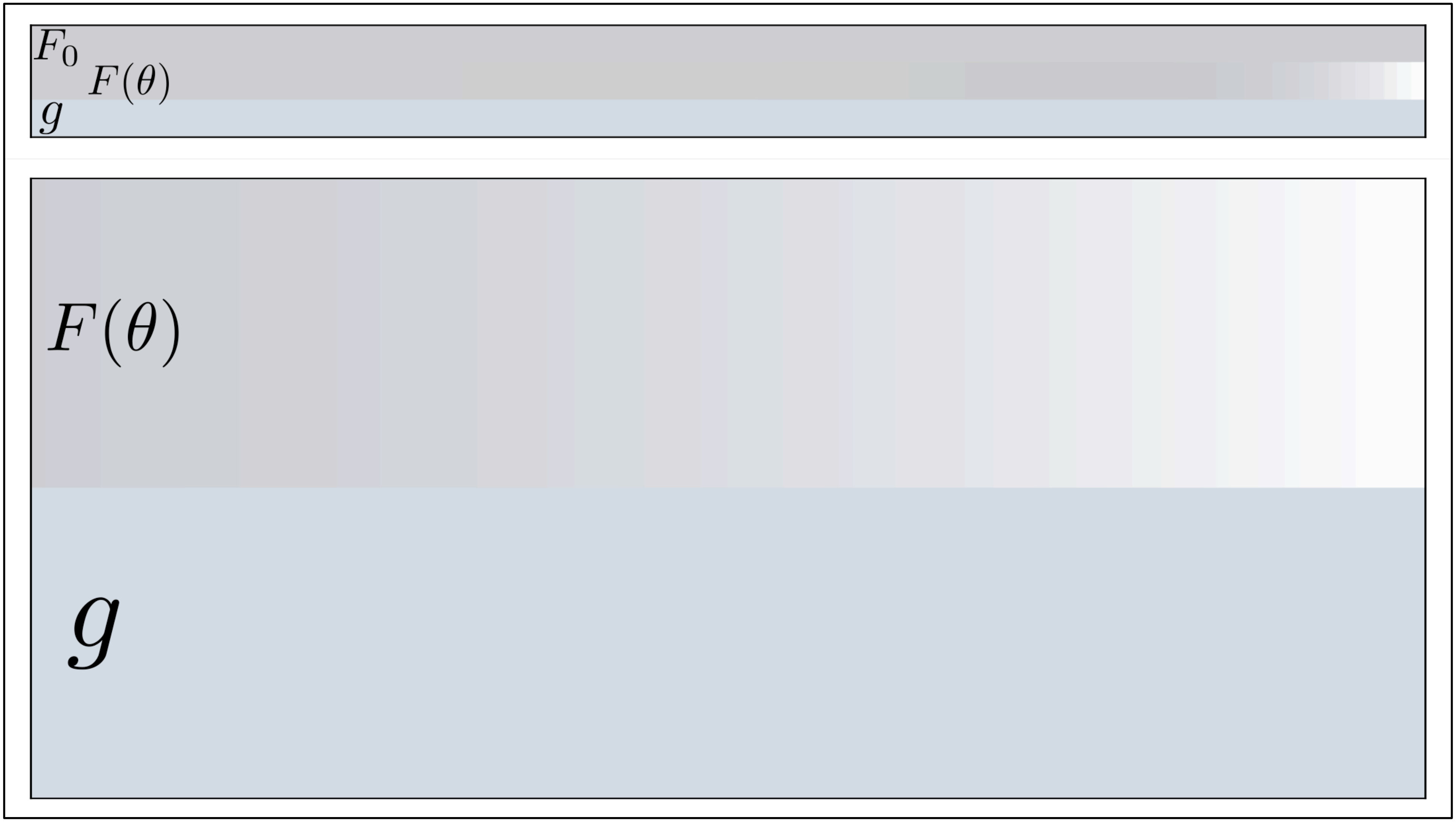
Parameter values must have an intuitive relationship to surface appearance. For a color parameter, that color should be visible on the surface. Diffuse albedo and F0 are good examples.

Unlike F0, the edge tint g is not guaranteed to be equal to the reflected color at <u>any</u> angle. I will demonstrate this with some color strips for a few metals. Each metal has three stacked strips: the middle one shows the ground-truth Fresnel color from 0 degrees on the left side to 90 degrees on the right. The top and bottom strips each have a solid color: F0 and g respectively. Note that the top and middle strips match closely over the left half of the strips, which shows why F0 is a good parameter for painting.

These strips tend to have a lot of variation close to 90 degrees, so I'll add a second figure zooming in on the last 10 degrees of the range, with the top strip removed since F0 isn't relevant for this angle range. These strips are for Chromium — we see that the g strip on the bottom doesn't match the Fresnel color (middle strip) at any angle; it has a colder hue.



This one is for Gold. We see that g has a lemony hue, which doesn't resemble the actual reflected color of gold at any angle.



And finally iron. Again the hue and saturation of g is different than the actual reflected colors. From these examples it is clear that the relationship between the value of g and surface appearance is not at all intuitive.

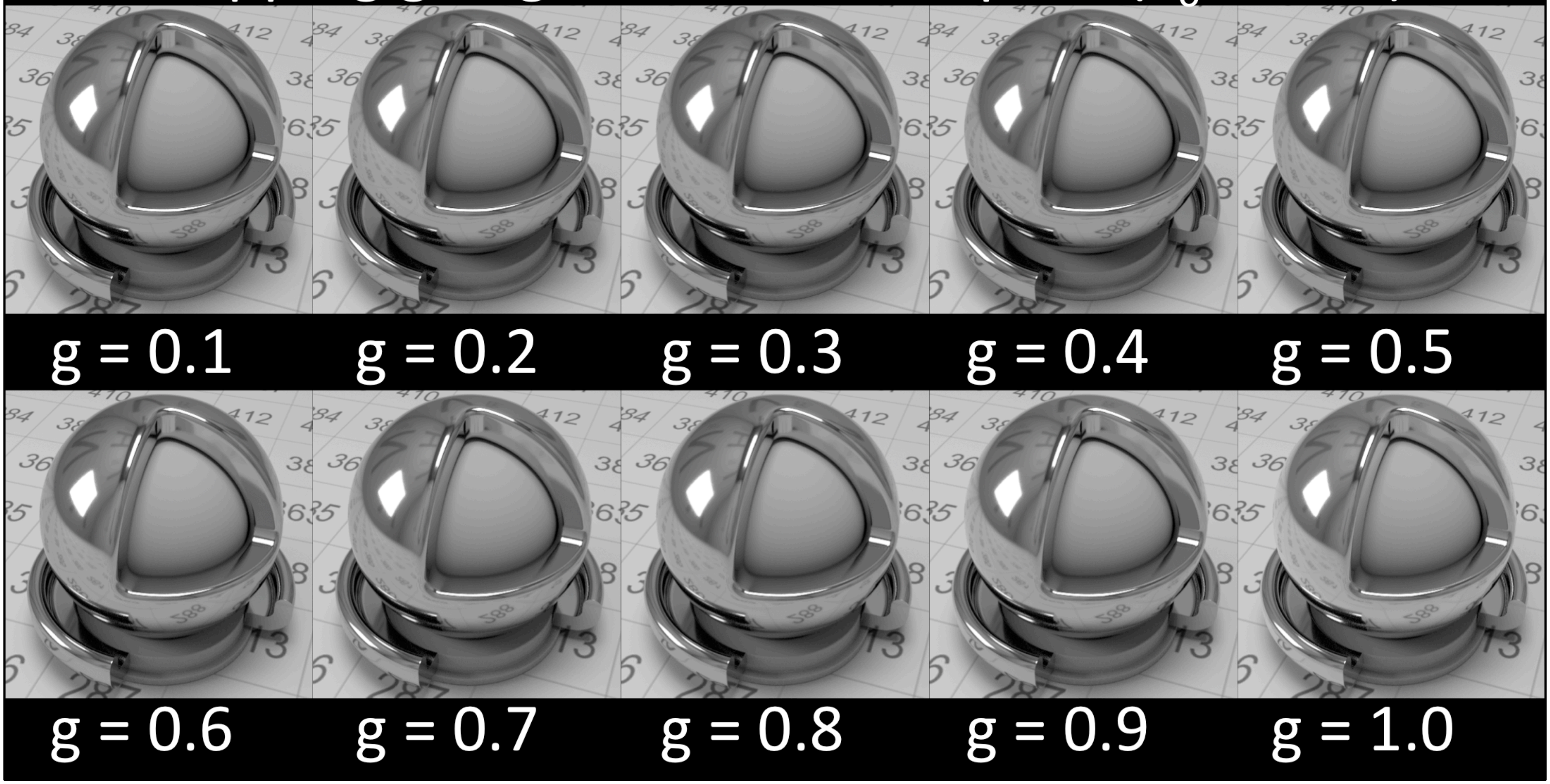
What makes a parameter artist friendly?

- Intuitive relationship between parameter value and surface appearance
  - If a color, that color should be visible on the surface
- Perceptually uniform effect over its range

The visual effect of changing a parameter by a certain amount should be roughly constant over the parameter range. A good example is GGX roughness with the commonly used remapping first suggested by Brent Burley*.

* B. Burley, Physically Based Shading at Disney. Practical Physically Based Shading in Film and Game Production, SIGGRAPH Course, 2012.

Stepping g edge tint in linear space ($F_0 = 0.50$)



This is not the case for g. These renders increase g in equal-sized steps across the 0 to 1 range. It's hard to see the difference between consecutive values for side-by-side images…

…so we'll show some pairwise flip comparisons. Here is a render with g set to 0…

…and here is g set to 0.1. **<u>Flipping between them</u>** shows literally no perceptible change.

Now let’s look at a pair closer to the middle of the range.

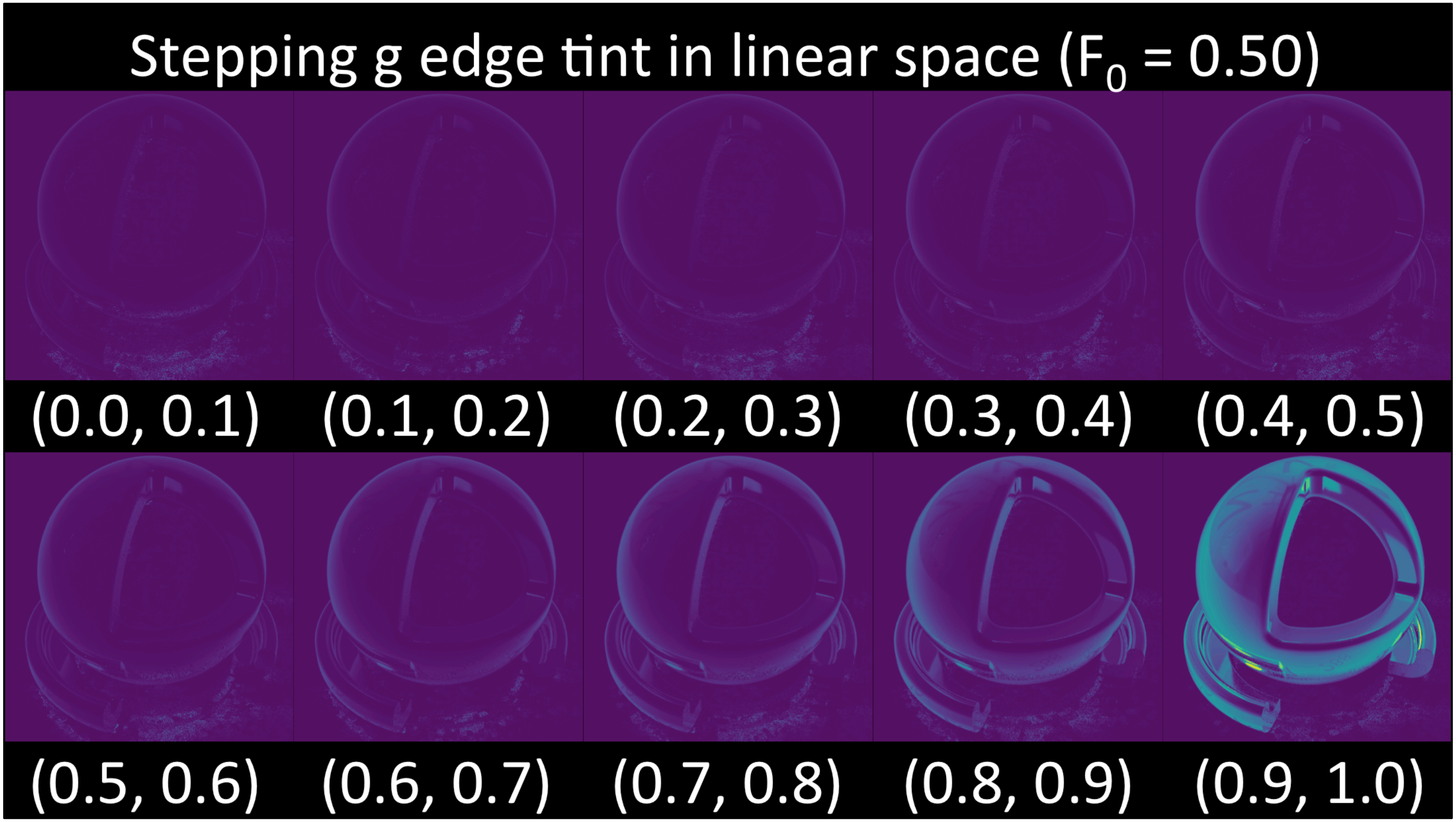
Here is a render with g set to 0.5…

…and 0.6. When we **<u>flip between this pair</u>** of images, there is a very subtle difference − you can see a bit of darkening on the lower left edge of the sphere if you look closely.

Now let’s look at a pair at the top of the range.

Here is a render with g set to 0.9…

…and to 1.0. When we **<u>flip between them</u>** we can see that the difference is quite noticeable. With an artist-friendly parameter, each of these pairs would have about the same visible difference.



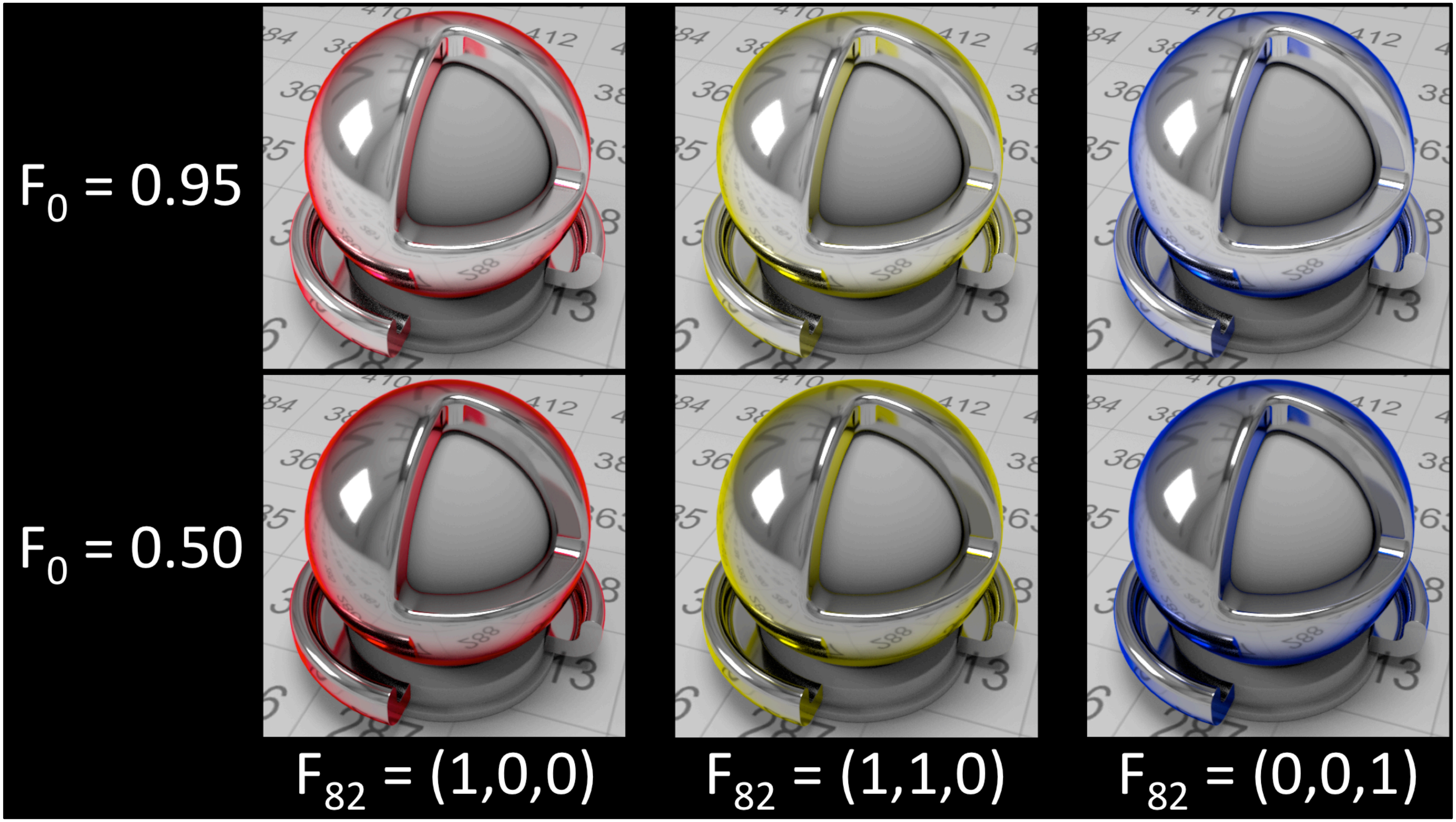
As an alternative visualization, here are color-mapped perceptual difference* images for each consecutive pair of renders. This matches the pairwise comparisons we've seen**.

* delta-E 2000.
** The issue is even worse when we consider that artists typically manipulate color values in a space such as sRGB. Here the steps are in linear space — if we did them in sRGB space the non-uniformity would be even larger.

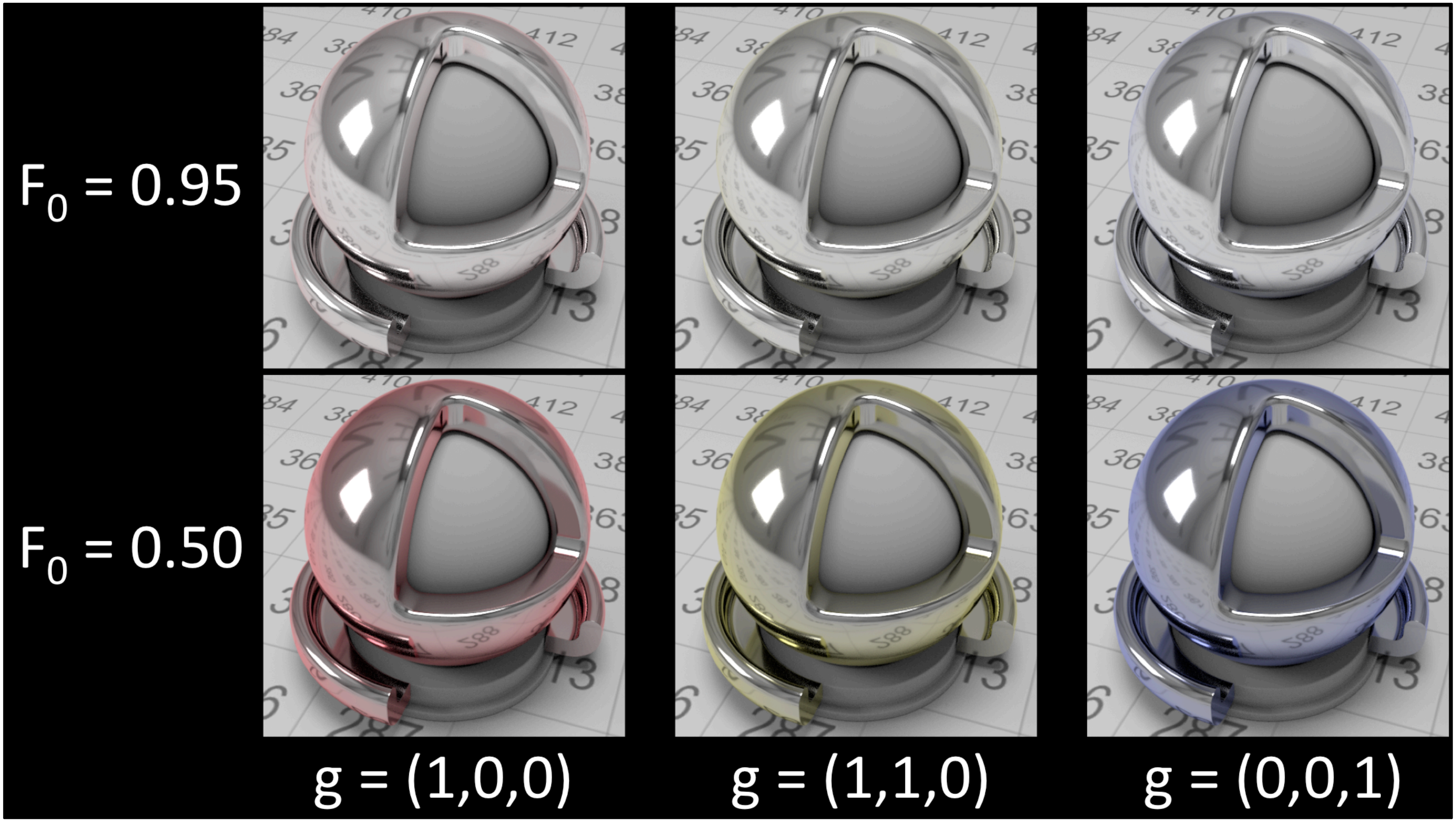
What makes a parameter artist friendly?

- Intuitive relationship between parameter value and surface appearance
  - If a color, that color should be visible on the surface
- Perceptually uniform effect over its range
- Orthogonal to other parameters

Ideally, parameters should be decoupled so that the visual effect of changing one does not depend on the value of another.



For example, the visual impact of the F82 parameter shown earlier is about the same regardless of whether the value of F0 is high or low.

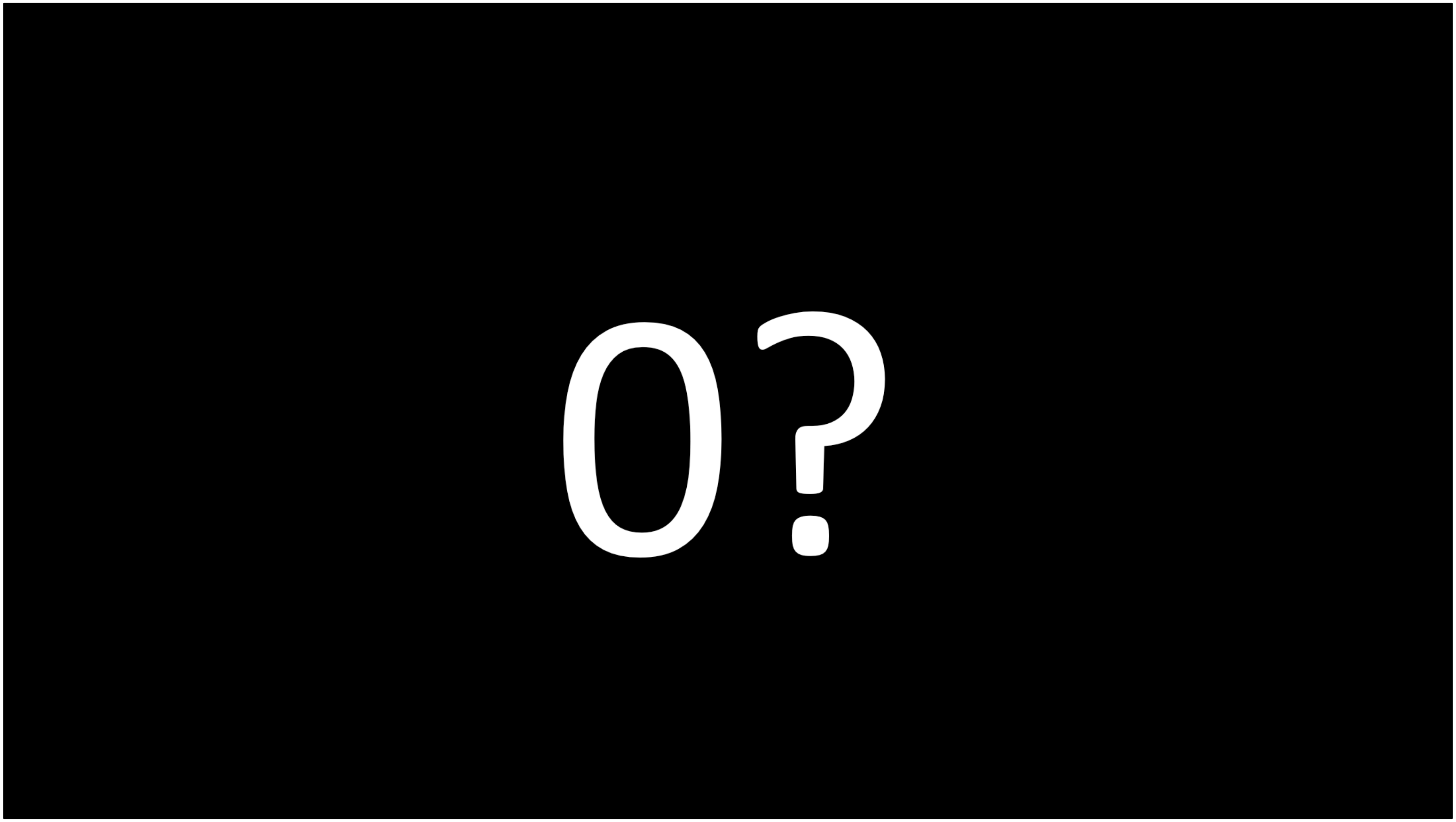


This is not the case for g. When the value of F0 is high, g barely has an effect. Its visual impact is much stronger when F0 is low.

What makes a parameter artist friendly?

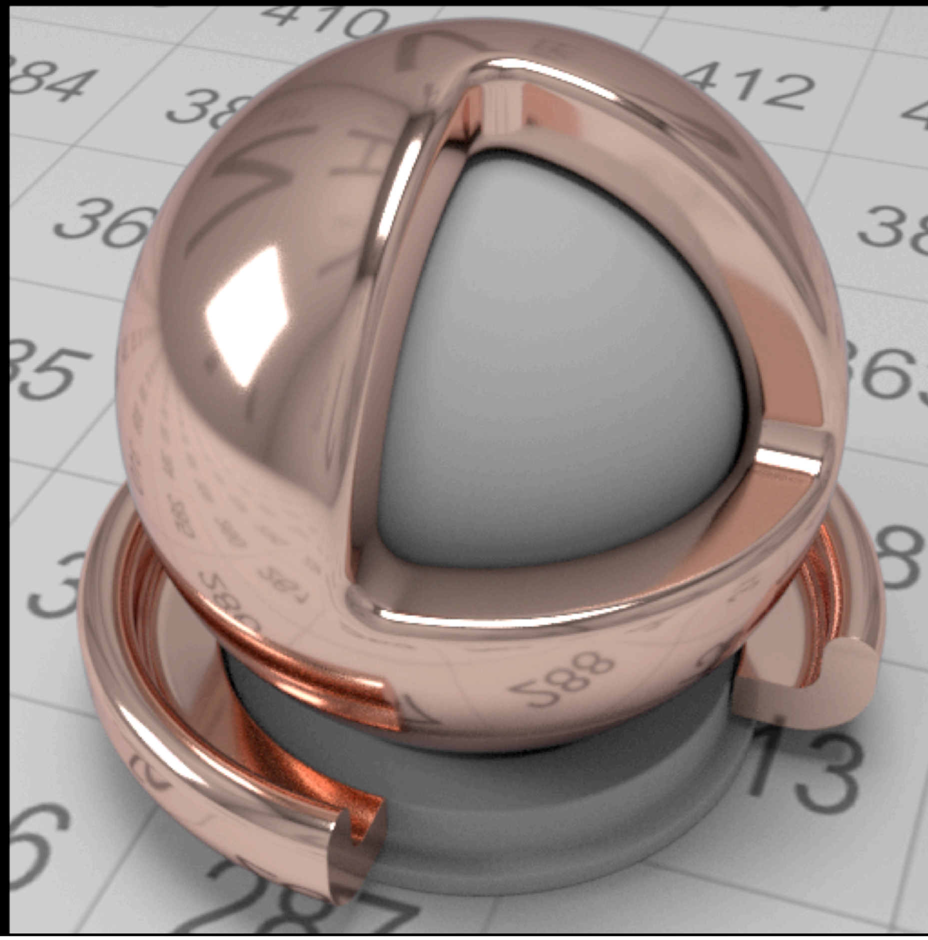
- Intuitive relationship between parameter value and surface appearance
  - If a color, that color should be visible on the surface
- Perceptually uniform effect over its range
- Orthogonal to other parameters
- Secondary parameters should have default values

A few primary parameters need to be set for every material: diffuse color and specular color, or base color and metalness, are good examples — specular roughness is another. In contrast, anisotropy and sheen are secondary parameters. It's clear that g is a secondary parameter, so it needs to have a default value that will yield a reasonable default appearance.



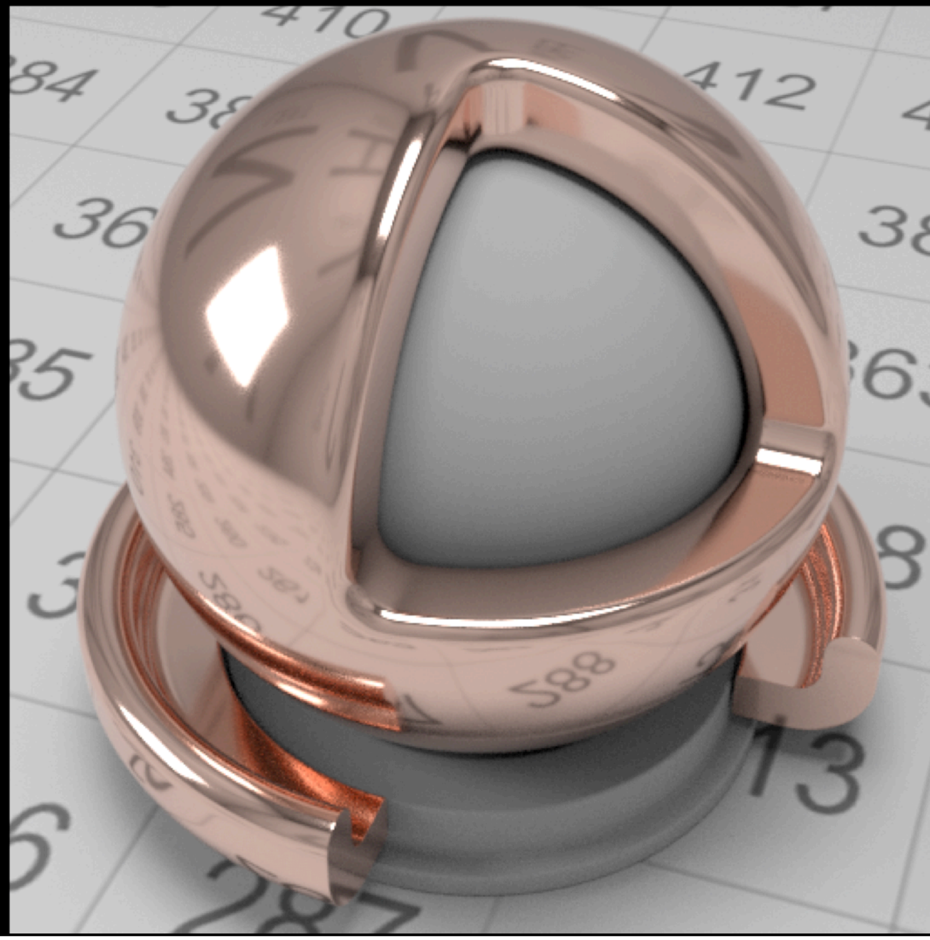
But such a default doesn’t exist for g. I sometimes hear that setting g to zero should yield reasonable default behavior. The thinking is that this is equivalent to setting kappa to zero, which implies a dielectric. And dielectrics have pretty reasonable visual behavior.

$g = 0$

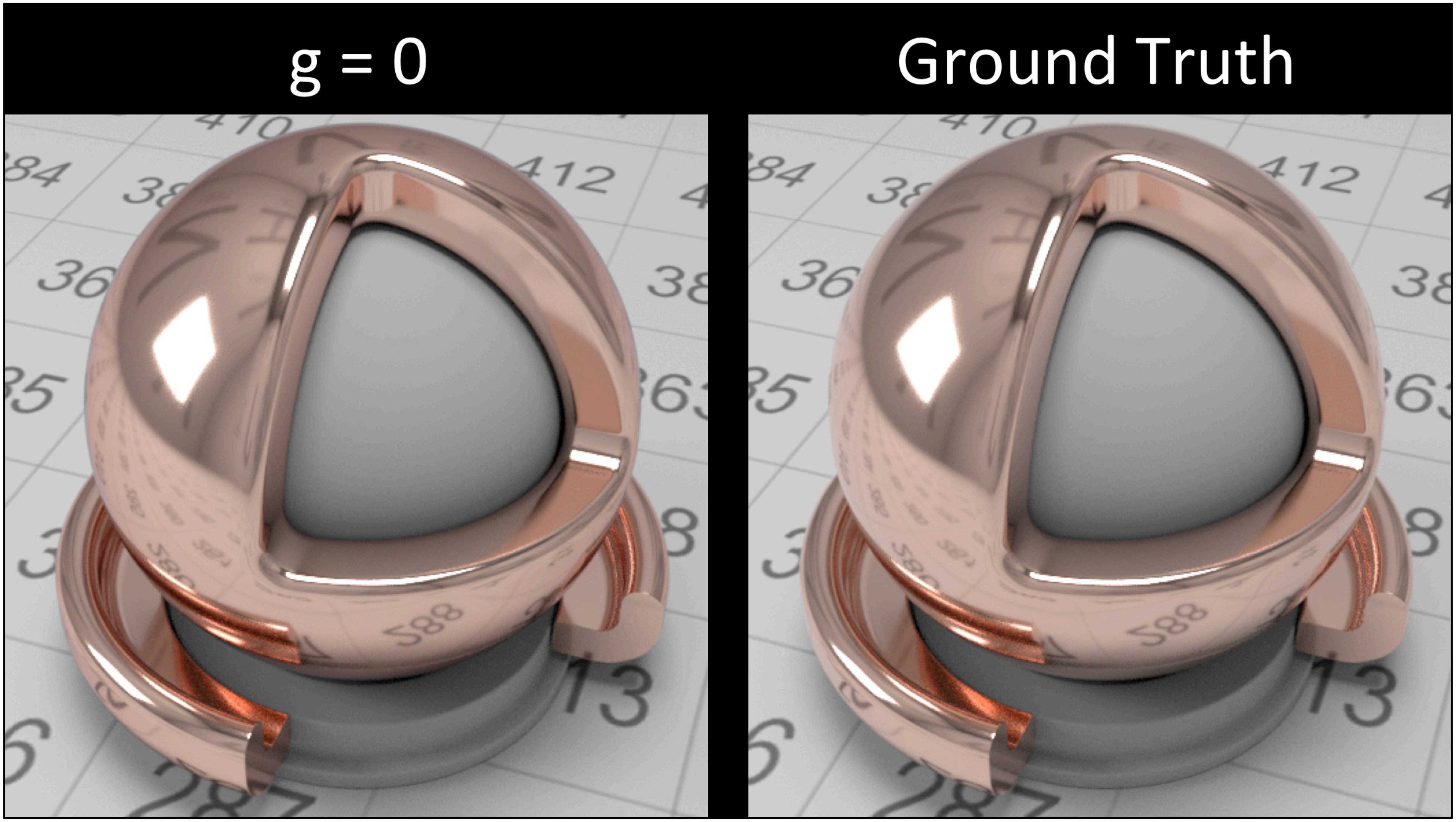


But setting g to zero doesn't get you something that looks like a dielectric, it results in an odd darkening and discoloration at the edge. Let's compare…

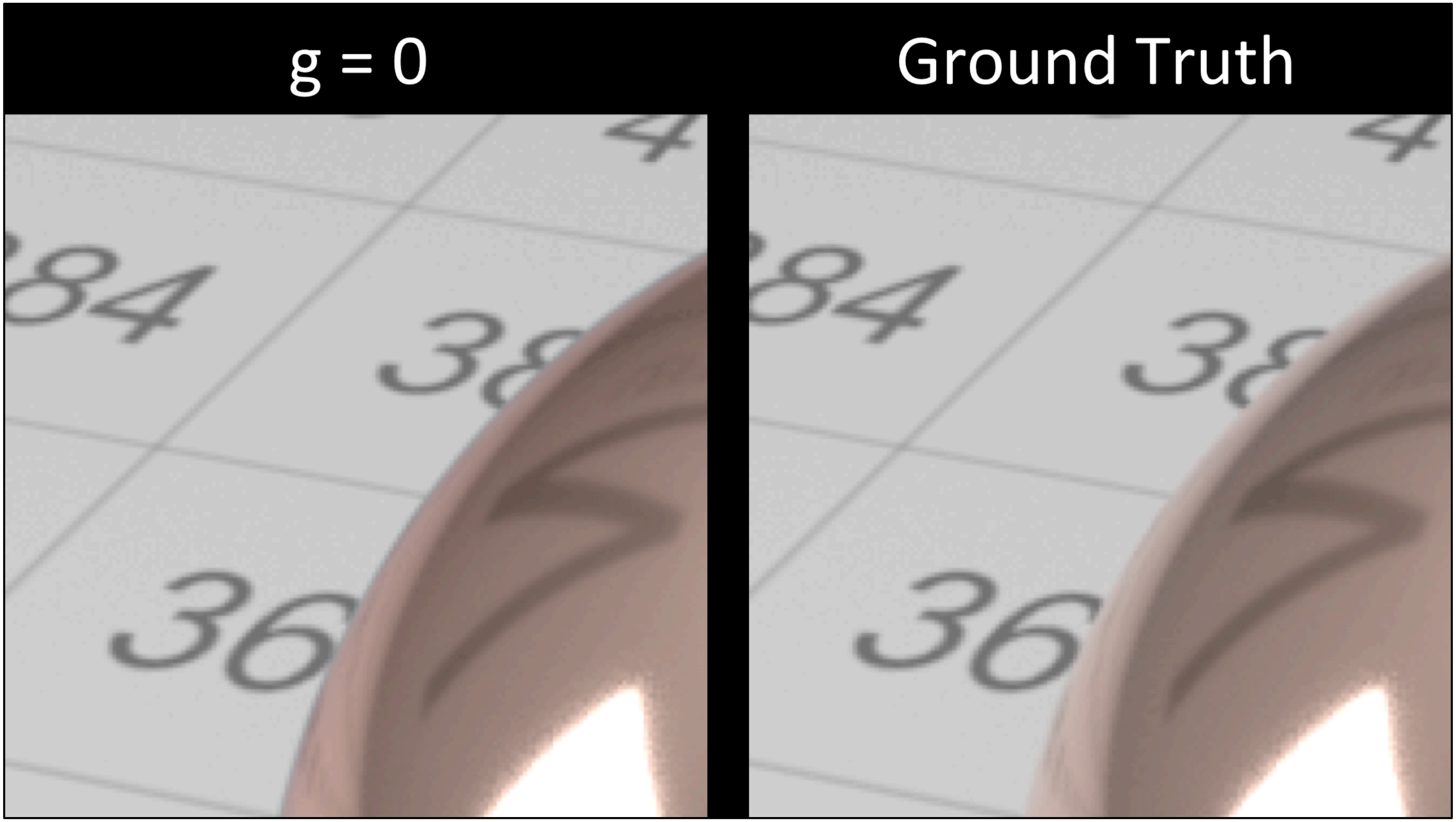
Ground Truth



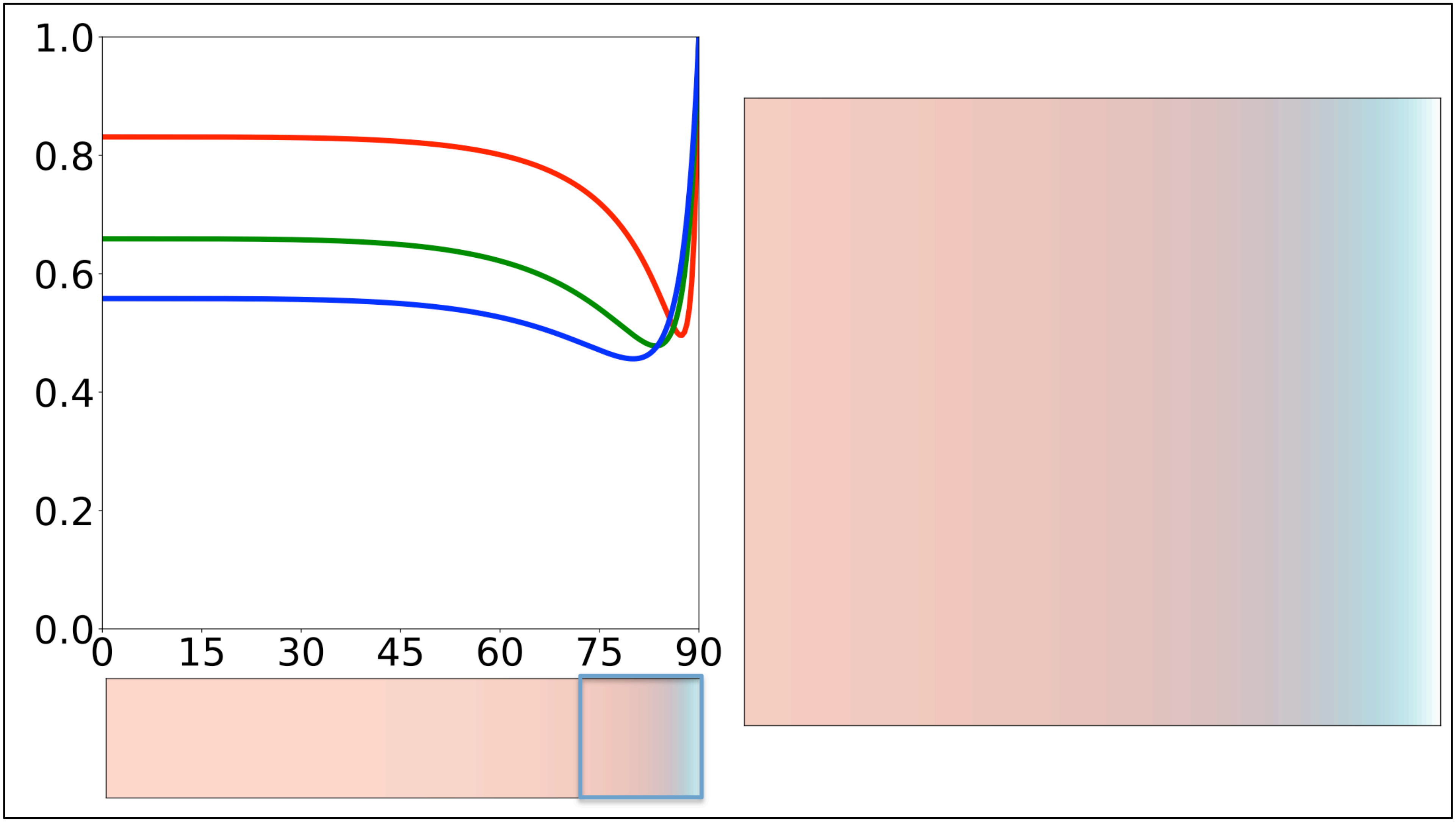
…with the ground truth curves. When we **flip between them**, the dark discolored edge is pretty noticeable, and it's…



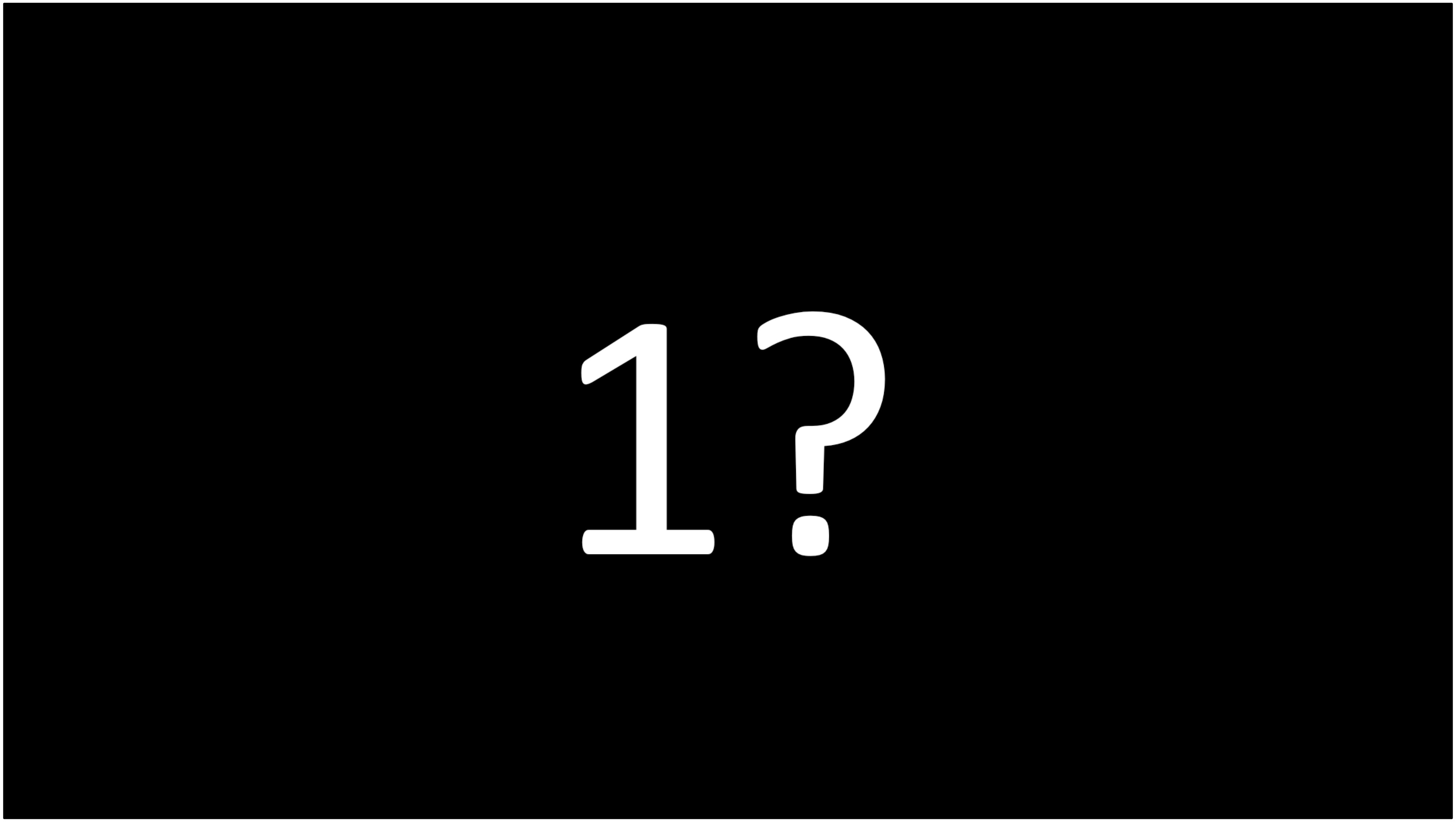
…even visible when we look at the renders side by side.



Here's a closer look — you can see a bluish rim in the "g equals 0" render.

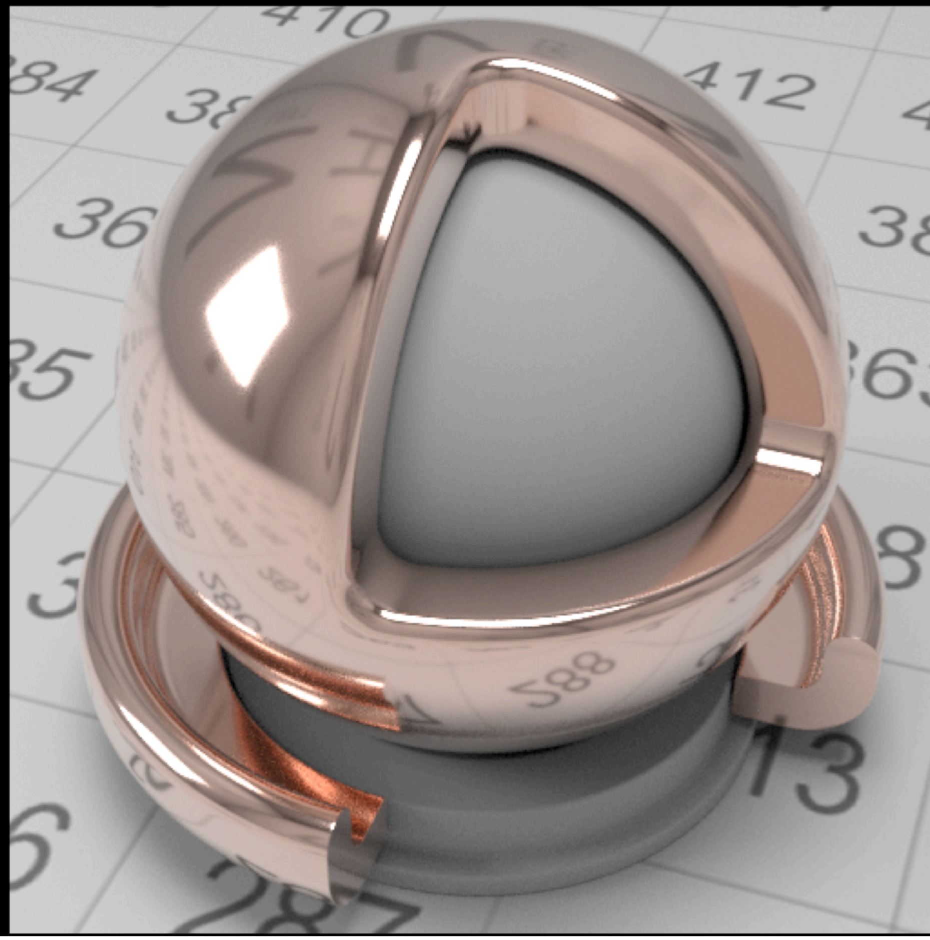


The reason is that the combination of high reflectivity and zero kappa implies an overly high value for eta, much higher than any dielectric or even metal. This causes an exaggerated dip, which is sharper for channels that are higher at 0 degrees, causing a color shift.



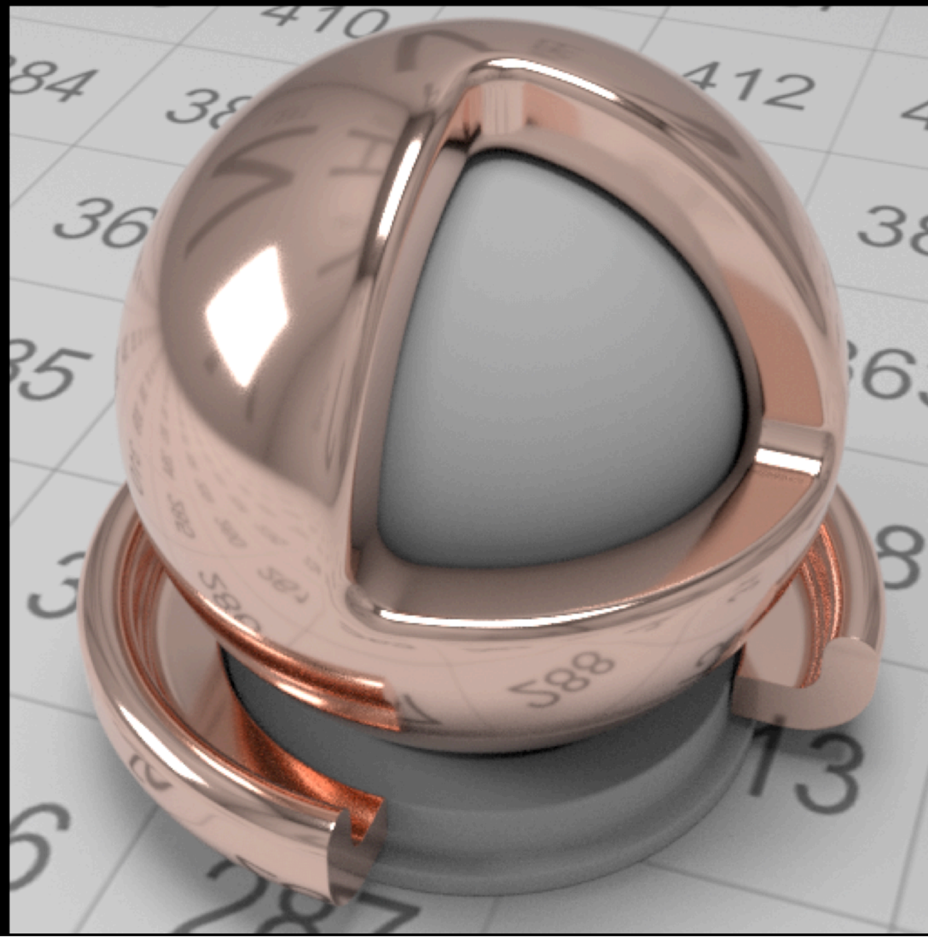
Well, how about setting g to one? That's the default that Autodesk Standard Surface uses.

g = 1

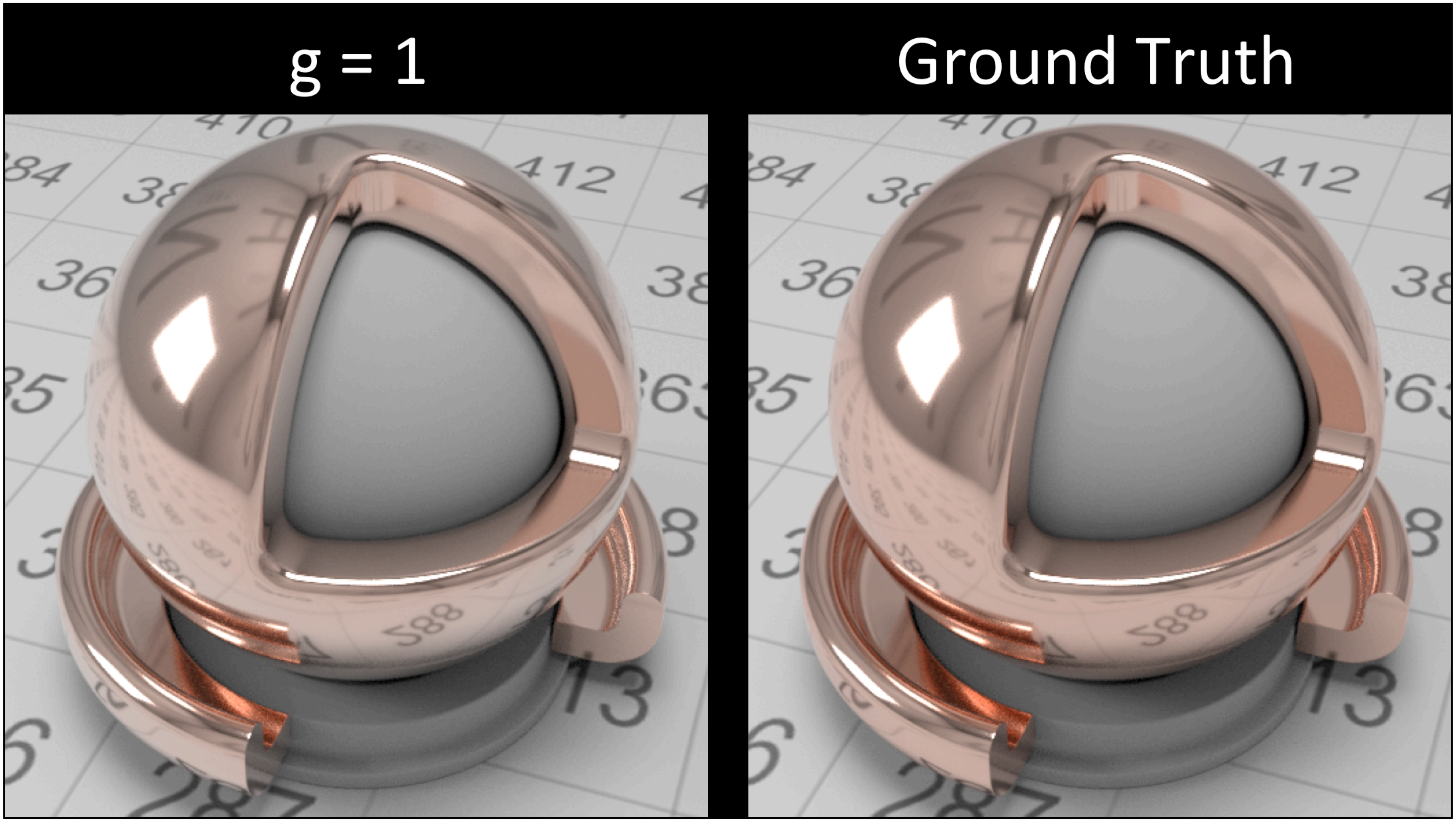


Setting g to one results in a washed out, desaturated appearance around the rim. Again, let's compare to…

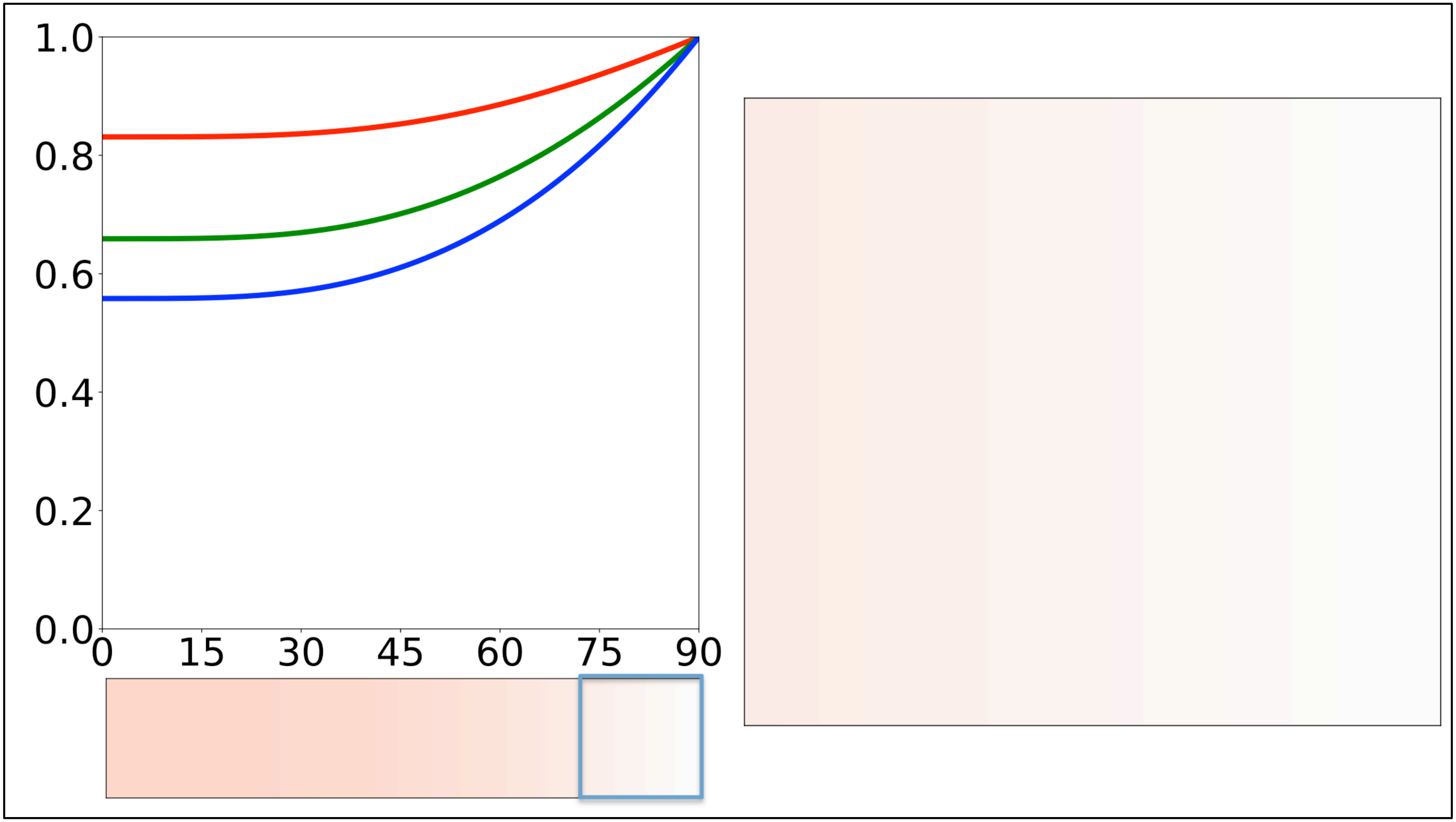
Ground Truth



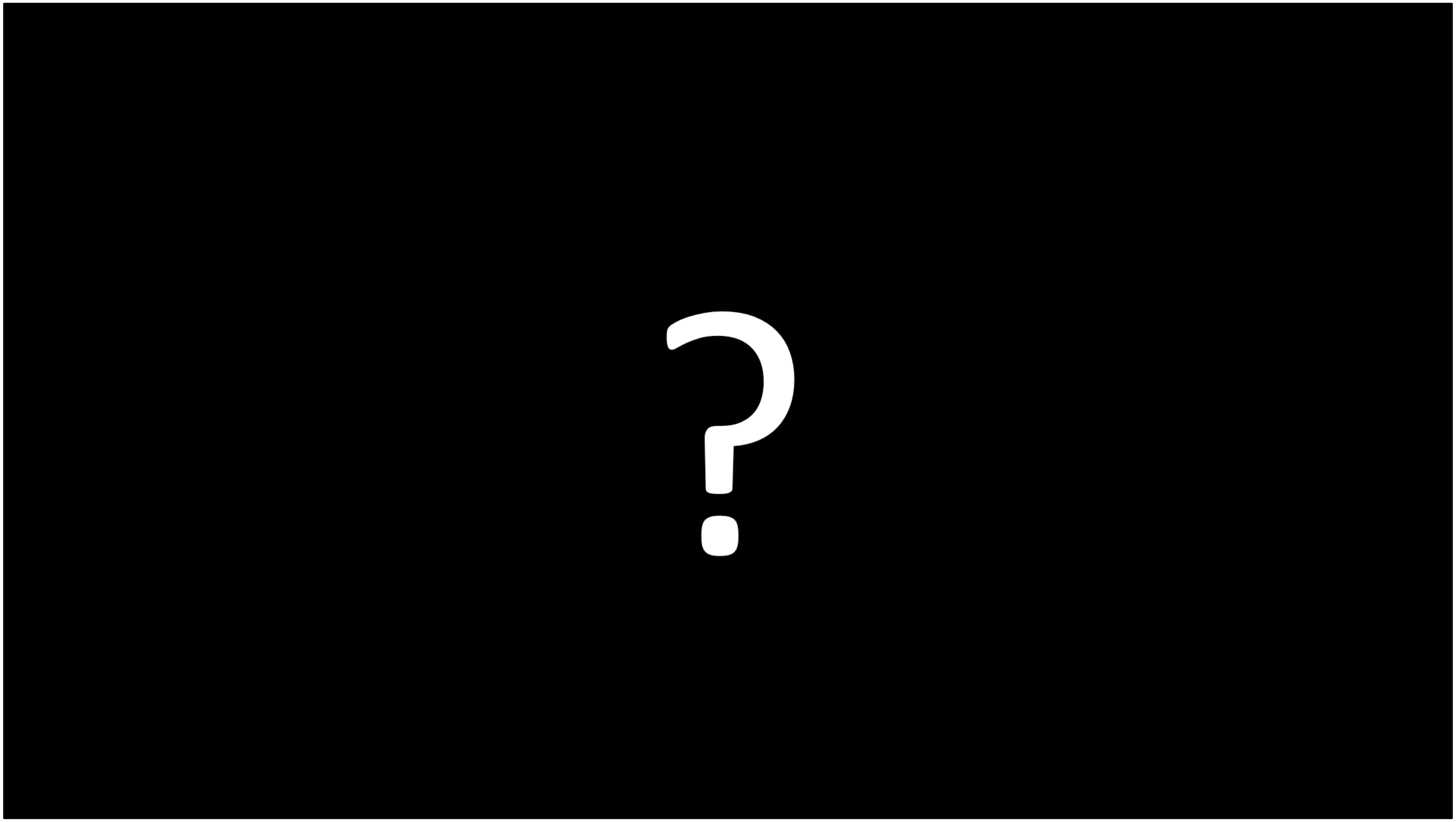
…the ground truth curves. **Flipping between them** shows a “washed out” effect that extends quite a bit inwards from the edge…



…and is visible even in side-by-side renders.



Again, looking at the curves can shed some light on what is going on. We see that they start going to white at around 30 degrees, which is a much smaller angle than usual. This causes the surface to appear overly desaturated.



Maybe some other default value? I don't know, I've tried using mid-grey and a bunch of other values. Each option looked wrong for some metals. This is a difficult situation for a surface painter: they can't ignore the edge tint parameter, and they need to spend a fair amount of effort to find a value that leads to a reasonable look.

What makes a parameter artist friendly?

- Intuitive relationship between parameter value and surface appearance
  - If a color, that color should be visible on the surface
- Perceptually uniform effect over its range
- Orthogonal to other parameters
- Secondary parameters should have default values

In conclusion, we can see that g doesn't fulfill any of the requirements for an artist-friendly parameter.

The Bitter Fruit of RGB Nonlinearity

Earlier I mentioned that performing nonlinear operations on RGB quantities can cause problems. I'll now elaborate on this a bit.

(Image credit: Momordica charantia, Cucurbitaceae, Bitter Melon, Bitter Gourd, fruit, splitting open. By H. Zell. Licensed CC BY-SA 3.0.)

Parameter Blending

- Compare the error (difference) between 2 cases:
  - Case A: For angles 0°–90°, evaluate Fresnel for two materials and blend the result 50/50.
  - Case B: Do a 50/50 blend on the parameters (F_0, g), and use the blended parameters to evaluate Fresnel for angles 0°–90°.

Parameter value blending is an extremely common operation. It happens as part of texture filtering, and when combining different materials when authoring. Ideally, blending the Fresnel parameters of two materials would give the same result as evaluating Fresnel for each material separately and blending the result. That is the case for the Schlick model since it is linear, but not for the g edge tint model.

Gold / Chromium	2.42
Gold / Iron	2.15
Copper / Chromium	2.02
Gold / Zinc	1.96
Brass / Chromium	1.95
Bronze / Chromium	1.94
Copper / Iron	1.94
Gold / Titanium	1.83
Bronze / Iron	1.78
Copper / Titanium	1.77
Brass / Iron	1.72
Zinc / Copper	1.68
Zinc / Bronze	1.66
Silver / Gold	1.57
Silver / Chromium	1.45
Aluminum / Gold	1.47
Gold / Platinum	1.47
Bronze / Titanium	1.49
Copper / Platinum	1.42
Zinc / Chromium	1.40

This table shows the errors* introduced by blending F0 and g values for different pairs of metals. To be fair, these errors aren't large — around one and a half to two times the just noticeable difference. But material authoring often involves multiple chained blends, so it is disconcerting to have errors introduced at each step, even if small.

* CIE delta-E 2000 differences, averaged over all input angles.

Color Gamut Conversion

- Compare 2 cases:
  - Case A: For angles 0°–90°, evaluate Fresnel in the ACEScg gamut, then convert to a new gamut.
  - Case B: Convert parameters (F_0, g) from ACEScg to a new color gamut, then evaluate Fresnel in that gamut for angles 0°–90°.
- We test conversion to two gamuts: one smaller than ACEScg (Rec.709) and one larger (AP0).

In production rendering, we sometimes need to convert parameters between color gamuts. To preserve the material appearance, shading with the converted parameters should give the same colors as shading in the original color space and then converting the result. This is again true for the Schlick model due to its linear nature, but not for g edge tint.

Metal	Gulbrandsen Rec.709	Gulbrandsen AP0
Silver	0.17	0.02
Copper	0.59	0.83
Gold	0.44	1.46
Bronze	1.38	0.79
Brass	1.89	0.54
Aluminum	0.08	0.04
Palladium	0.04	0.02
Nickel	0.07	0.03
Titanium	0.05	0.02
Mercury	0.11	0.05
Platinum	0.05	0.03
Iron	0.06	0.02
Zinc	0.74	0.39
Chromium	0.06	0.03

Overall, these errors are smaller than the blending case. Some of the more strongly colored metals have errors of about one and a half times the just noticeable difference, which isn't that bad. Still, with a linear model like Schlick's these errors don't exist at all — color-space conversion is a lossless operation, as it should be.

Computational Cost

Finally, using the Fresnel equations in your shading model is just plain expensive.

Computational Cost

$$F(\theta) \approx F_0 + (1 - F_0)(1 - \cos\theta)^5$$

Instead of something nice and cheap like Schlick…

Computational Cost

$$F(\theta) = \frac{F_s(\theta) + F_p(\theta)}{2}$$

$$F_s(\theta) = \frac{a^2 + b^2 - 2a\cos\theta + \cos^2\theta}{a^2 + b^2 + 2a\cos\theta + \cos^2\theta}$$

$$F_p(\theta) = F_s(\theta)\frac{a^2 + b^2 - 2a\sin\theta\tan\theta + \sin^2\theta\tan^2\theta}{a^2 + b^2 + 2a\sin\theta\tan\theta + \sin^2\theta\tan^2\theta}$$

$$2a^2 = \sqrt{(\eta^2 - \kappa^2 - \sin^2\theta)^2 + 4\eta^2\kappa^2} + (\eta^2 - \kappa^2 - \sin^2\theta)$$

$$2b^2 = \sqrt{(\eta^2 - \kappa^2 - \sin^2\theta)^2 + 4\eta^2\kappa^2} - (\eta^2 - \kappa^2 - \sin^2\theta)$$

…you get a wall of math for Fresnel's equations themselves…

Computational Cost

$$F(\theta) = \frac{F_s(\theta) + F_p(\theta)}{2}$$

$$F_s(\theta) = \frac{a^2 + b^2 - 2a\cos\theta + \cos^2\theta}{a^2 + b^2 + 2a\cos\theta + \cos^2\theta}$$

$$F_p(\theta) = F_s(\theta)\frac{a^2 + b^2 - 2a\sin\theta\tan\theta + \sin^2\theta\tan^2\theta}{a^2 + b^2 + 2a\sin\theta\tan\theta + \sin^2\theta\tan^2\theta}$$

$$2a^2 = \sqrt{(\eta^2 - \kappa^2 - \sin^2\theta)^2 + 4\eta^2\kappa^2} + (\eta^2 - \kappa^2 - \sin^2\theta)$$

$$2b^2 = \sqrt{(\eta^2 - \kappa^2 - \sin^2\theta)^2 + 4\eta^2\kappa^2} - (\eta^2 - \kappa^2 - \sin^2\theta)$$

$$\eta(F_0, g) = g\frac{1 - F_0}{1 + F_0} + (1 - g)\frac{1 + \sqrt{F_0}}{1 + \sqrt{F_0}}$$

$$\kappa(F_0, \eta) = \sqrt{\frac{1}{1 - F_0}\Big(F_0(\eta + 1)^2 - (\eta - 1)^2\Big)}$$

…with some extra bricks for converting the edge tint parameterization to eta and kappa. This math is commonly used in a dedicated conductor lobe…

Computational Cost

$$F(\theta) = \frac{F_s(\theta) + F_p(\theta)}{2}$$

$$F_s(\theta) = \frac{a^2 + b^2 - 2a\cos\theta + \cos^2\theta}{a^2 + b^2 + 2a\cos\theta + \cos^2\theta}$$

$$F_p(\theta) = F_s(\theta)\frac{a^2 + b^2 - 2a\sin\theta\tan\theta + \sin^2\theta\tan^2\theta}{a^2 + b^2 + 2a\sin\theta\tan\theta + \sin^2\theta\tan^2\theta}$$

$$2a^2 = \sqrt{(\eta^2 - \kappa^2 - \sin^2\theta)^2 + 4\eta^2\kappa^2} + (\eta^2 - \kappa^2 - \sin^2\theta)$$

$$2b^2 = \sqrt{(\eta^2 - \kappa^2 - \sin^2\theta)^2 + 4\eta^2\kappa^2} - (\eta^2 - \kappa^2 - \sin^2\theta)$$

$$\eta(F_0, g) = g\frac{1 - F_0}{1 + F_0} + (1 - g)\frac{1 + \sqrt{F_0}}{1 + \sqrt{F_0}}$$

$$\kappa(F_0, \eta) = \sqrt{\frac{1}{1 - F_0}\Big(F_0(\eta + 1)^2 - (\eta - 1)^2\Big)}$$

$$F(\theta) = \frac{F_s(\theta) + F_p(\theta)}{2}$$

$$F_s(\theta) = \left(\frac{a - \cos\theta}{a + \cos\theta}\right)^2$$

$$F_p(\theta) = F_s(\theta)\left(\frac{a - \sin\theta\tan\theta}{a + \sin\theta\tan\theta}\right)^2$$

$$a^2 = \eta^2 - \sin^2\theta$$

…so we <u>also</u> need to calculate the dielectric version of the equations, and mix the results. It would be one thing to spend all this extra computation if there were a significant benefit, but I hope I have shown that the benefit is dubious, perhaps even negative.

* Offline renderers will sometimes stochastically select one lobe rather than computing both, but that adds an additional source of noise.

Conclusions

- The switch from Schlick's approximation to Fresnel's equations resulted in significant drawbacks and few (if any) benefits.
- In many cases staying with Schlick (or going back to it) is the best choice.
- If more control and accuracy than Schlick is needed, there are other alternatives.

To conclude, the switch away from Schlick's approximation to models based on direct use of Fresnel's equations was unfortunate. For many applications, the Schlick model — either in its original or generalized form — is in the sweet spot of expressivity, cost and simplicity. When the Schlick model isn't enough, it's worth exploring alternatives such as reparameterized Lazányi, or the models presented in the next talk. Or come up with your own model — if you find a good one, please let the rest of the industry know about it.

Acknowledgements

- André Mazzone for feedback, and assistance implementing the custom shader node.
- Stephen Hill, Jonathan Stone, Ian Comley, Jonathan Moulin, Thomas Schaedlich, Zap Andersson, Eugene d'Eon, Iliyan Georgiev, Niklas Harrysson, Alan King, and Laurent Belcour for discussion and feedback.
- Rendered scene inspired by the Maxwell shaderball scene created by Thomas Anagnostou.

And finally, I'd like to acknowledge some of the people who helped me on this project. And thank you for listening.

Appendix

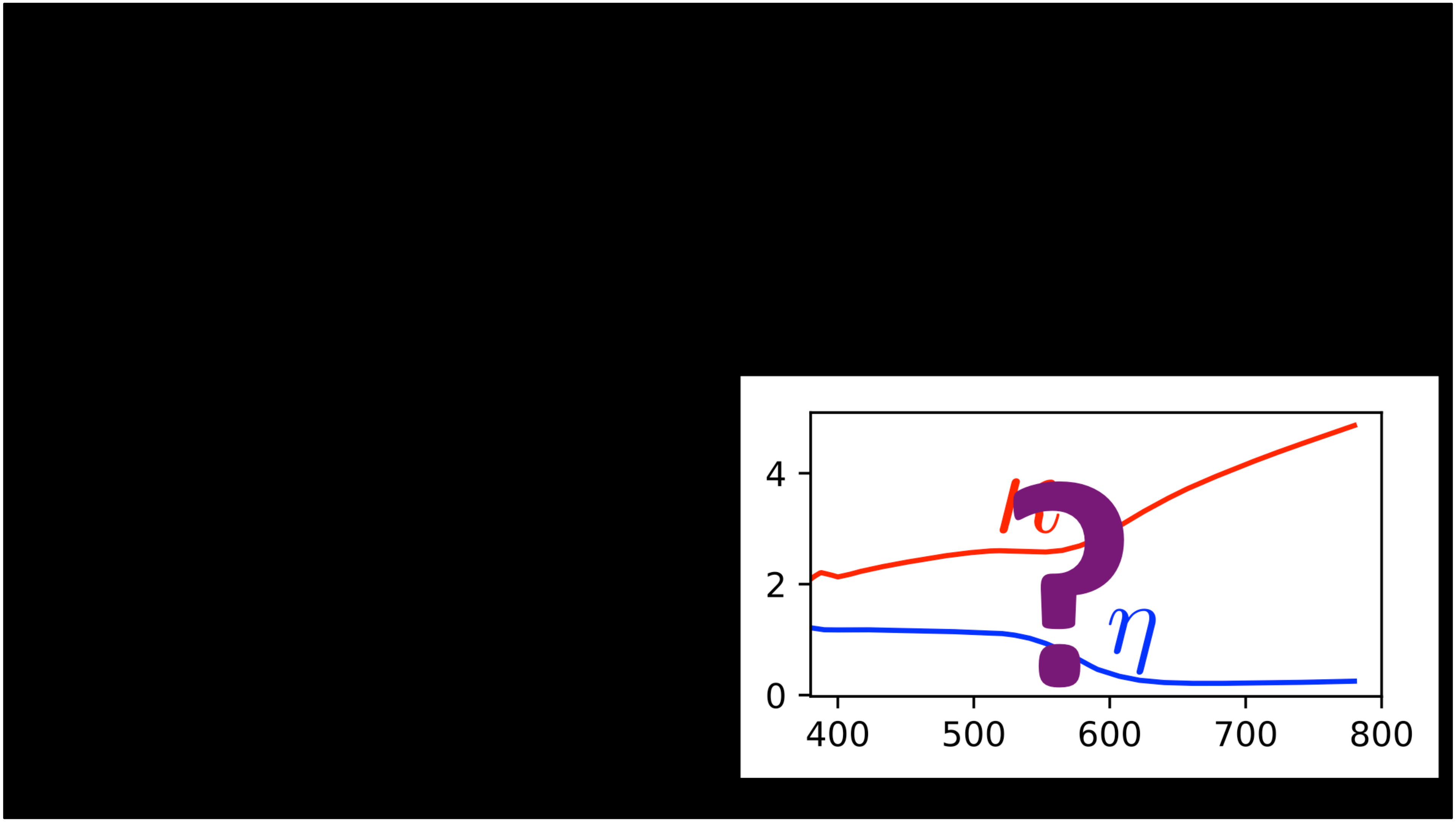
The following section was not presented at the conference. It is included in the course notes as optional material for further reading.

More Details on the Cook-Torrance Paper

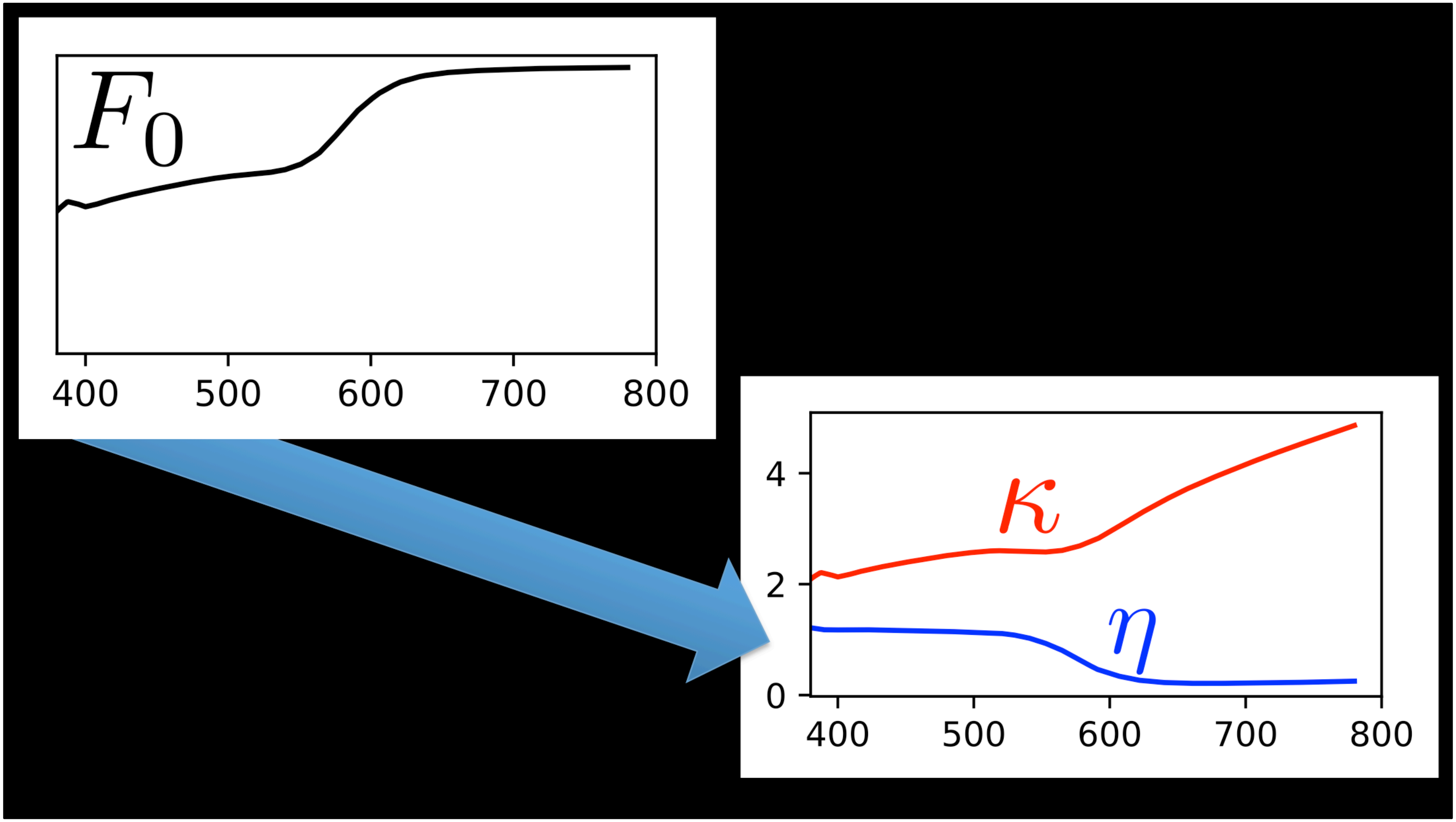
There are some details regarding the Cook-Torrance paper that I only had to to refer to briefly in the talk.

1. Full procedure: spectral computation of reflected light spectral power distribution (SPD), then conversion to RGB pixel color. ✓
2. An approximation of the above, based on RGB interpolation. ✓
3. A procedure for approximating spectral η and κ from spectral F0. ✗

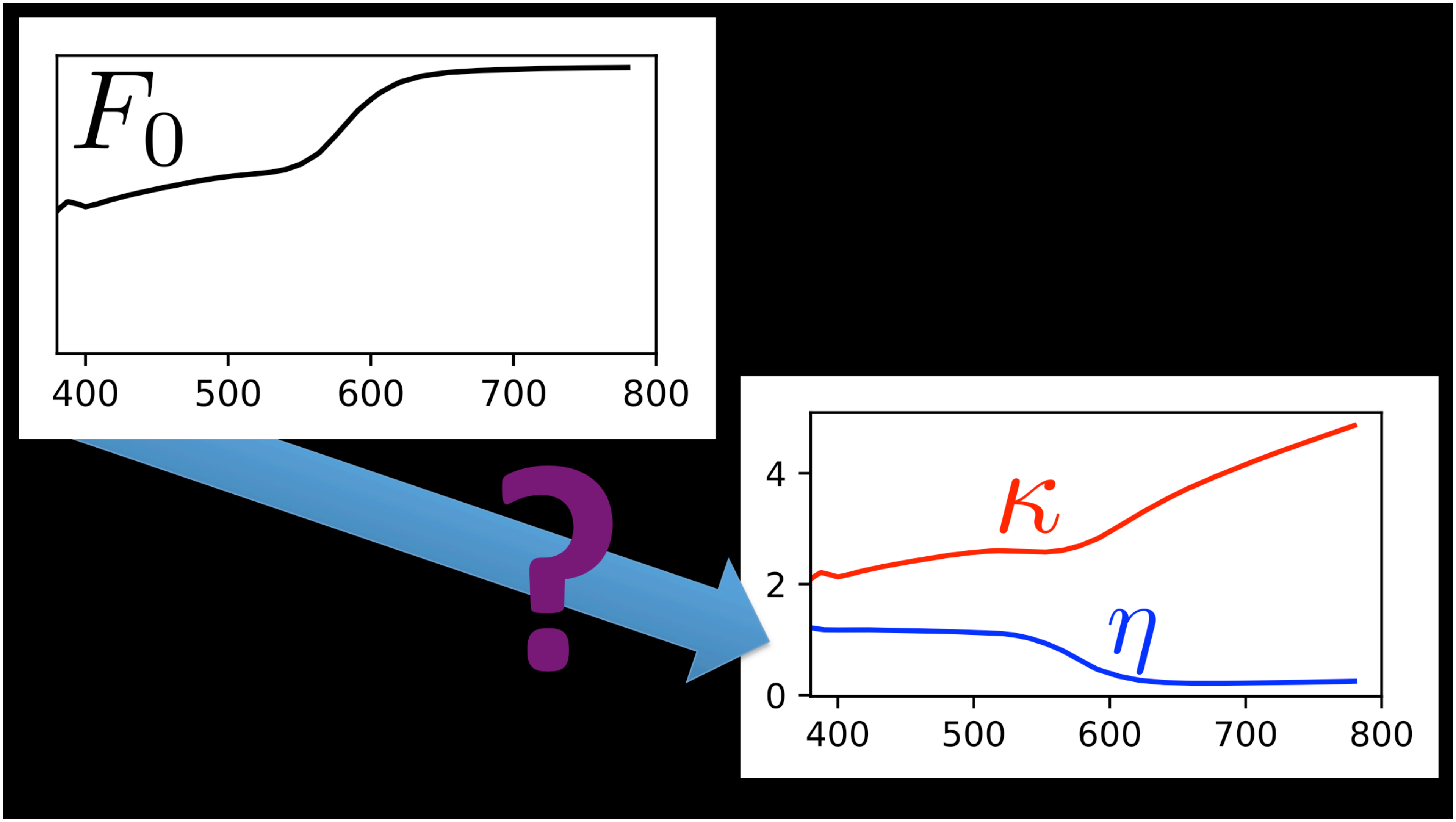
As mentioned earlier, the Cook-Torrance paper included a procedure for inferring approximate values for spectral eta and kappa from spectral F0 data.



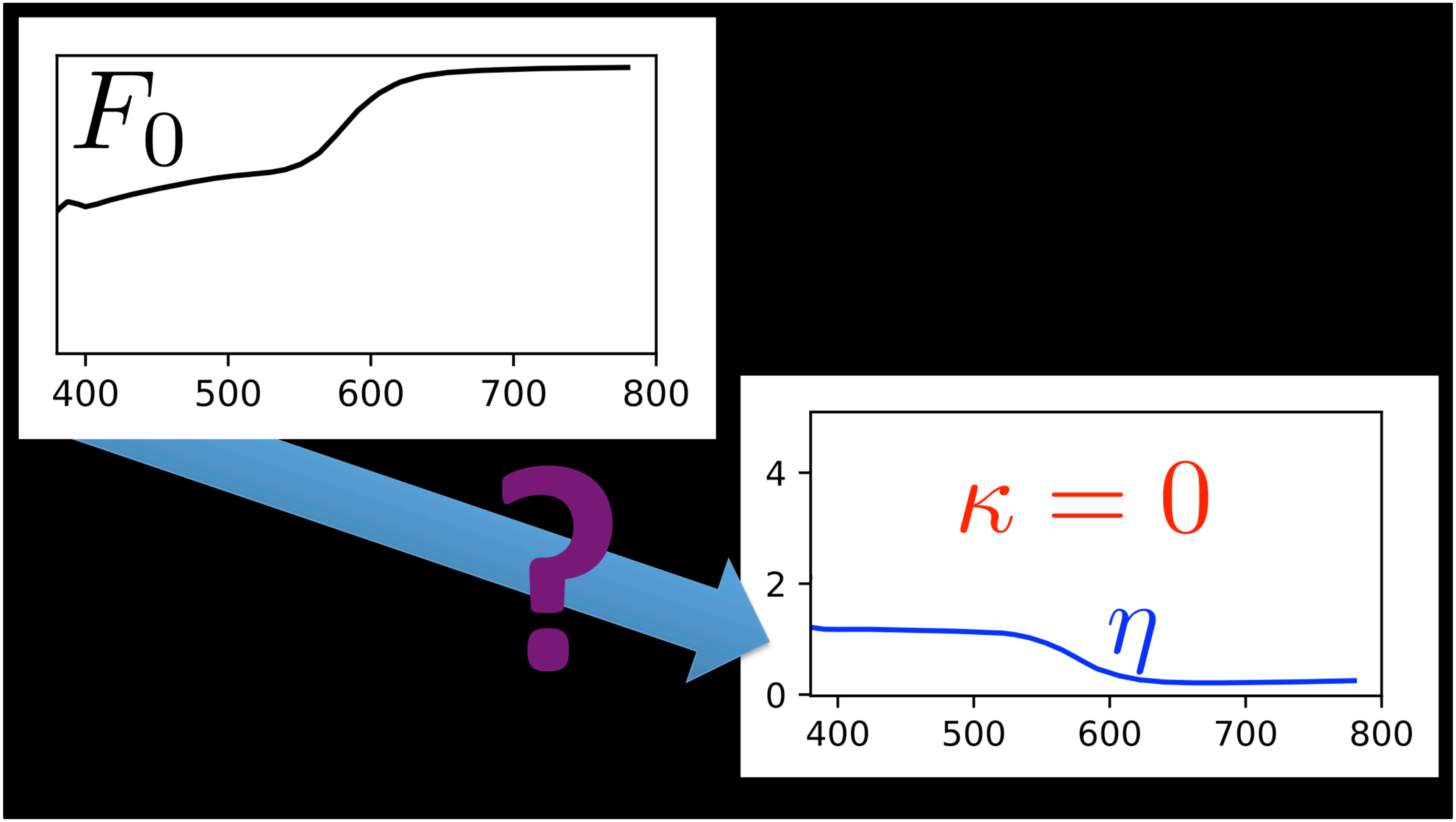
The motivation for this procedure is the fact that measured spectral IOR data does not exist for most materials — only for a handful, typically in a laboratory-grade pure form.



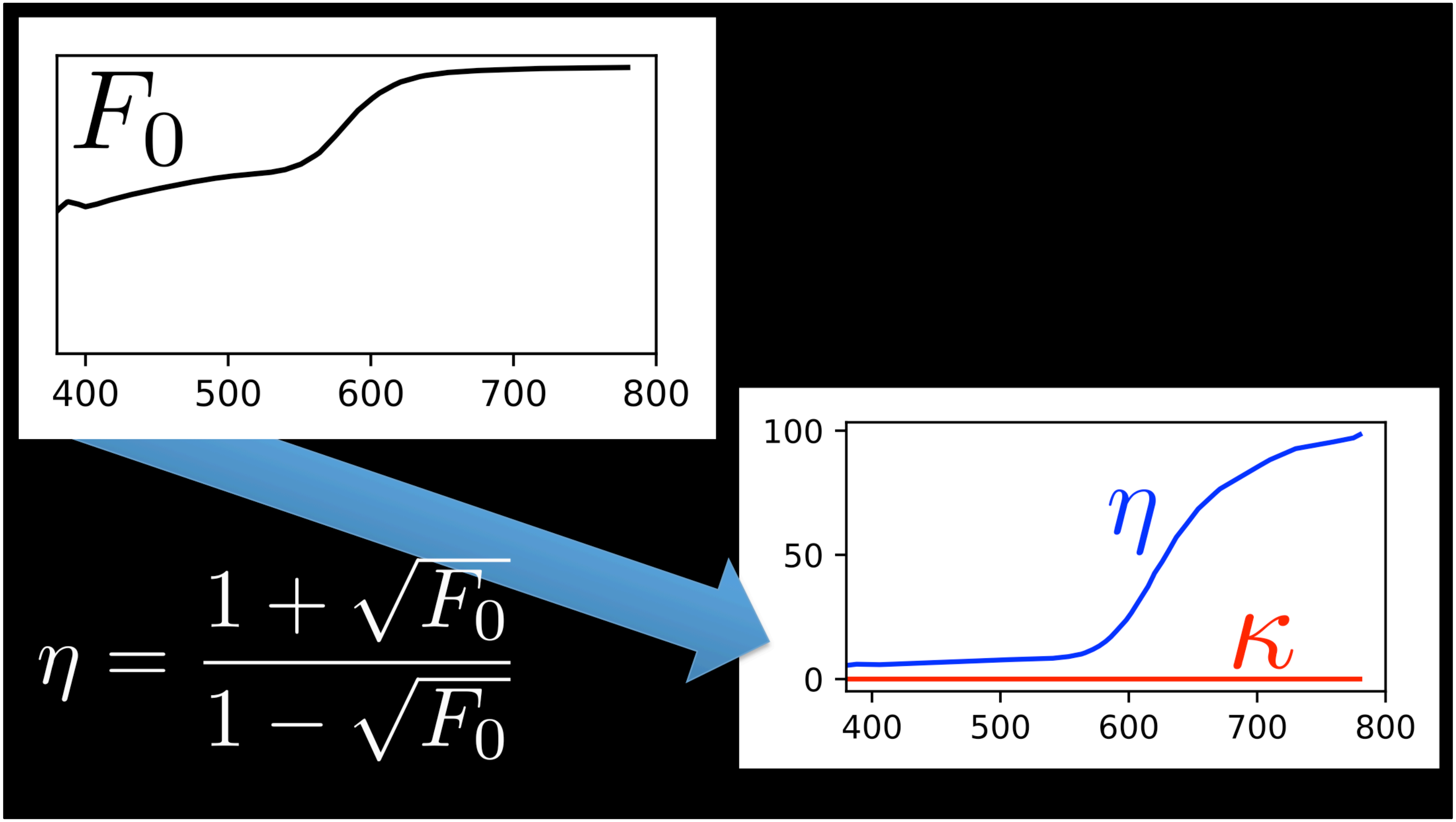
As a solution, Cook and Torrance suggested starting from measured values of F0, the reflectance at normal incidence (0 degrees), and using them to infer the IOR values. F0 values are more commonly available than IOR data.



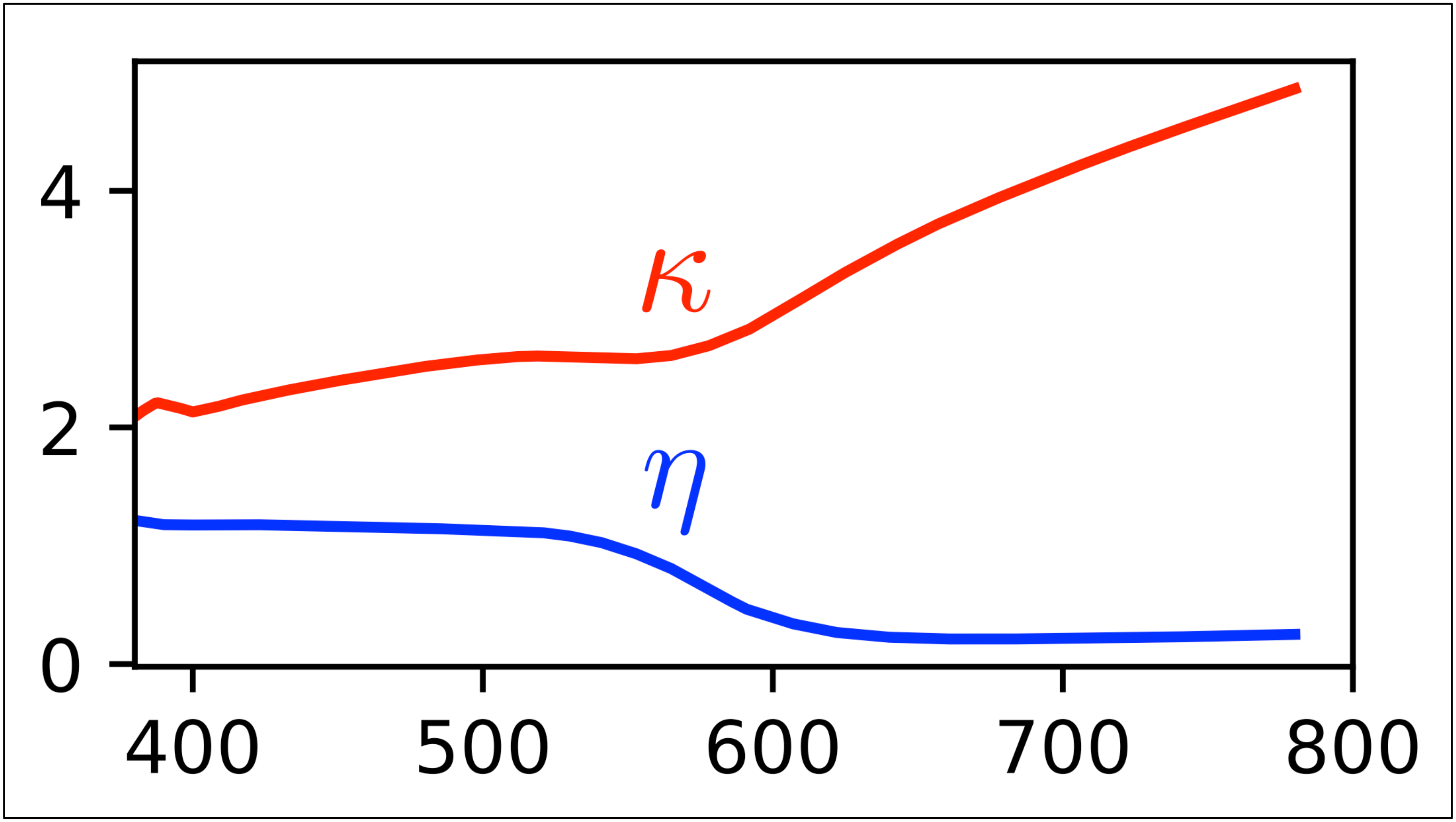
But how do we infer IOR from reflectance? The problem is underdetermined: there are infinite eta-kappa pairs that correspond to a given reflectance value. Cook and Torrance noted that dielectrics are straightforward, since we know…



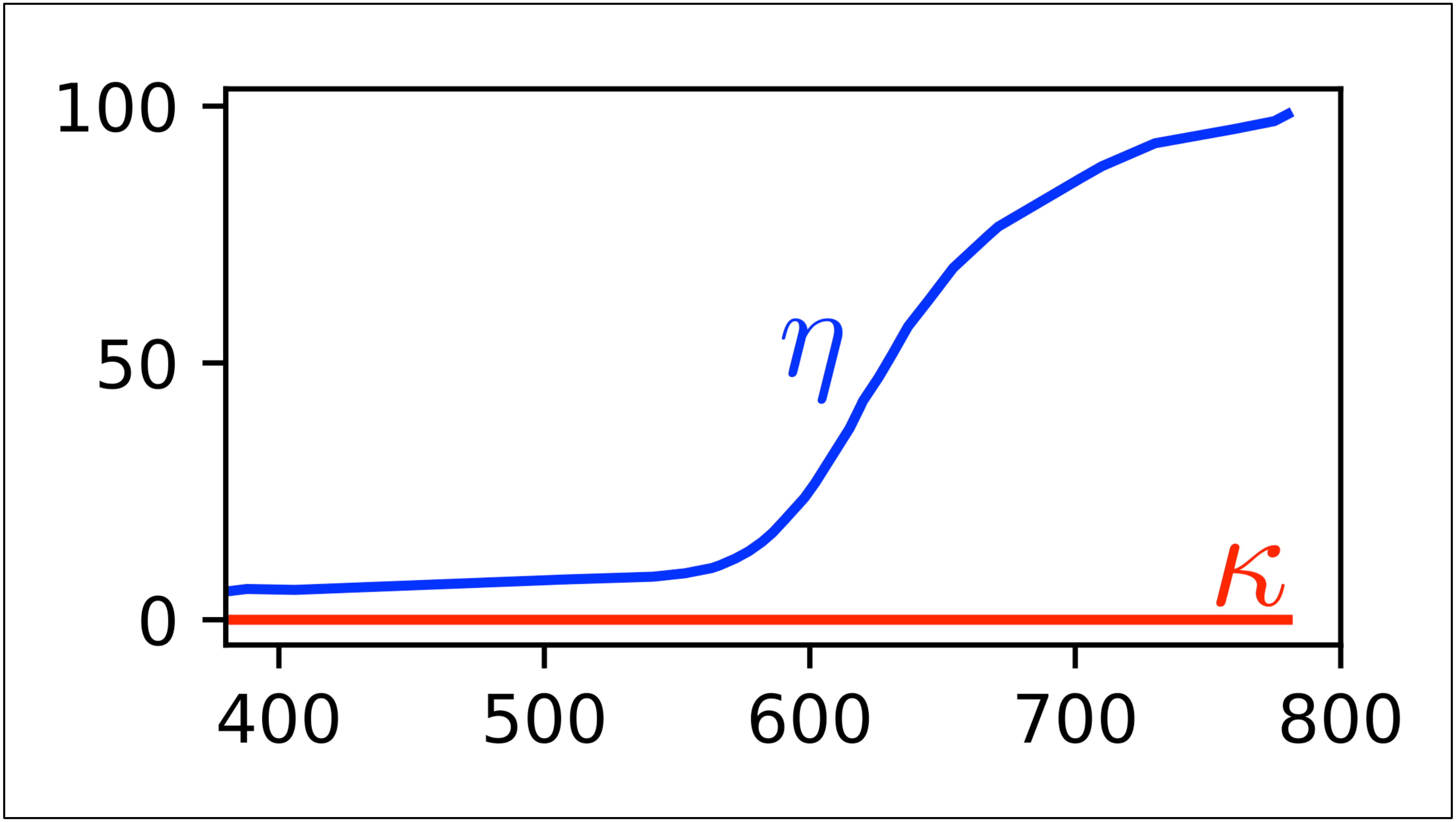
…that their kappa is zero.



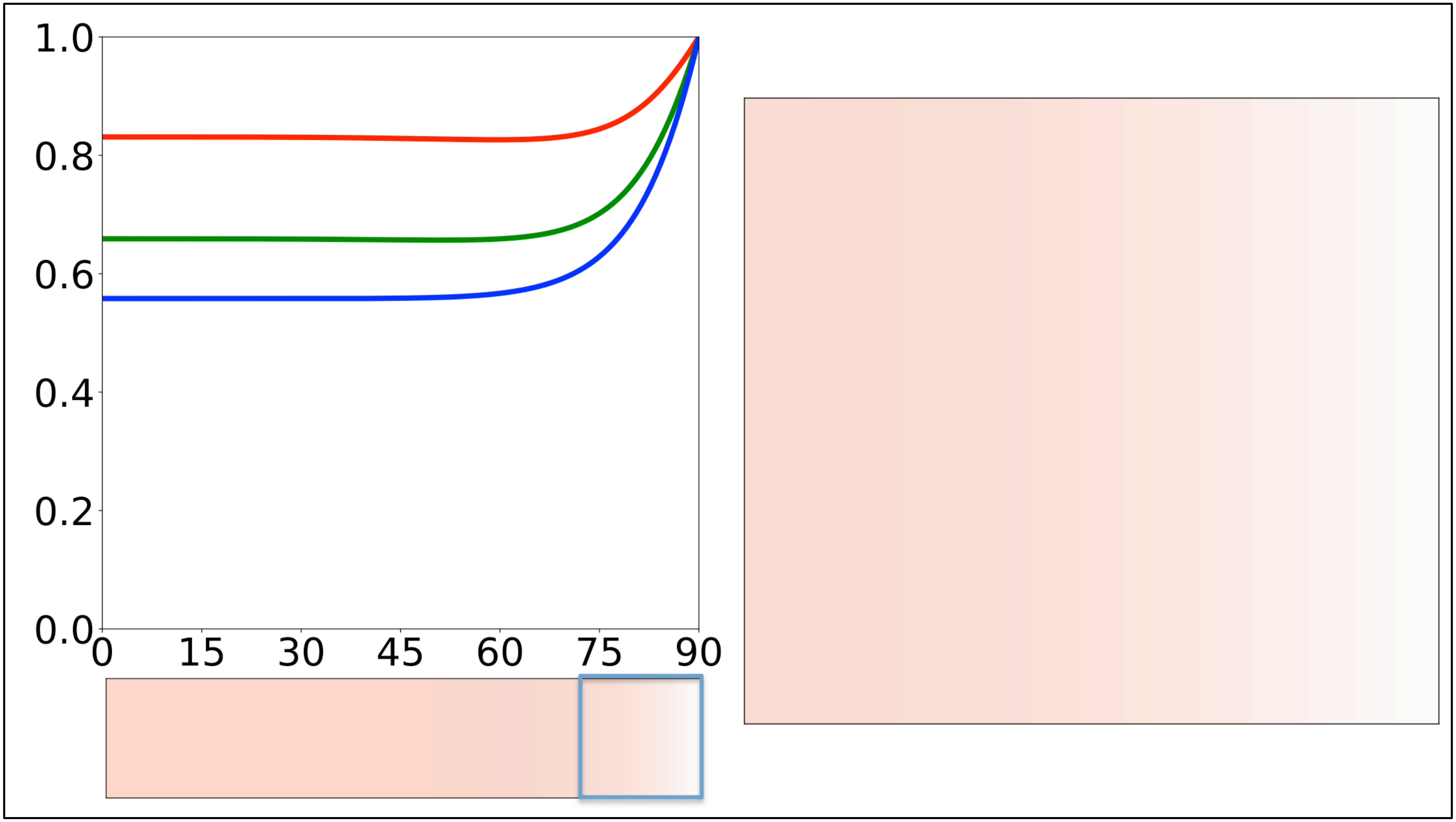
In this case, there is a simple equation to compute eta from normal-incidence reflectance. Despite only being correct for dielectrics, the paper recommended using this method for metals as well. Their reasoning was that the result for normal incidence will be correct, and that the angular variation depends relatively weakly on kappa. However, the authors neglected to take into account that applying this equation to metals results in hugely exaggerated values of eta.



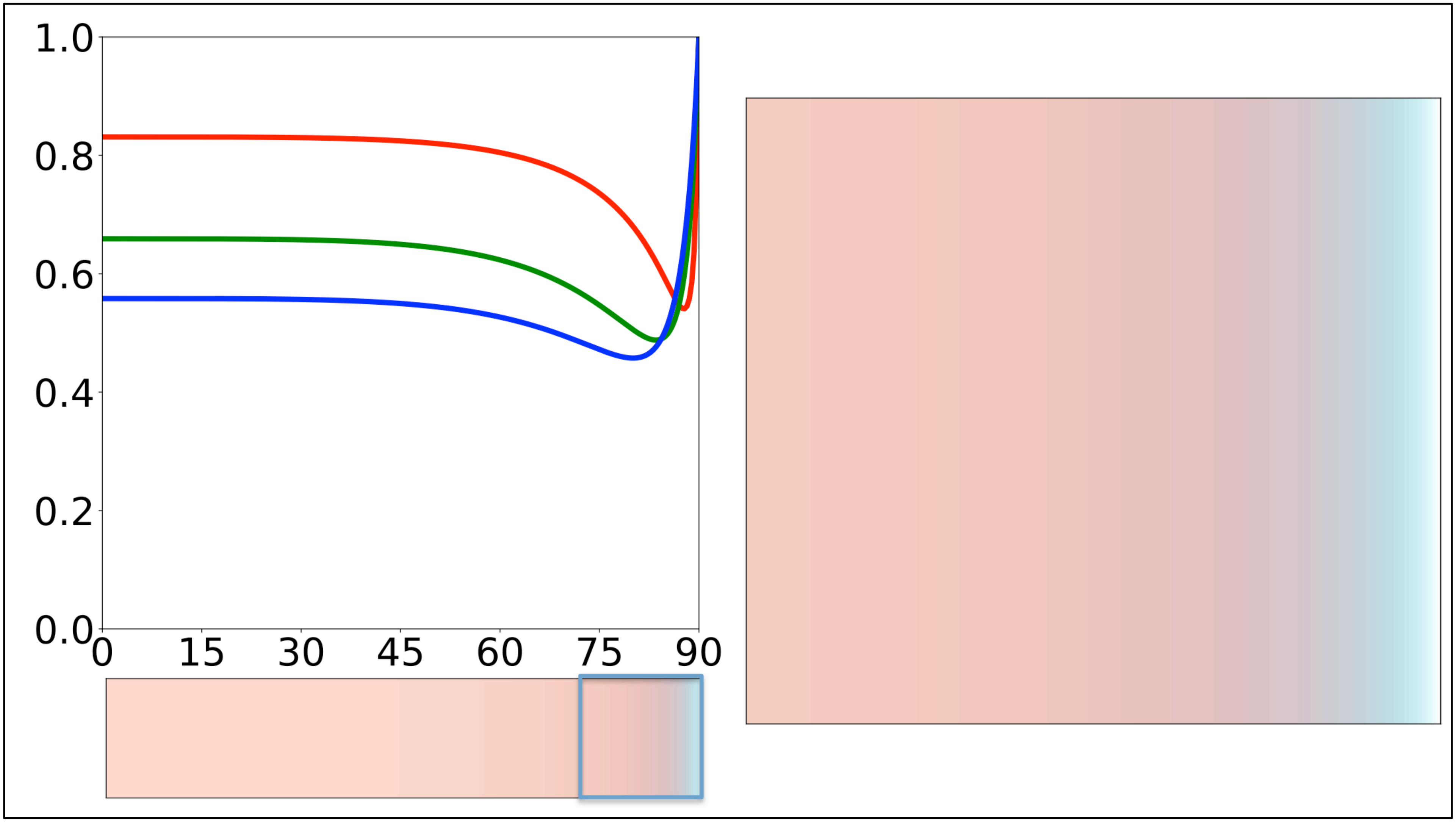
These are the actual spectral values of eta and kappa for copper. Note that neither of them go above 5.



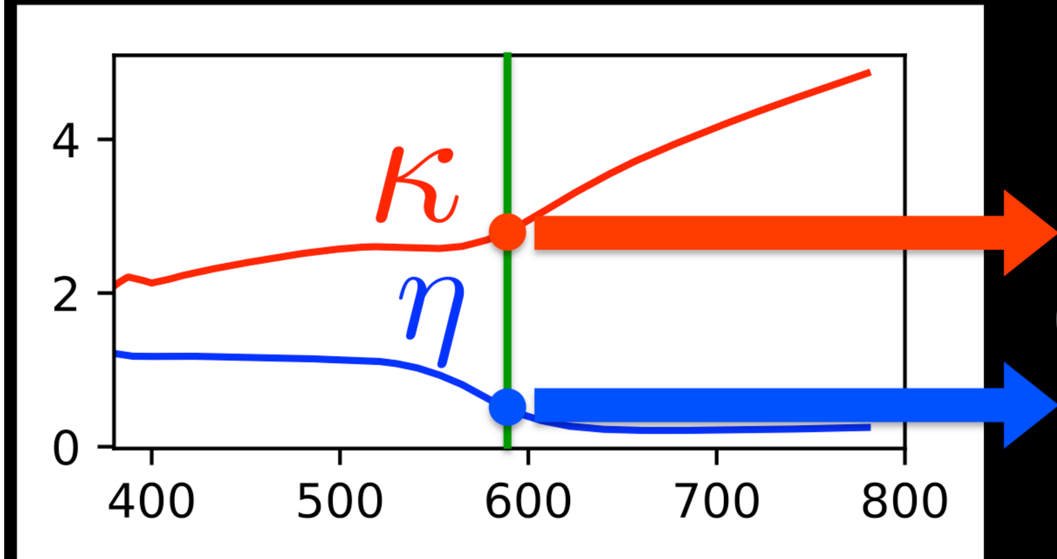
And these are the values resulting from forcing kappa to 0 while keeping F0 constant. The value of eta approaches 100 at the red end of the spectrum. No metal or dielectric has values anywhere near this high.



As an example, we'll again use copper. These are the correct RGB curves and colors. In contrast, using the "kappa equals 0" assumption…



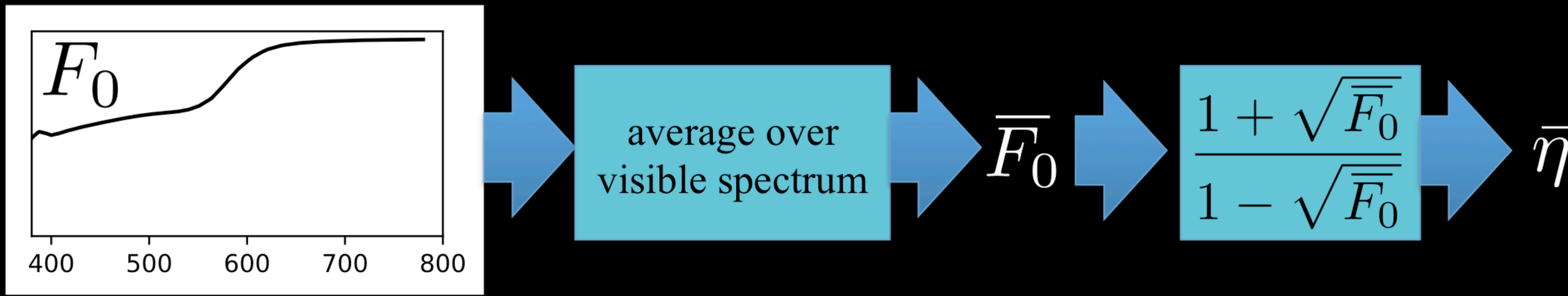
…results in this. Flipping between them clearly shows a gross error that makes this method unusable. Note that this approximation is not exactly the same as using black for the edge tint g parameter (since it is applied spectrally), but the two methods are close both conceptually and in their results.



$$(\eta', \kappa') \Rightarrow F'(\theta)$$

$$F(\theta) \approx F_0 + (1 - F_0)\frac{\max(0, F'(\theta) - F'(0^\circ))}{1 - F'(0^\circ)}$$

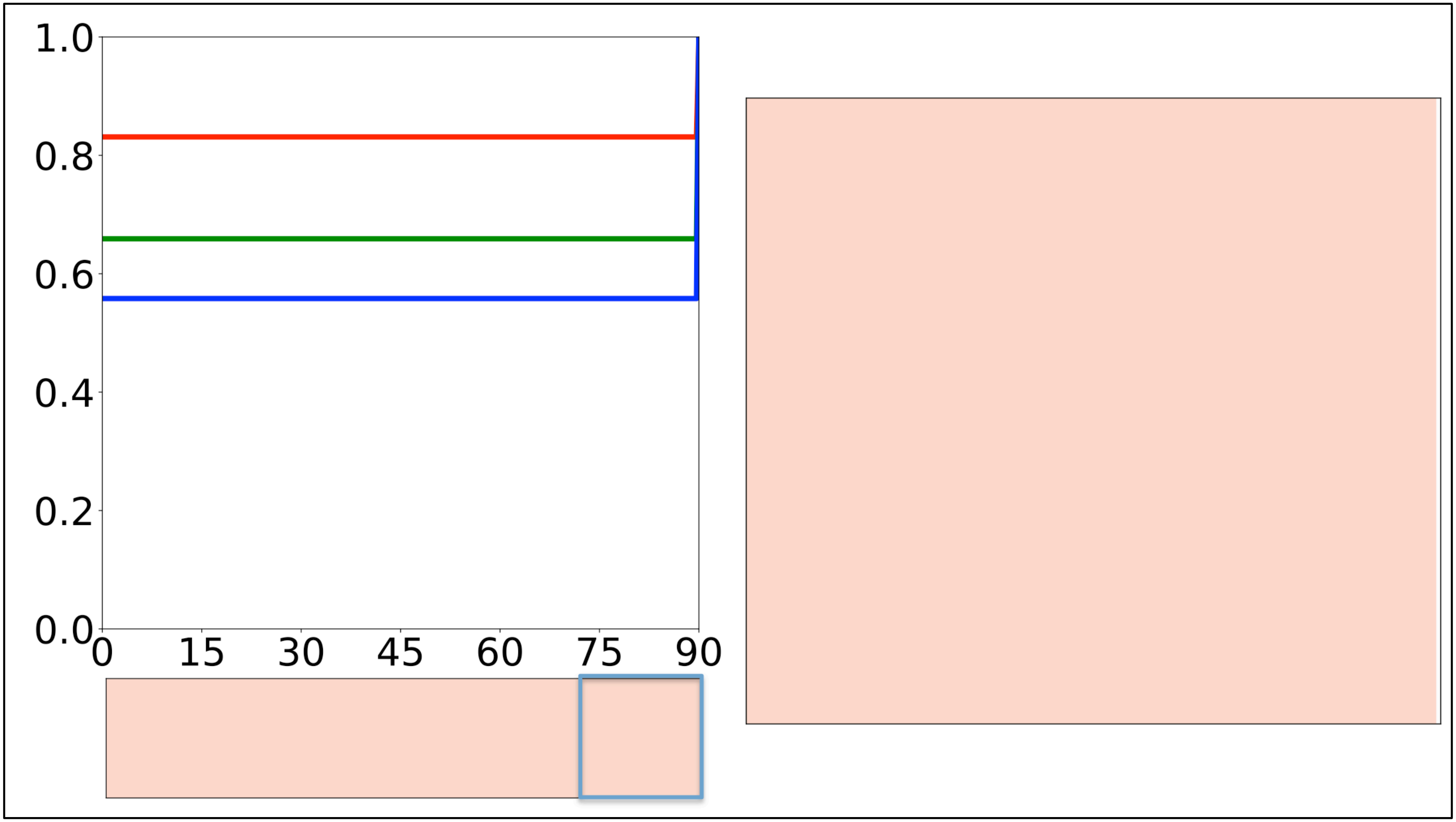
The Cook-Torrance paper also included this flawed procedure in the original version of their color-interpolation approximation. The version we presented earlier is shown here — it uses a representative single complex IOR value (eta and kappa). However, the original version was a bit different…



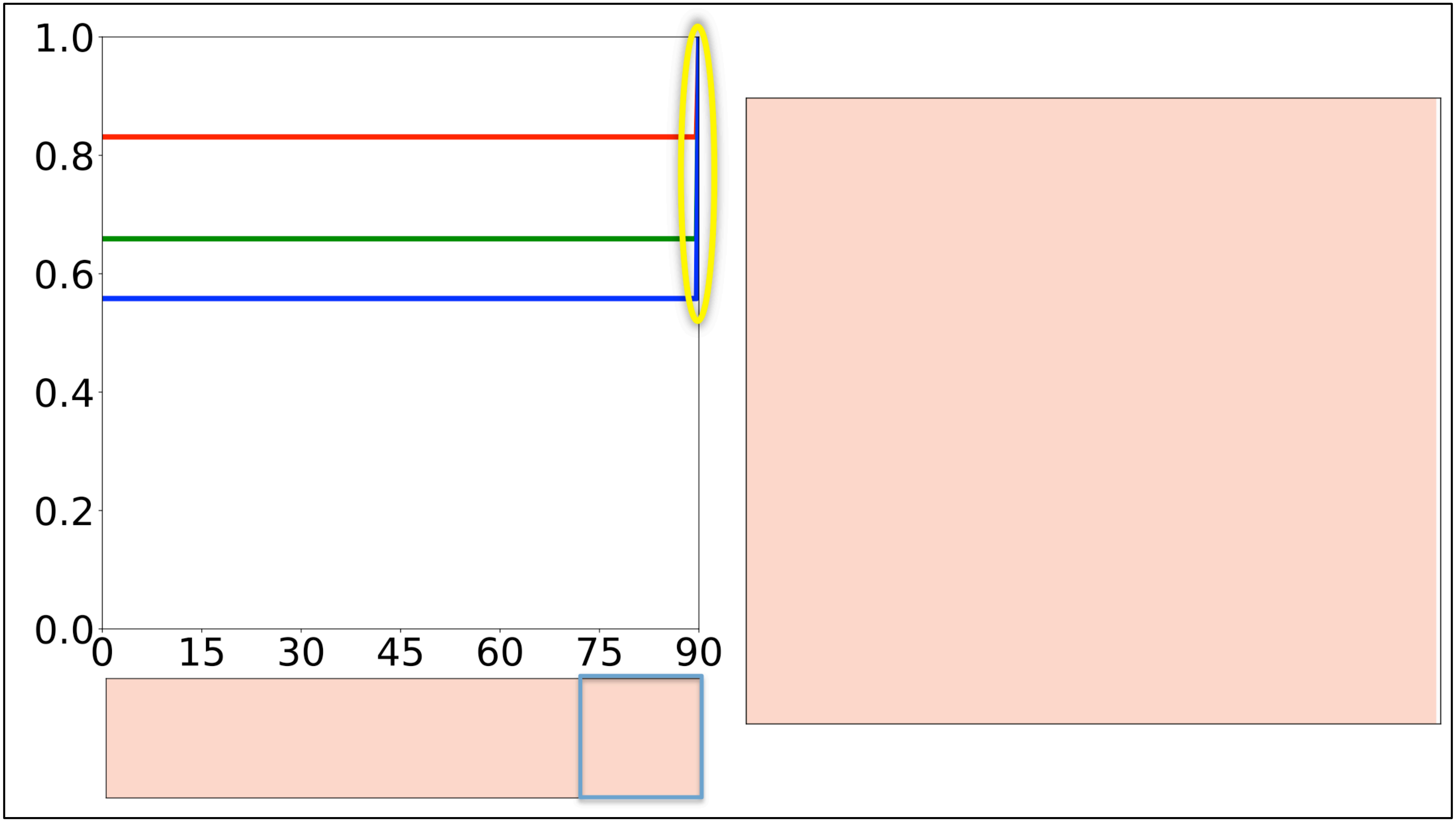
$$\bar{\eta} = \frac{1+\sqrt{\overline{F_0}}}{1-\sqrt{\overline{F_0}}}$$

$$F(\theta) \approx F_0 + (1 - F_0)\frac{\max(0, F_{\bar{\eta}}(\theta) - F_{\bar{\eta}}(0^\circ))}{1 - F_{\bar{\eta}}(0^\circ)}$$

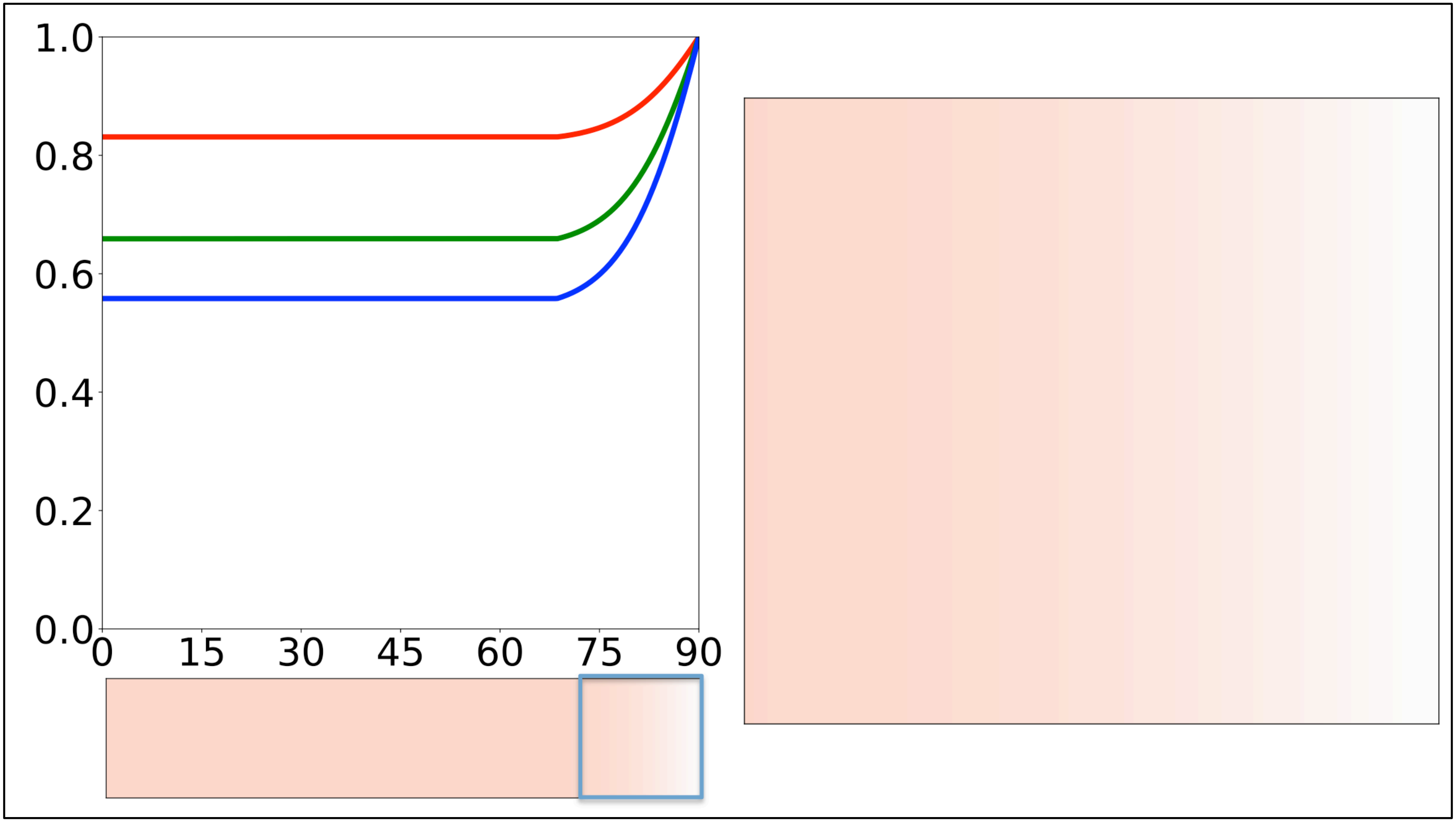
…Cook and Torrance took the average value of F0, and used the “kappa equals 0” assumption to infer a value for eta, which they then used for the interpolation. This version produces significantly worse results:…



…the strong dip resulting from the "kappa equals zero" assumption is clamped, resulting in basically constant reflectance until just before 90 degrees…



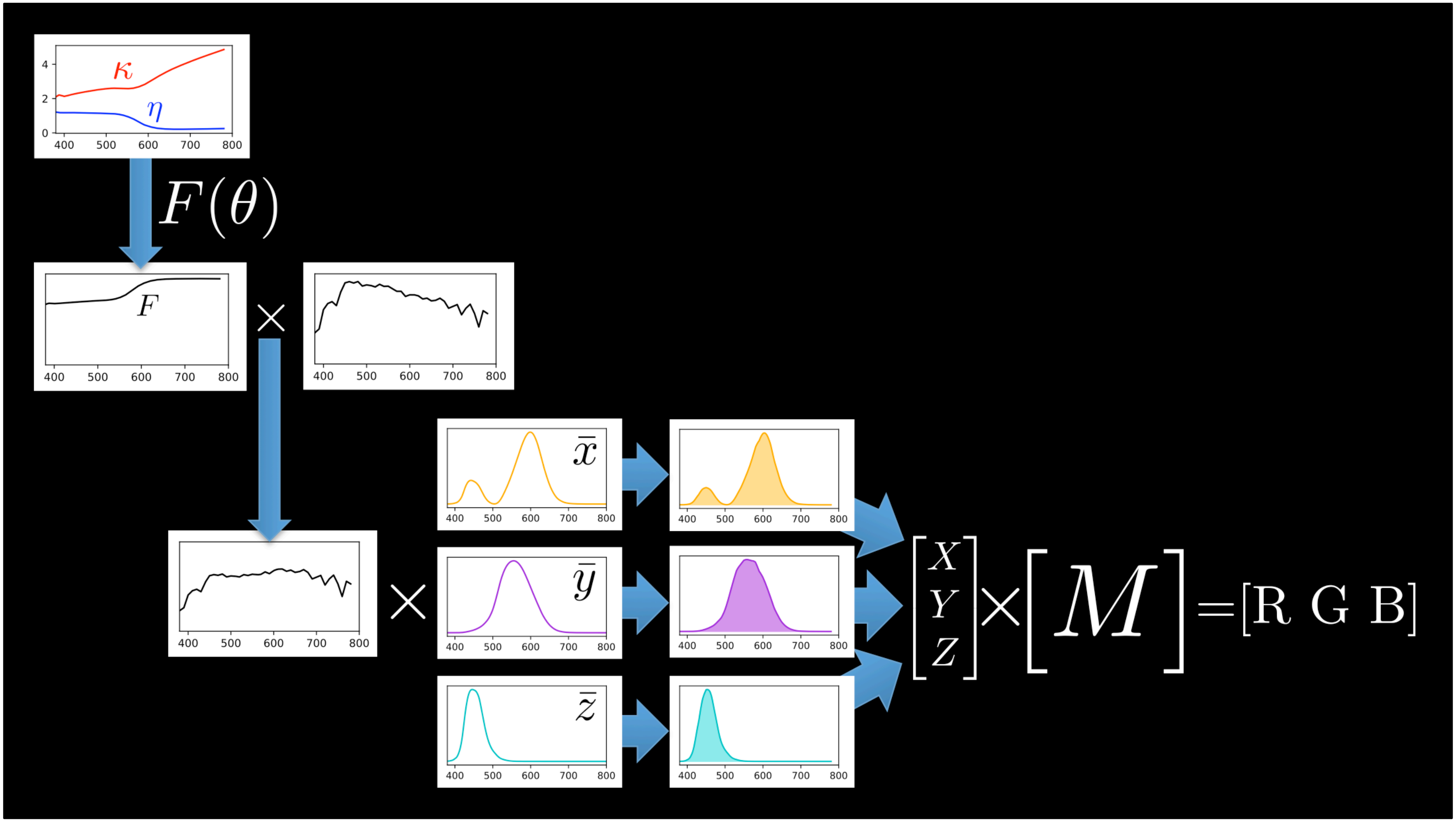
…as marked in yellow at the very end of the graph.



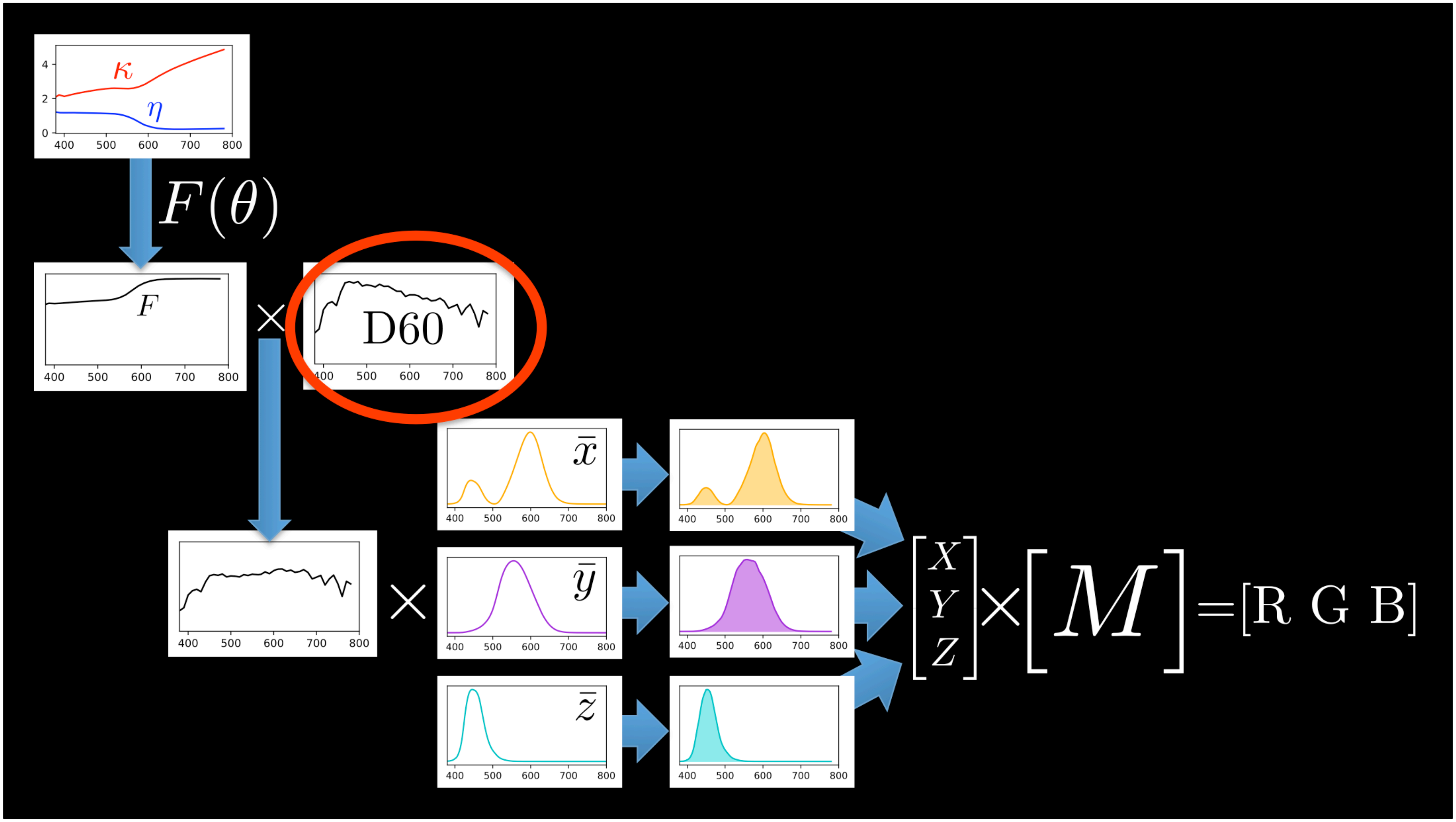
The modified version preserves the spirit of the original, with much better results.

Computing RGB Parameter Values From Spectral IOR Data

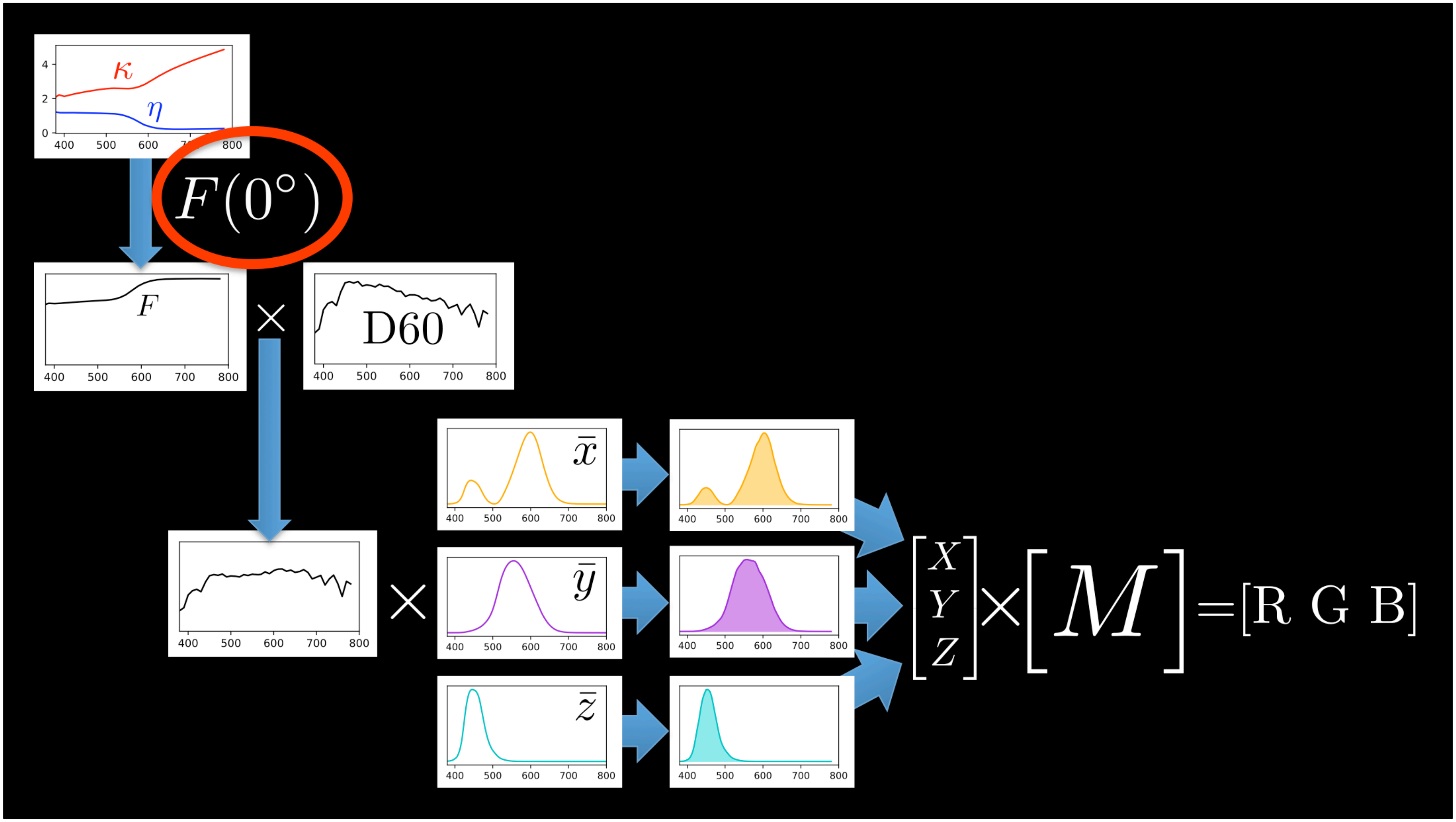
Another topic I didn’t have time to cover is how to compute RGB parameter values from measured spectral data. Although, as I mentioned, the value of laboratory measurements is often overestimated, they can sometimes be useful to generate material “presets” for painters to use as a starting point.



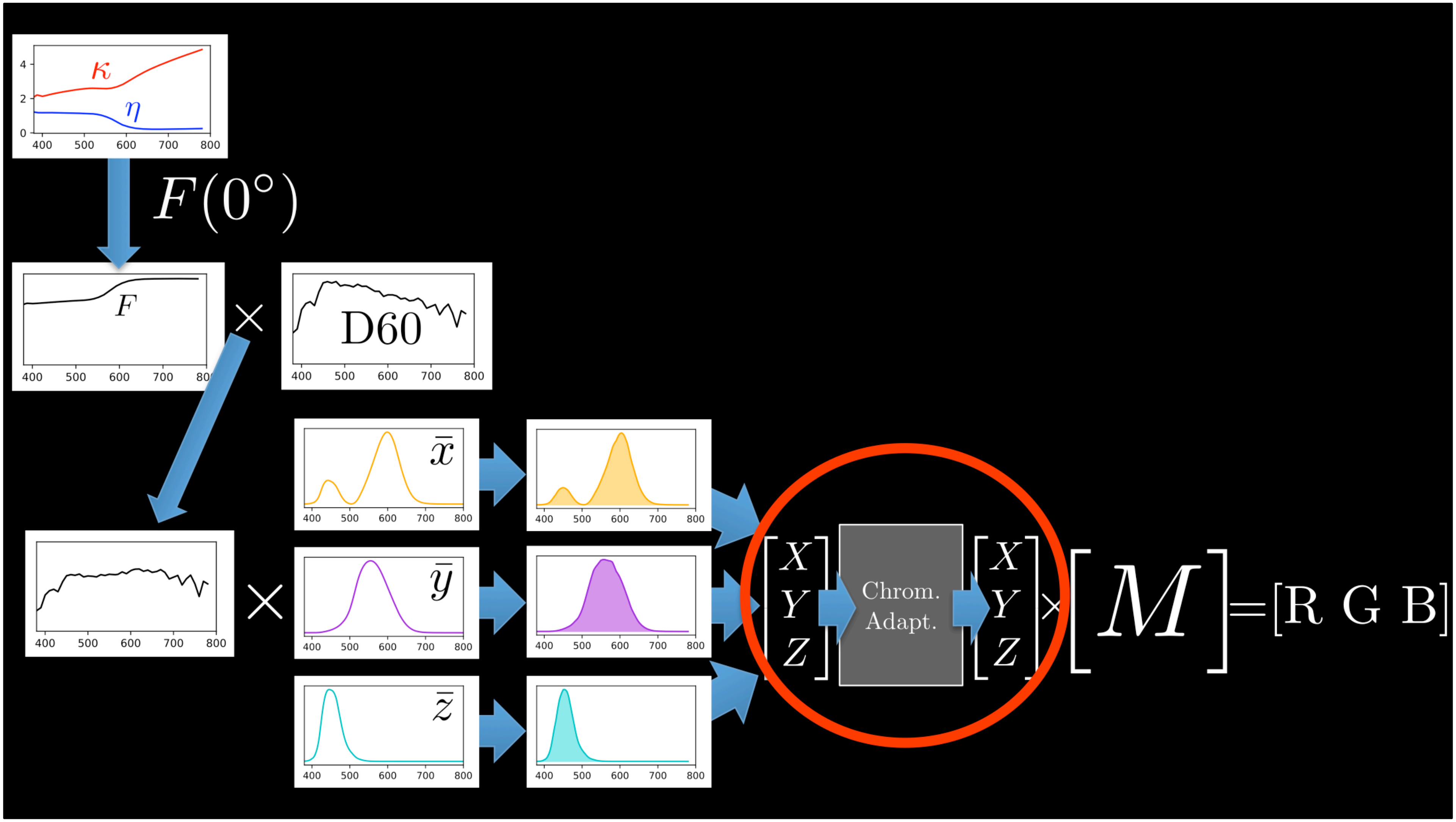
The simplest case is a parameter that represents the reflected color at a given angle, such as F0 or F82. In this case we simply follow the "full procedure" for spectral Fresnel computation detailed earlier, in a specific way.



First, instead of an arbitrary illuminant SPD we use the reference white illuminant for the working color space. In the case of the ACEScg color space, this is D60.

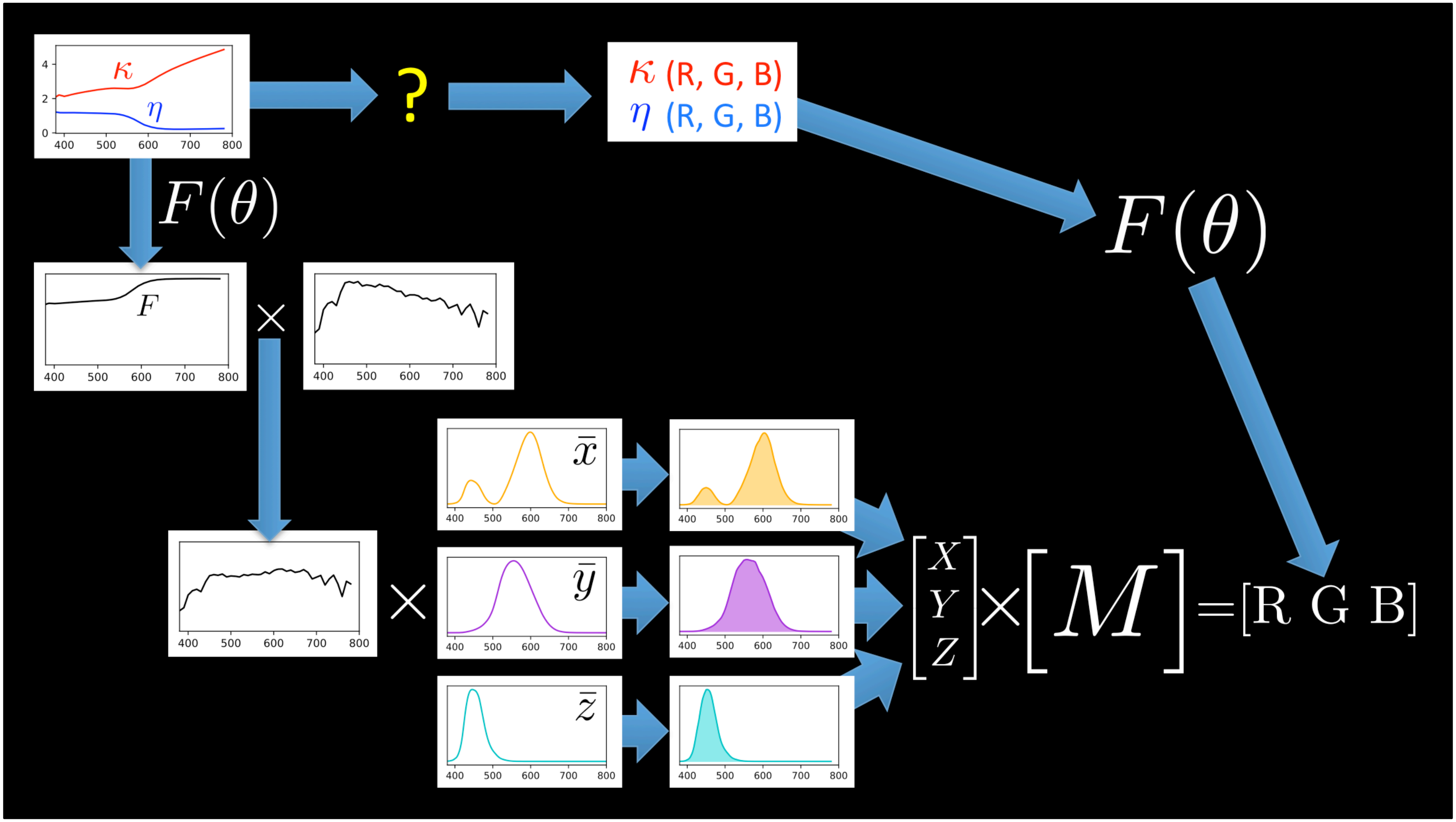


Also, instead of an arbitrary incidence angle we of course use the specific reference angle for the parameter: 0 degrees in the case of F0.

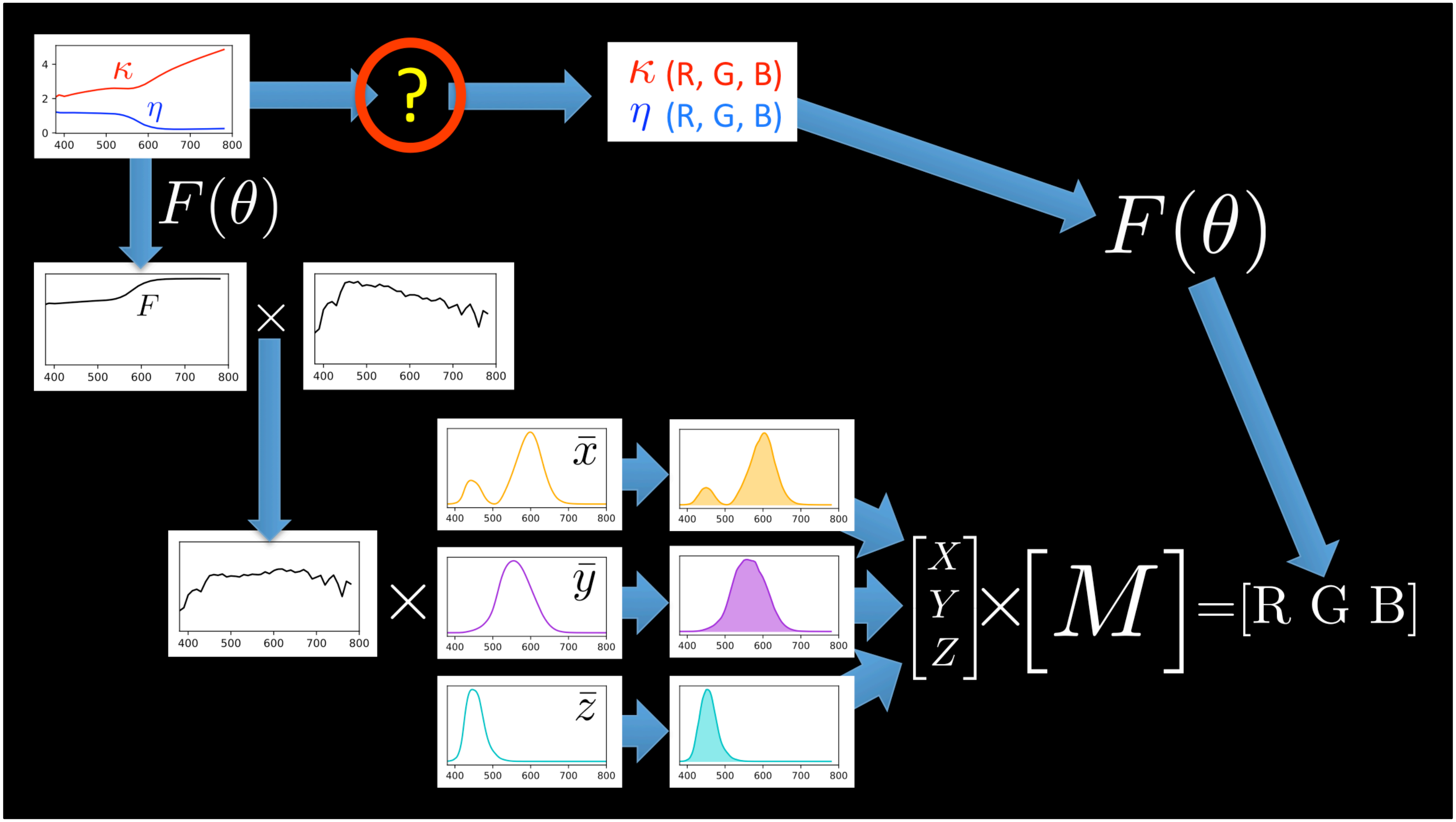


Finally, a few color spaces (such as ACEScg*) have white points that do not match that of their reference illuminant. In this case, a chromatic adaptation step must be applied to the XYZ values before matrix multiplication.

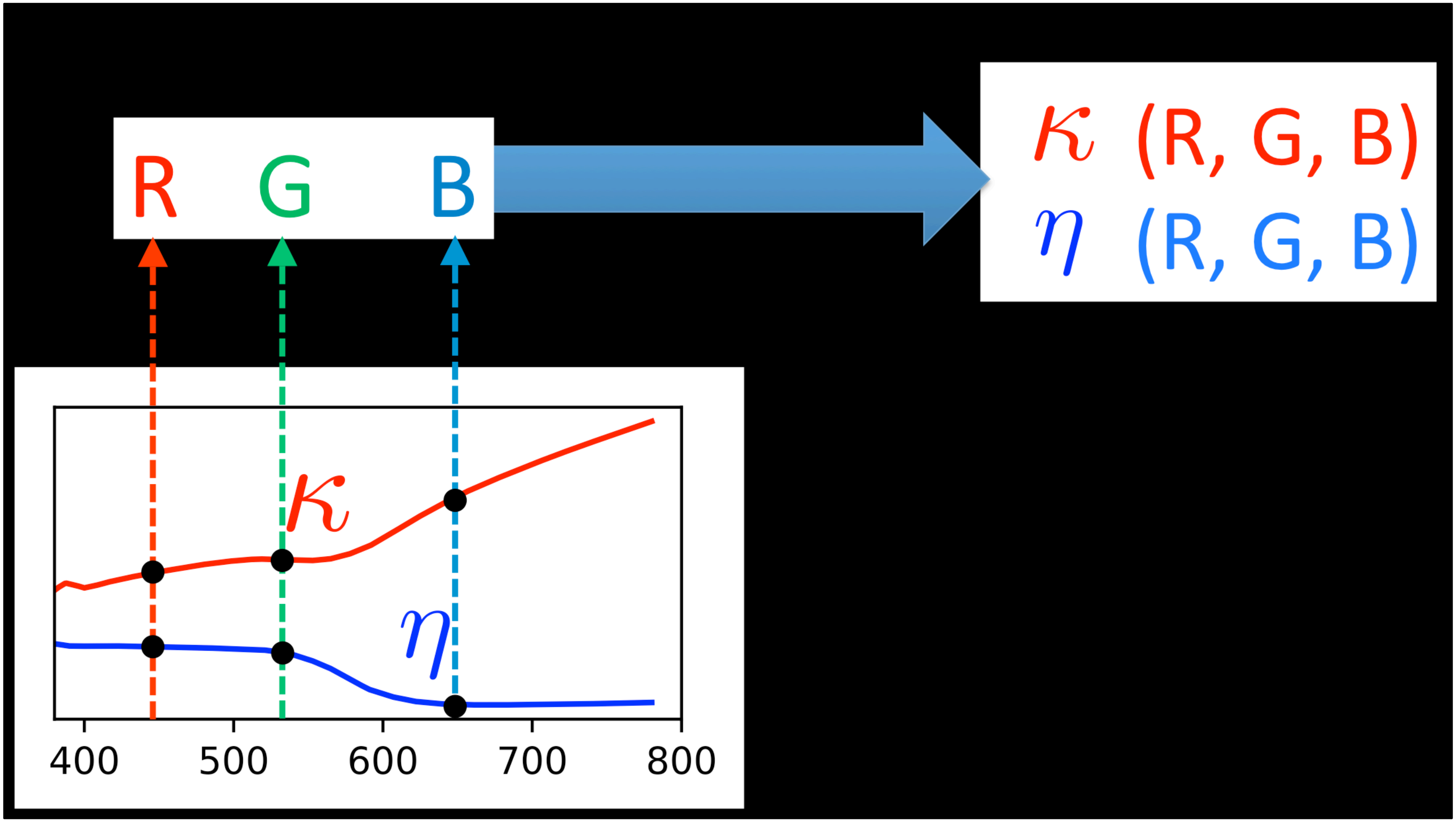
* TB-2018-001: Derivation of the ACES White Point Chromaticity Coordinates. AMPAS Science & Technology Council, 2018.



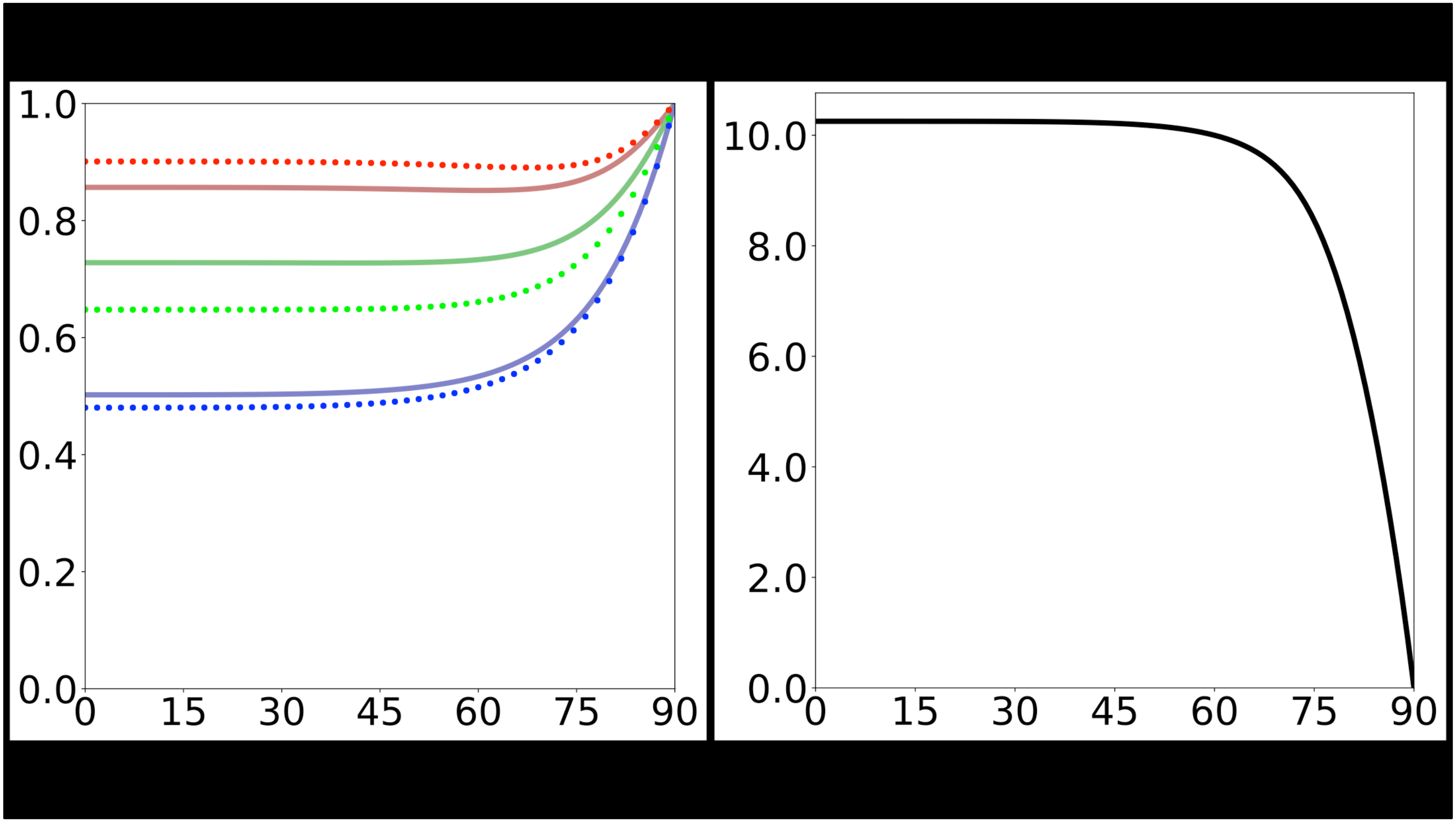
For eta and kappa — or the edge tint g parameter, since they are equivalent — the situation is far less straightforward.



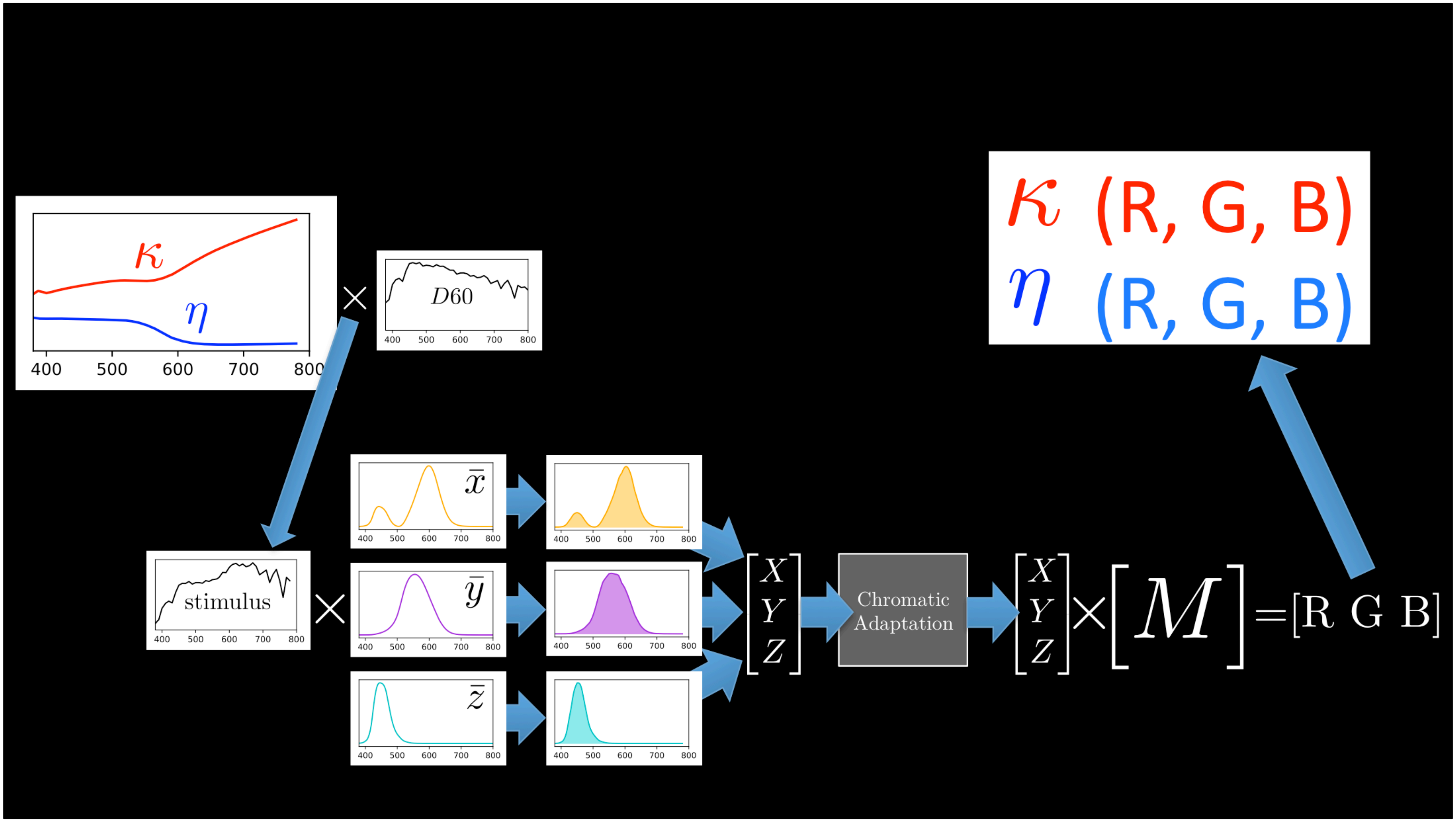
We are in the position of implementing this question mark. This step is represented by a question mark since there is no correct answer. The practical answer we will use is "whatever procedure yields an RGB result closest to the ground truth". Note that there is no guarantee that ANY set of RGB eta, kappa values exists that can produce an exact match. Fortunately, it is possible to get very close, but not easily.



Perhaps the most commonly used approach is point-sampling. In other words, pick representative wavelengths for R, G, and B, and use the eta and kappa values for those wavelengths.

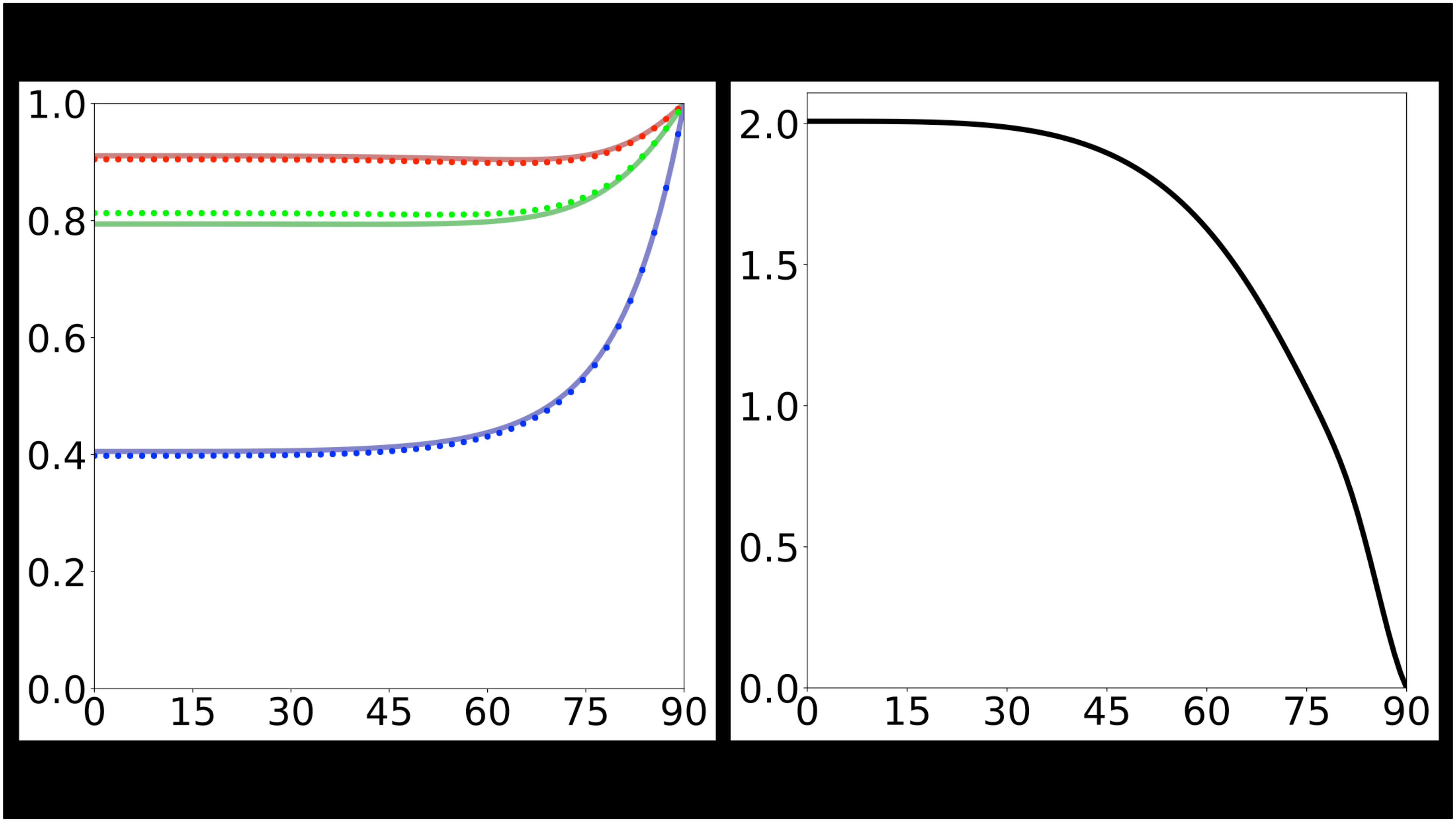


This can lead to quite large errors, significantly larger than the Schlick approximation (this is bronze, with an RGB plot on the left and a CIE delta-E 2000 plot on the right).



Another approach, suggested by Wenzel Jakob in a 2015 SIGGRAPH presentation*, is to process the spectral eta and kappa values as if they were spectral reflectance values.

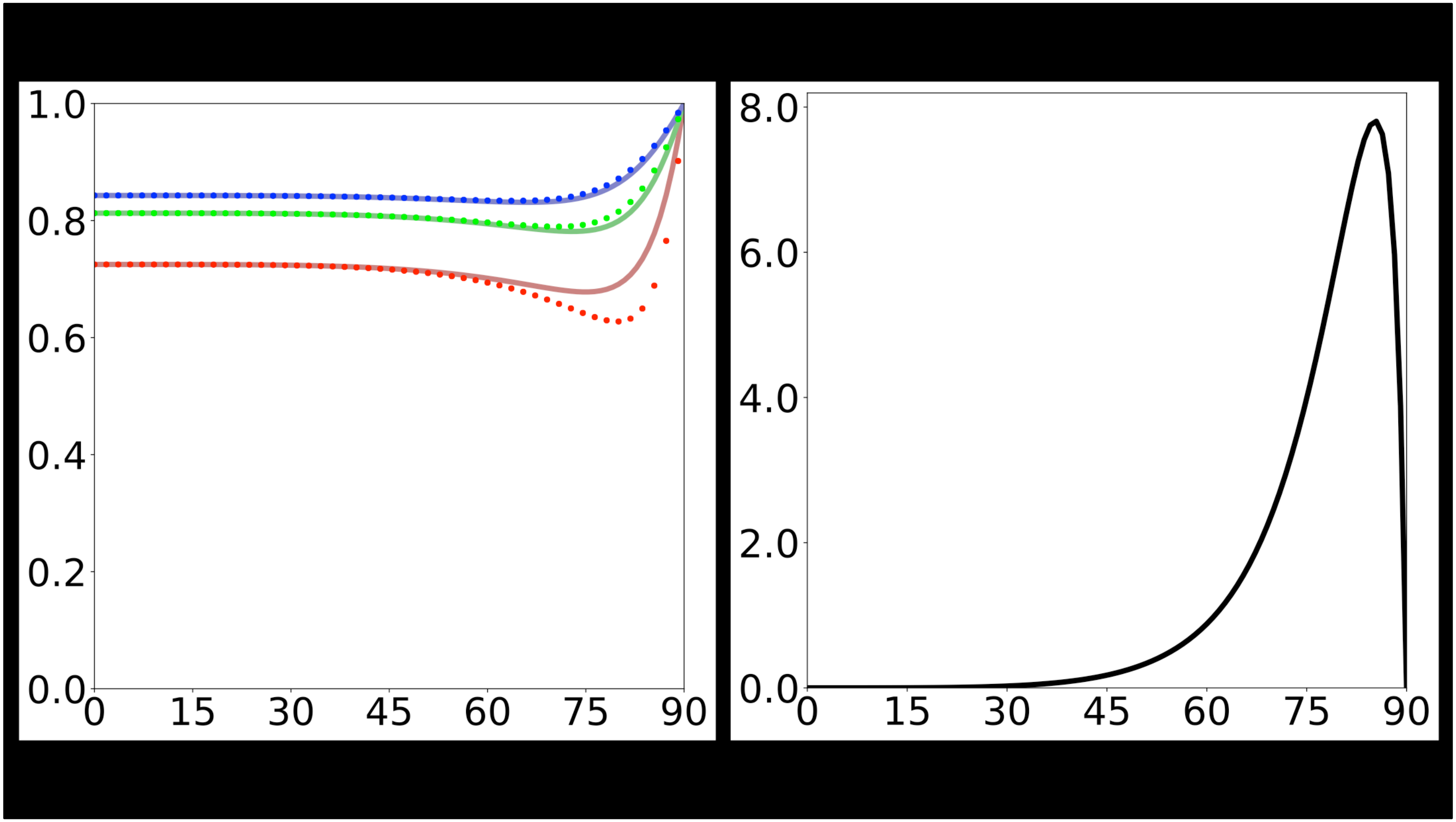
* JAKOB W.: layerlab: A Computational Toolbox for Layered Materials. In Physically Based Shading in Theory and Practice, ACM SIGGRAPH 2015 Courses (2015), SIGGRAPH ’15, ACM. URL: http://selfshadow.com/ publications/s2015-shading-course/



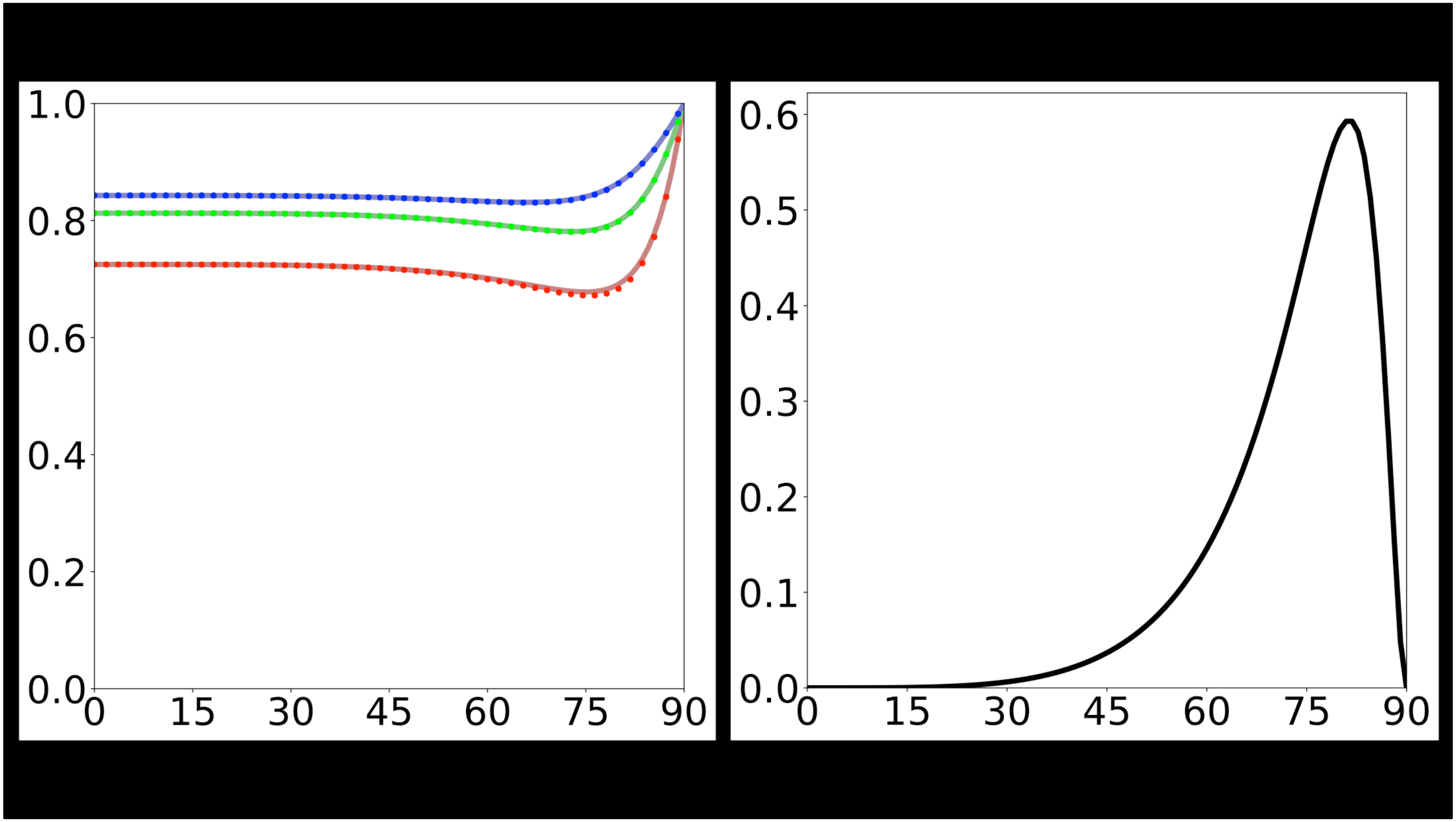
The errors are smaller than with point sampling (here we see plots for Gold), roughly of similar magnitude to the Schlick errors, but more apparent since they occur for facing angles.

$$g(F_0, \eta) = \frac{\dfrac{1+\sqrt{F_0}}{1-\sqrt{F_0}} - \eta}{\dfrac{1+\sqrt{F_0}}{1-\sqrt{F_0}} - \dfrac{1-F_0}{1+F_0}}$$

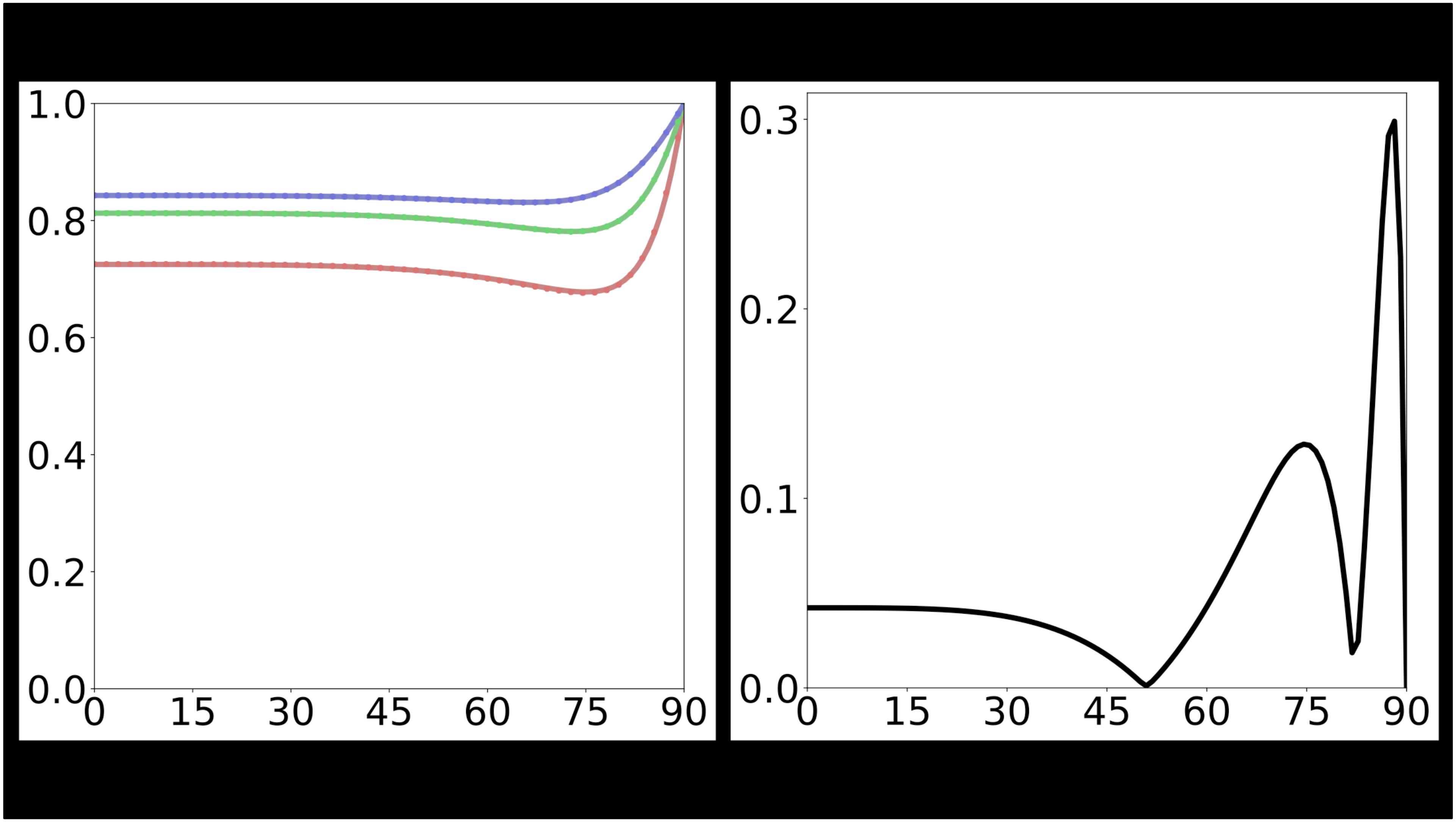
The Gulbrandsen parameterization can help. Compute F0 as shown earlier, and compute RGB values for eta using one of the other methods. Then use this equation to compute the edge tint g from F0 and eta.



Here we point-sample eta, and combine it with the correct value of F0 to compute the edge tint g (here we see plots for Zinc). The reflectivity parameter ensures that the facing angles have the correct color. There are still some large errors at glancing angles. Overall the error here is about as bad as Schlick on average; larger for some metals, smaller for others.



And here we process eta as if it were spectral reflectance (using color-matching functions, etc.), and combine it with the correct value of F0 to get the value of the edge tint g (here again we have plots for Zinc). Here the delta-E error is well under the visibility threshold. However, getting this level of accuracy requires careful processing, including chromatic adaptation and an extra normalization step to account for rounding errors in the RGB matrix. Note that this approach still isn't physically meaningful.



Since all approaches are unprincipled, the "correct" method is whatever yields the smallest error. It turns out that we can get the smallest error by forgetting about trying to compute the values in any meaningful way and simply doing a "black box" numerical fitting (here we have Zinc yet again).

Metal	Schlick	PS η, κ	Refl. η, κ	Refl. F_0 PS η	Refl. F_0 Refl. η	Fitted η, κ
Silver	0.27	0.99	0.19	0.08	0.04	0.02
Copper	0.69	8.49	0.90	2.26	0.06	0.03
Gold	0.69	6.05	1.70	1.23	0.07	0.05
Bronze	0.78	9.47	0.15	1.52	0.05	0.02
Brass	1.04	5.49	0.39	0.62	0.08	0.04
Aluminum	1.12	1.09	0.06	0.69	0.02	0.01
Palladium	1.70	2.61	0.07	0.46	0.03	0.02
Nickel	1.82	3.80	0.14	0.65	0.03	0.02
Titanium	1.87	3.34	0.02	0.96	0.02	0.01
Mercury	1.89	0.97	0.11	1.00	0.04	0.02
Platinum	2.08	3.01	0.10	0.58	0.03	0.02
Iron	2.42	0.65	0.15	0.54	0.03	0.01
Zinc	2.44	8.67	1.55	2.70	0.23	0.08
Chromium	2.90	0.36	0.13	0.24	0.05	0.02

Here are the average error for all measured metals and fitting approaches. The first column shows the error between Schlick and ground truth, and the others use Fresnel equations with different methods for computing the parameters. "PS" means point-sampling, and "Refl." means processing the parameter in question as if it were a reflectance value. The takeaway is that for the Fresnel equations to be consistently more accurate than Schlick, the parameter values need to be carefully calculated, ideally numerically fitted. If fitting isn't feasible, then computing F0 and eta as reflectance parameters also comes pretty close.