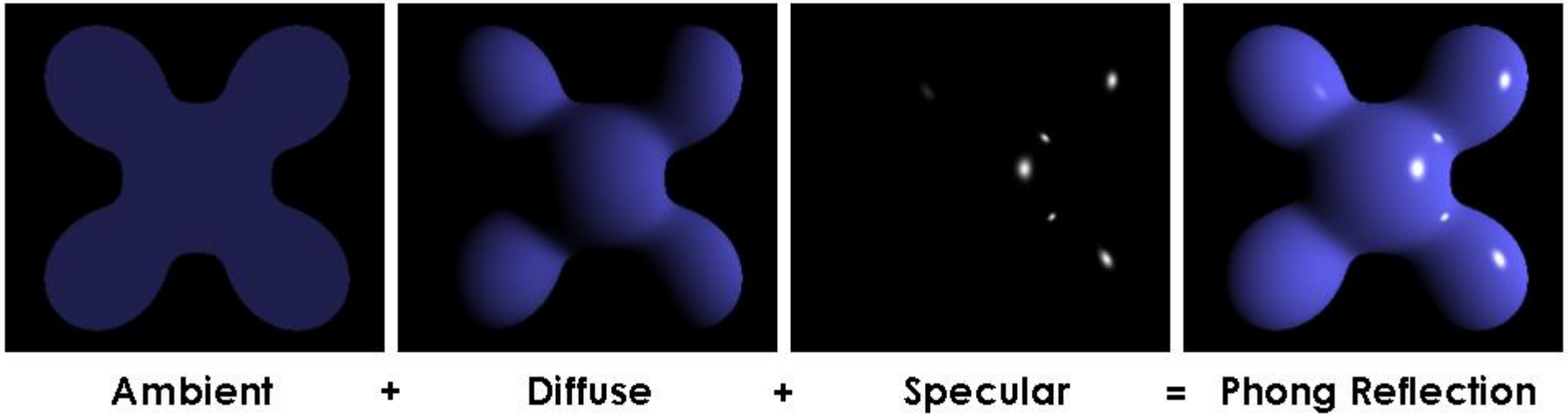
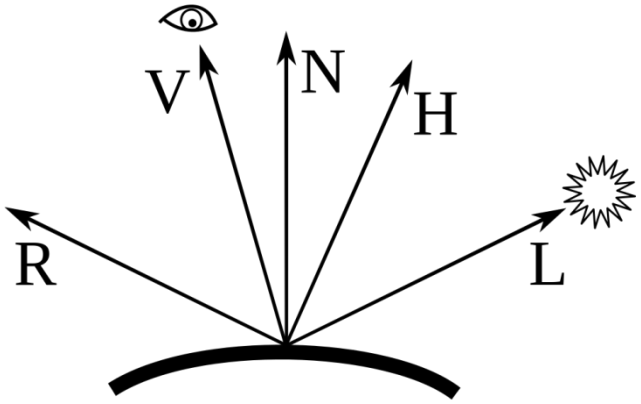
GRK 7

Dr Wojciech Palubicki

Phong model of lighting





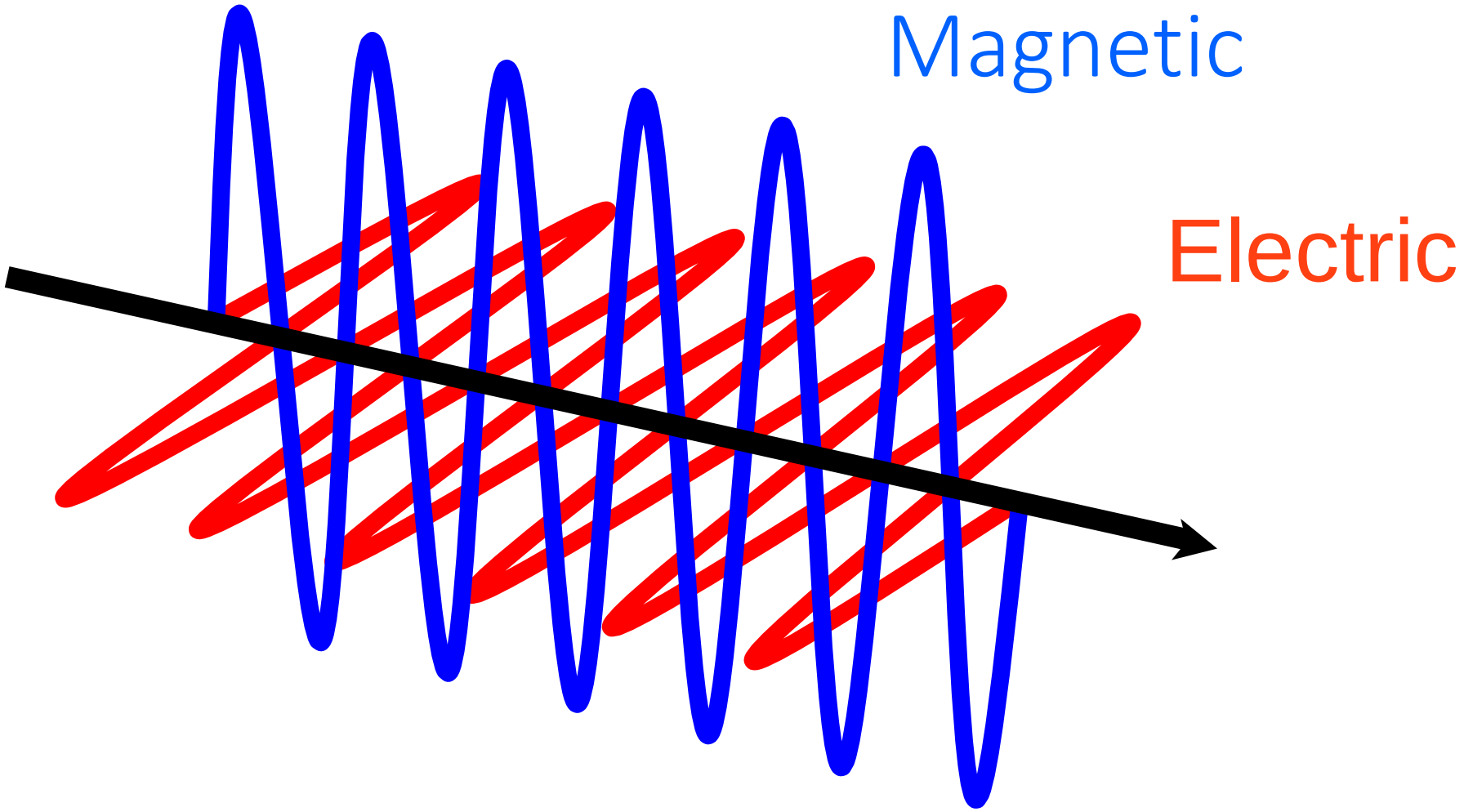

PHONG


PHONG
PBR

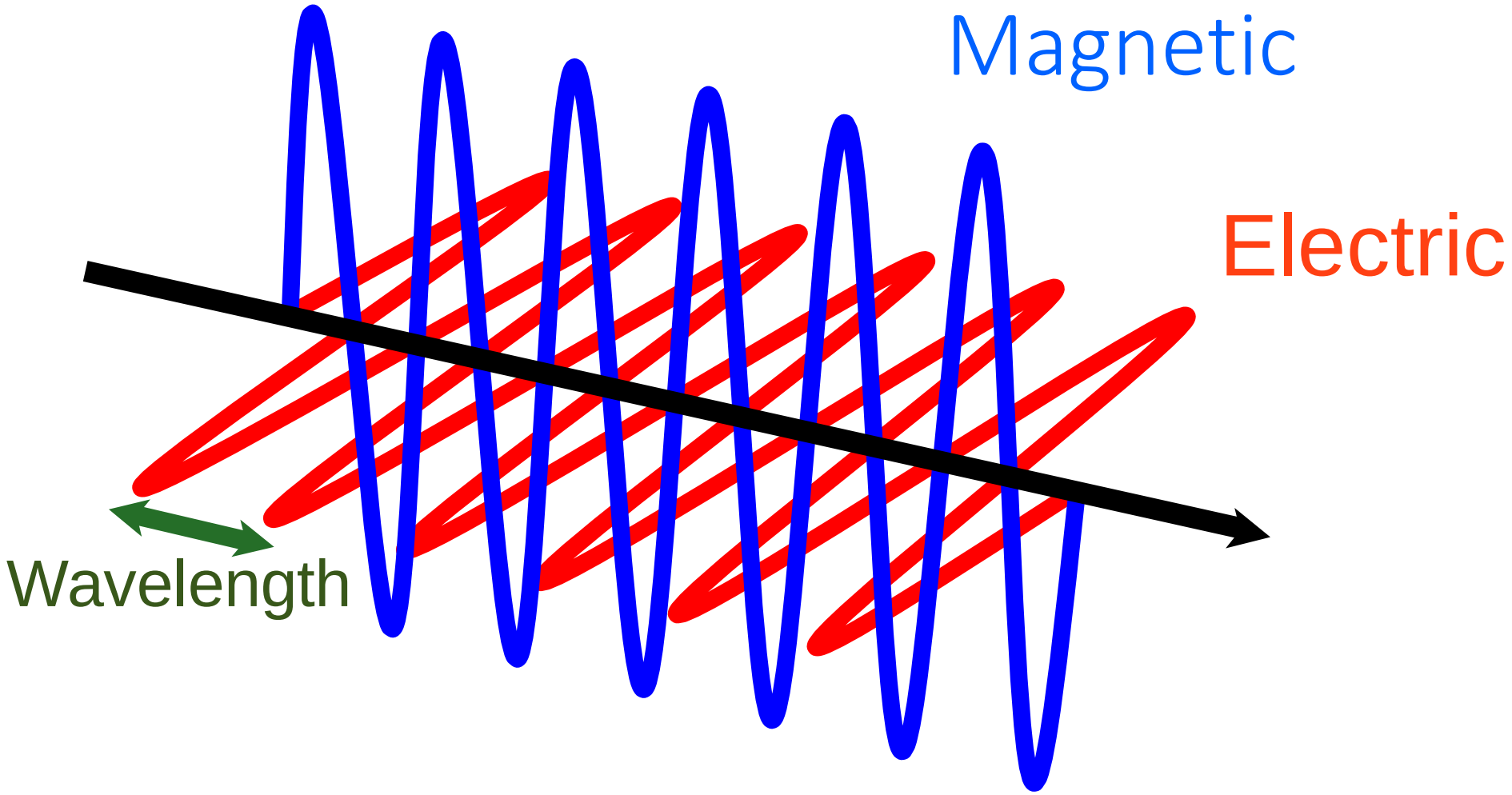
Physically based rendering (PBR)

- “Real-Time Rendering, 3rd Edition”, A K Peters 2008
- Physics of Light
- Geometric Optics
- Mathematical description for real-time lighting (micro-facet BRDF)

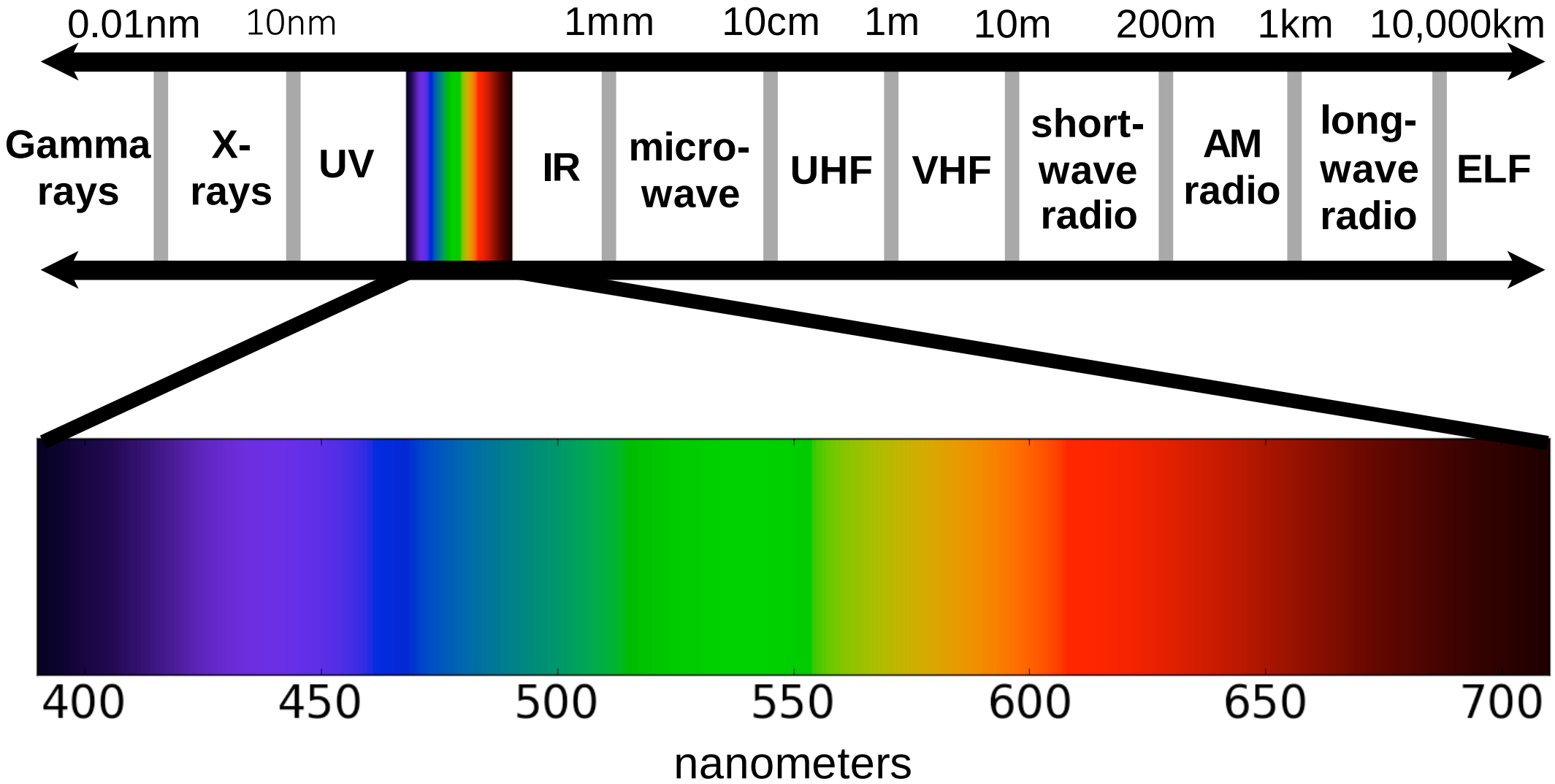
Light – physical point of view

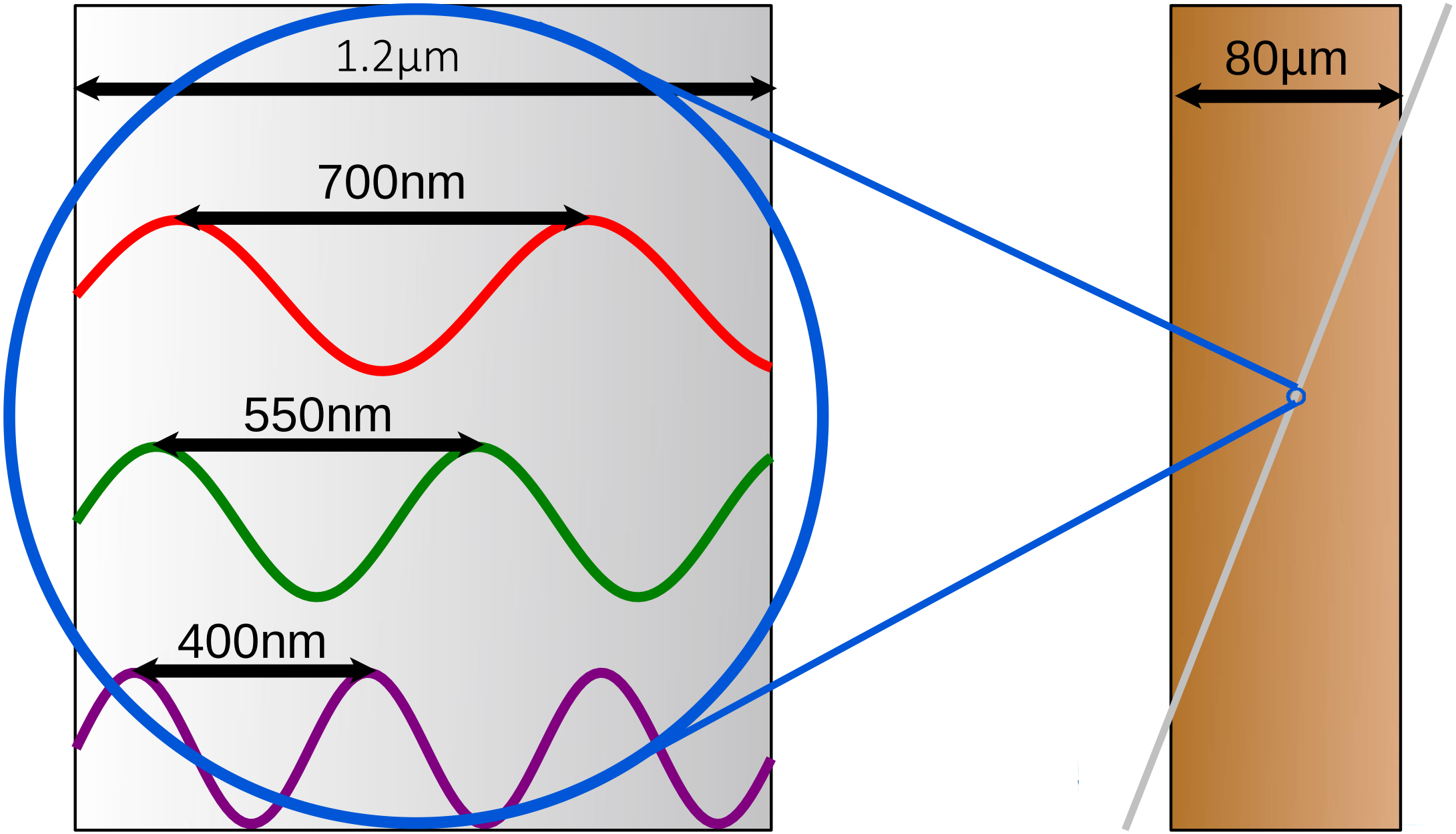


Light – physical point of view

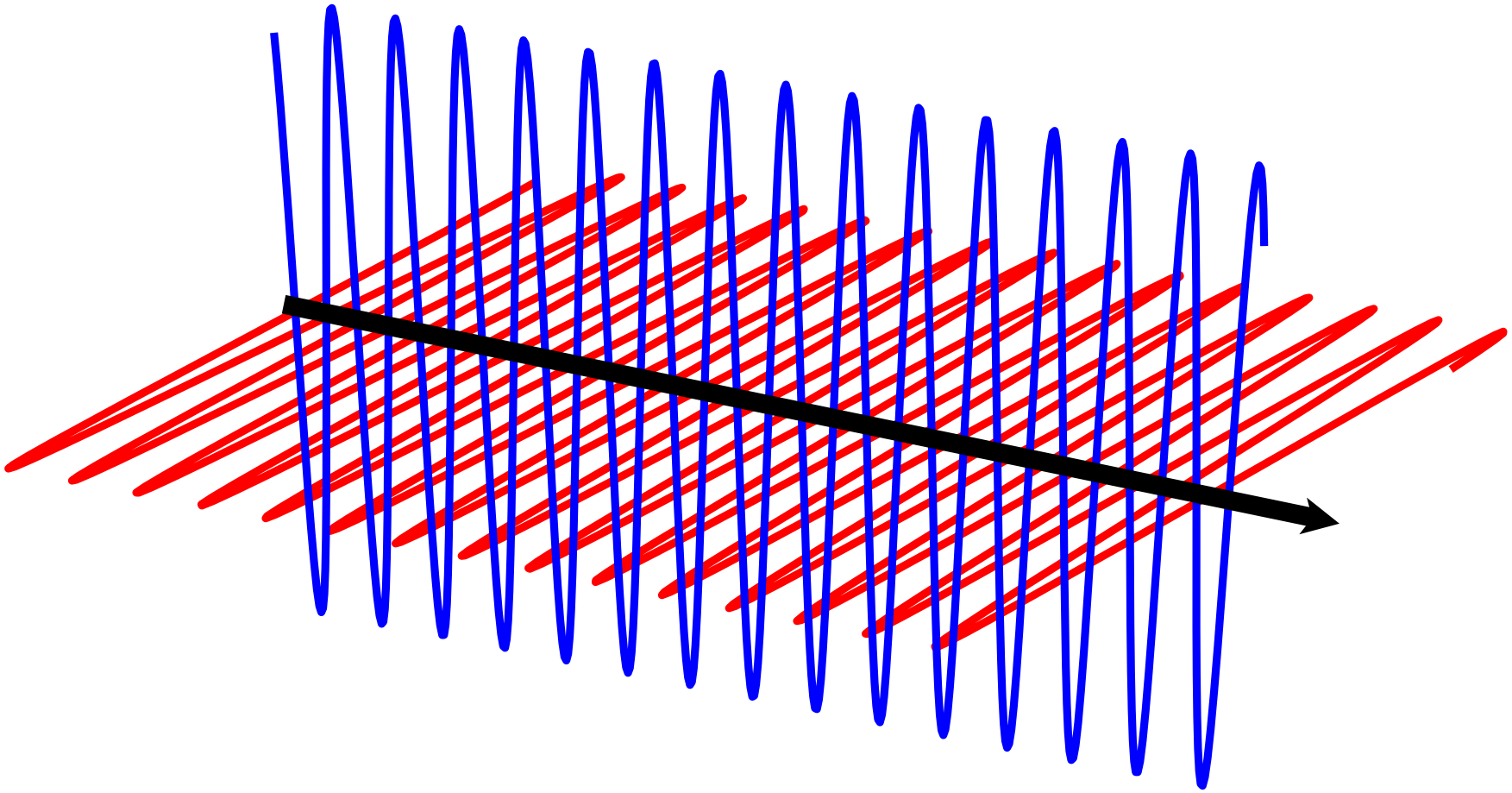


Electromagnetic wavelengths

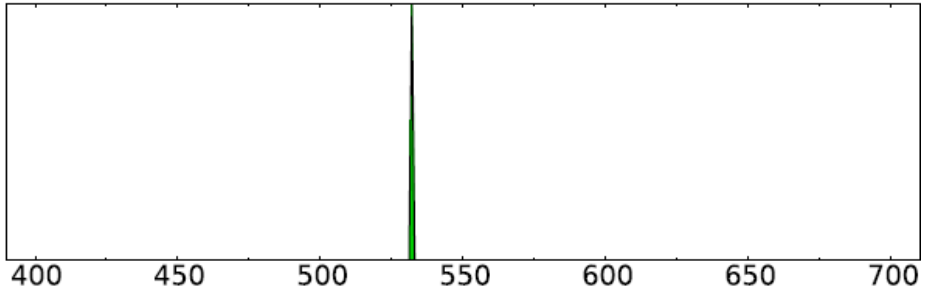



1.2µm
700nm
550nm
400nm
80µm

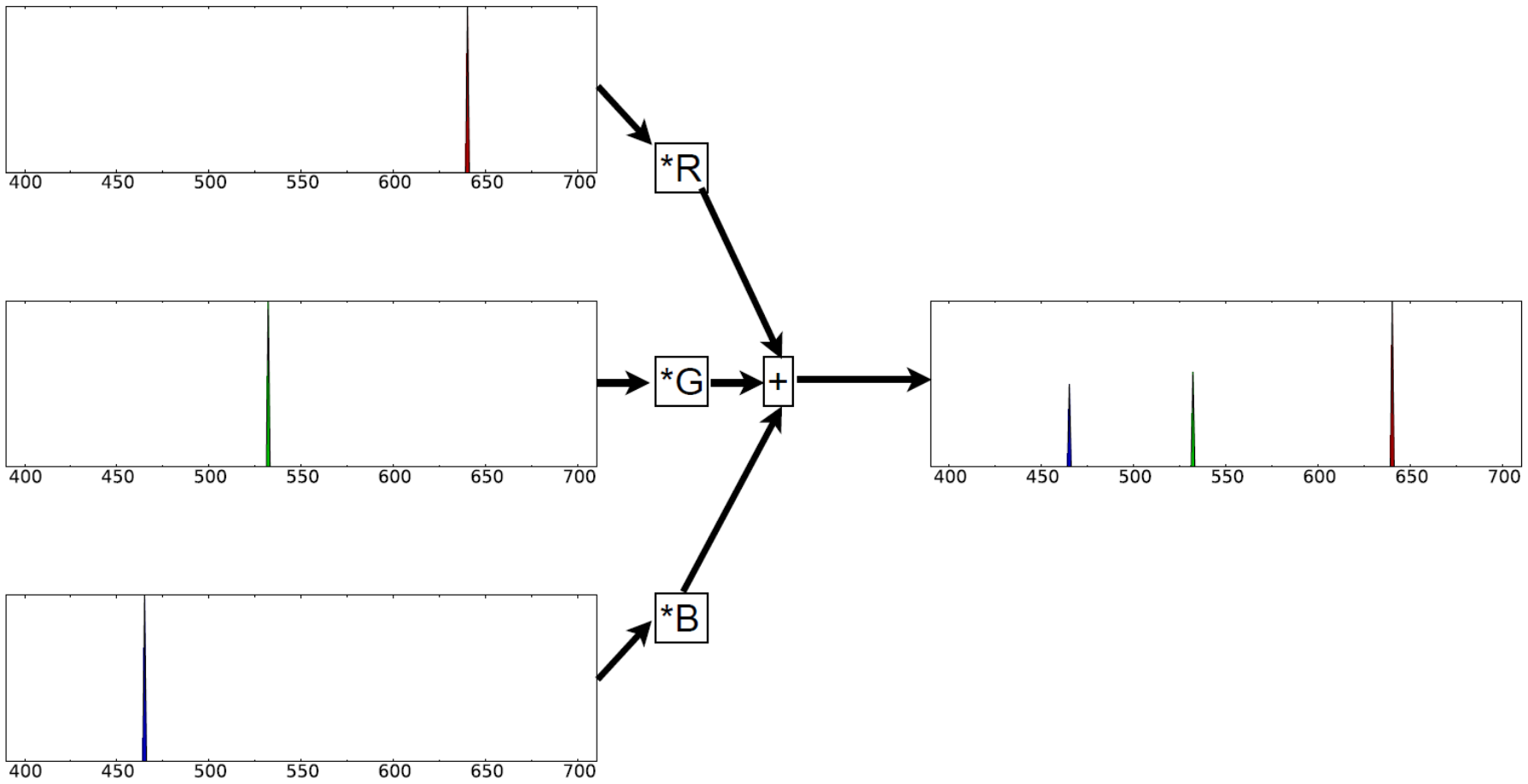
Wavelengths



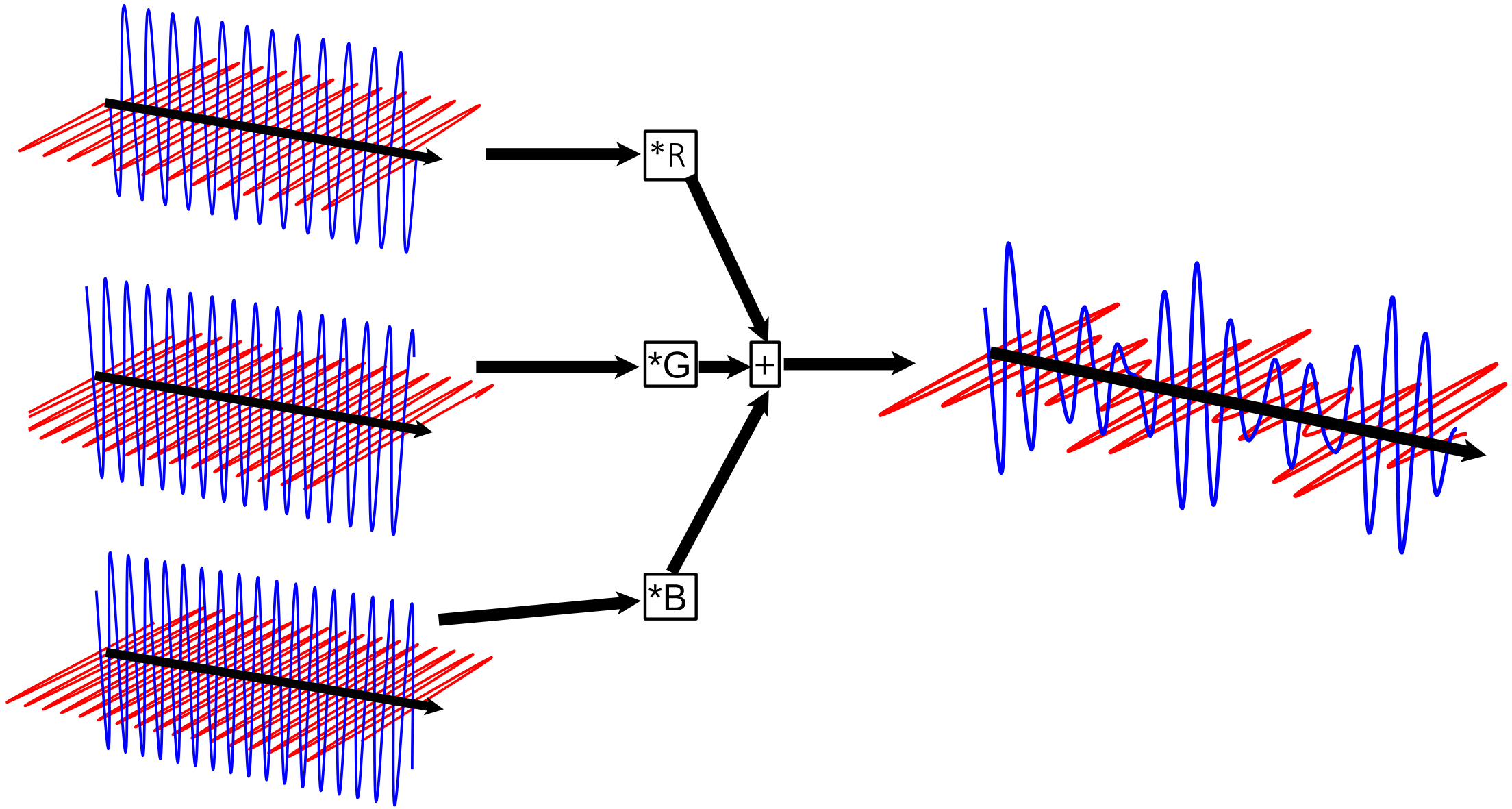
Spectral Power Distribution (SPD)



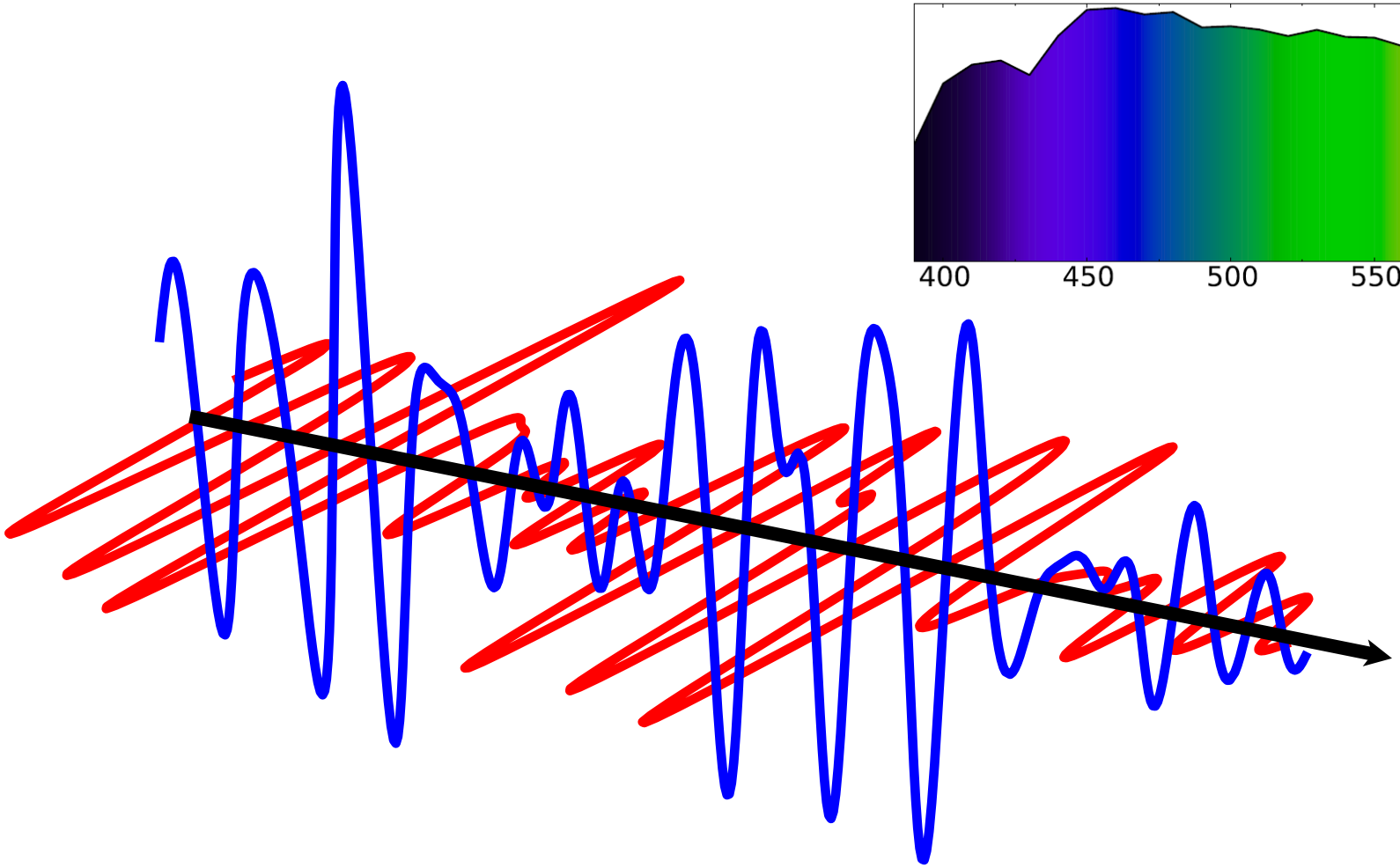
Example: RGB Laser Projector

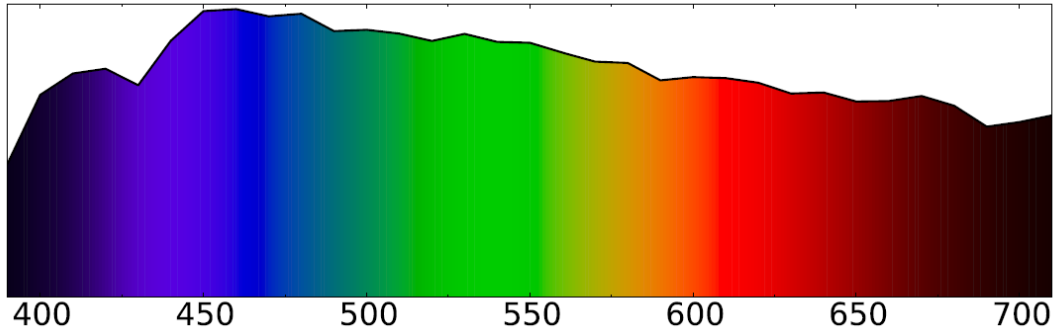


Resulting wave form

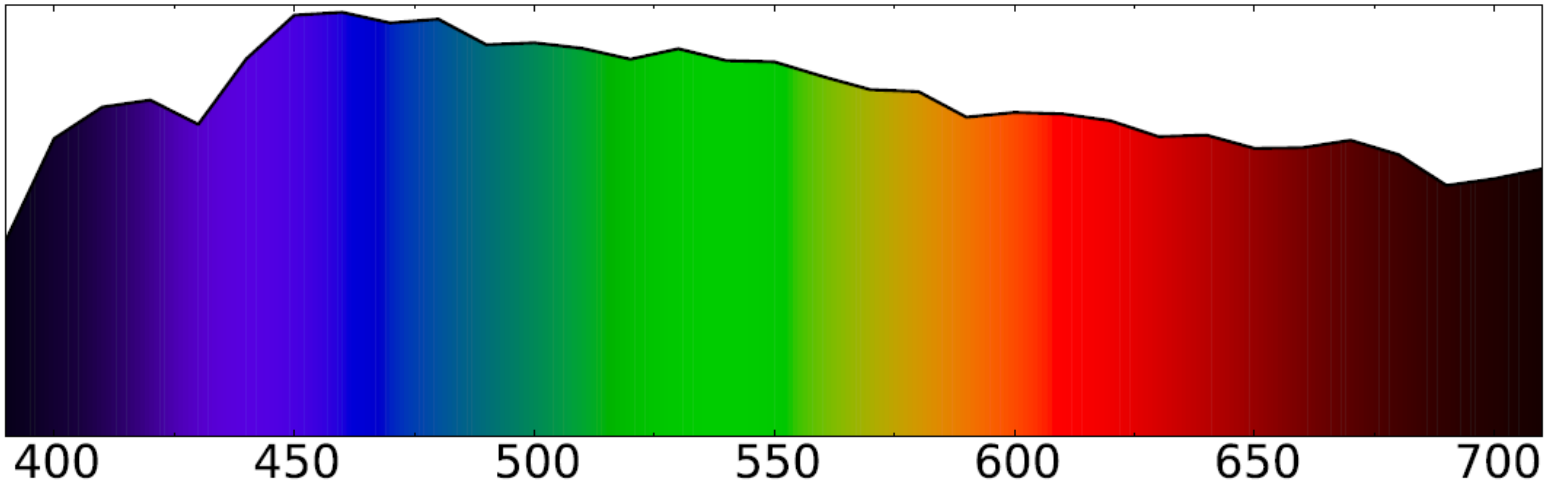


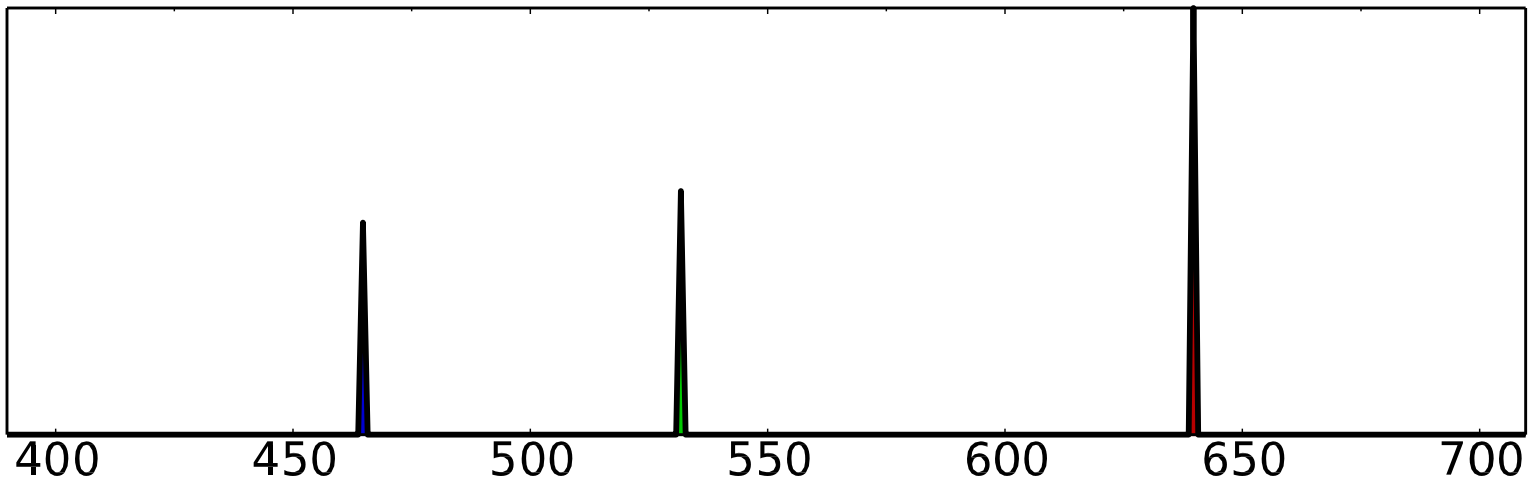
White light wave form



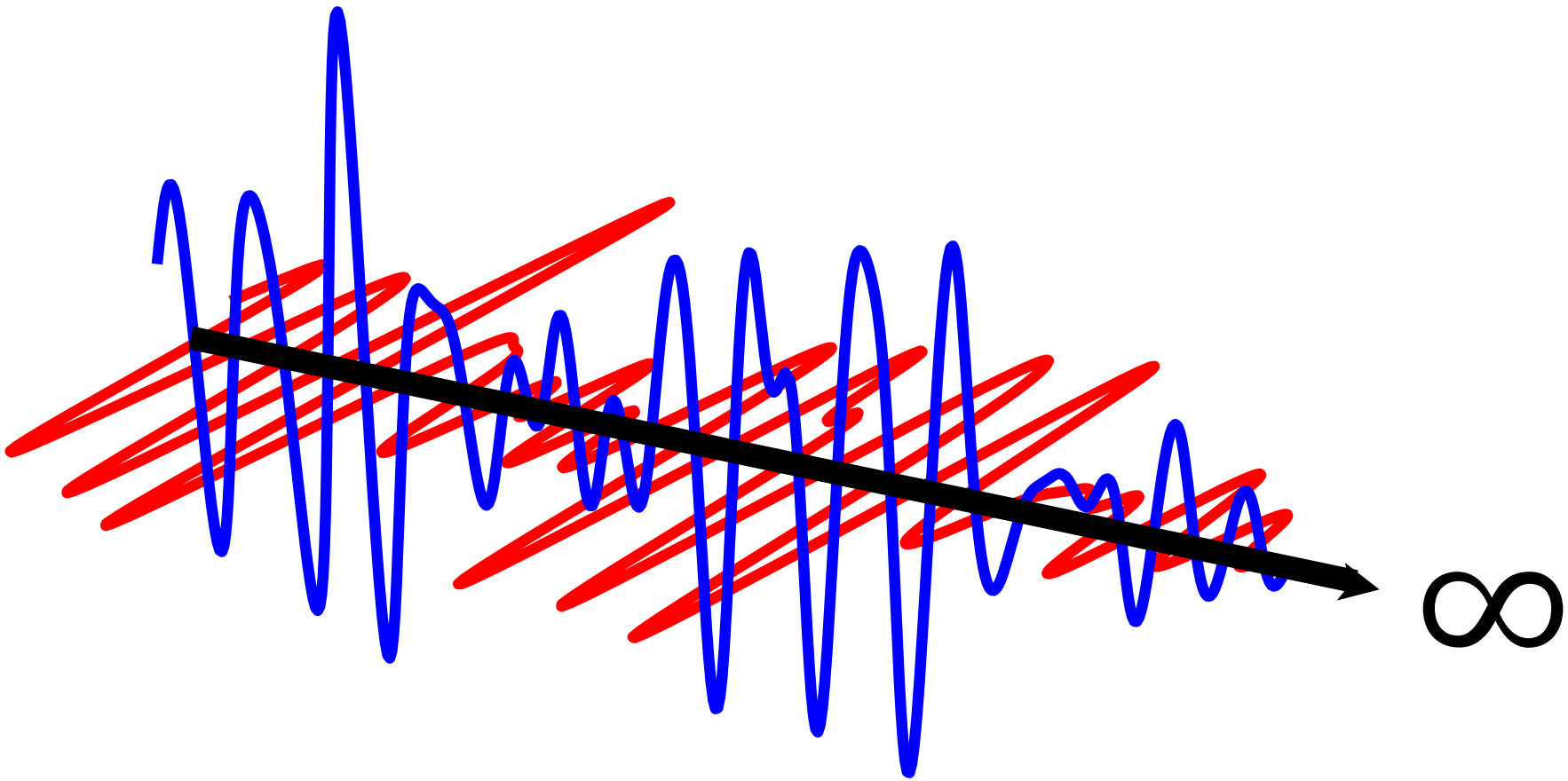


White light and laser projector light comparison

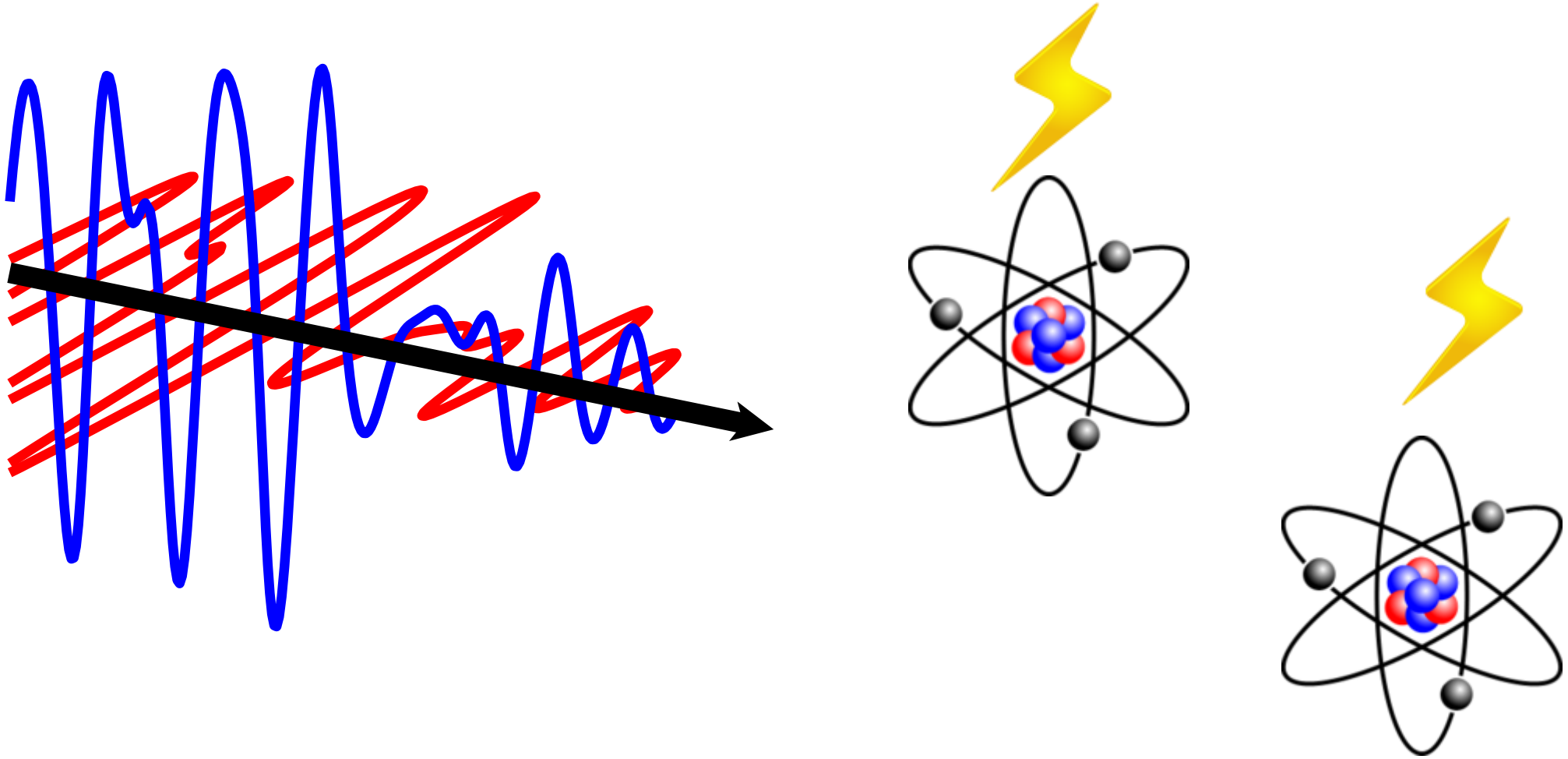




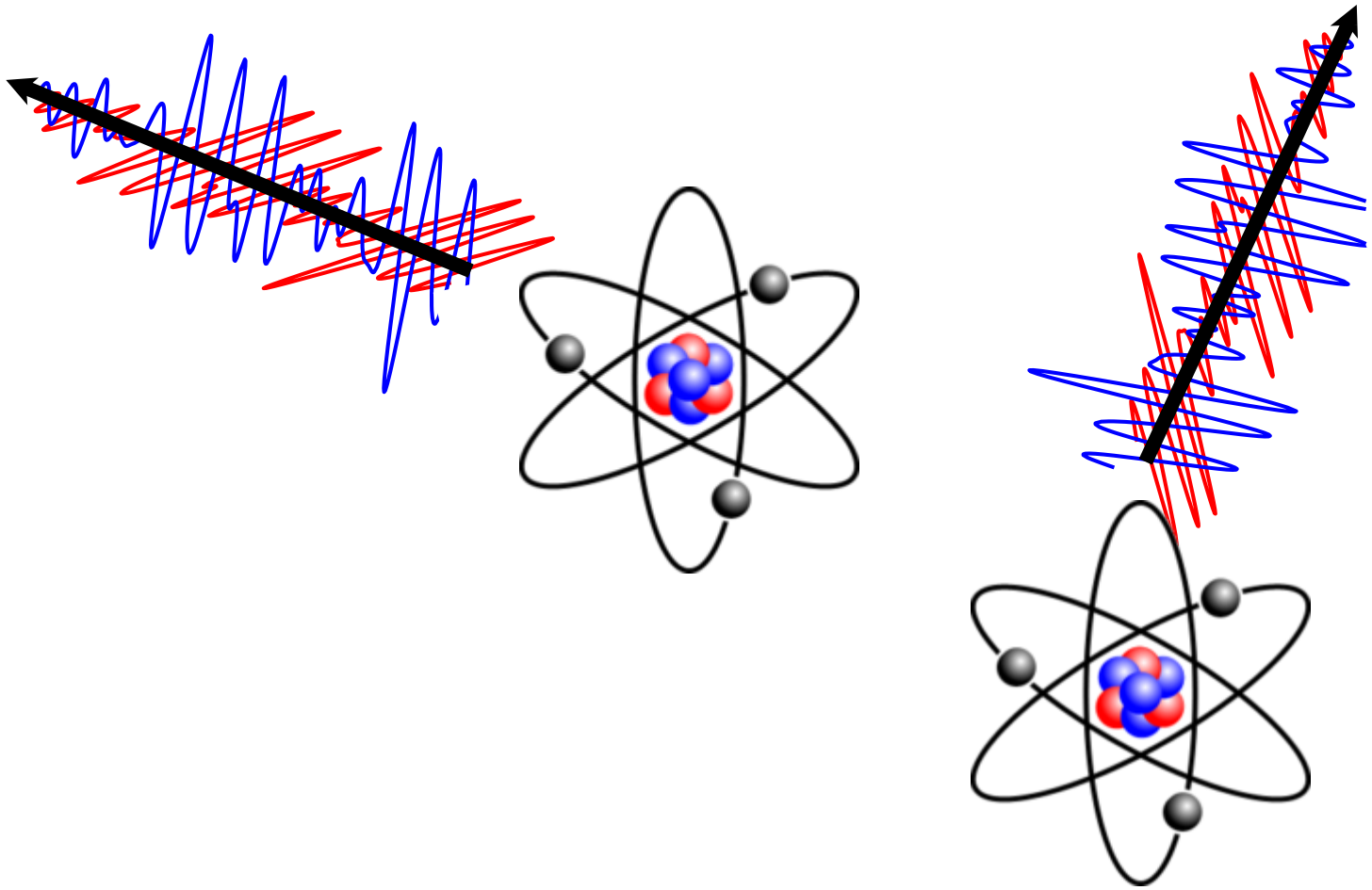
In a vacuum light propagates to infinity



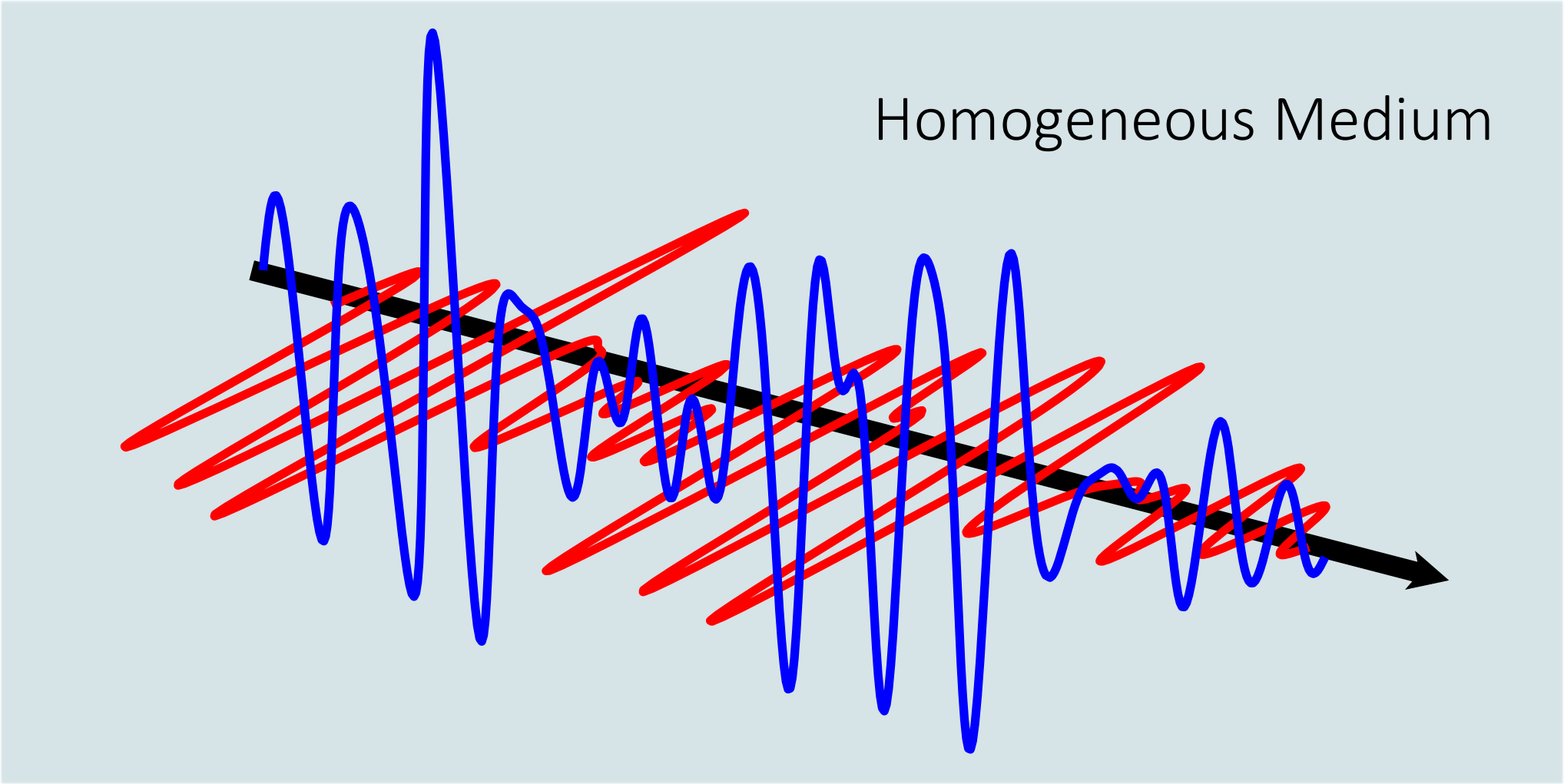
When interacting with atoms it energizes them



This energy is absorbed and re-radiated as light

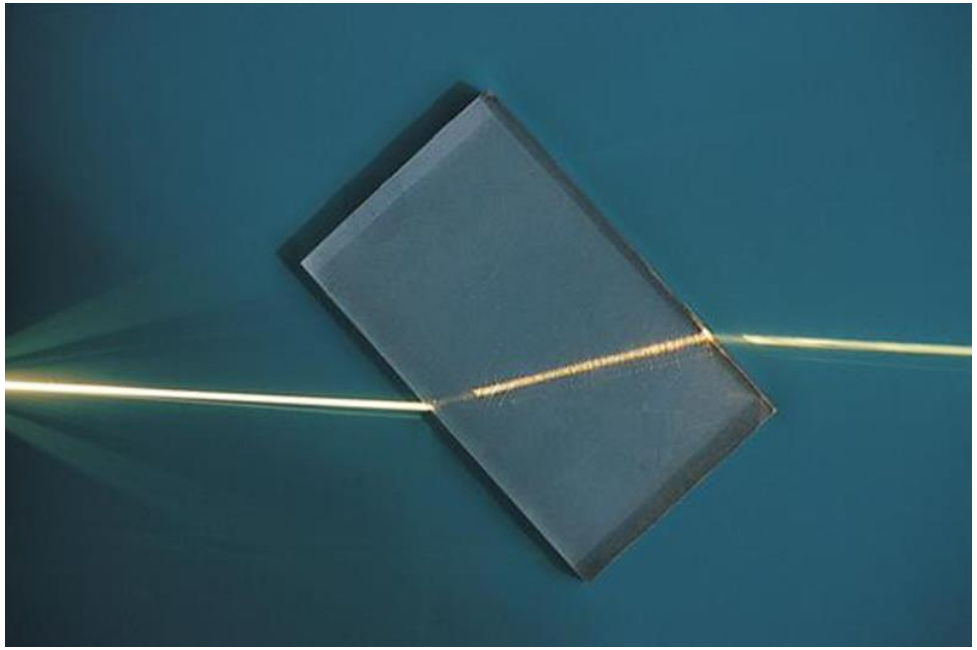


Simplification: Wave Optics

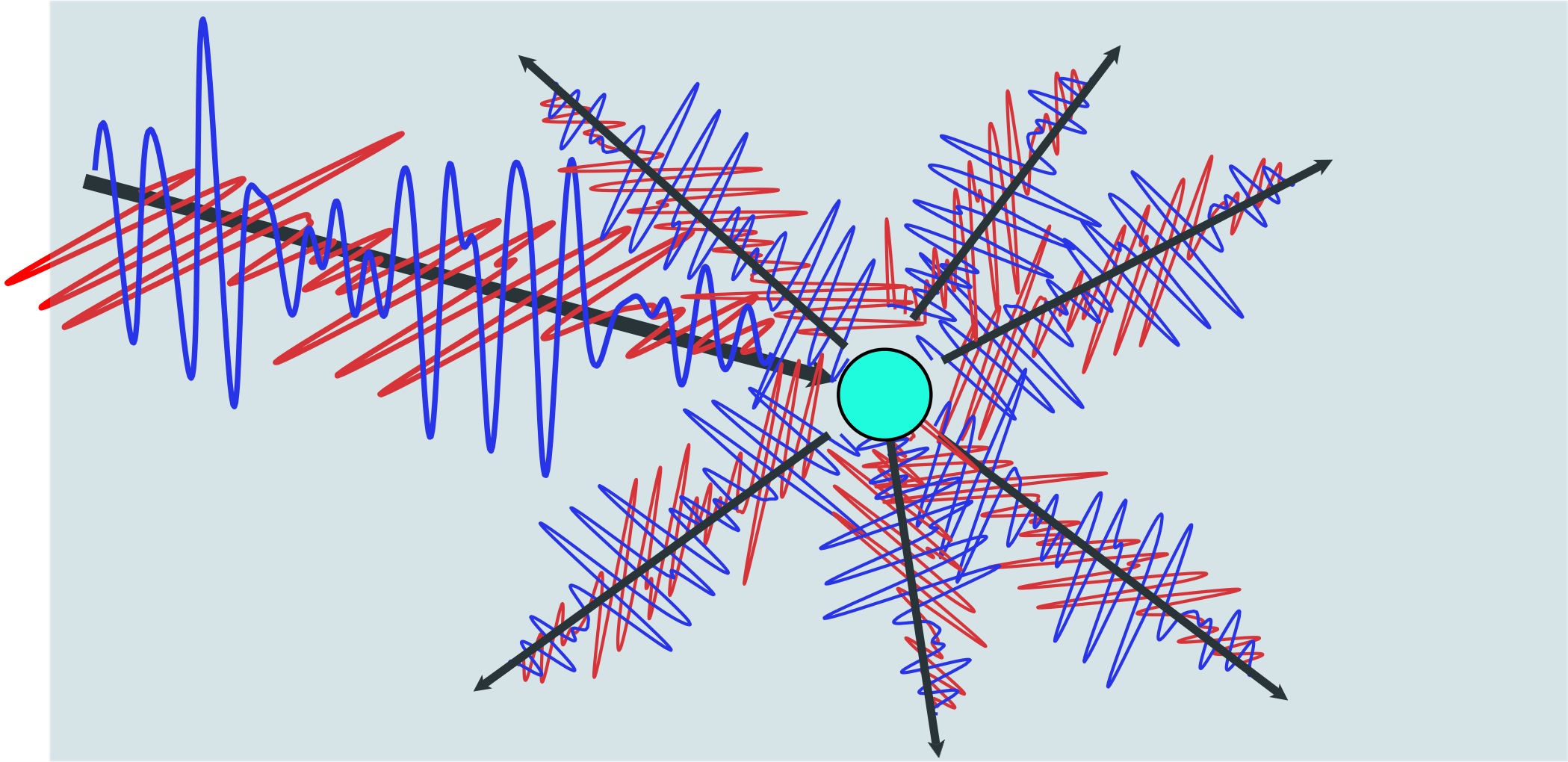


Refractive Index n (dimensionless)

- $n = \frac{c}{v}$
- c is the speed of light in vacuum
- v is the phase velocity of light in the medium
- Measures the **absorption** of light by a medium



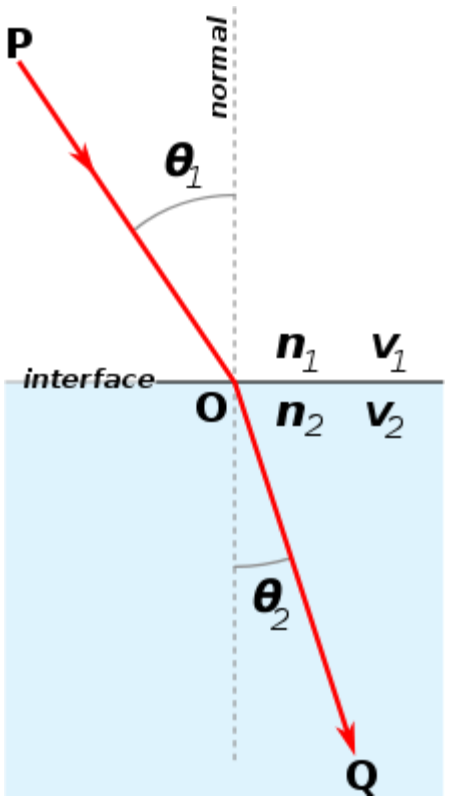
Scattering particle



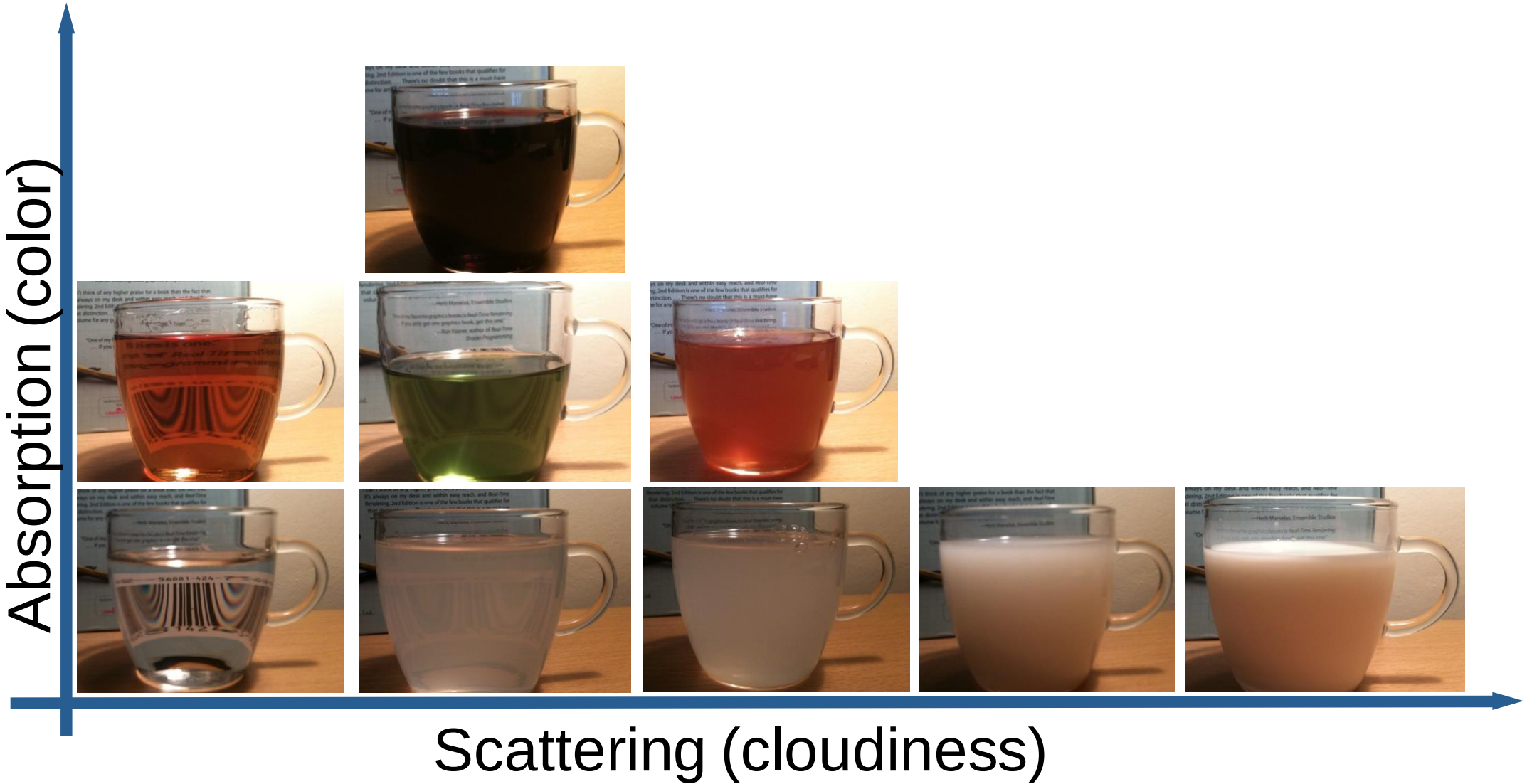
Snell's Law

- describes the relationship between the angles of incidence and refraction, when referring to light or other waves passing through a boundary between two different isotropic media, such as water, glass, or air

$$\frac{\sin\theta_2}{\sin\theta_1} = \frac{v_2}{v_1} = \frac{n_1}{n_2}$$

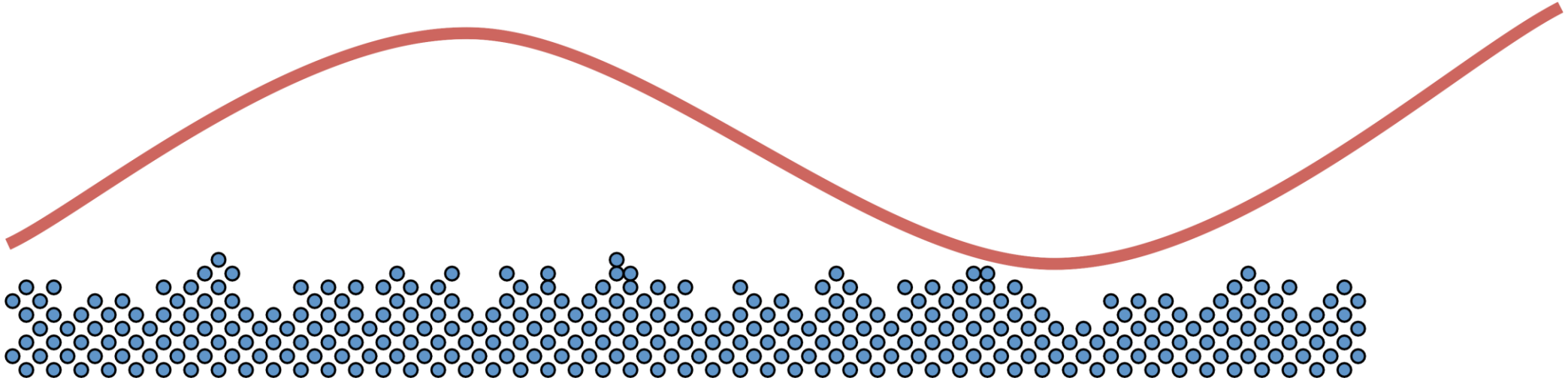


Appearance of a medium

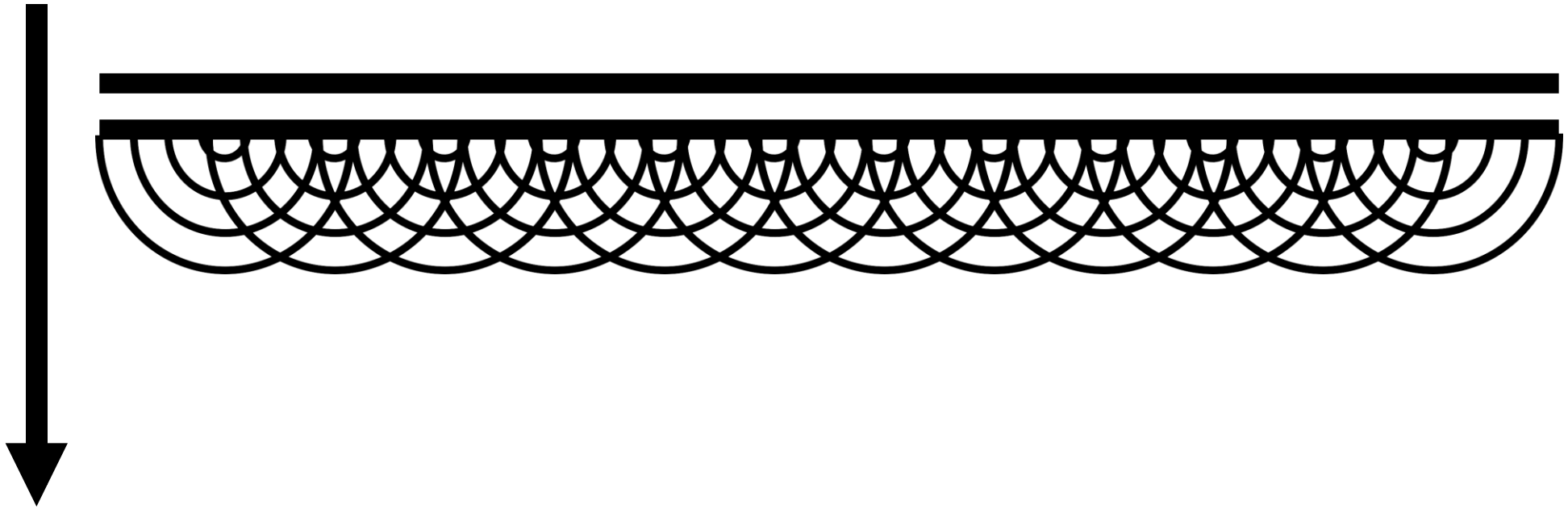


Object surfaces

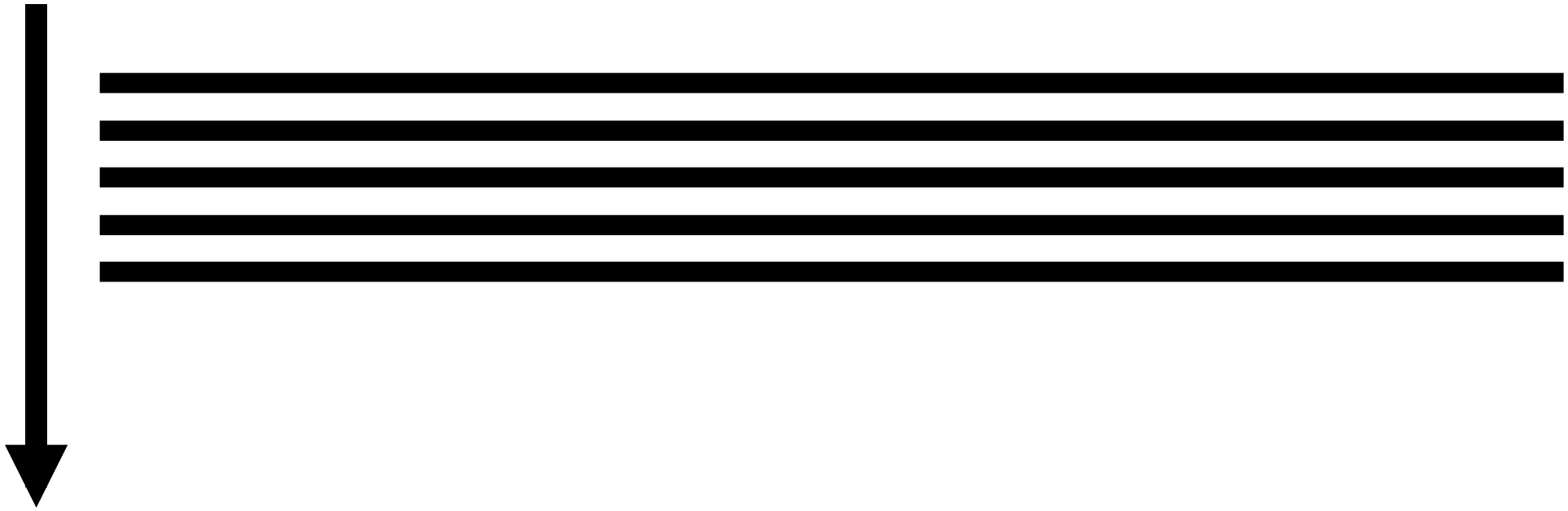
Nanogeometry



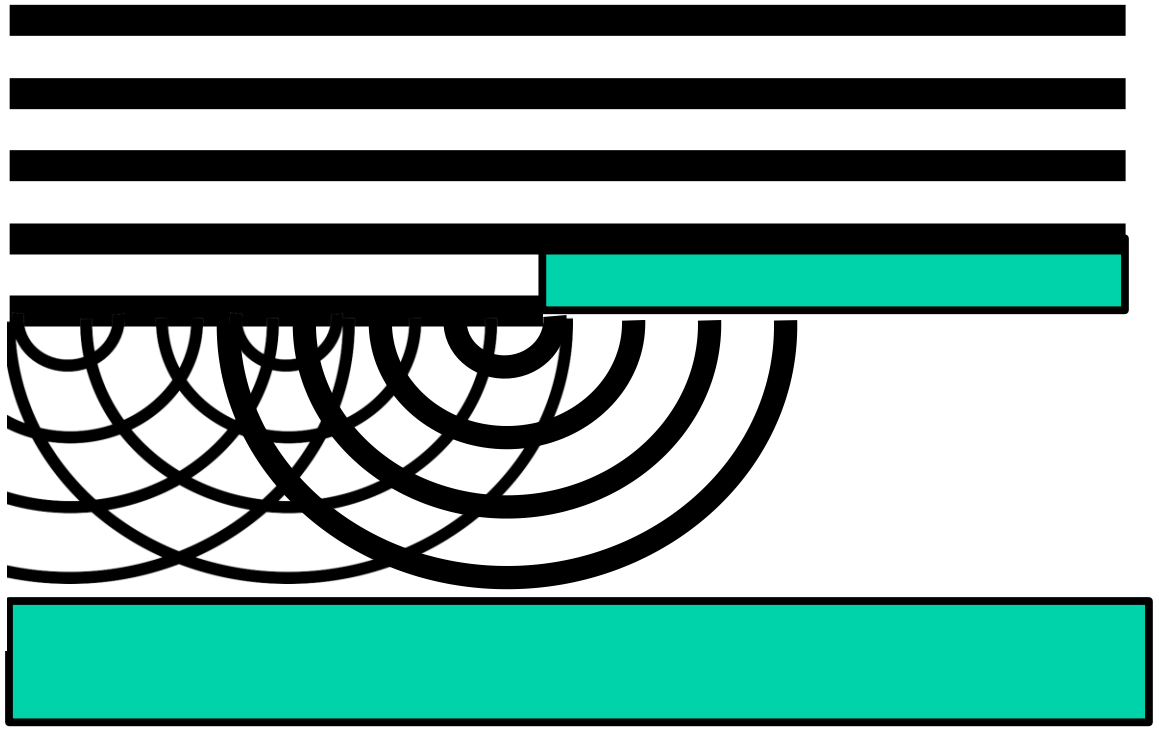
Huygens-Fresnel Principle



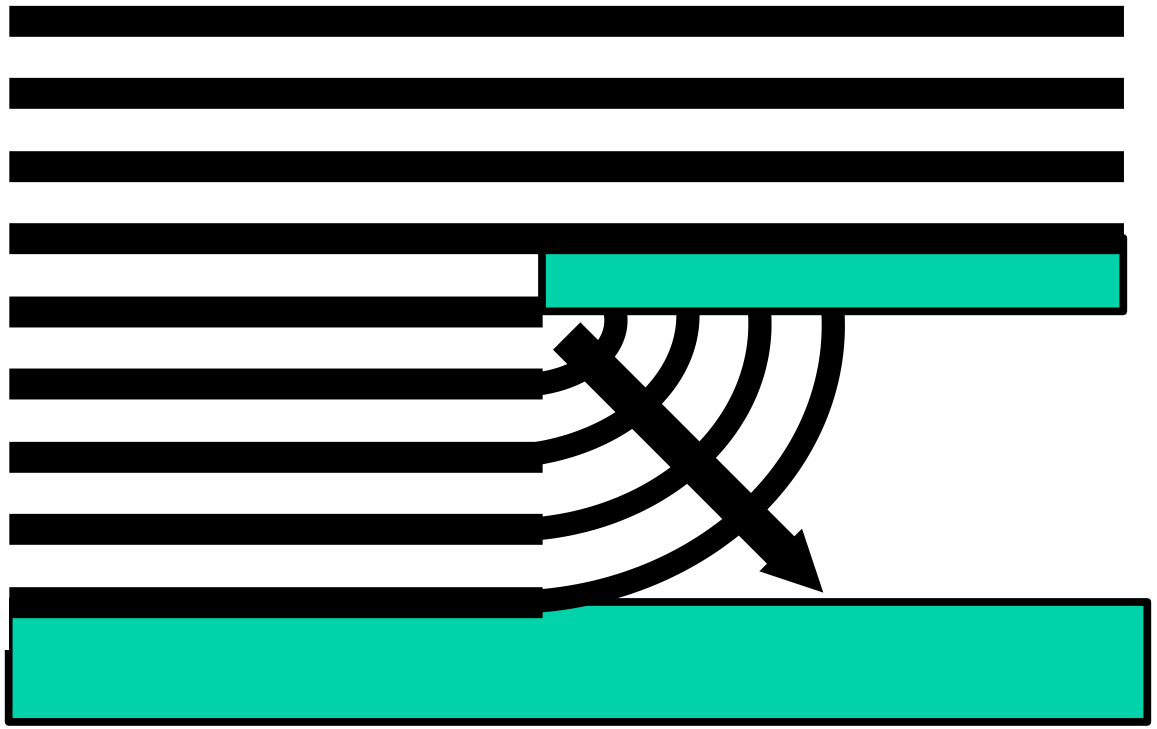
Huygens-Fresnel Principle



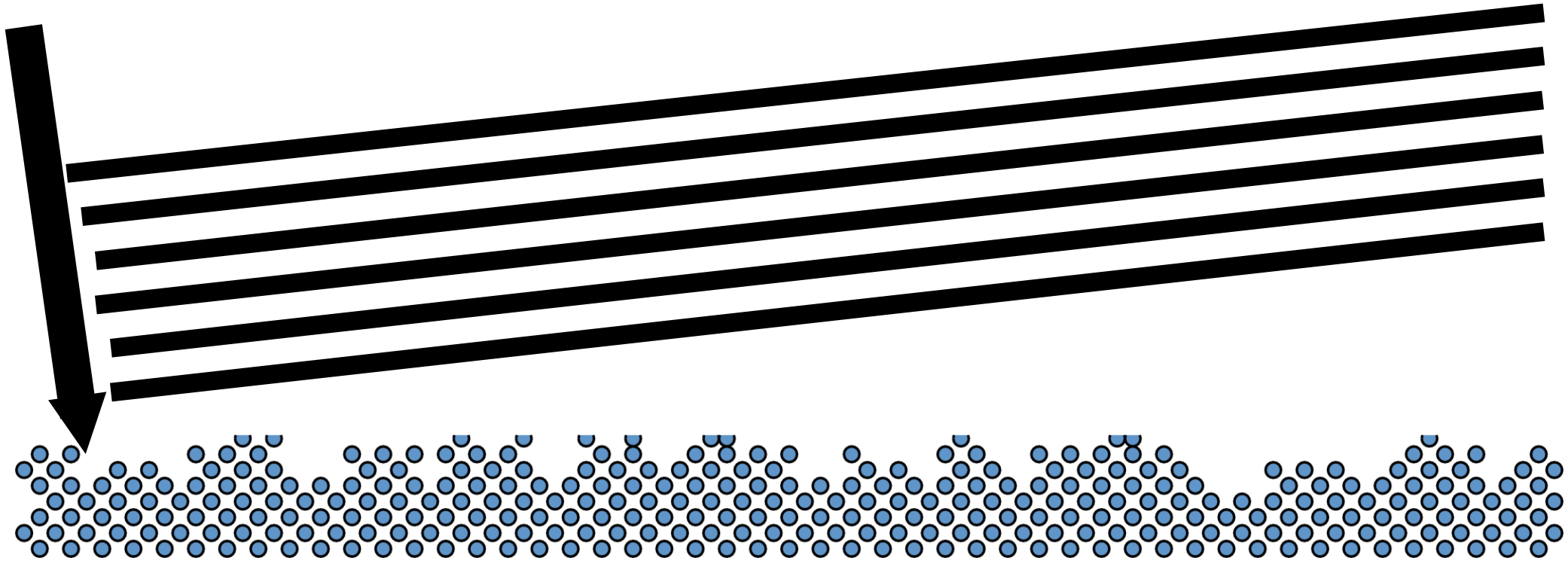
Diffraction



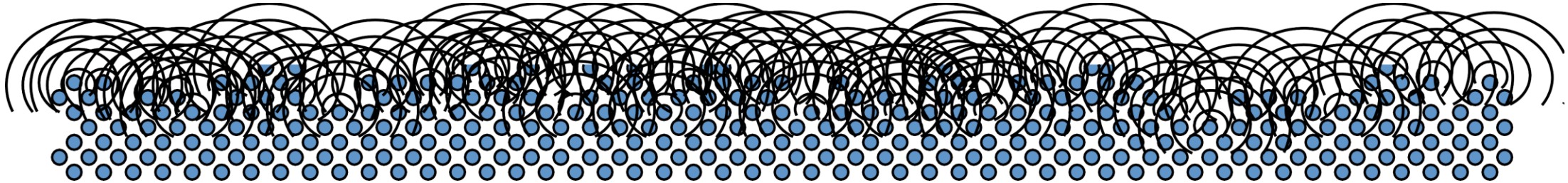
Diffraction



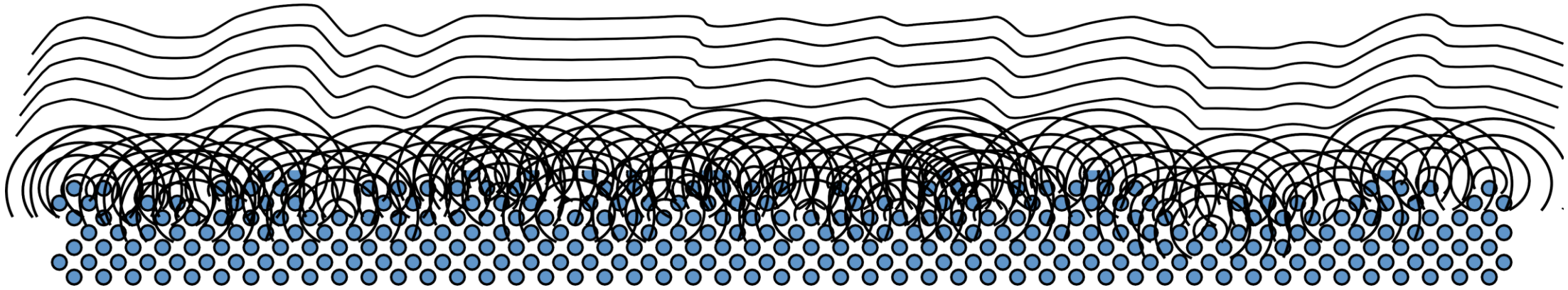
Diffraction from Optically-Smooth Surface



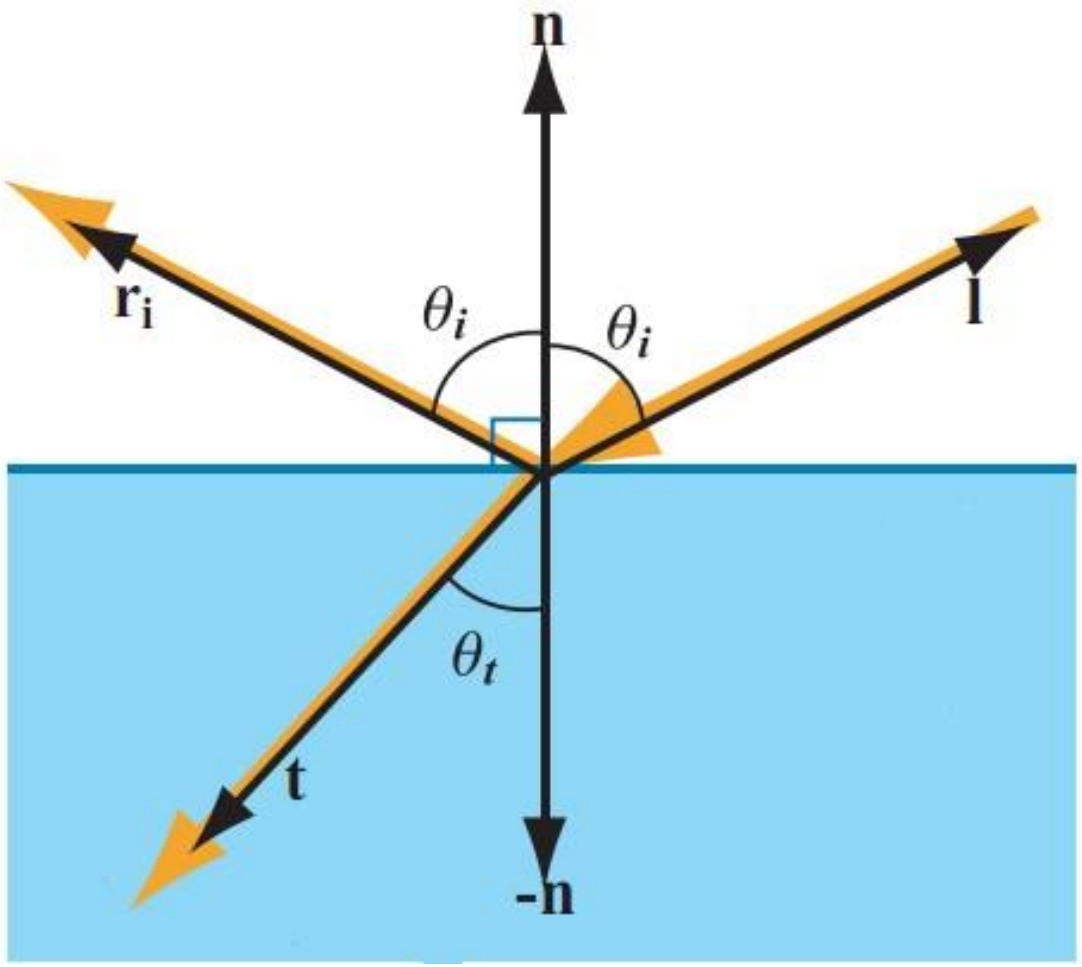
Diffraction from Optically-Smooth Surface



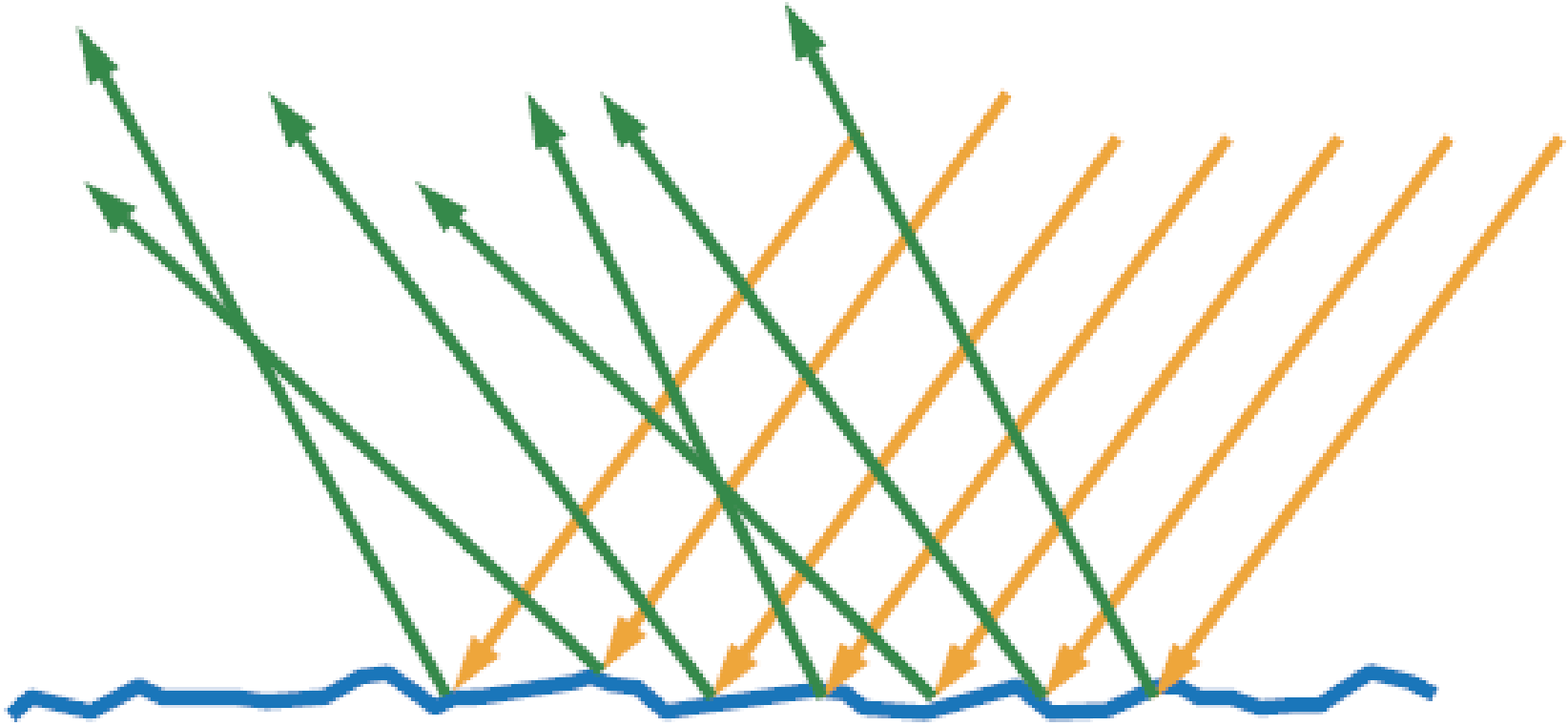
Diffraction from Optically-Smooth Surface



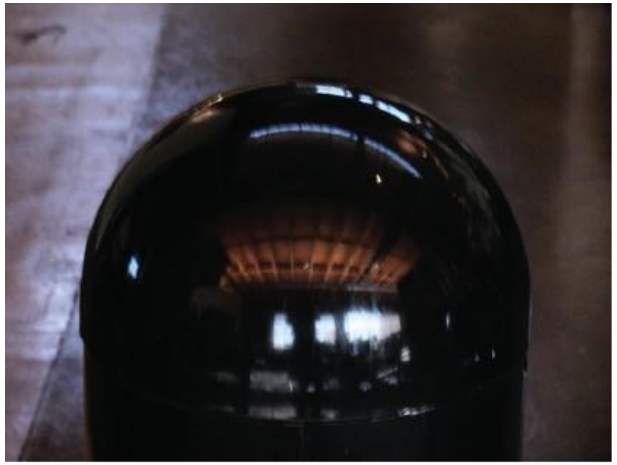
Geometric Optics

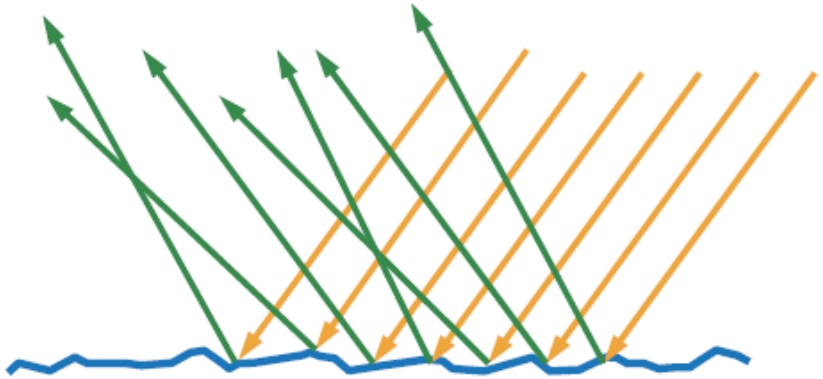


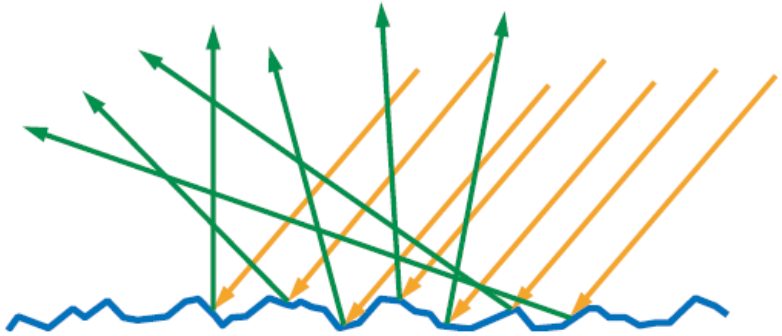
Microgeometry



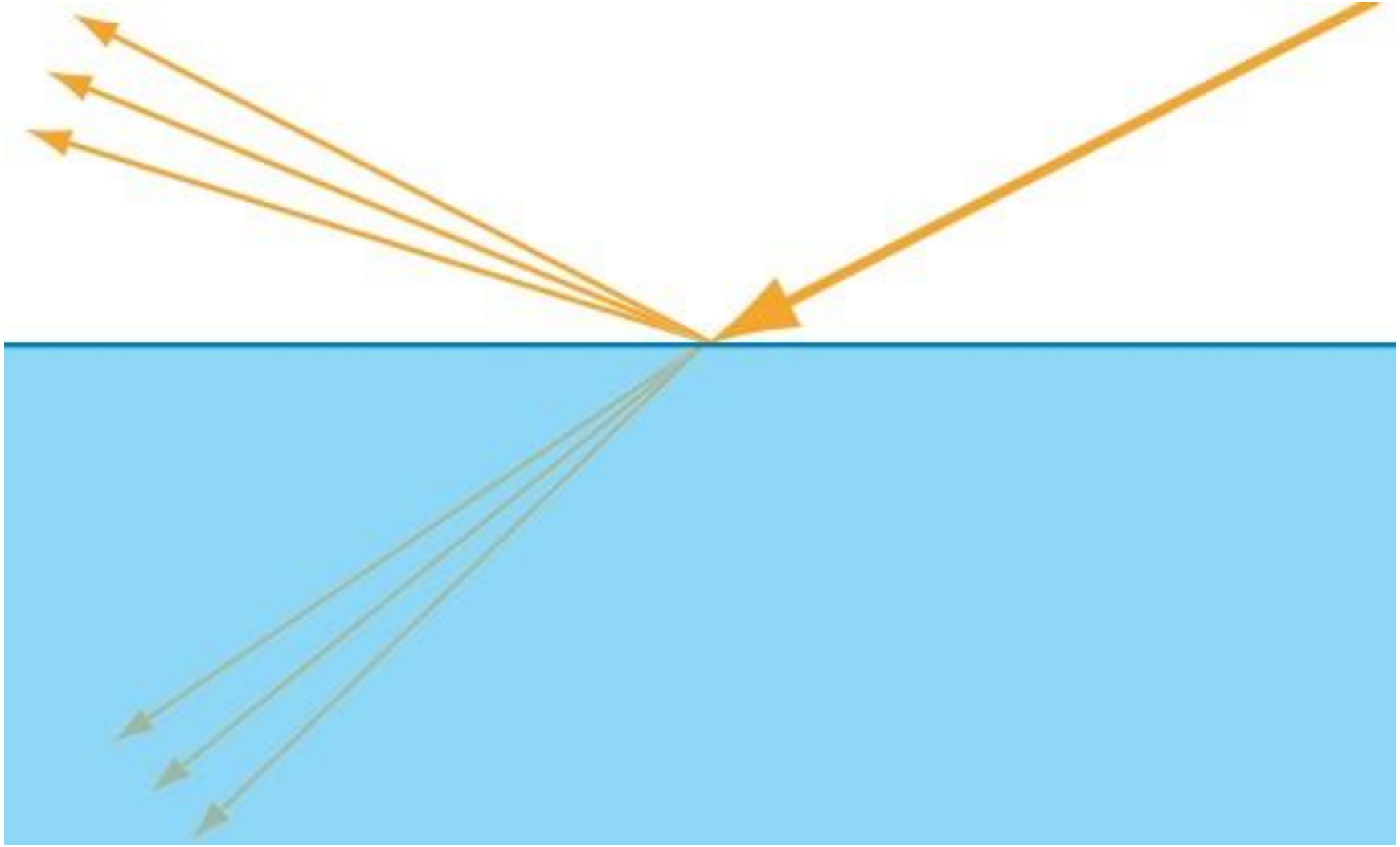
Rougher = Blurrier Reflections



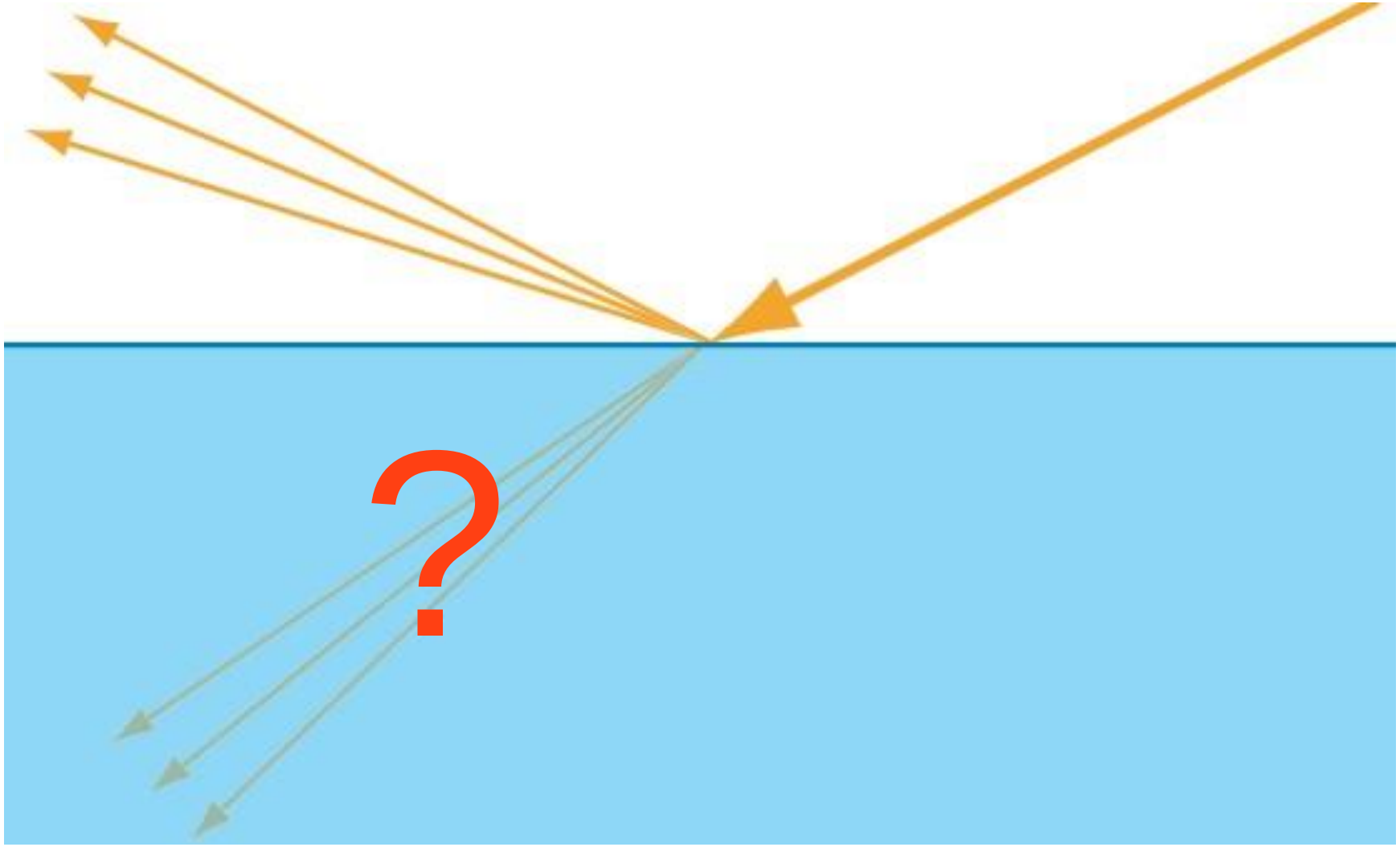




Statistical macroscopic view



What happens to the refracted light?



Metals (Conductors)

Dielectrics (Insulators)

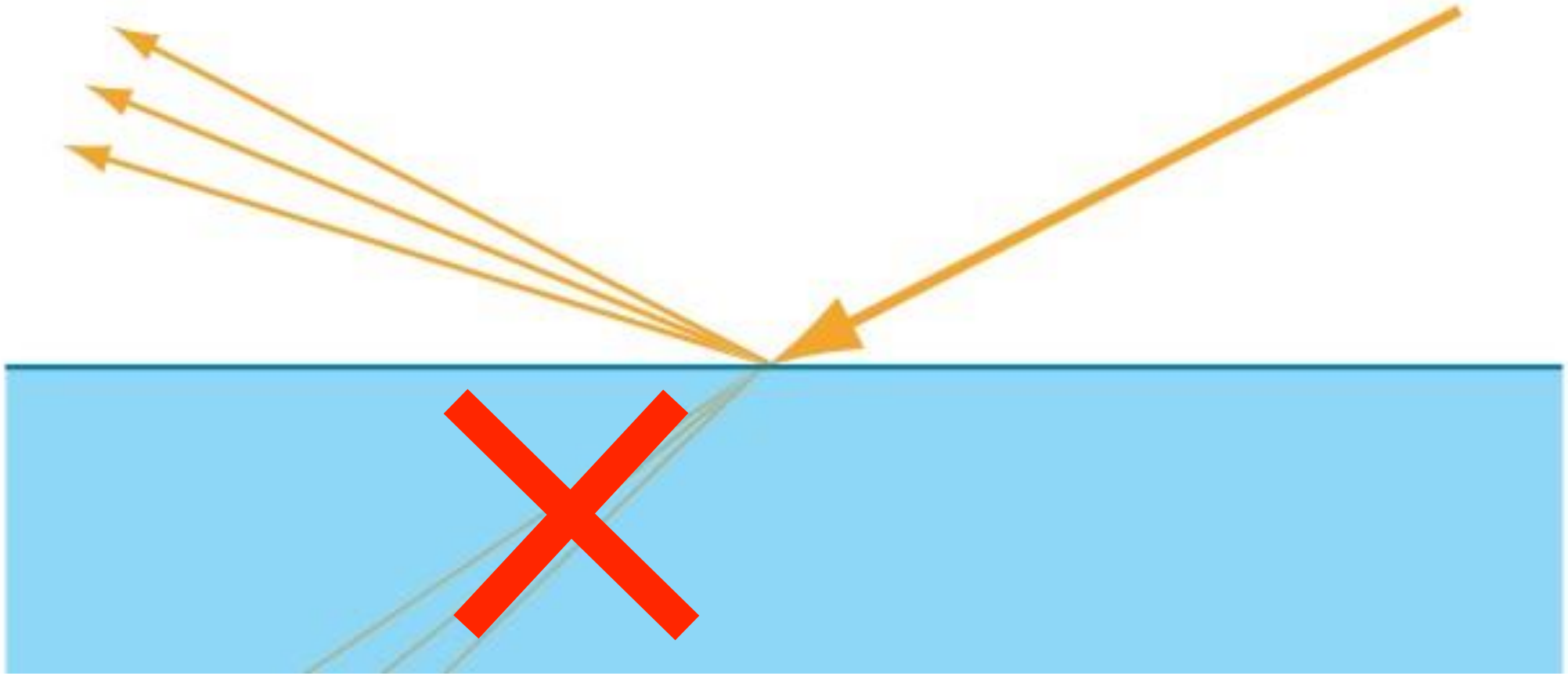
Semiconductors

Metals

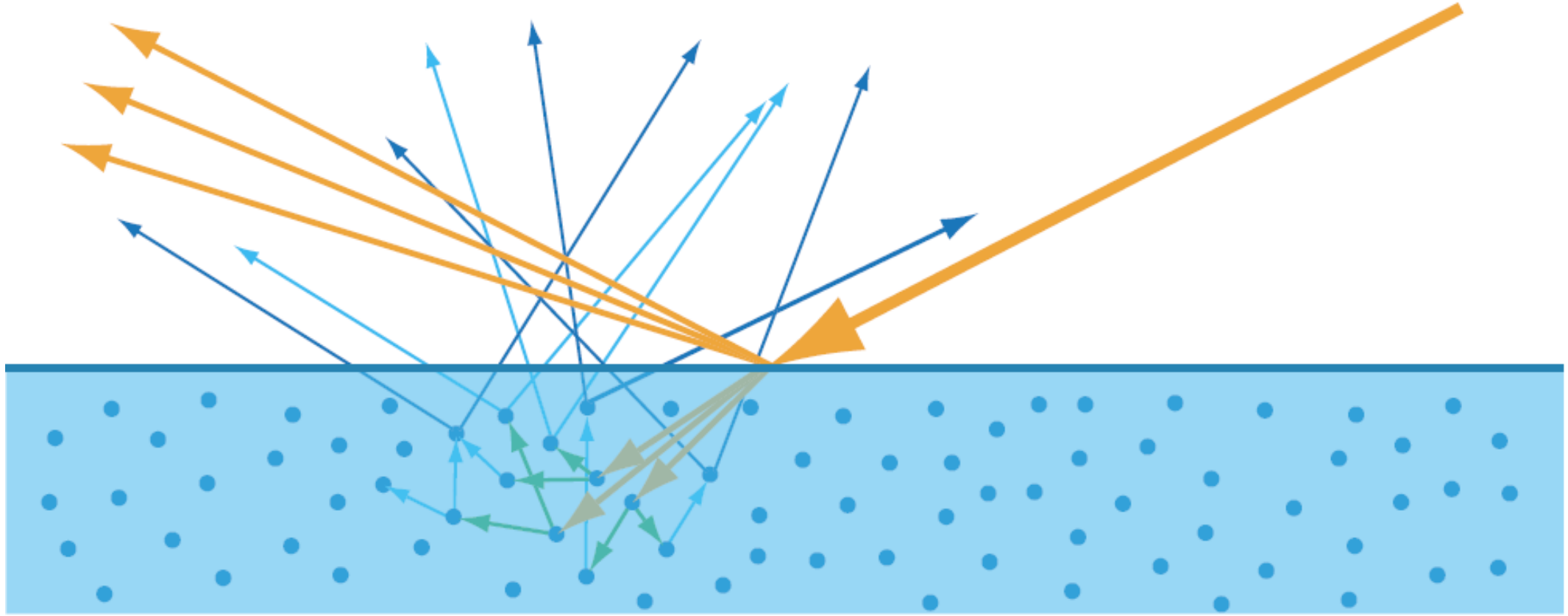
Non-Metals

~~Semiconductors~~

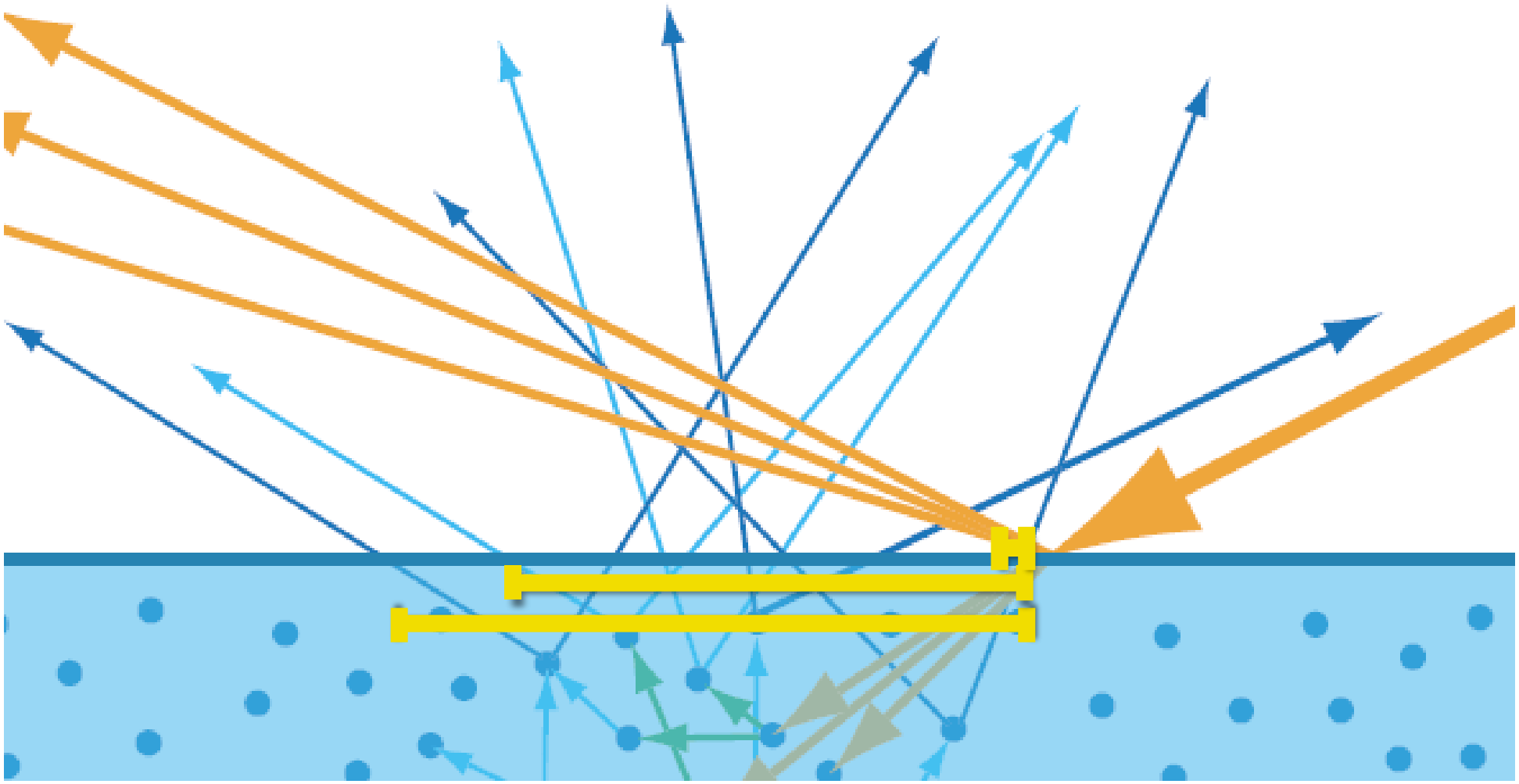
Metals



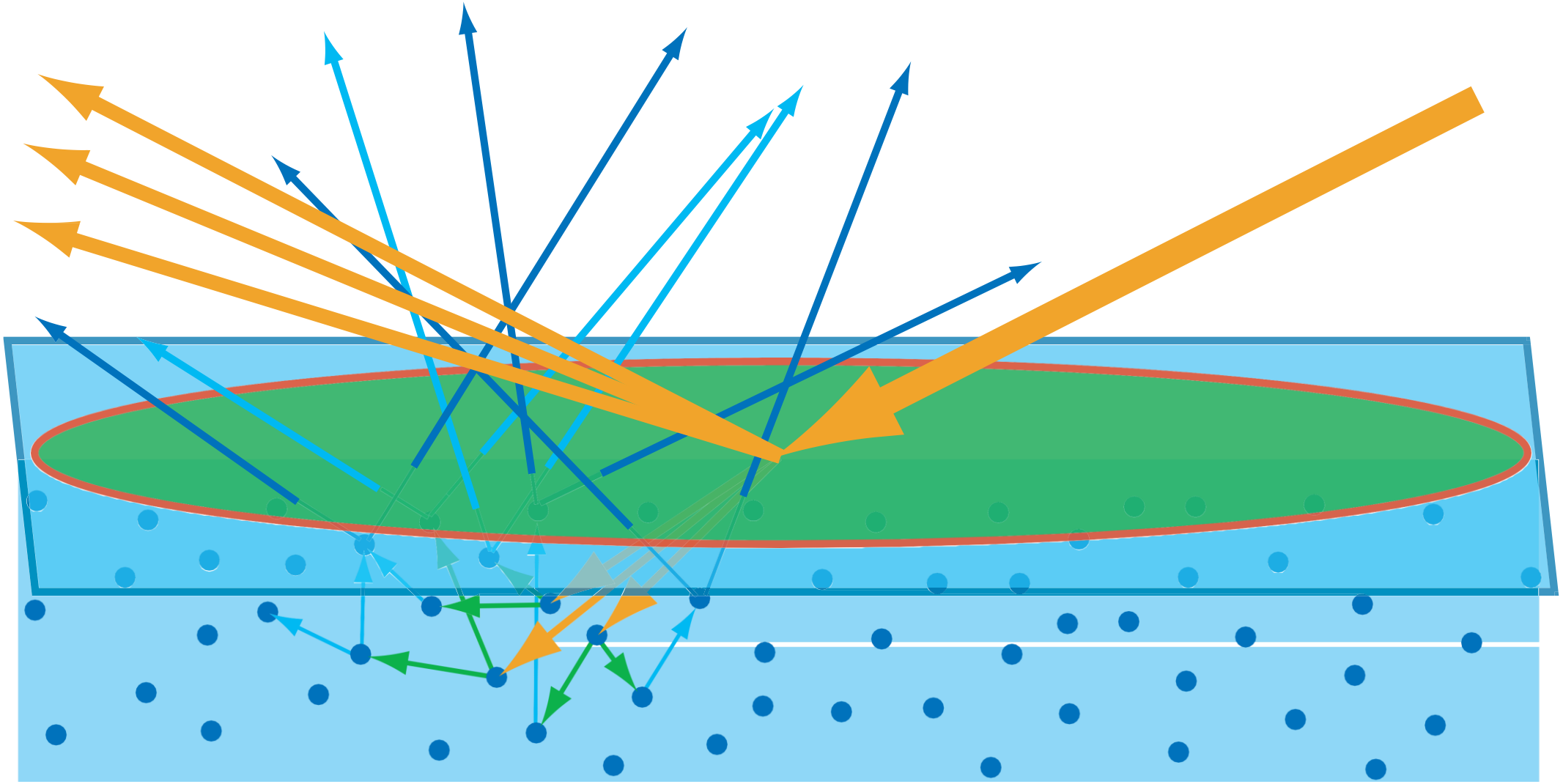
Non-Metals



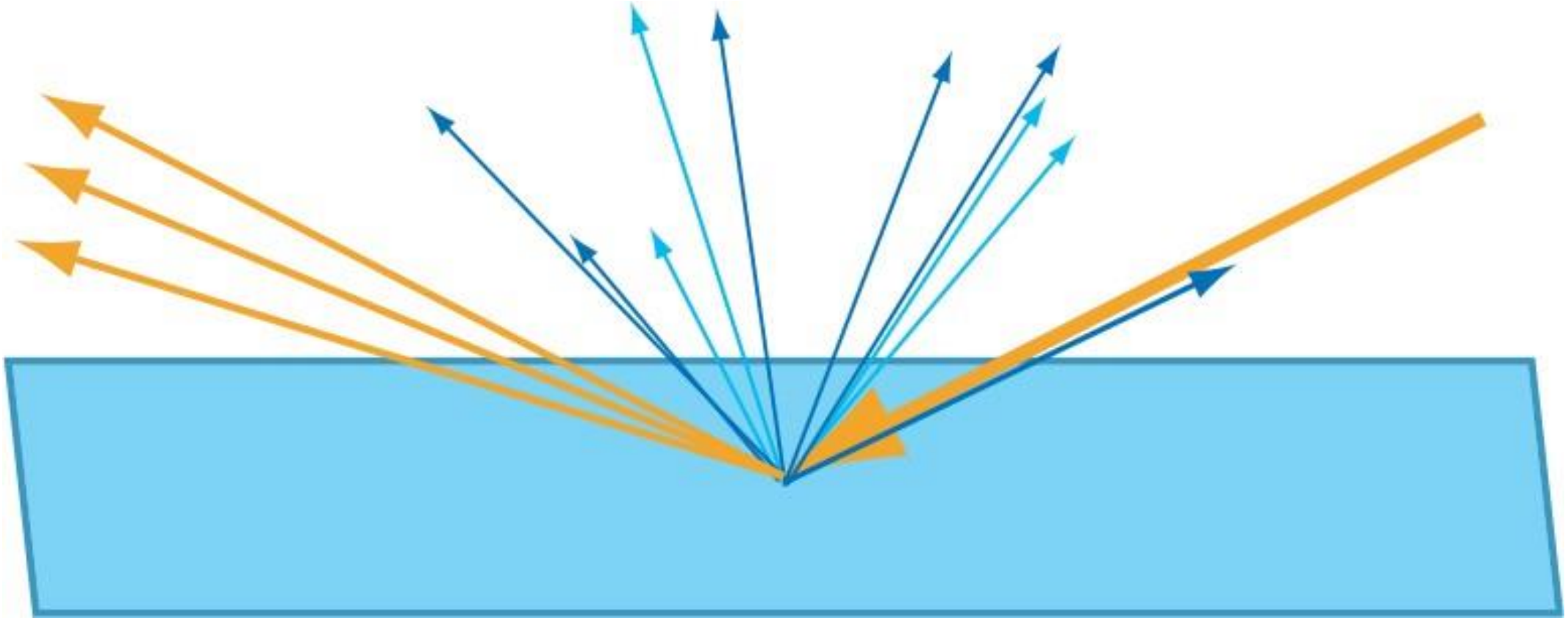
Refracted light



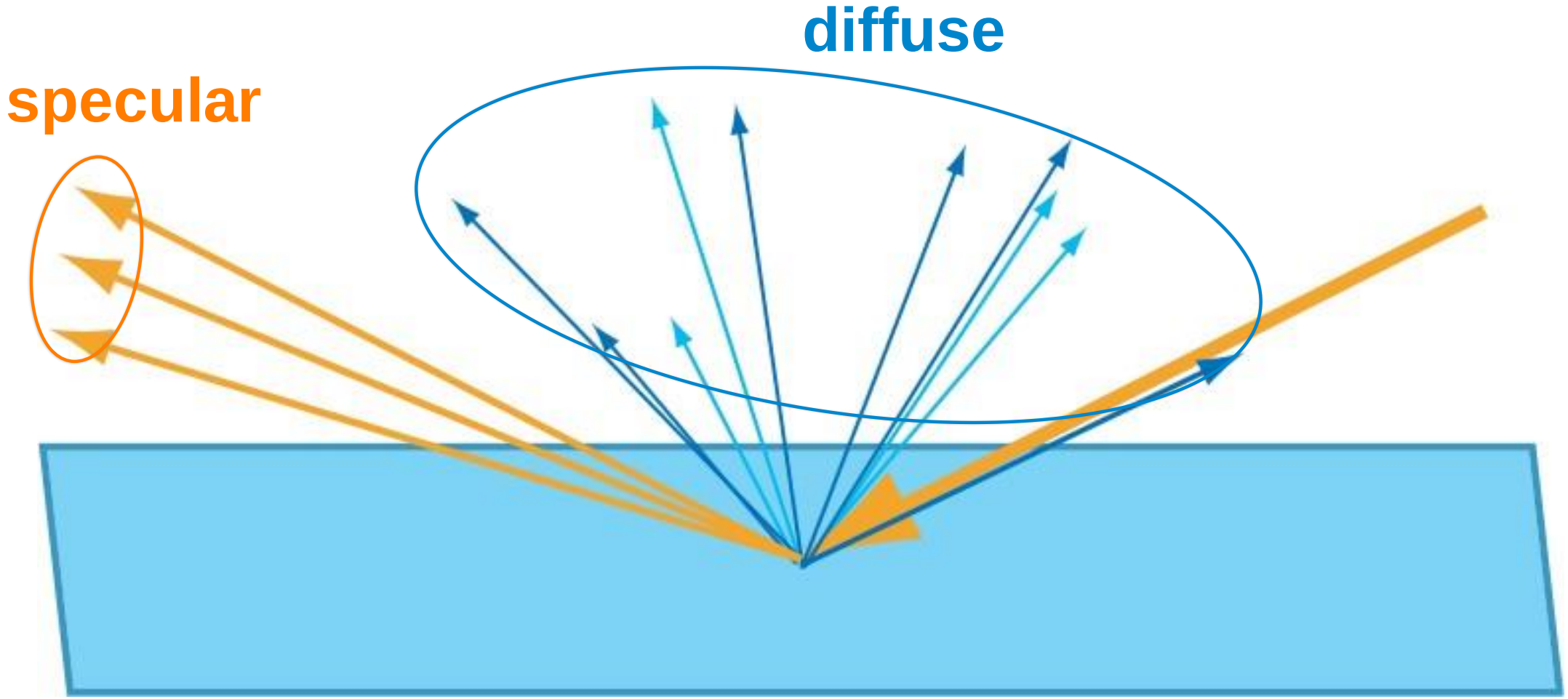
Large sub-Surface scattering area



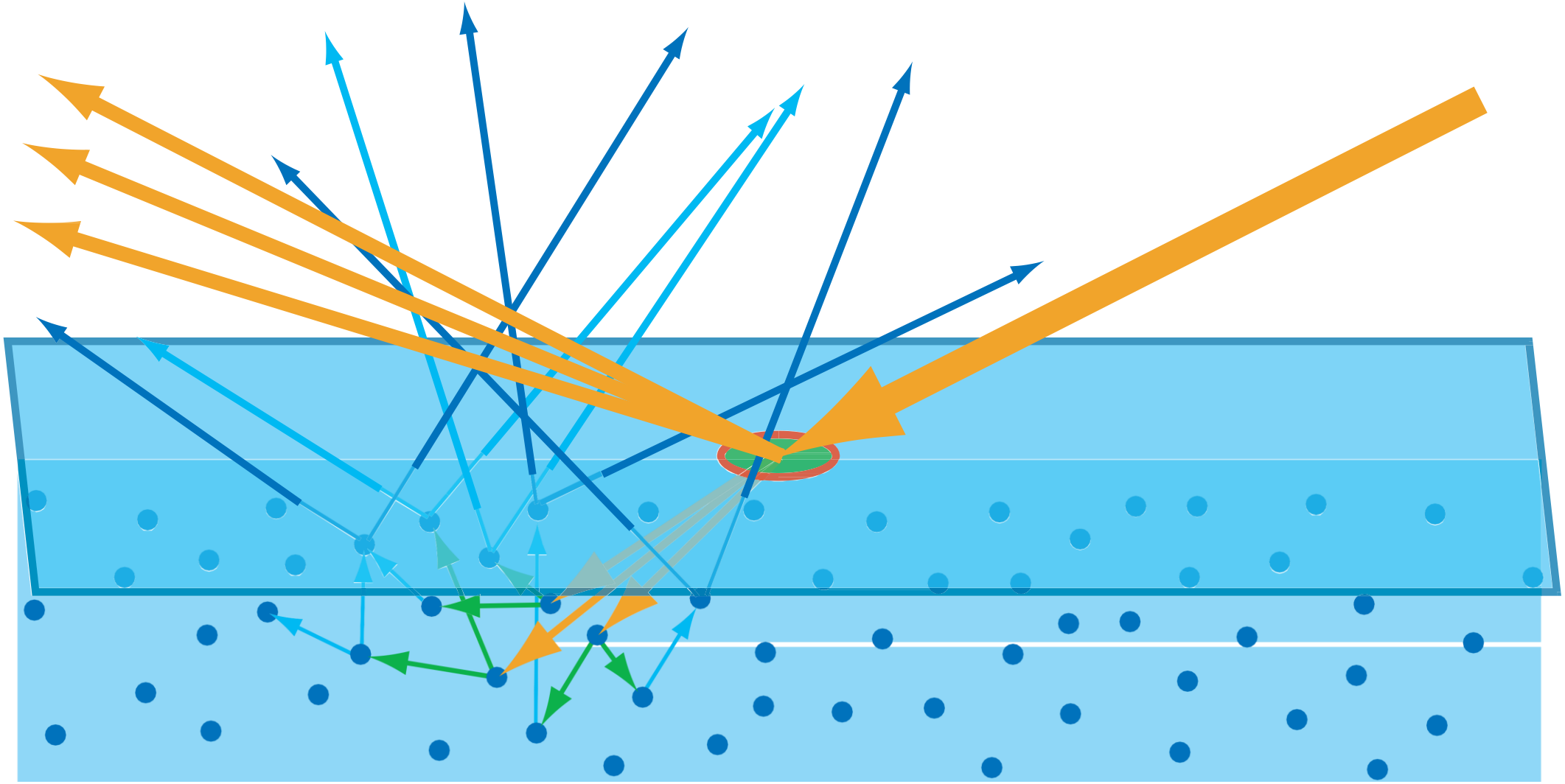
Ignoring sub-surface scattering

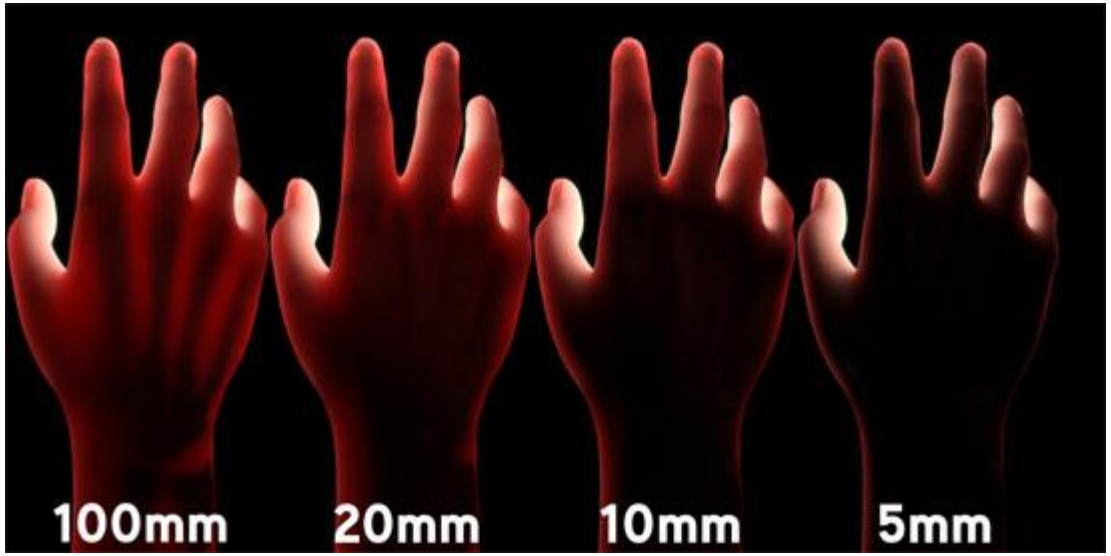


Divide into specular and diffuse light



Small sub-surface scattering area




100mm
20mm
10mm
5mm


SSS OFF
SSS ON

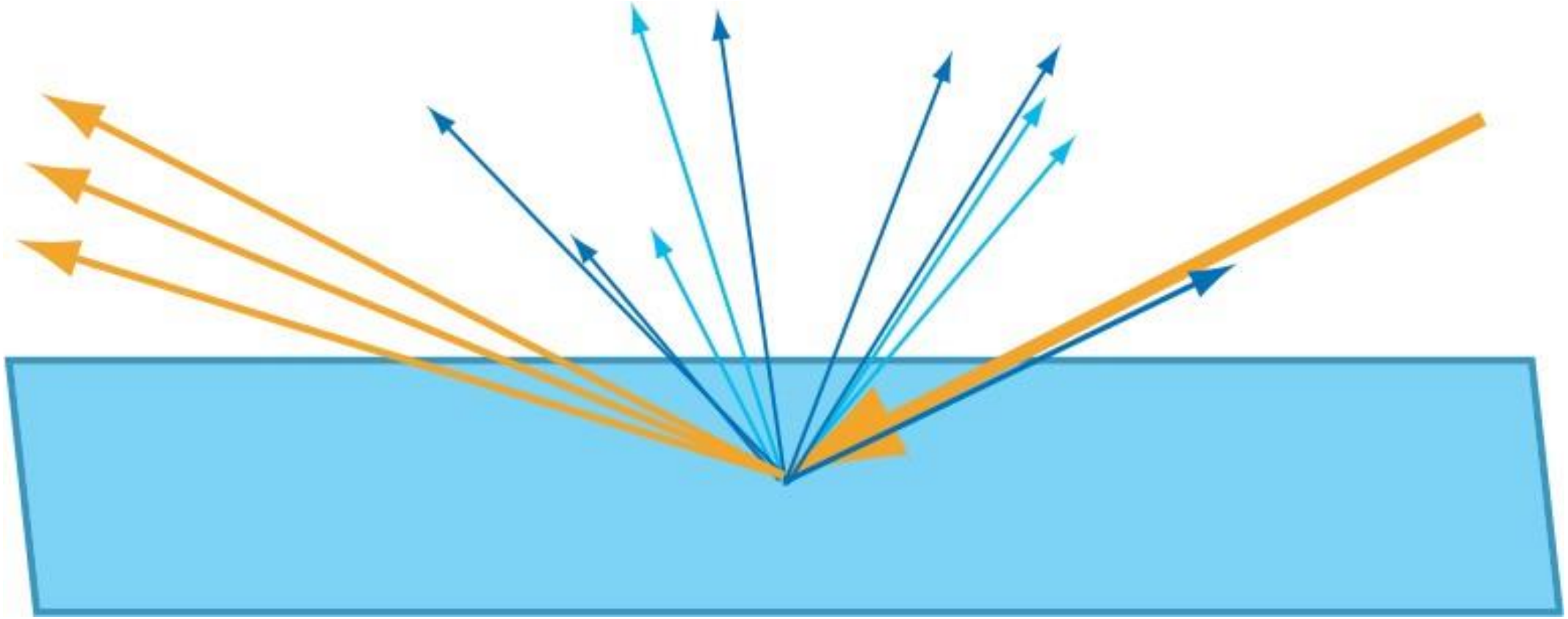
Mathematical model

Radiance

Single Ray

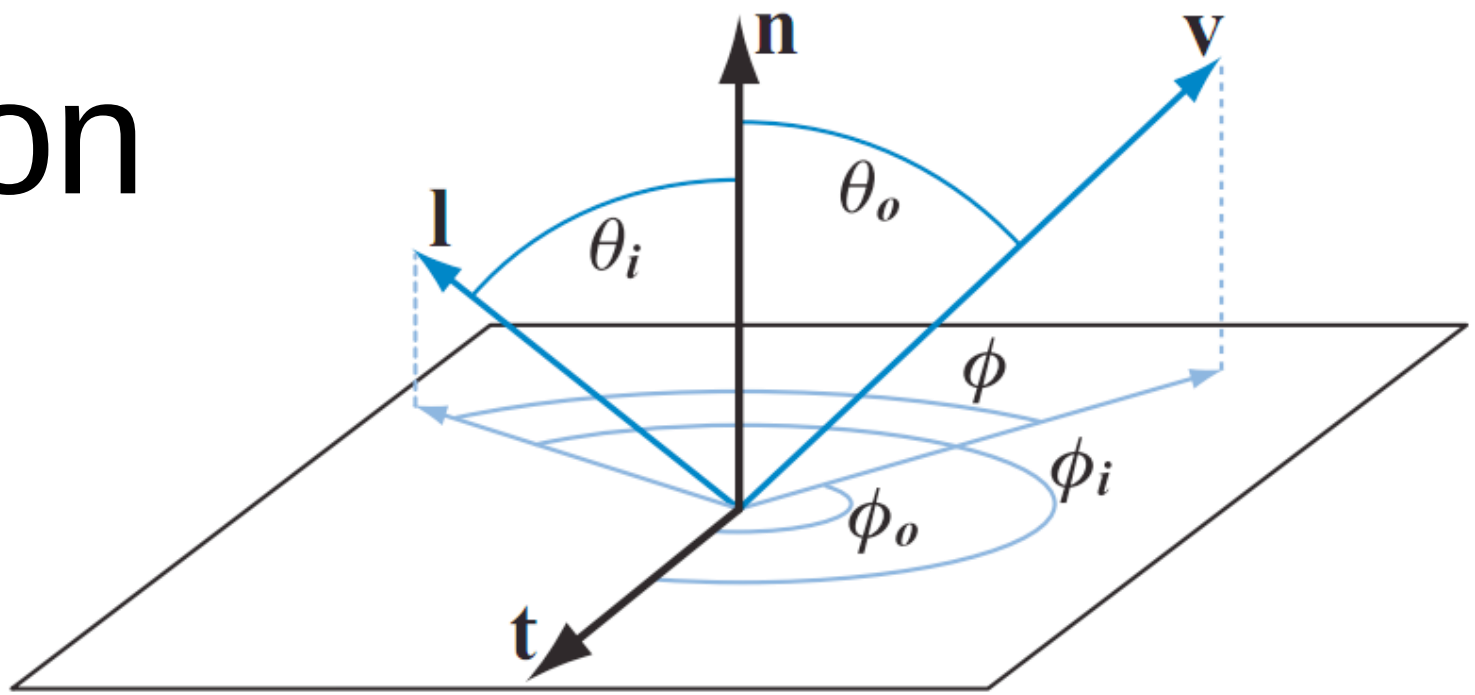
Spectral/RGB

Depends only on light and view directions



Bidirectional
Reflectance
Distribution
Function

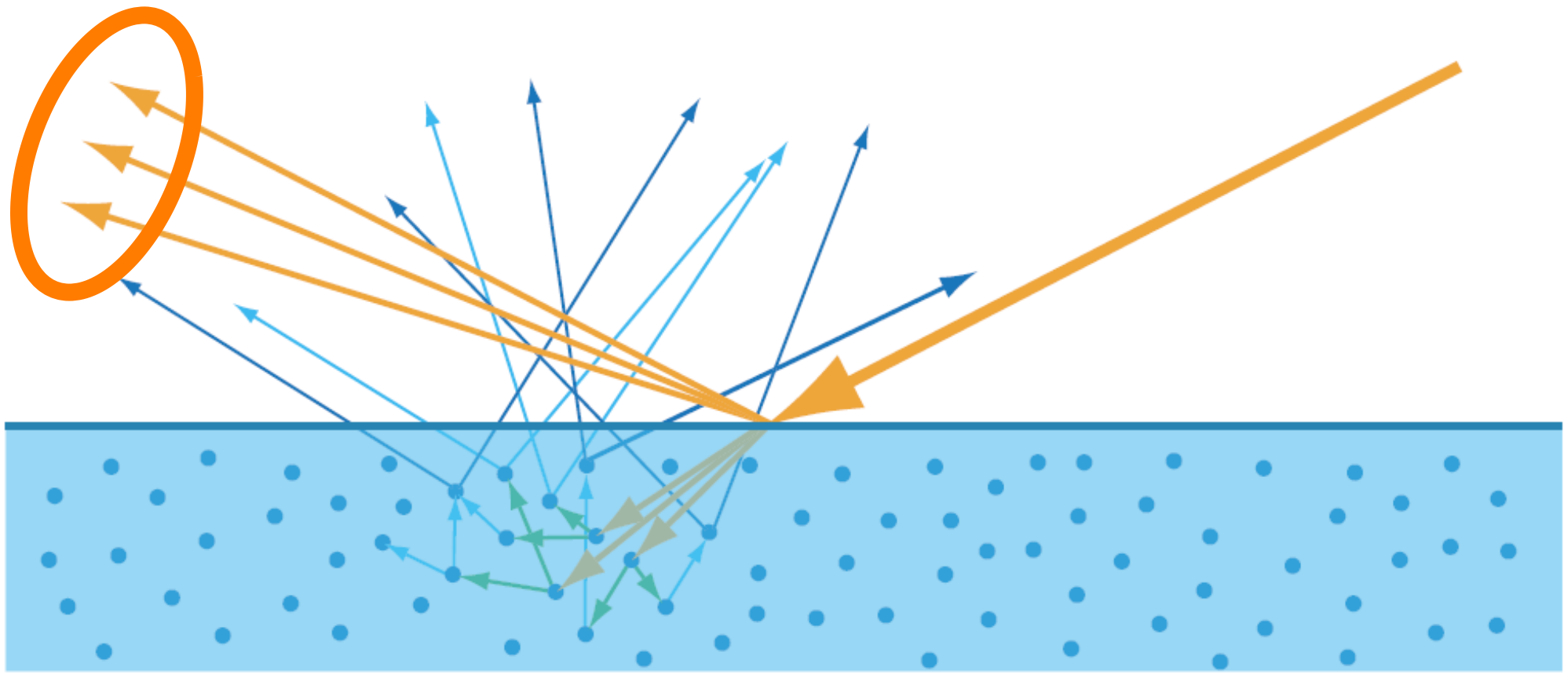
$$f(\mathbf{l}, \mathbf{v})$$



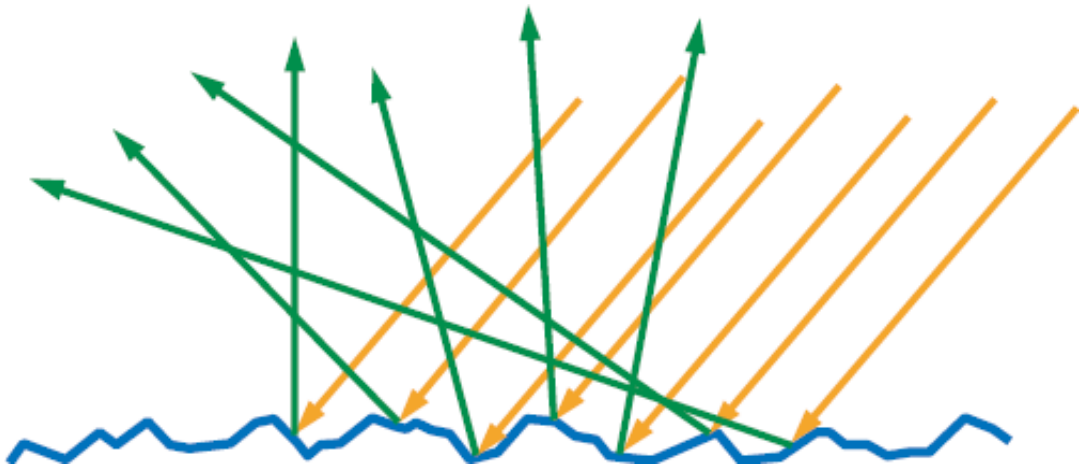
The Reflectance Equation

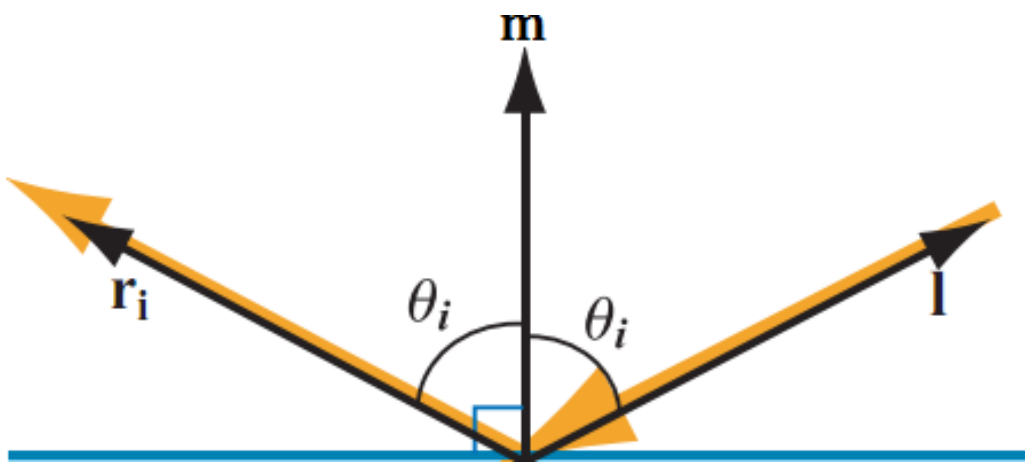
$$L_o(\boldsymbol{v}) = \int_{\Omega} f(\boldsymbol{l}, \boldsymbol{v}) \otimes L_i(\boldsymbol{l})(\boldsymbol{n} \cdot \boldsymbol{l}) d\omega_i$$

Surface Reflection (Specular Term)

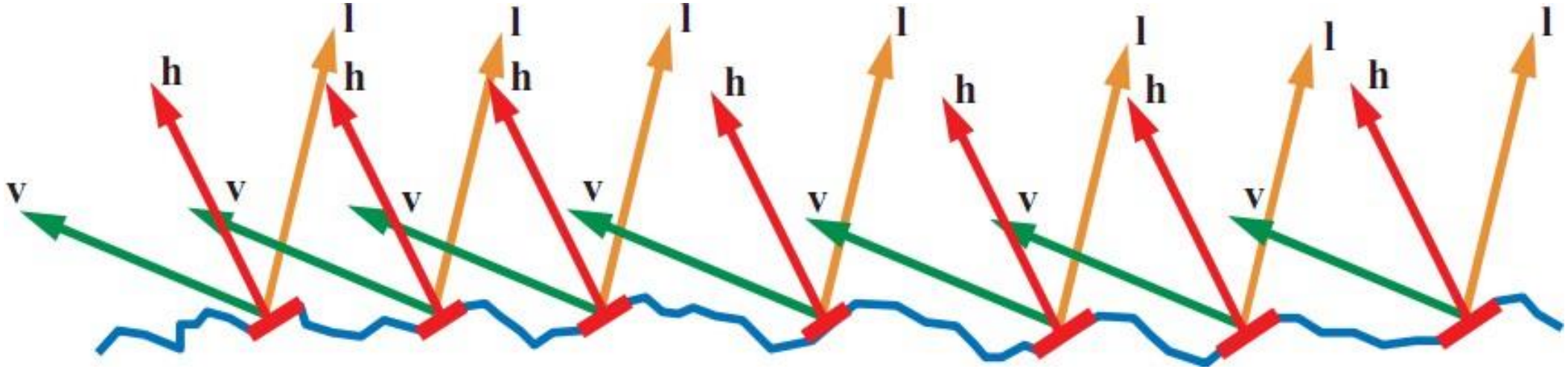


Microfacet Theory

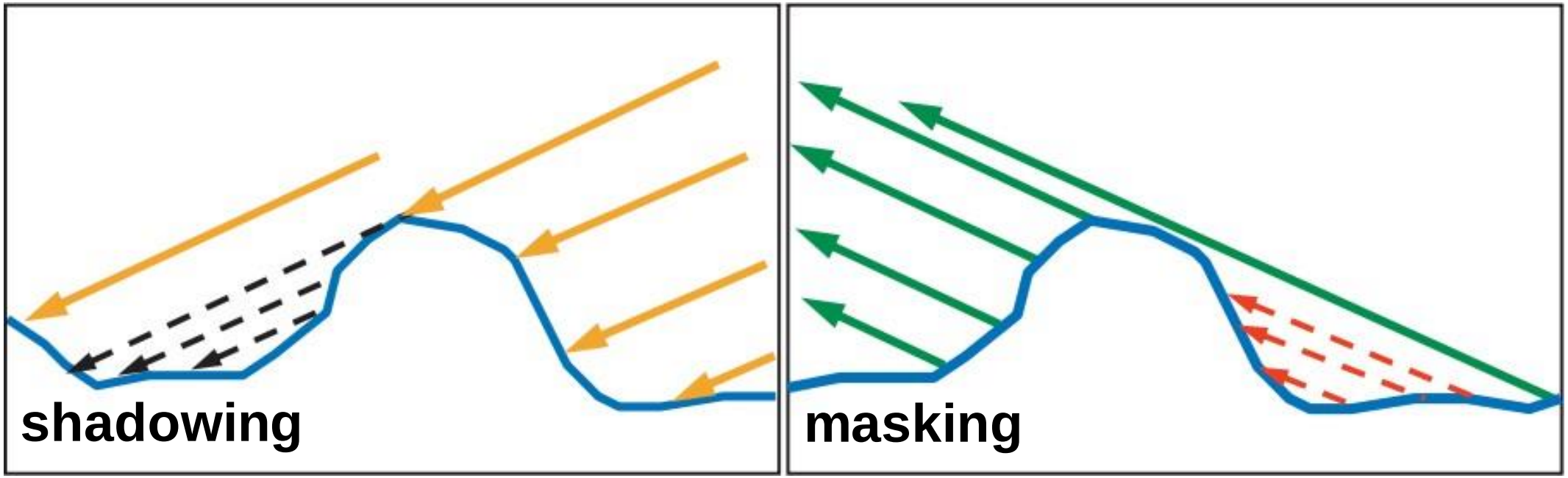




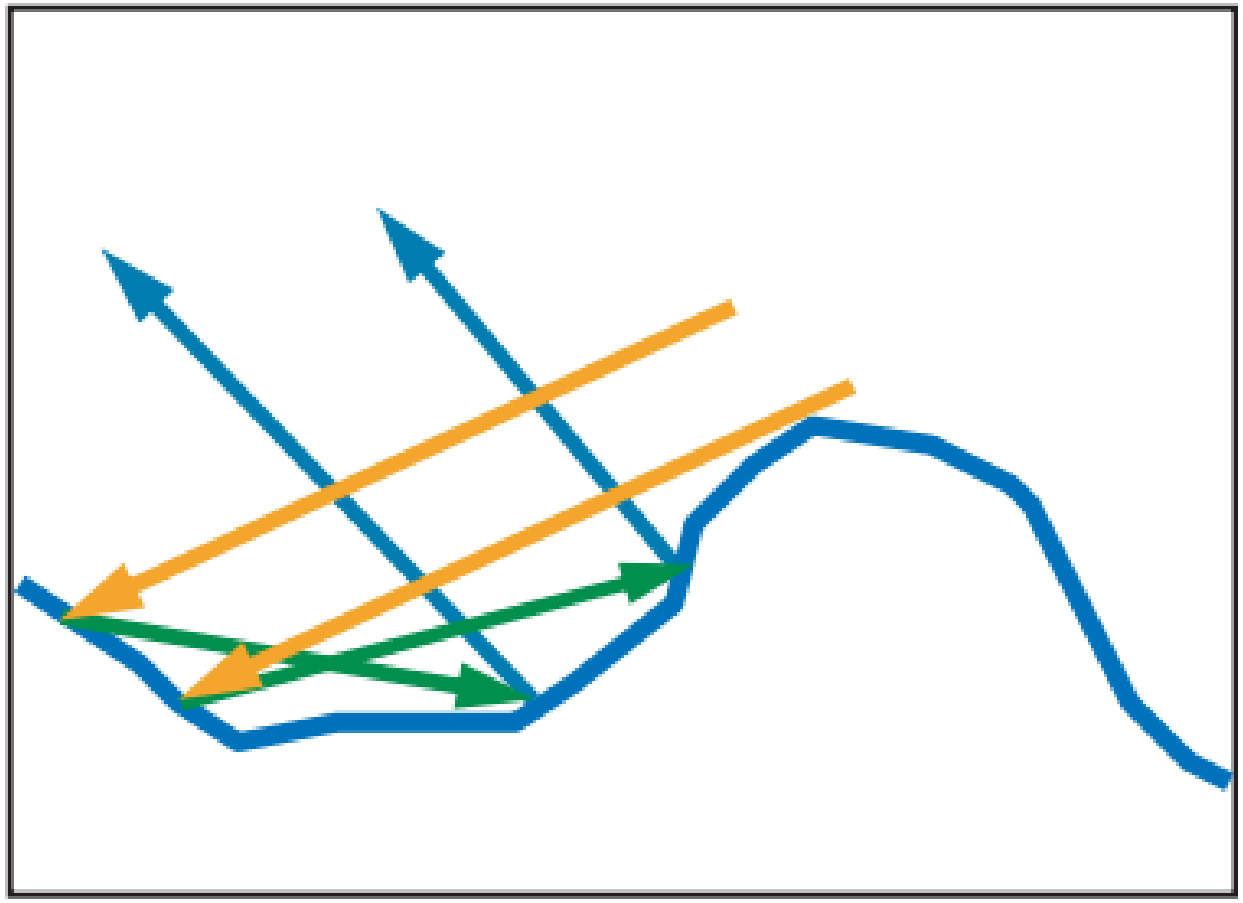
The Half Vector



Shadowing and Masking



Multiple Surface Bounces



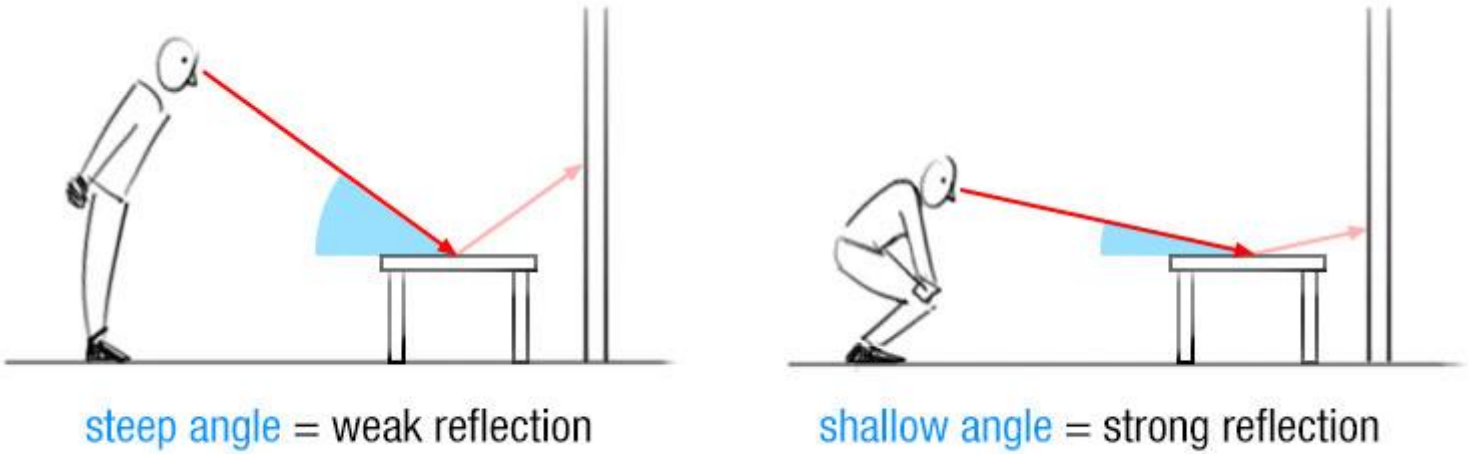
Microfacet Specular BRDF

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

Fresnel Reflectance

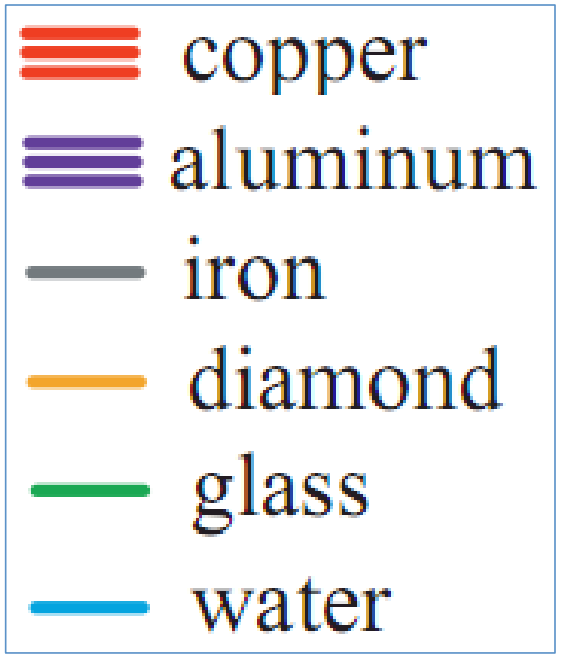
$$f(\mathbf{l}, \mathbf{v}) = \frac{\boxed{F(\mathbf{l}, \mathbf{h})} G(\mathbf{l}, \mathbf{v}, \mathbf{h}) D(\mathbf{h})}{4(\mathbf{n} \cdot \mathbf{l})(\mathbf{n} \cdot \mathbf{v})}$$

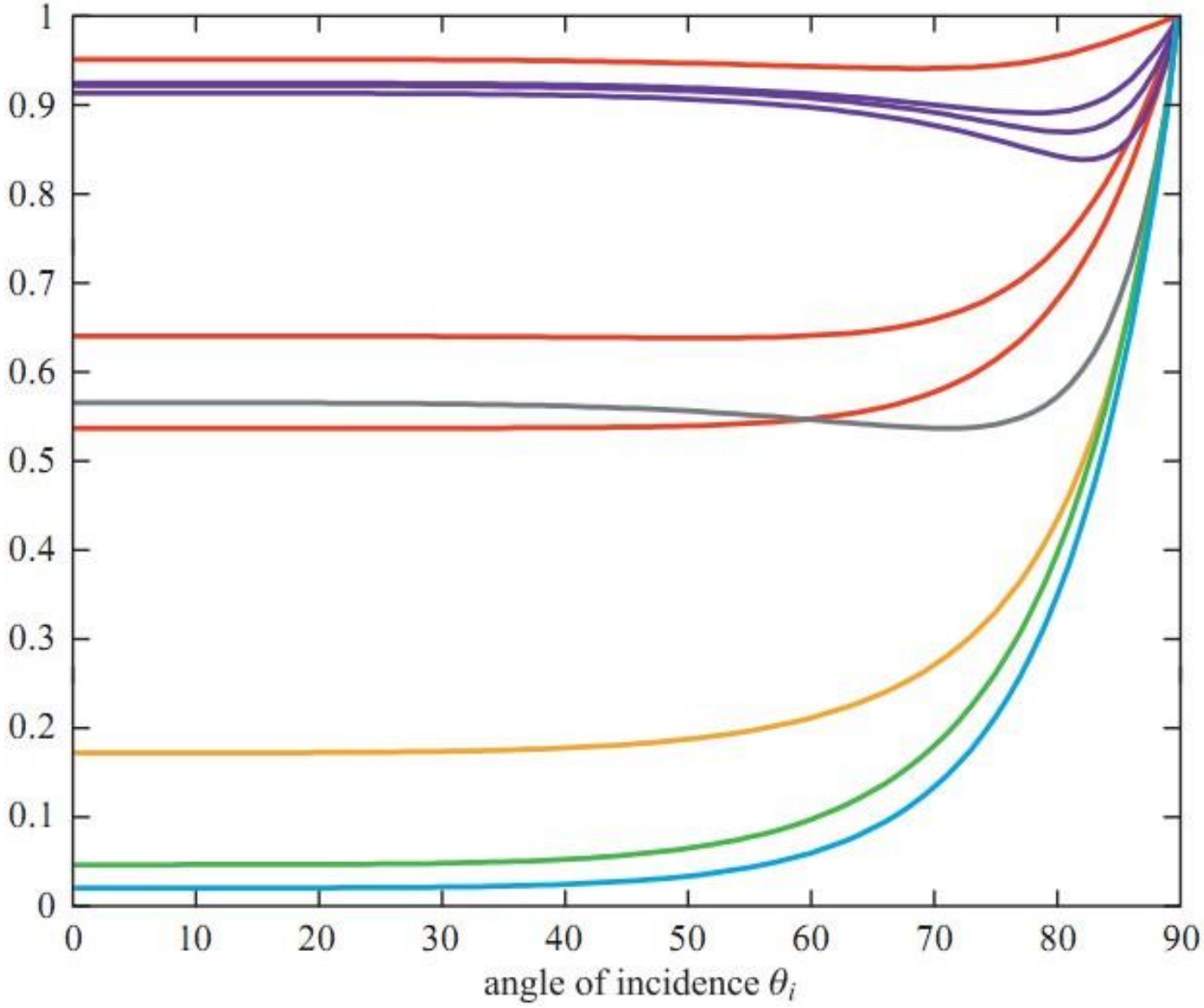
Fresnel Reflectance

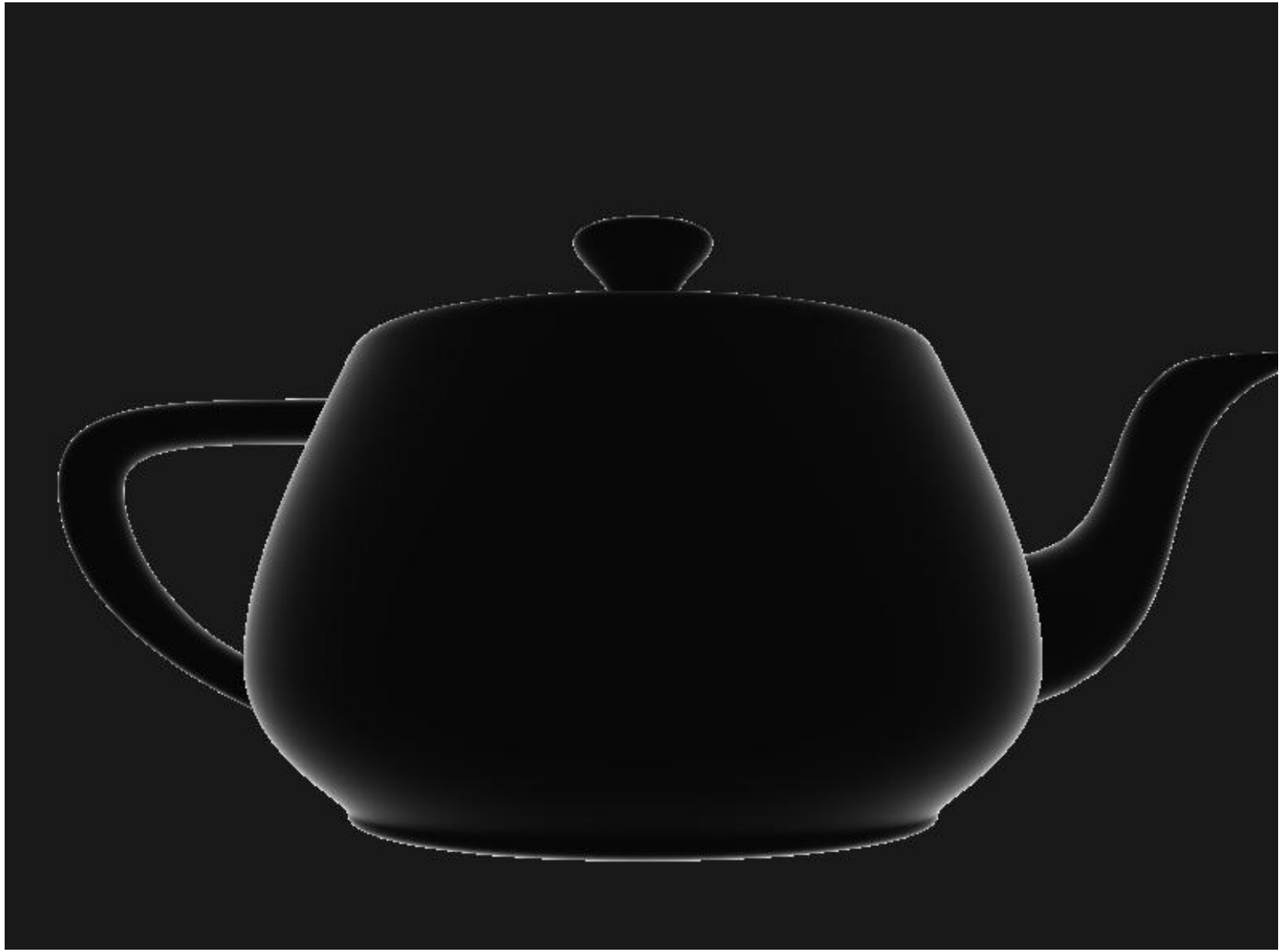


Example

Fresnel Reflectance





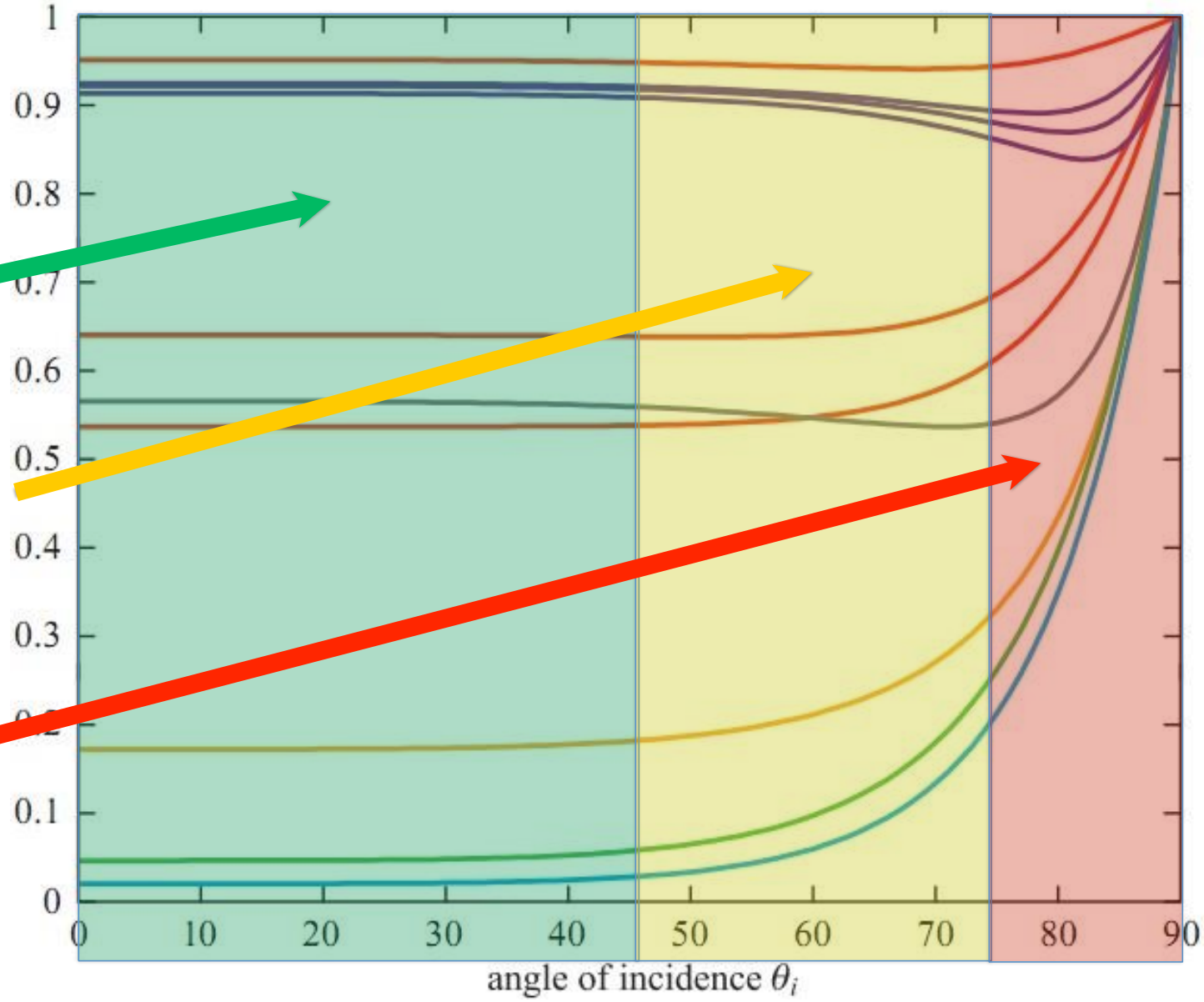


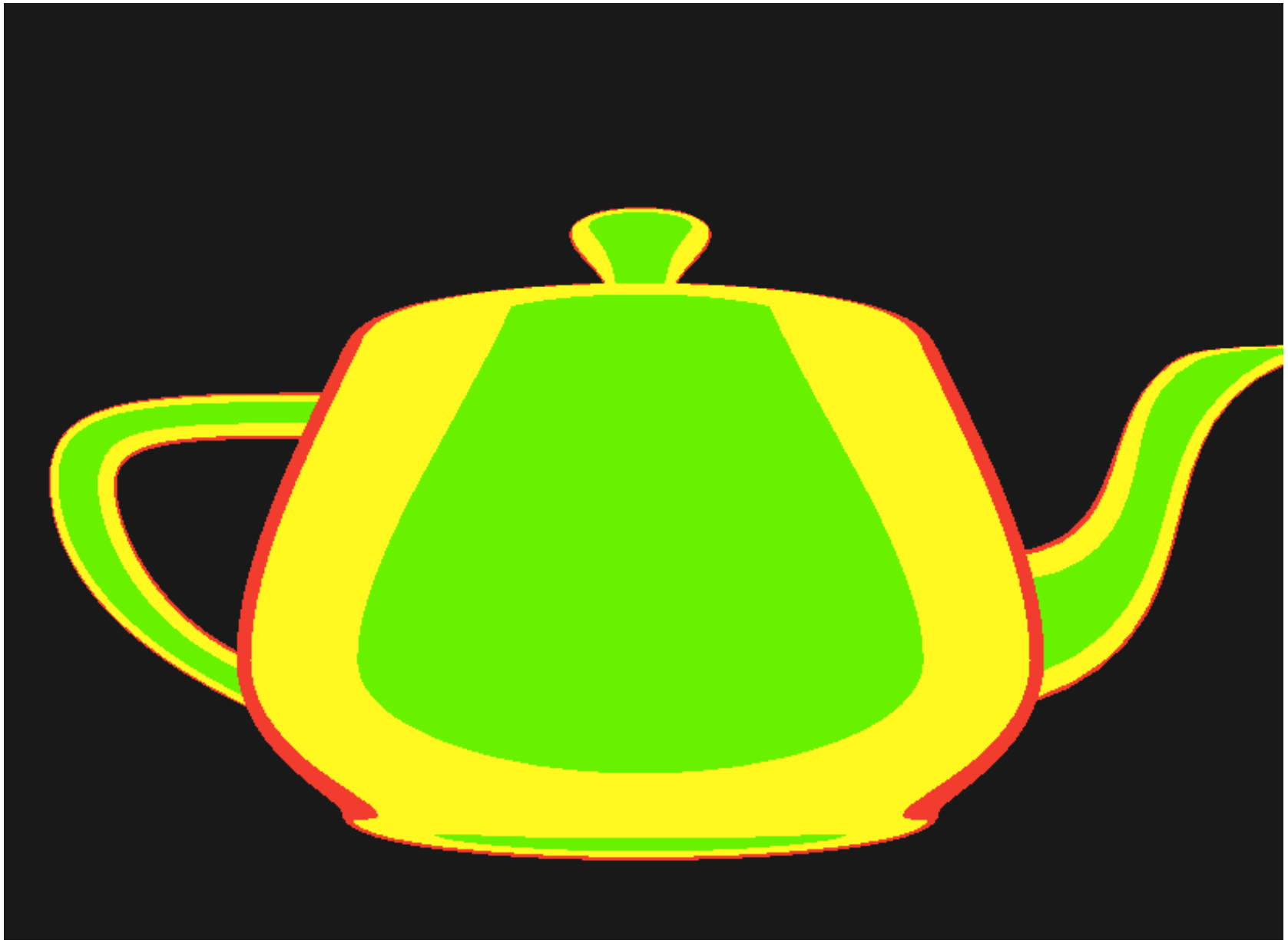
Fresnel Reflectance

barely changes

changes somewhat

goes rapidly to 1

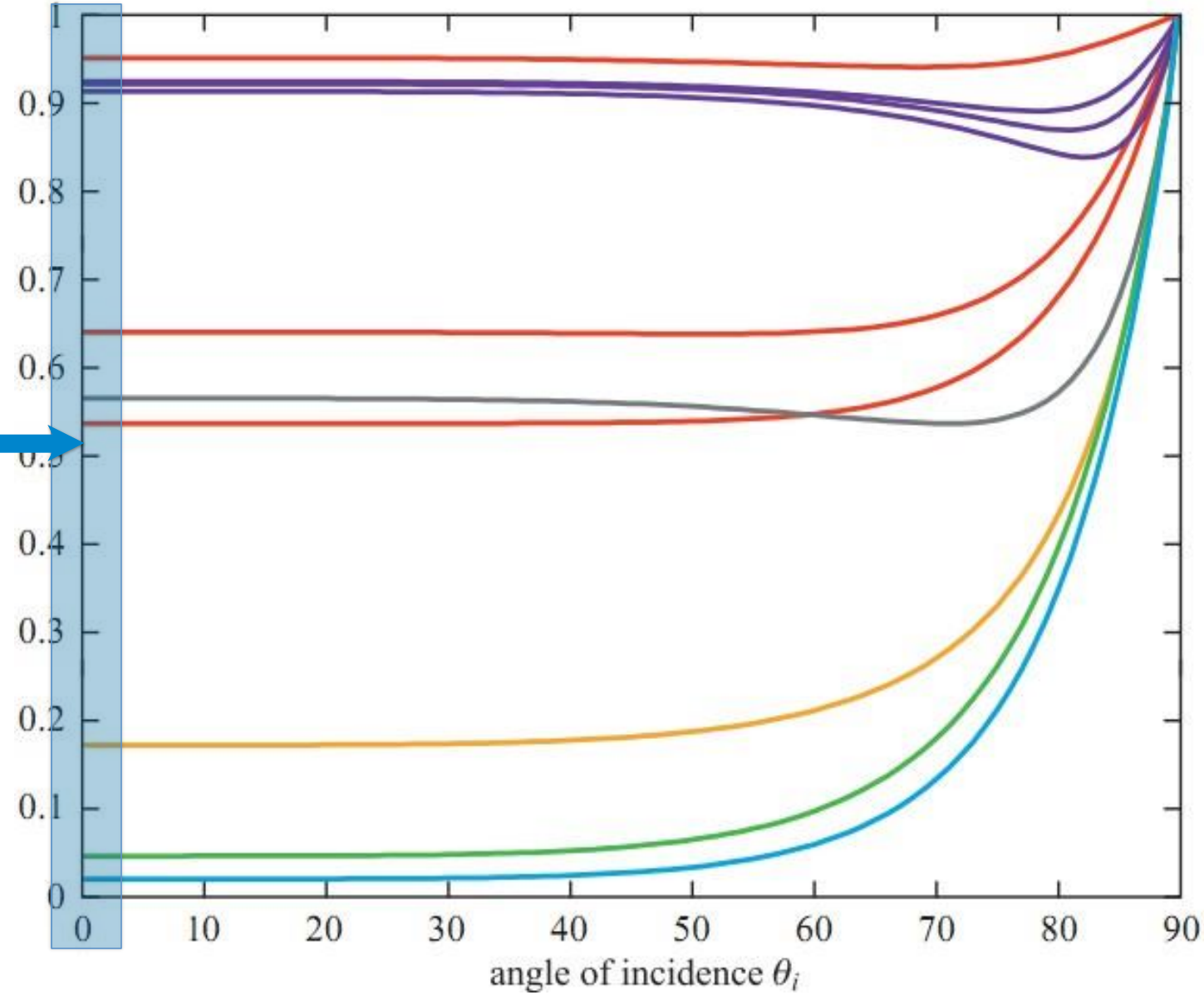




Fresnel Reflectance

$$F_0 = F(0°)$$

Is the surface's characteristic specular color



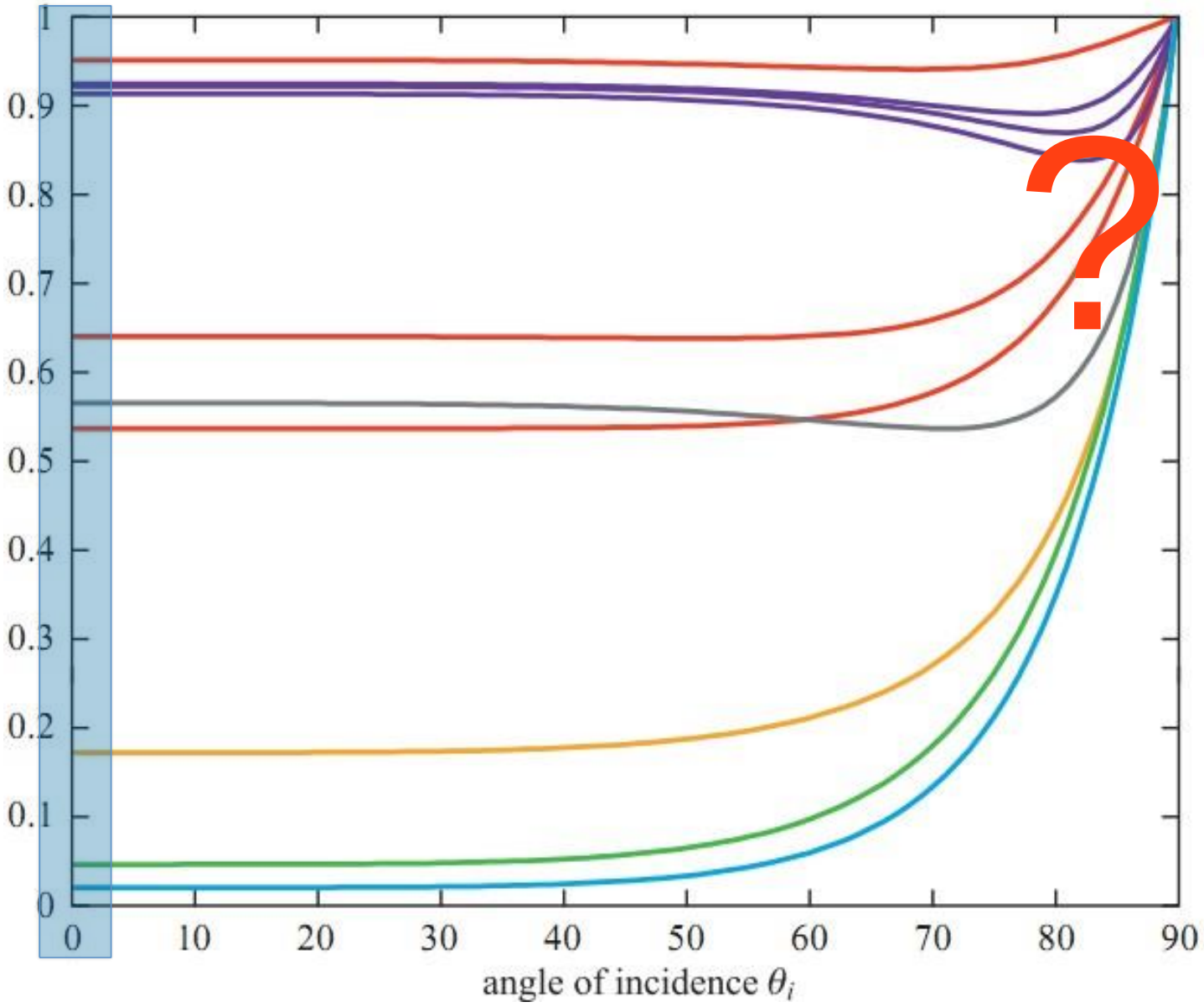
Metal	F_0 (Linear,Float)	F_0 (sRGB,U8)	Color
Titanium	0.542,0.497,0.449	194,187,179	
Chromium	0.549,0.556,0.554	196,197,196	
Iron	0.562,0.565,0.578	198,198,200	
Nickel	0.660,0.609,0.526	212,205,192	
Platinum	0.673,0.637,0.585	214,209,201	
Copper	0.955,0.638,0.538	250,209,194	
Palladium	0.733,0.697,0.652	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	

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	

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	

Fresnel Reflectance



The Schlick Approximation to Fresnel

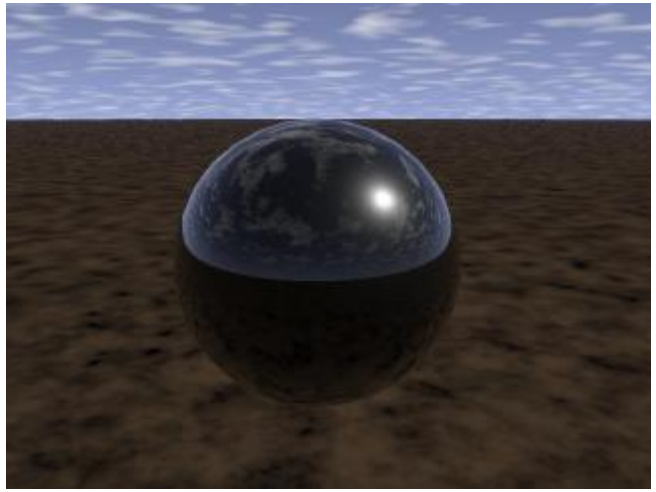
- Fairly accurate, cheap, parameterized by F_0

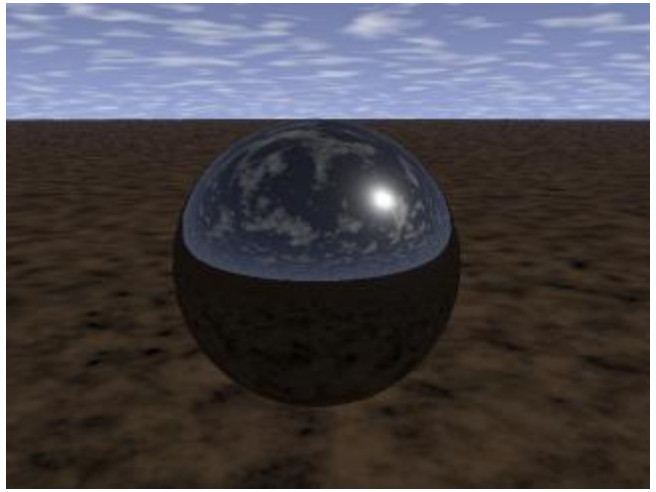
$$F_{\text{Schlick}}(F_0, \mathbf{l}, \mathbf{n}) = F_0 + (1 - F_0)(1 - (\mathbf{l} \cdot \mathbf{n}))^5$$

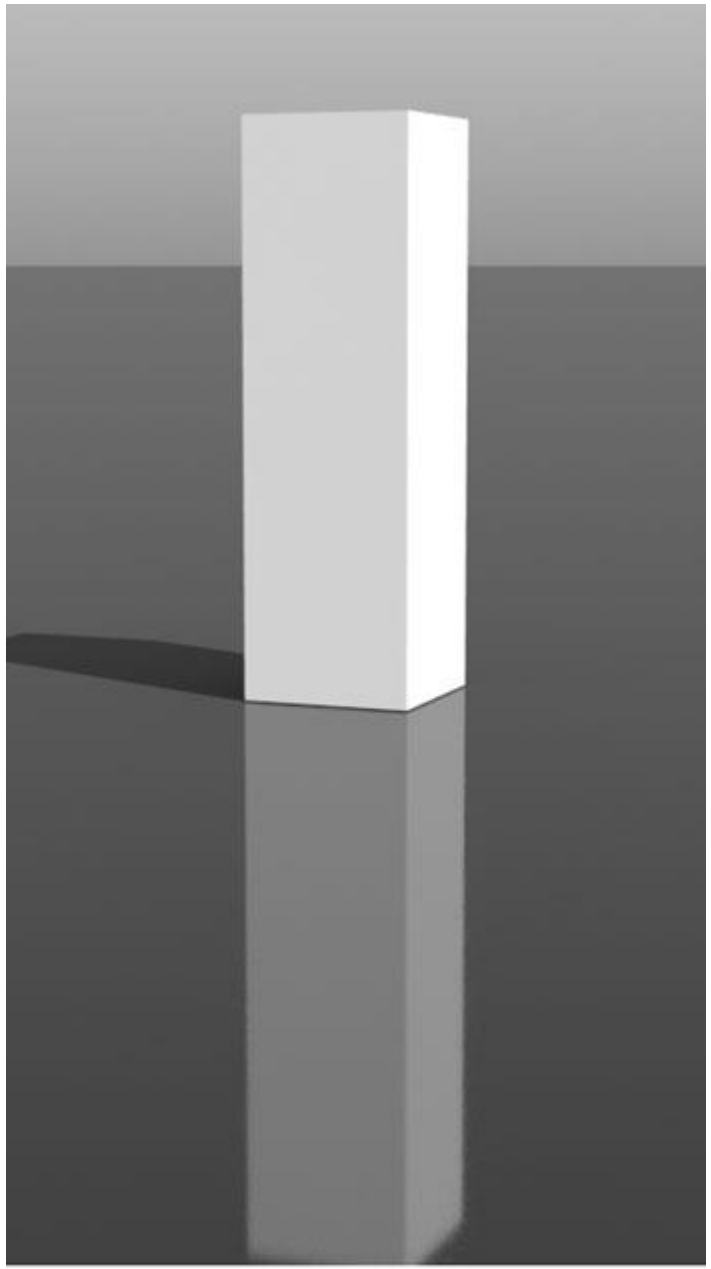
- For microfacet BRDFs ($\mathbf{n} = \mathbf{h}$):

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

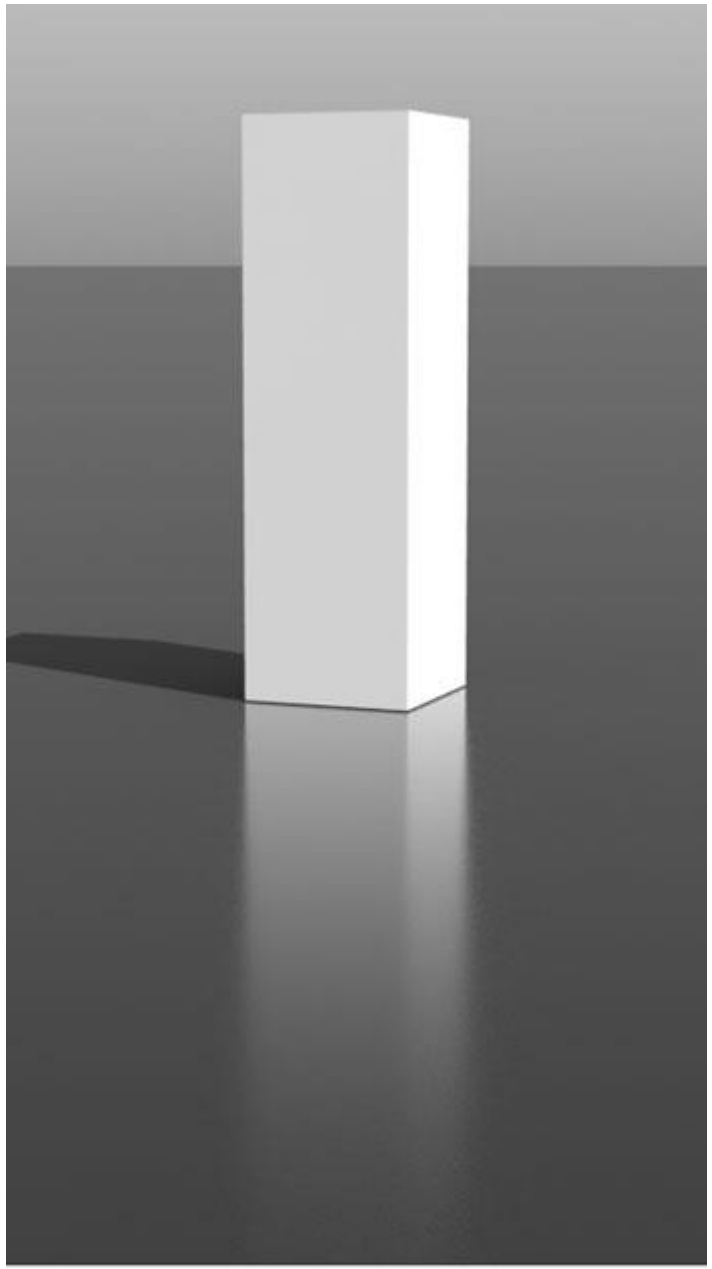
With and without Fresnel reflectance







without fresnel



with fresnel

Normal Distribution Function

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

Different Normal Distribution Functions

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

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

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

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

Let $D_{BlinnPhong}(h) = K(\underline{n.h})^{\alpha}$

$$\int_{\Theta} D_{BlinnPhong}(h)(n.h)\,dh = 1$$

$$\int_{\Theta} K(\underline{n.h})^{\alpha}(n.h)\,dh = 1$$

$$K\int_0^{2\pi}\int_0^{\pi}(\underline{n.h})^{\alpha}(n.h)\sin\theta\ d\theta d\varphi = 1$$

$$K\int_0^{2\pi}\left(\int_0^{\pi/2}(\underline{n.h})^{\alpha}(n.h)\sin\theta\ d\theta + \int_{\pi/2}^{\pi}(\underline{n.h})^{\alpha}(n.h)\sin\theta\ d\theta\right)d\varphi = 1$$

$$K\int_0^{2\pi}\left(\int_0^{\pi/2}(n.h)^{\alpha+1}\sin\theta\ d\theta + \int_{\pi/2}^{\pi}0\,(n.h)\sin\theta\ d\theta\right)d\varphi = 1$$

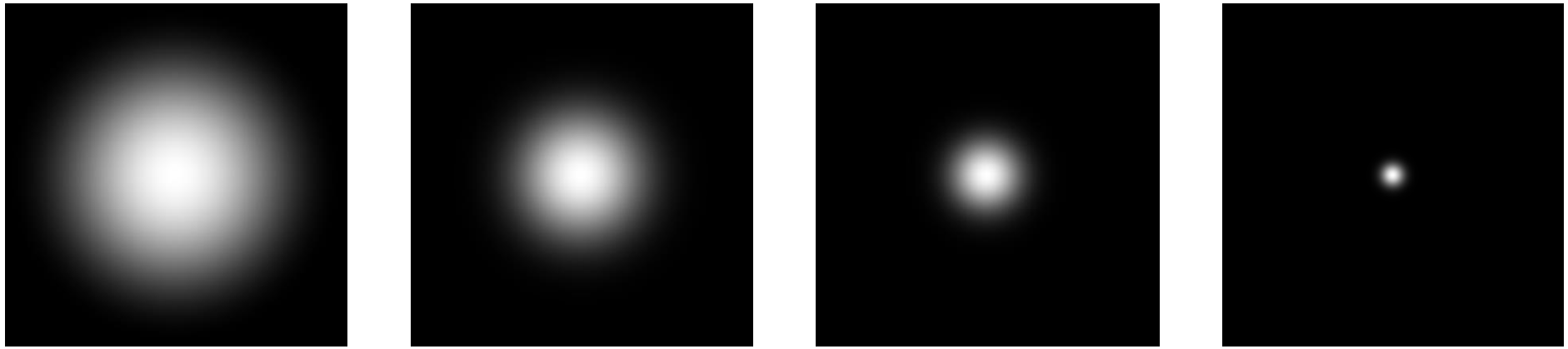
$$K\,2\pi\int_0^{\pi/2}(\cos\theta)^{\alpha+1}\,d(-\cos\theta) = 1$$

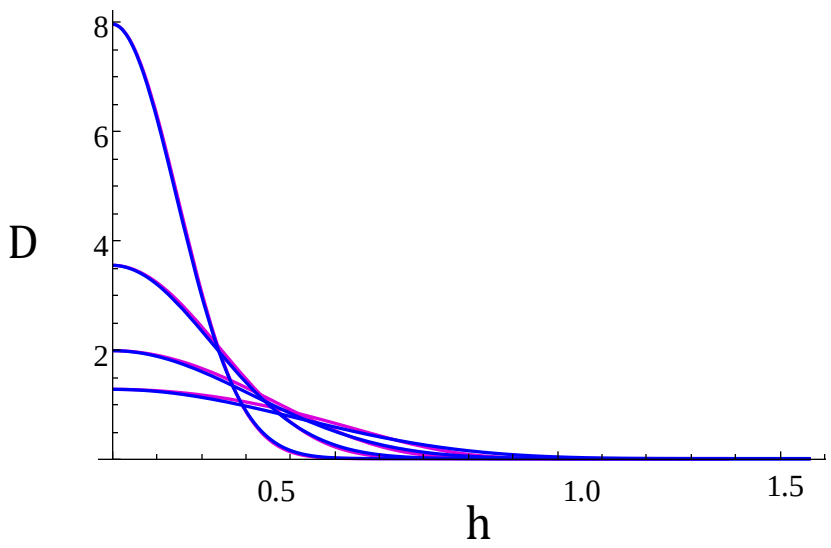
$$-K\,2\pi\left[\frac{\cos^{\alpha+2}\theta}{\alpha+2}\right]_0^{\pi/2} = 1$$

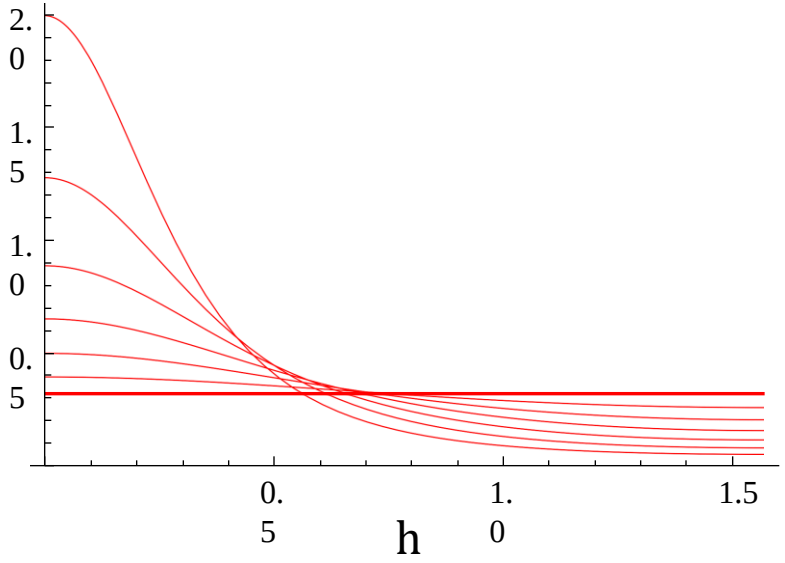
$$-K\,2\pi\frac{-1}{\alpha+2} = 1$$

$$K = \frac{\alpha+2}{2\pi}$$

$$\therefore\ D_{BlinnPhong}(h) = \frac{\alpha+2}{2\pi}(\underline{n.h})^{\alpha}$$

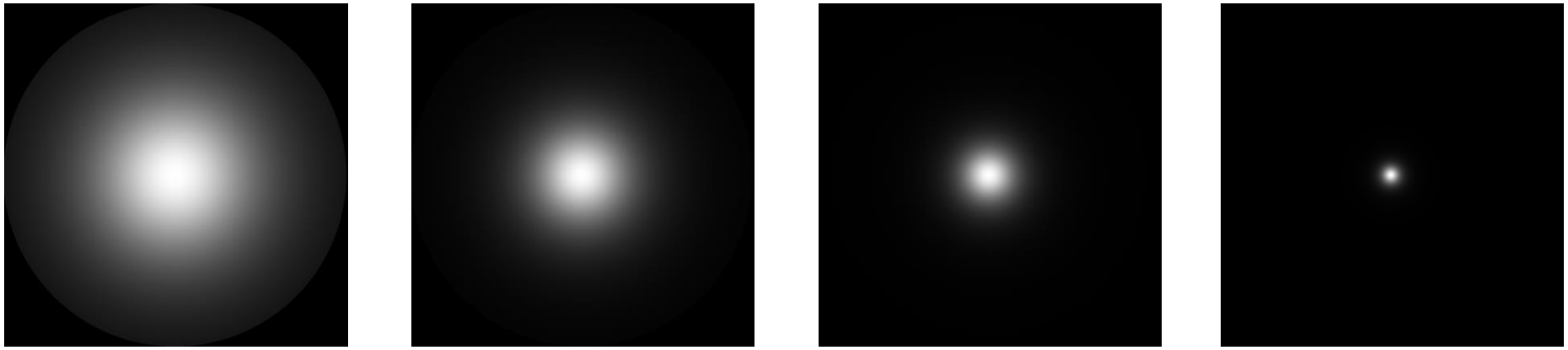


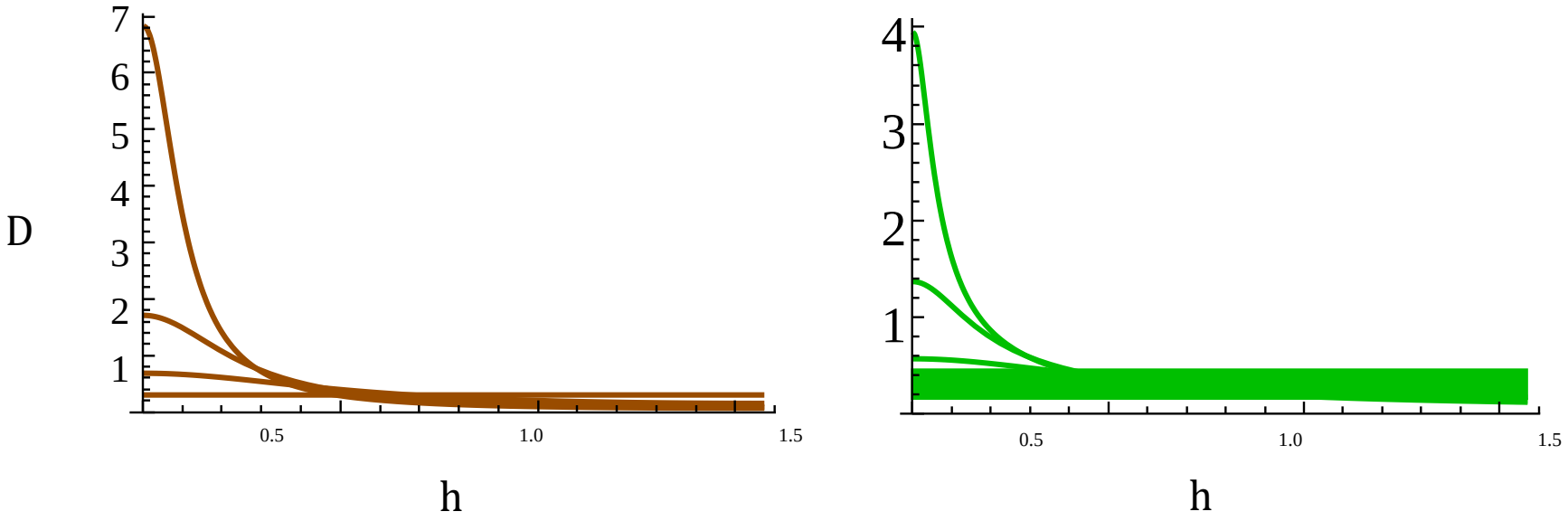




.

Blobby highlights: Beckmann, Phong, Blinn-Phong

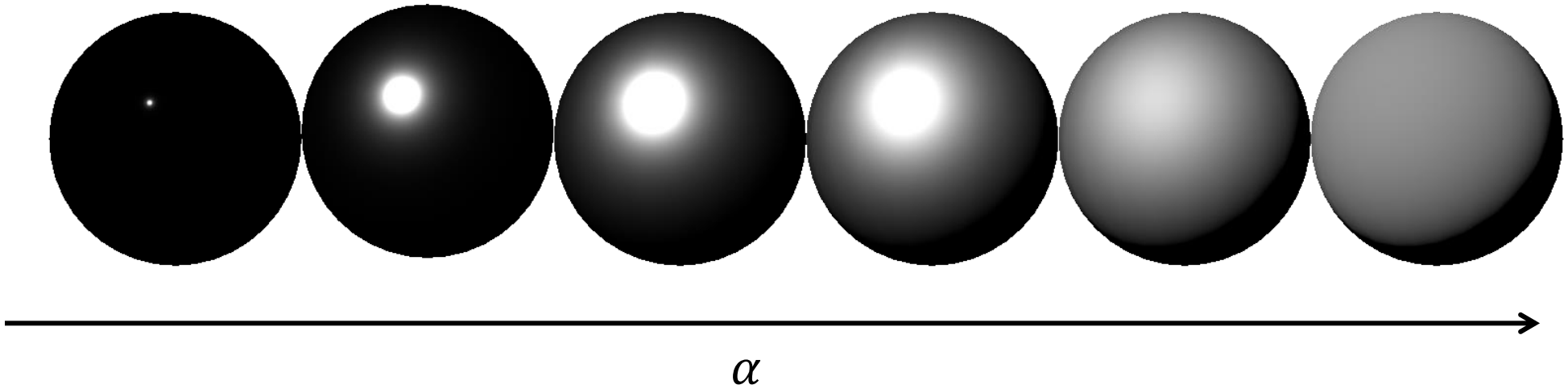




Sharp highlights: GGX

Normal Distribution Function

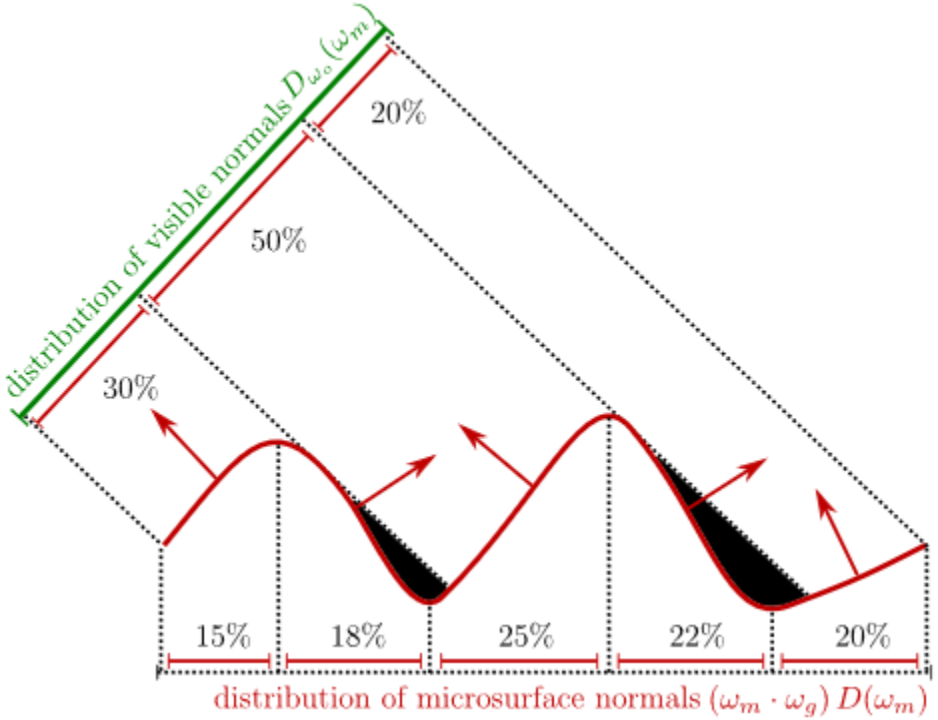
- approximates the relative surface area of microfacets aligned to the (halfway) vector
- D(n, h, α) where α indicates surface roughness



Geometry Function

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

Shadowing and Masking from View Direction



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

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

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

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

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

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

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

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

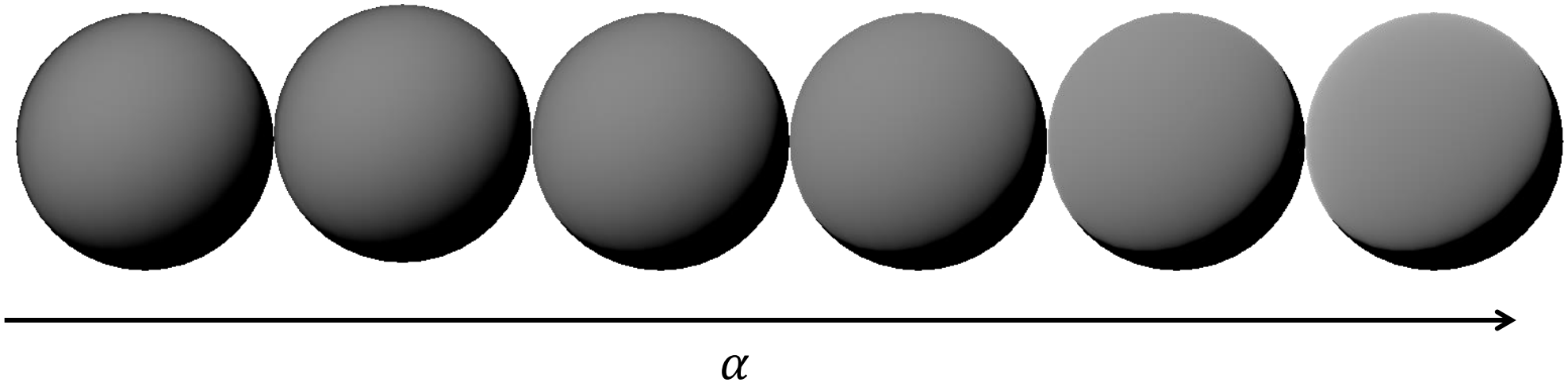
Heitz 2014, Journal of Computer Graphics Techniques Vol. 3, No. 2, 2014

G_{s1}

$$G_{1-Schlick}(v, h) = \frac{(n.v)}{(n.v)(1-k)+k}$$

$$, where\ k = m\sqrt{\frac{2}{\pi}}, m\ is\ the\ rms\ roughness$$

Schlick method results



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

Concentration of active microfacets

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

Visibility of microfacets

Concentration of active microfacets

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

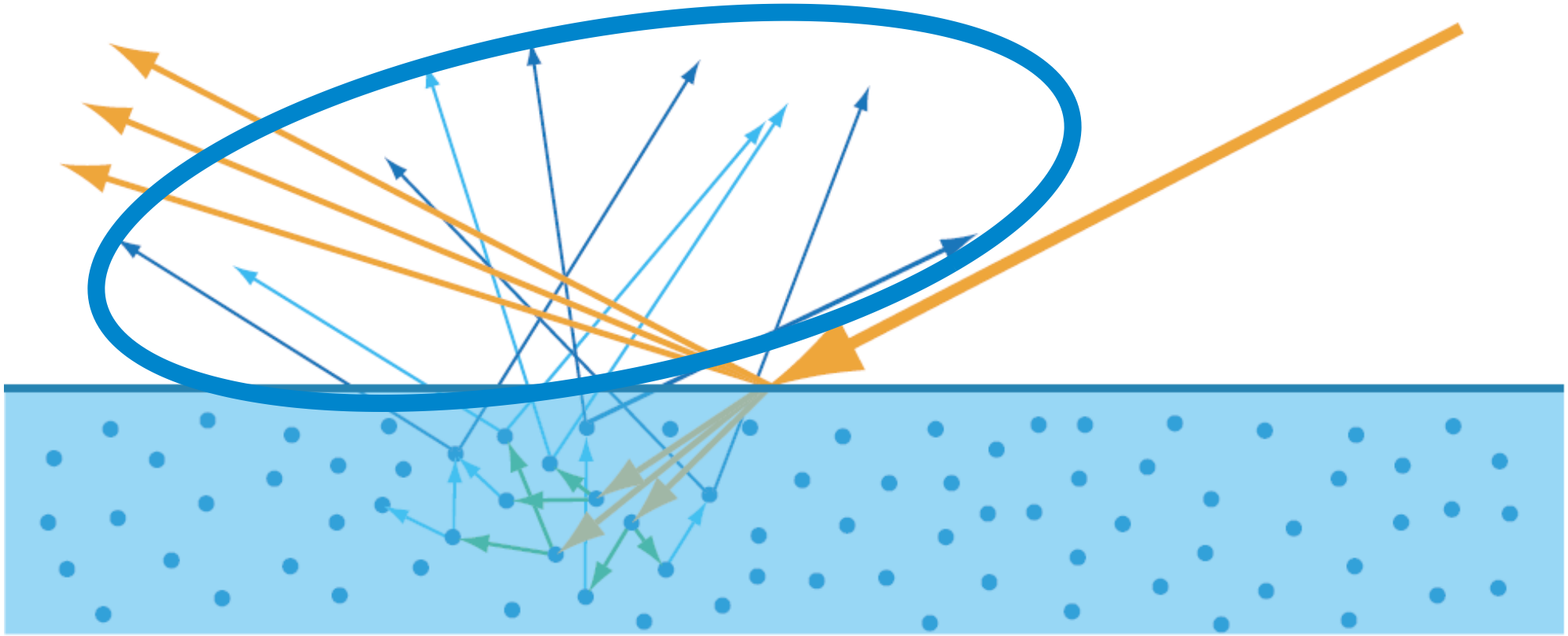
Fresnel reflectance

Visibility of microfacets

Concentration of active microfacets

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

Subsurface Reflection (Diffuse Term)



Lambert

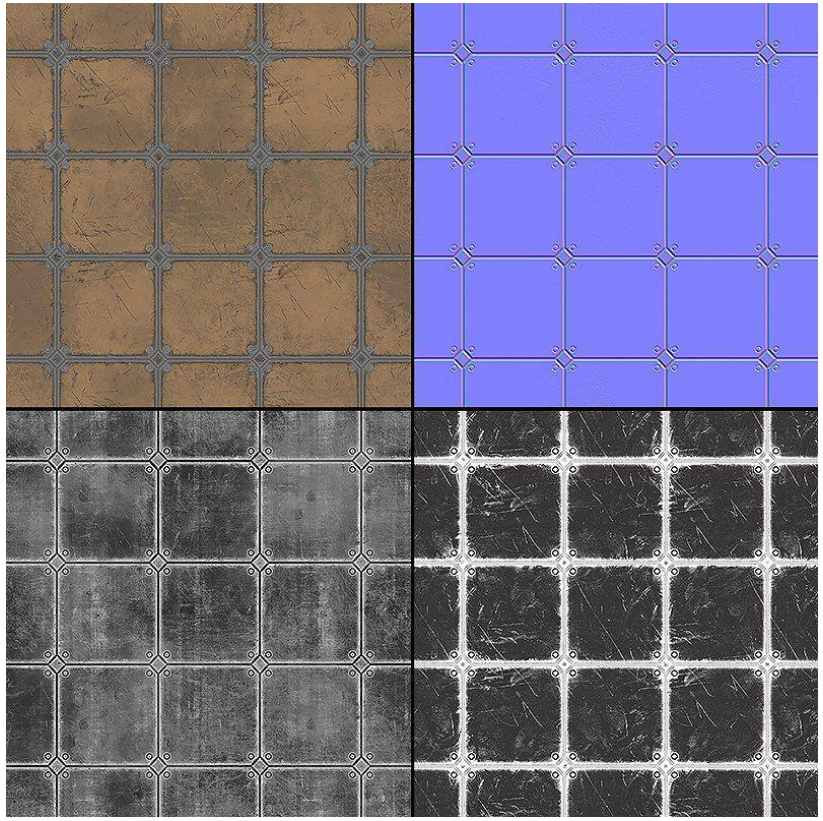
- Constant value ($\mathbf{n} \bullet \mathbf{l}$ is part of reflectance equation):

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

- $\mathbf{c}_{\mathrm{diff}}$: fraction of light reflected, or diffuse color, also called **albedo**

Textures

- Albedo
- Normal
- Roughness
- Metallic

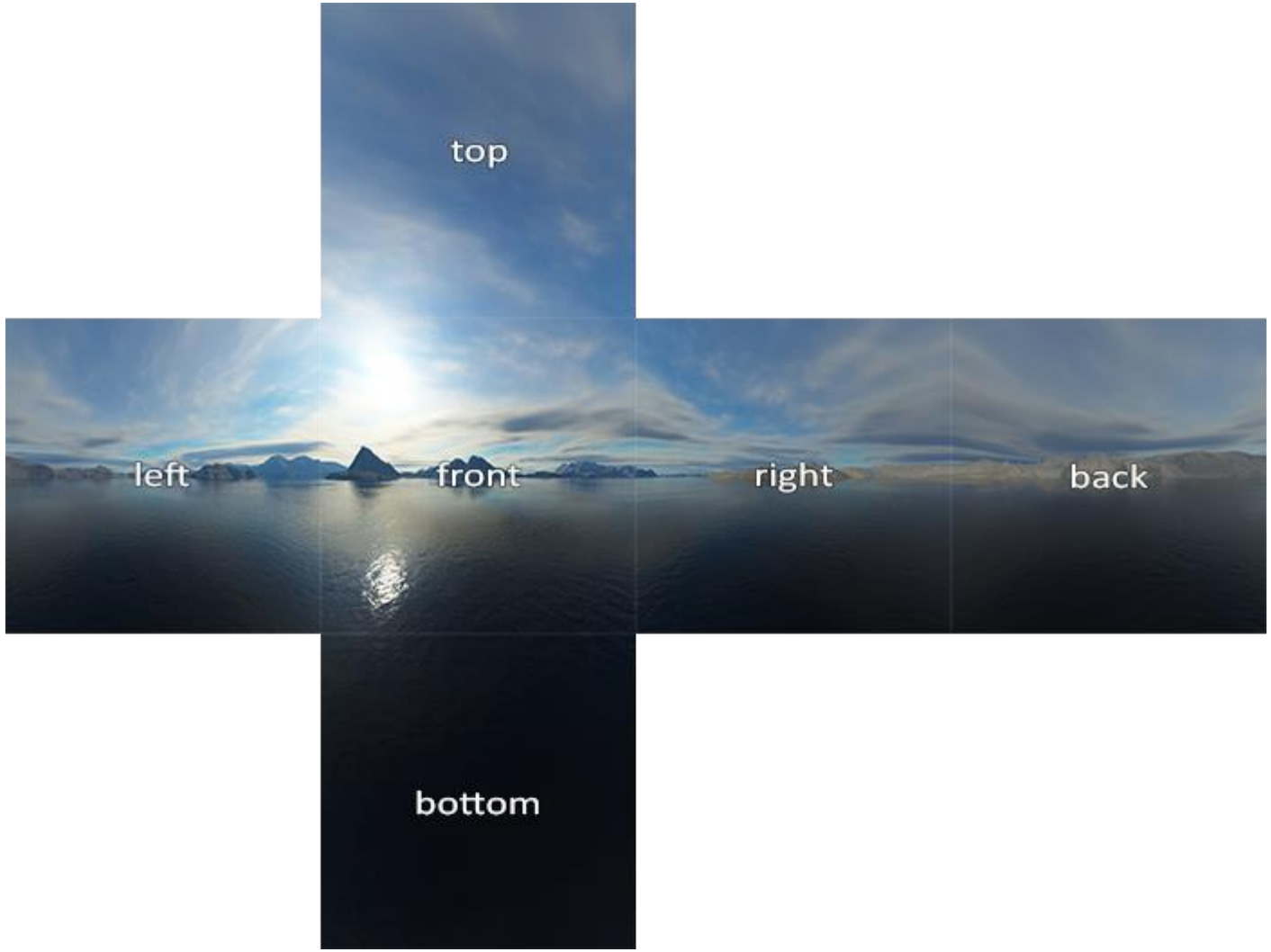


Example microfacet BRDF

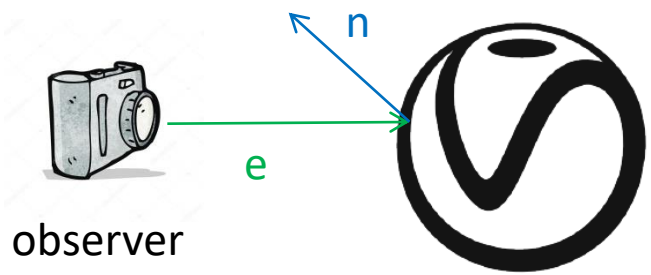
- http://simonstechblog.blogspot.com/2011/12/microfacet-brdf.html
- https://learnopengl.com/PBR/Theory

Environment mapping

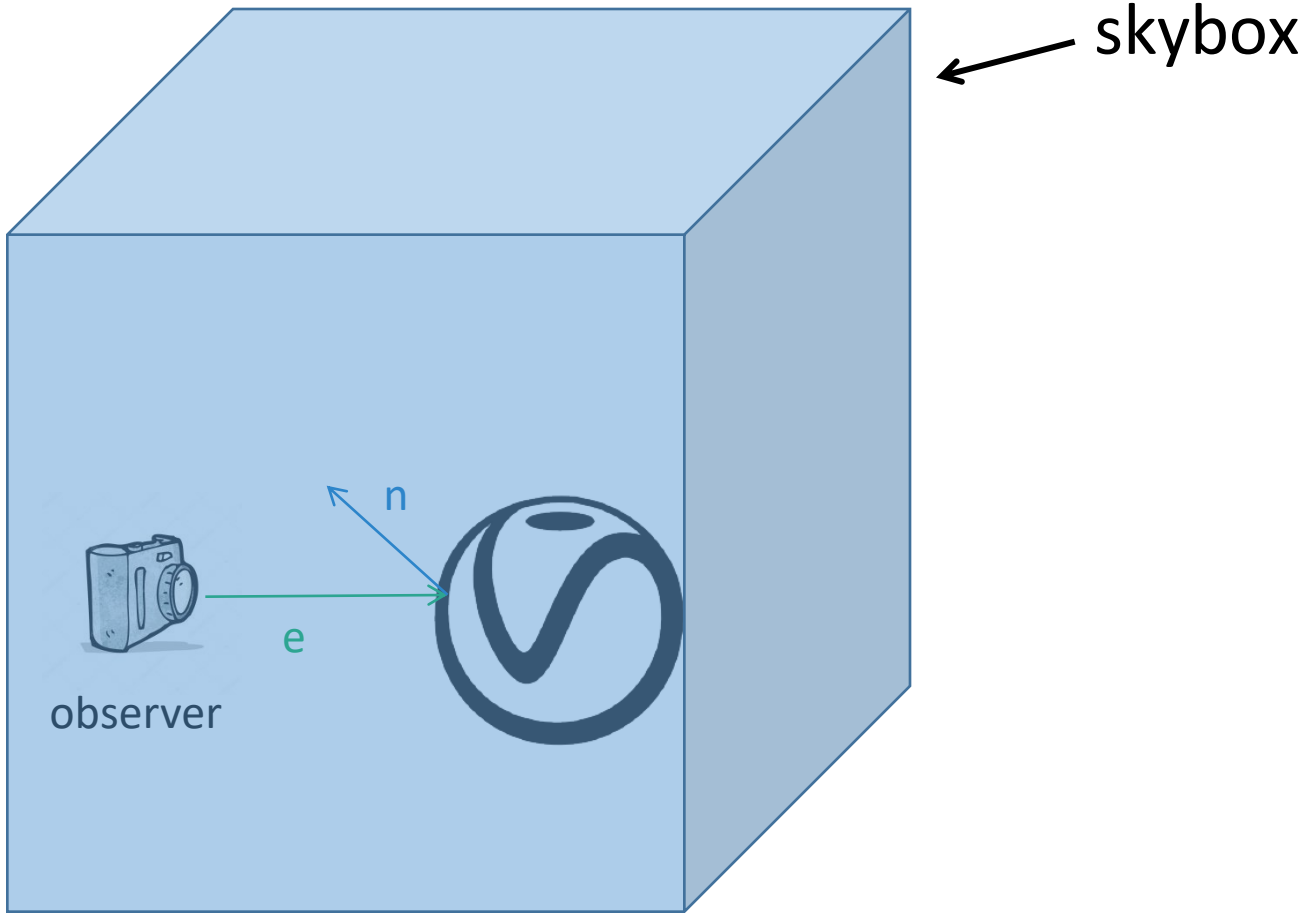
Skybox



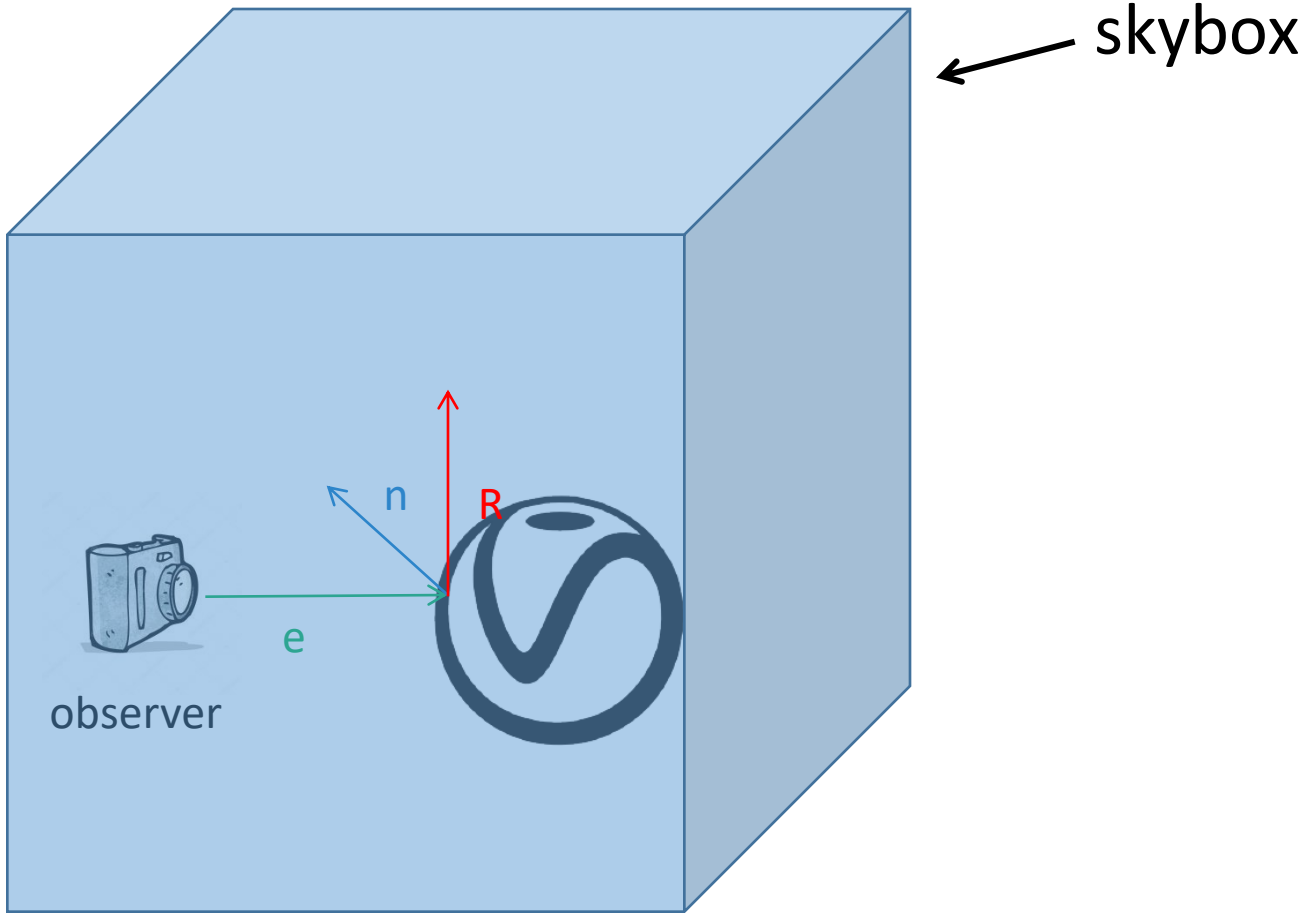
Idea: environment mapping



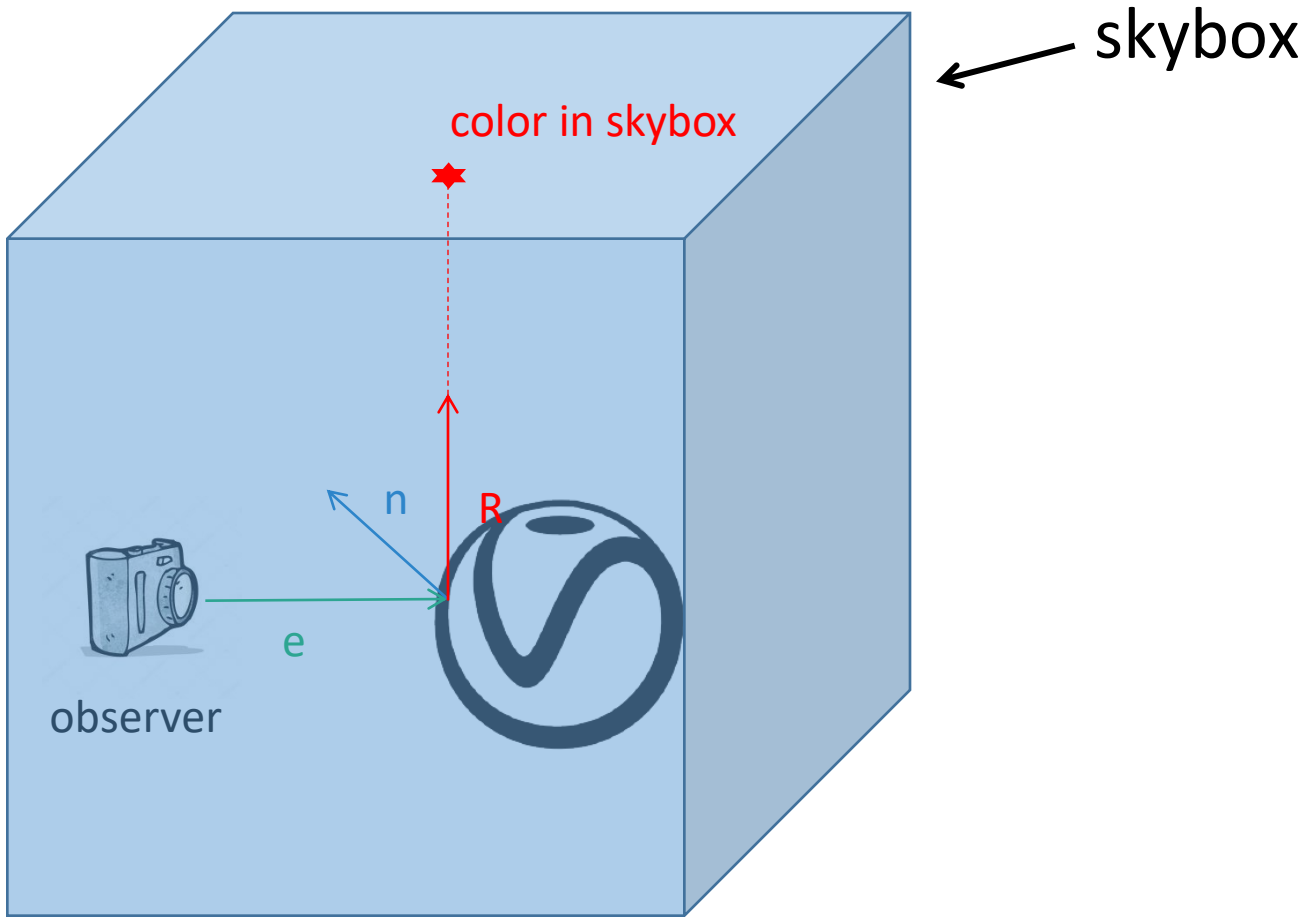
Idea: environment mapping



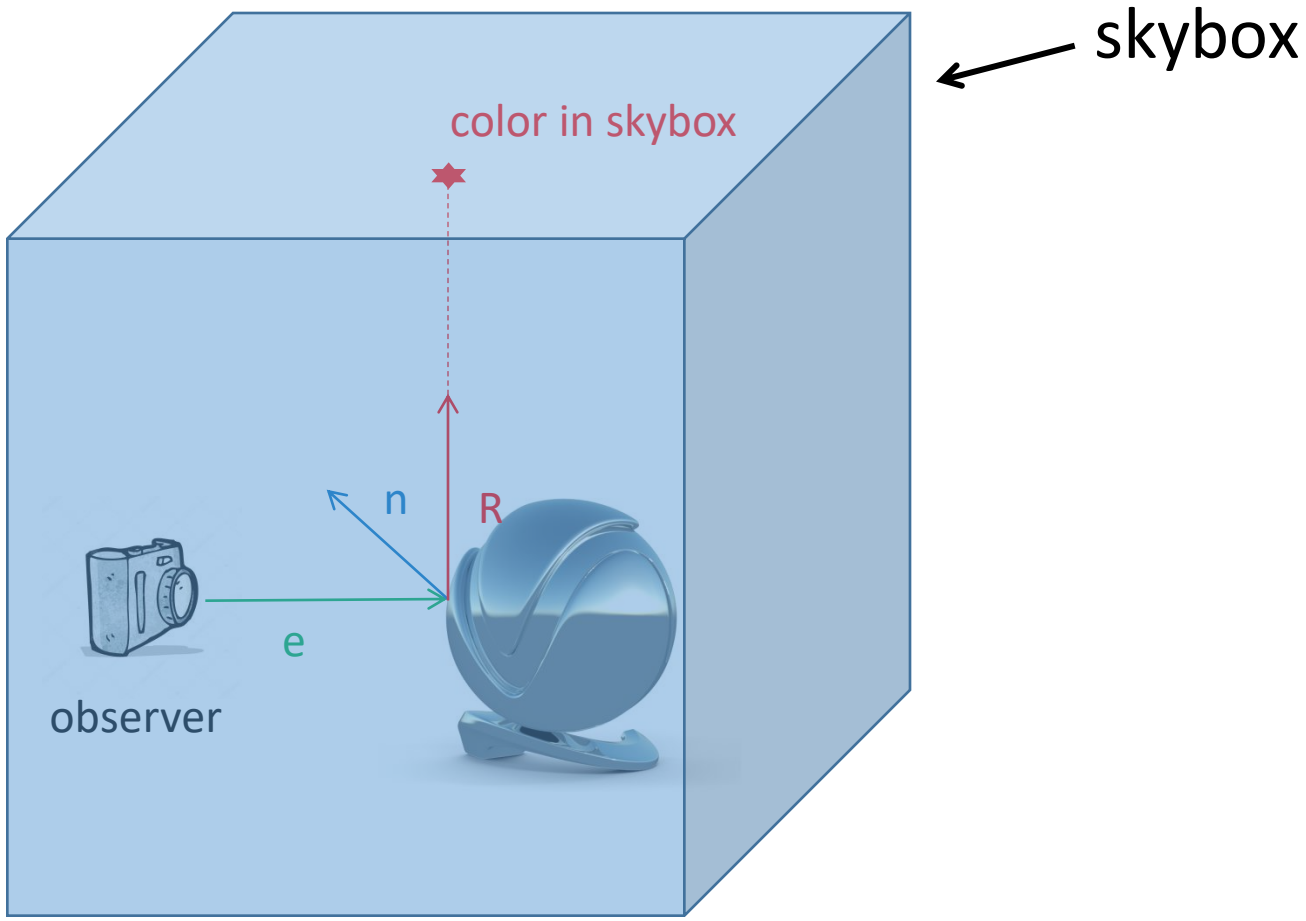
Idea: environment mapping



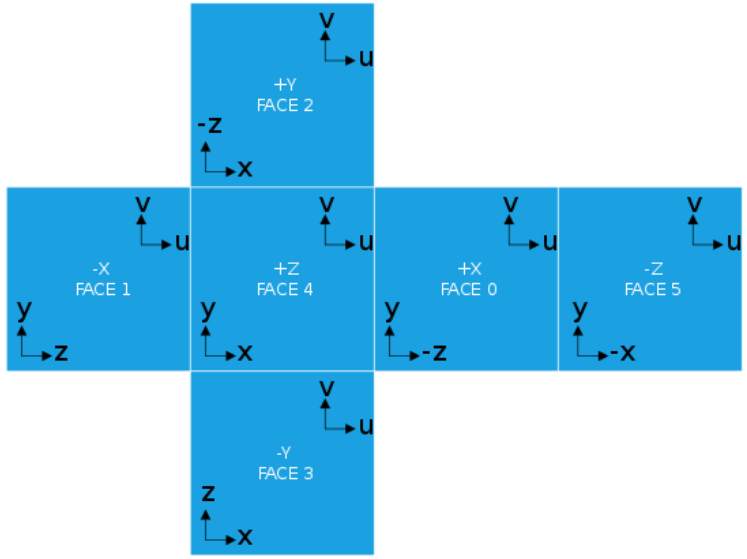
Idea: environment mapping



Idea: environment mapping



6 Textures define cube map



OpenGL:

glTexImage2D() // sends texture data to GPU

GL_TEXTURE_CUBE_MAP_POSITIVE_X, GL_TEXTURE_CUBE_MAP_NEGATIVE_X,
GL_TEXTURE_CUBE_MAP_POSITIVE_Y, GL_TEXTURE_CUBE_MAP_NEGATIVE_Y,
GL_TEXTURE_CUBE_MAP_POSITIVE_Z, GL_TEXTURE_CUBE_MAP_NEGATIVE_Z

Fragment shader:

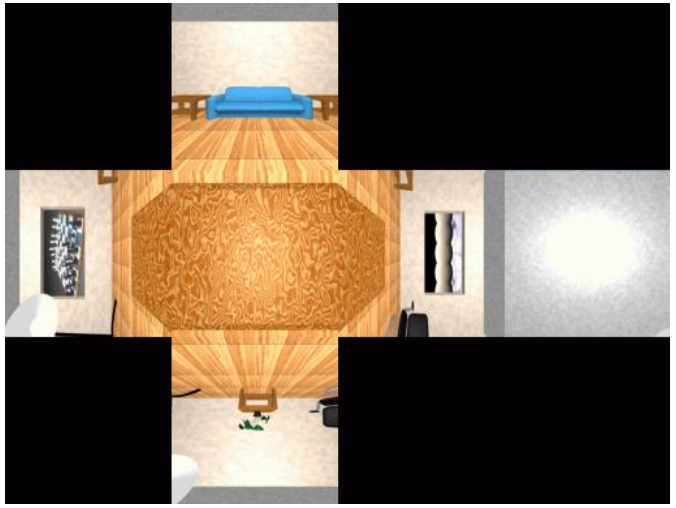
type samplerCube

compute texture coordinates in cube map using reflected vector

Sampling cubemap in GLSL

```
in vec3 texCoord;
out vec4 fragColor;
uniform samplerCube cubemap;

void main (void)
{
    fragColor = texture(cubemap, texCoord);
}
```

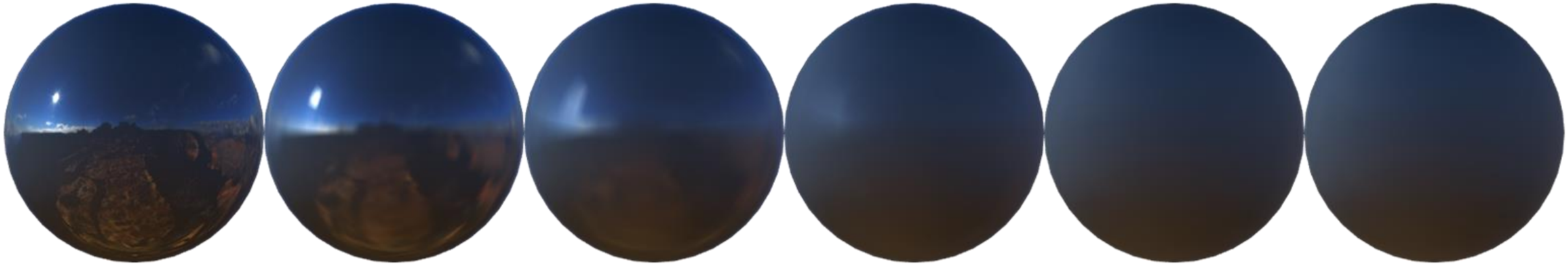
Environment mapping



Teapot environment

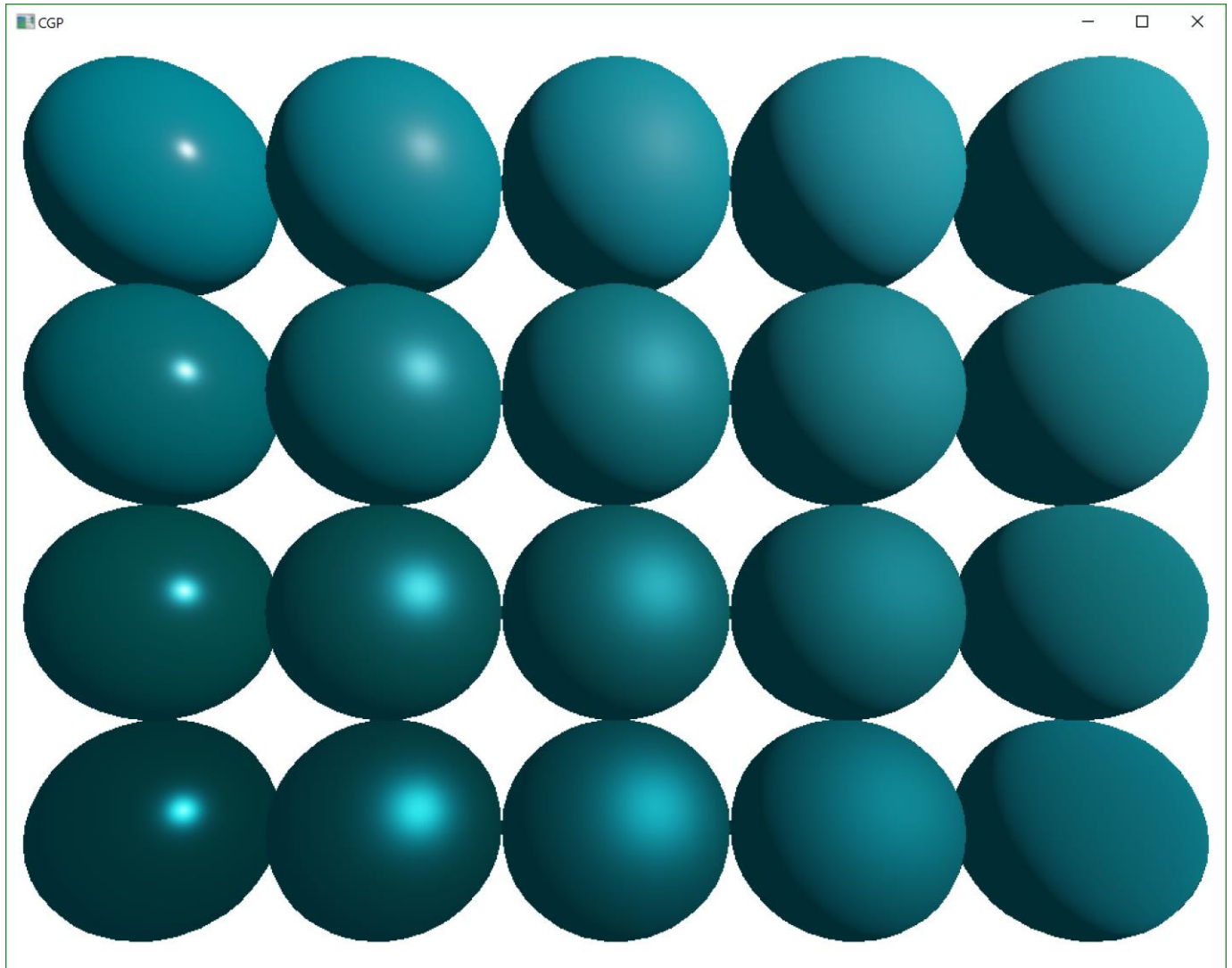
Final effect

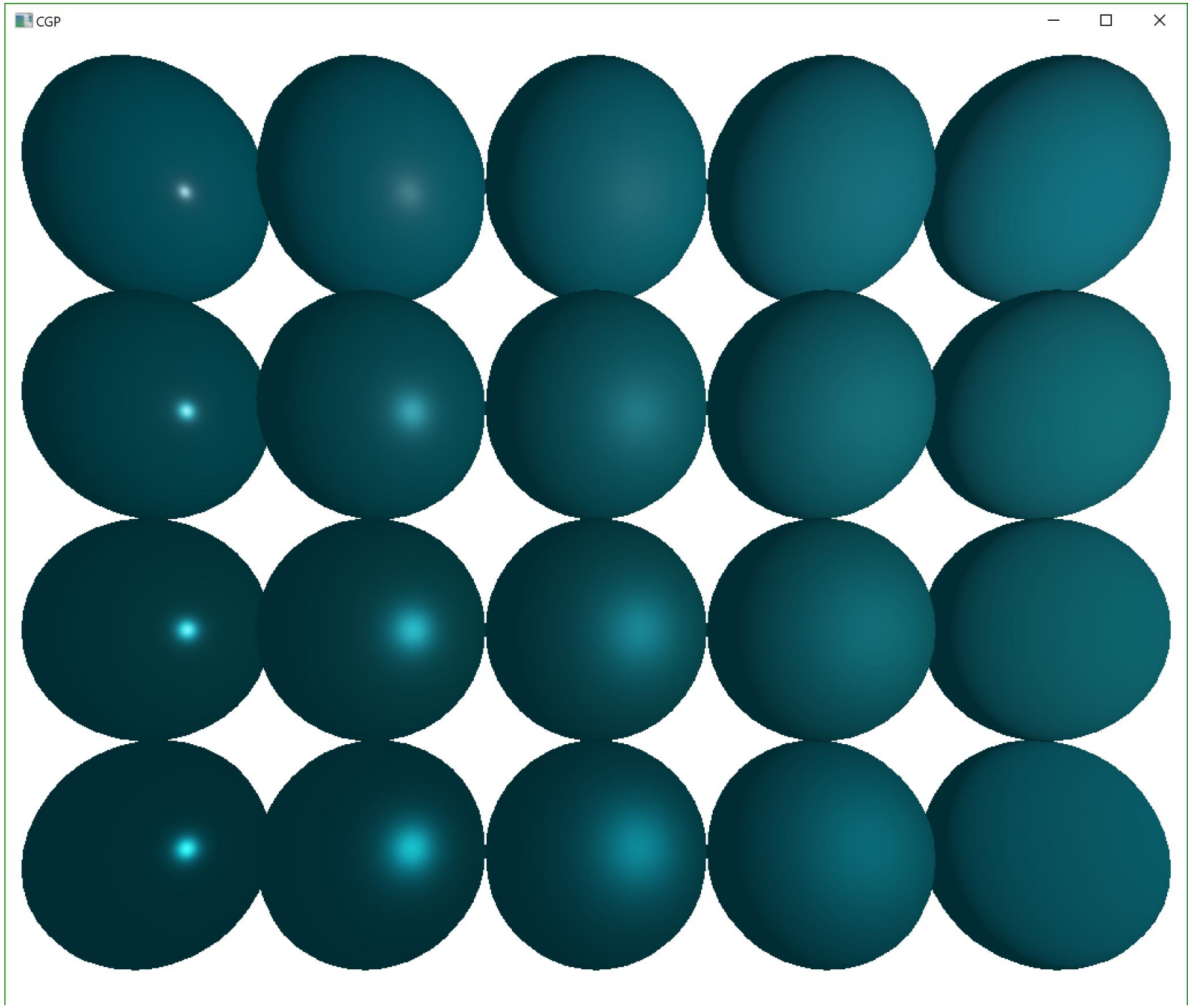
Environment mapping and roughness

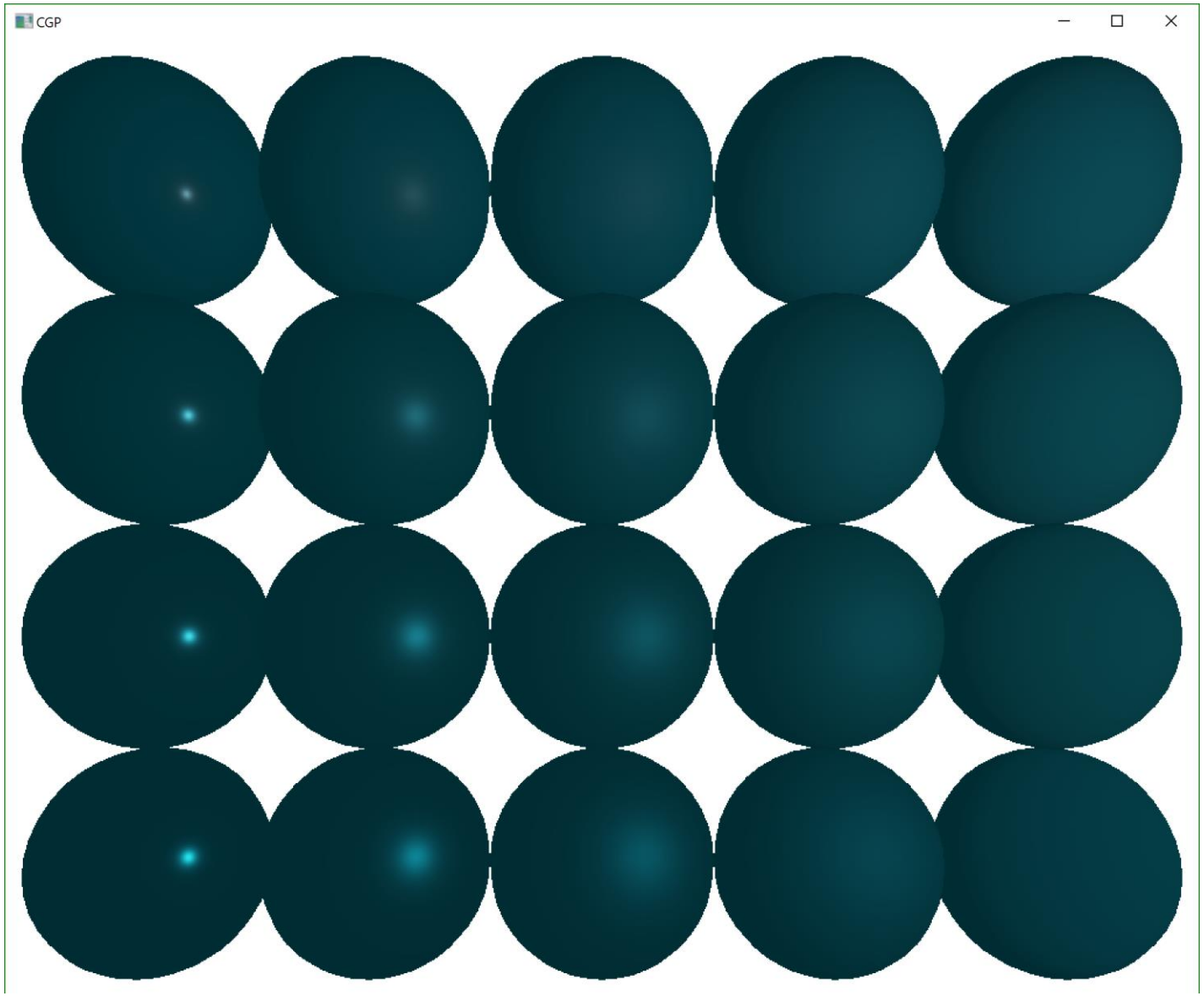


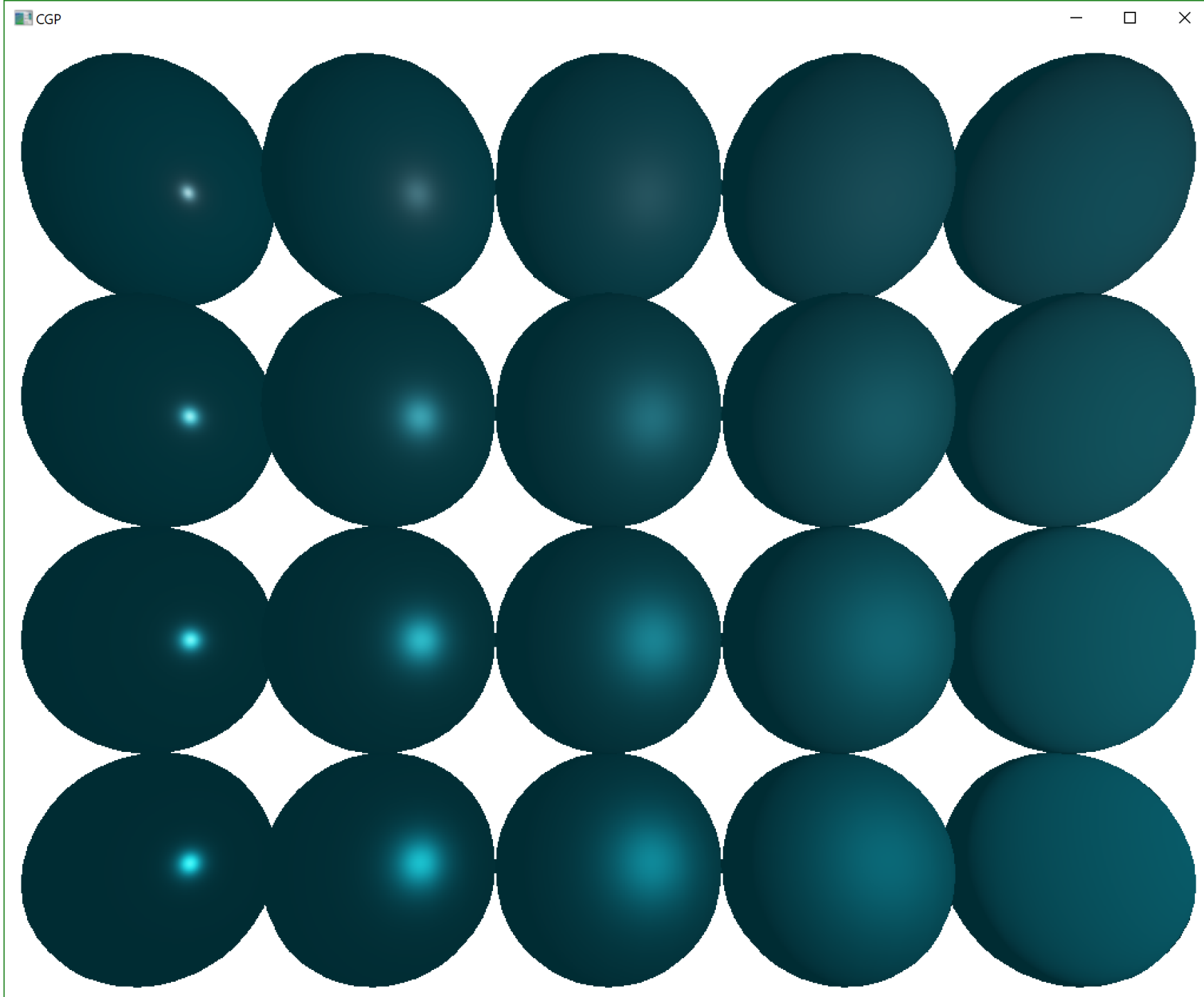
Exercise

- Render 25 different spheres with varying metallic and roughness parameter values using a BRDF of your choice.
- How is F0 calculated?
- How would you add a texture? What information would be necessary?




CGP


CGP


CGP

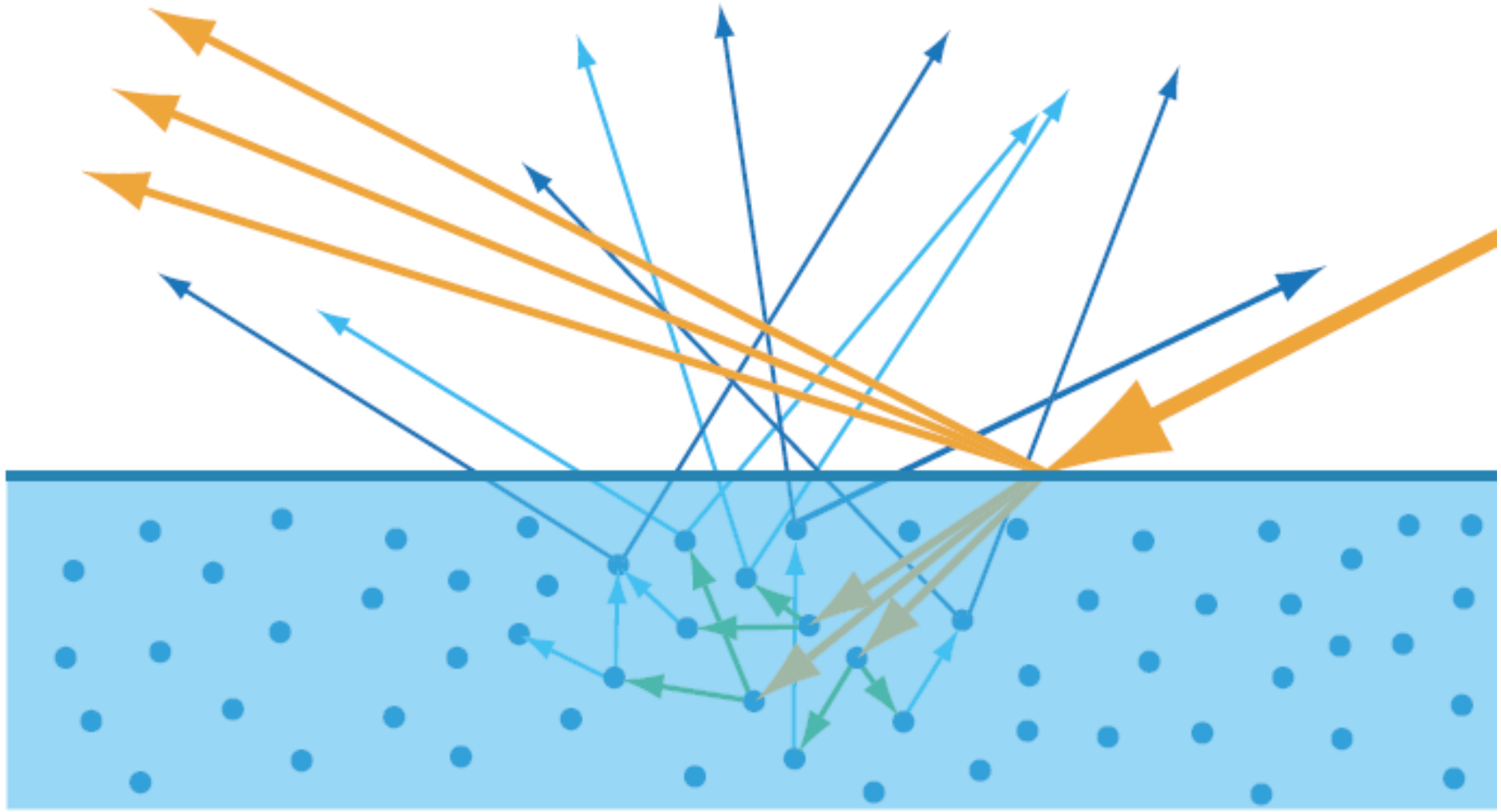
Implementation Overview

```
void main()
{
    vec3 normal = normalize(Normal);

    vec4 BaseColor = vec4(1.0f, 0.0f, 0.0f, 1.0f);
    vec4 SpecularColor = vec4(1.0f, 1.0f, 1.0f, 1.0f);

    vec3 LightDirection = normalize(vec3(0, 4, 4) - Position);
    vec3 ViewDirection = normalize(uCameraPosition - Position);
    vec3 HalfVector = normalize(ViewDirection + LightDirection);
    float Roughness = 0.04f;

    float RefractiveIndex = 0.24f;
    float F0 = pow(((1.0f - RefractiveIndex) / (1.0f + RefractiveIndex)), 2);

    float NdotL = saturate(dot(LightDirection, normal));
    float NdotV = abs(dot(ViewDirection, normal)) + EPSILON;
    float LdotH = saturate(dot(LightDirection, HalfVector));
    float NdotH = saturate(dot(normal, HalfVector));

    float DiffuseFactor = BRDF_Lambert(NdotL);
    float SpecularFactor = 0.0f;
    if(DiffuseFactor > 0.0f)
    {
        SpecularFactor = BRDF_Specular(NdotV, NdotL, NdotH, LdotH, Roughness, F0);
    }
    glFragColor = BaseColor * DiffuseFactor + SpecularColor * SpecularFactor;
}
```

Diffuse Roughness = Specular Roughness

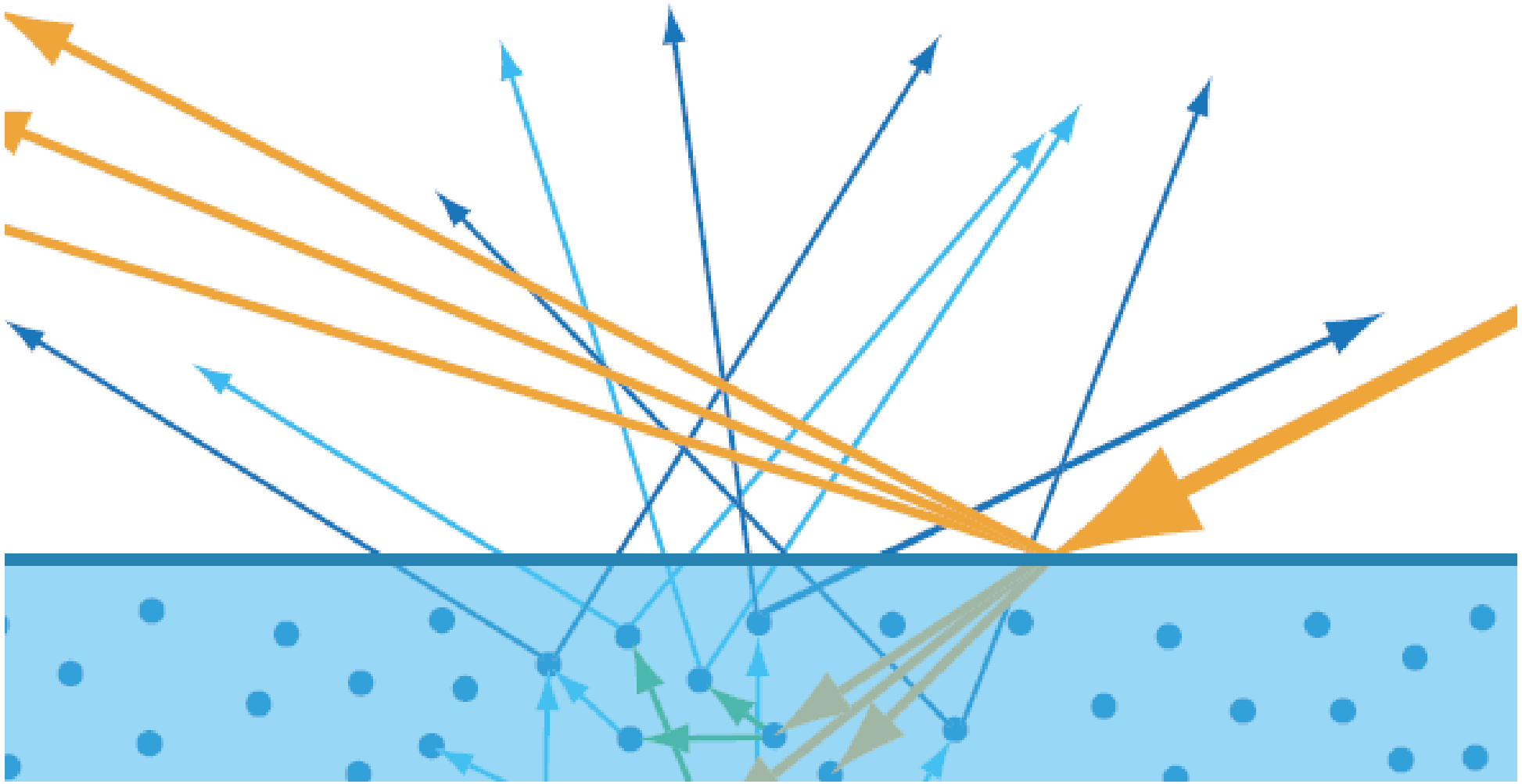
It's become common to use rough diffuse models like Oren-Nayar or "Disney diffuse" for all surfaces, and to plug the specular roughness into them. But I want to take this opportunity to point out a problem with this approach.

Beyond Lambert: Diffuse-Specular Tradeoff



Oren-Nayar reflectance model

Beyond Lambert: Surface Roughness

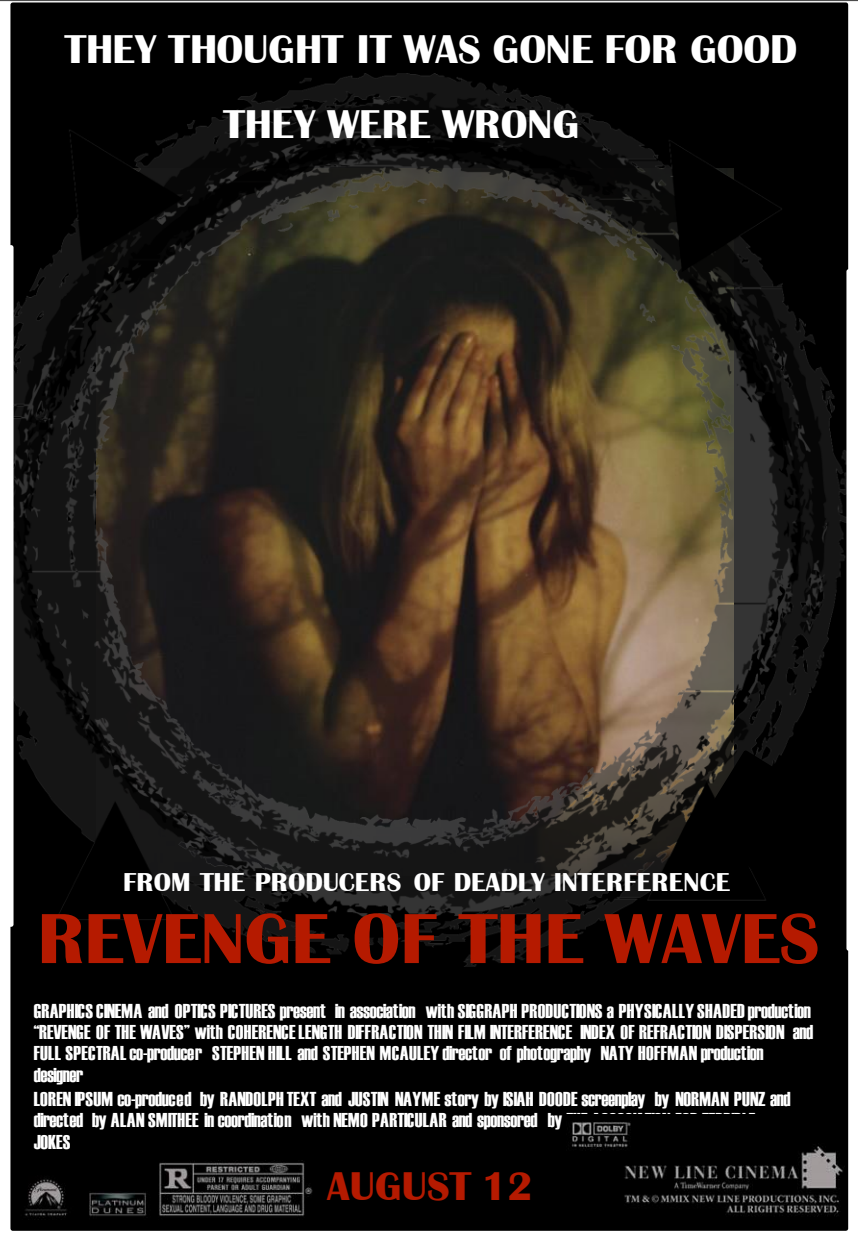


Oren-Nayar reflectance model

Diffuse Roughn

Specular R ess

It's been known for a while that diffuse response effectively smooths out small bumps, as can be seen from LightStage's separate diffuse and specular normal maps. But this applies even more strongly to roughness. Ideally you should use a separate roughness value for these models; otherwise use them sparingly, only for materials where you know the microgeometry is larger than the scattering distance.

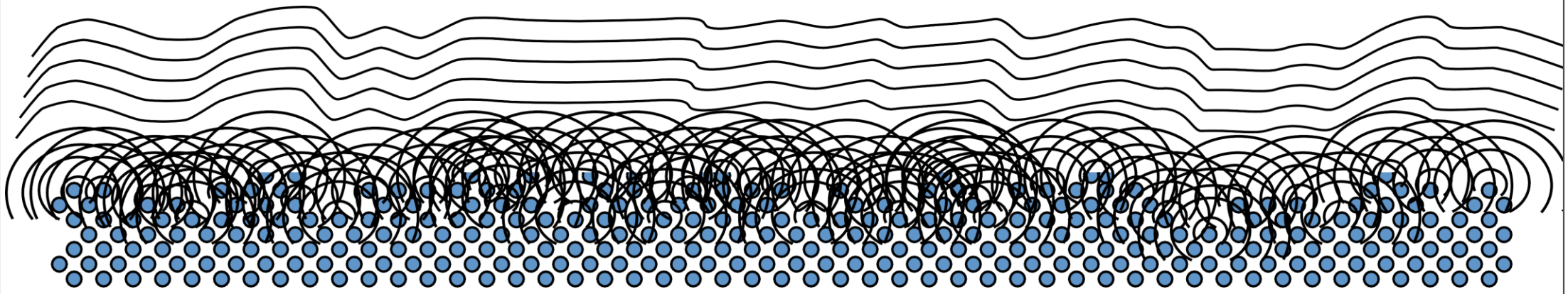


SIGGRAPH2015

I'll briefly go back to wave optics, which I'm sure you thought I had forgotten about after I abandoned it earlier in the talk.

Image by flickr user 55Laney69; licensed CC-Attribution (https://creativecommons.org/licenses/by/2.0/)

Diffraction from Optically-Smooth Surface



With few exceptions, the computer graphics community has either ignored the effects of nanogeometry diffraction or has asserted their insignificance. However, at the recent Material Appearance Modeling symposium, Holzschuch and Pacanowski showed convincing evidence that part of visible BRDF behavior (the “long tail” of the highlights in particular) was due to this phenomenon.

Microgeometry & Nanogeometry

Microgeometry	Nanogeometry
Lobe shape determined by surface statistics (micro-scale NDF)	Lobe shape determined by surface statistics (nano-scale SPD)
No wavelength dependence	Strong wavelength dependence
Incidence angle may affect surface statistics via visibility	Incidence angle may affect surface statistics via foreshortening

It appears that in many materials, reflectance is affected by roughness on both the micro- and nano- scales. I'll go over some high-level differences between the two; for more detail see Holzschuch & Pacanowski's talk. The Nano-scale lobe shape is controlled by the surface SPD (similar to the SPDs we saw earlier, but with respect to 2D surface spatial frequency rather than 1D wave temporal frequency). <*read rest*>

Math → Rendering

Once you have the math, the next step is to implement it in a film or game renderer. My course notes have a bit of background on this, and the industry talks in this course (this year as well as previous years) include many details.

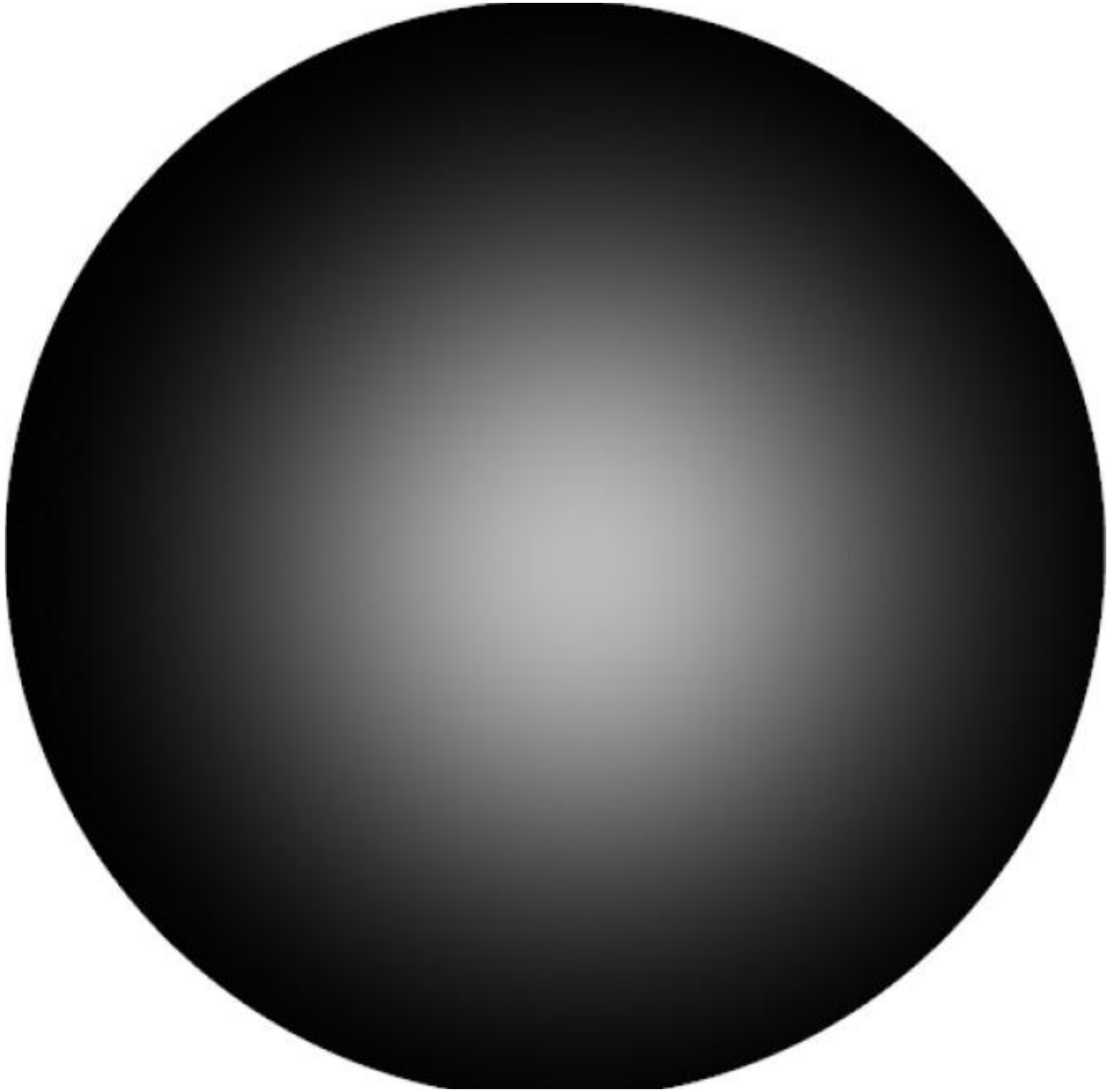


Image from “Rendering Glints on High-Resolution Normal-Mapped Specular Surfaces”, Yan et al., SIGGRAPH 2014

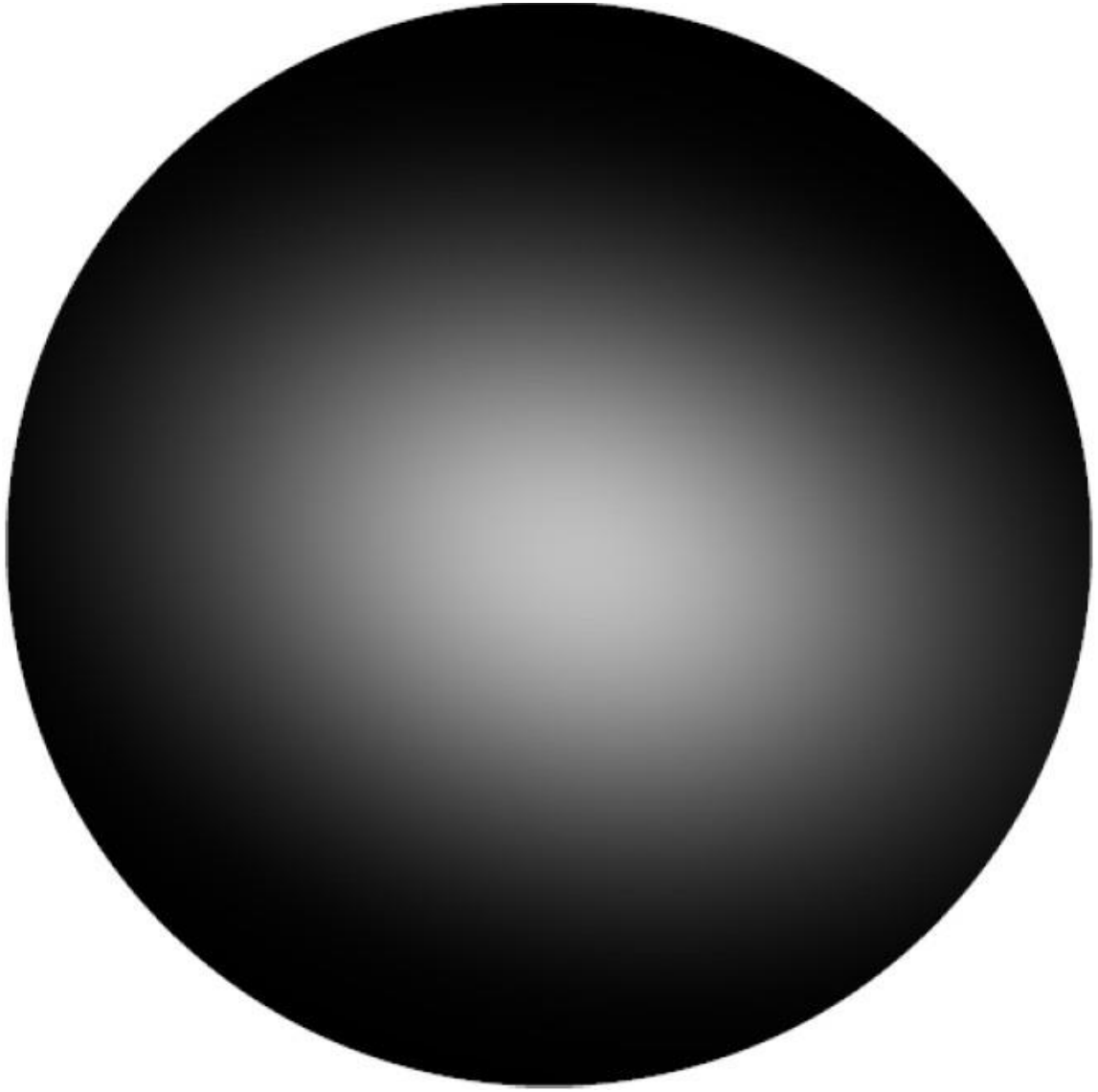


Image from “Rendering Glints on High-Resolution Normal-Mapped Specular Surfaces”, Yan et al., SIGGRAPH 2014

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

...this, causing a “glinty” appearance. Though last year’s “glint rendering” paper offered a solution that could be used for film production, it’s too costly for game use. Games will continue to use more ad-hoc methods; the snow sparkle talk from this year’s “Advances in Real-Time Rendering” course is a good example of the current state of the art.

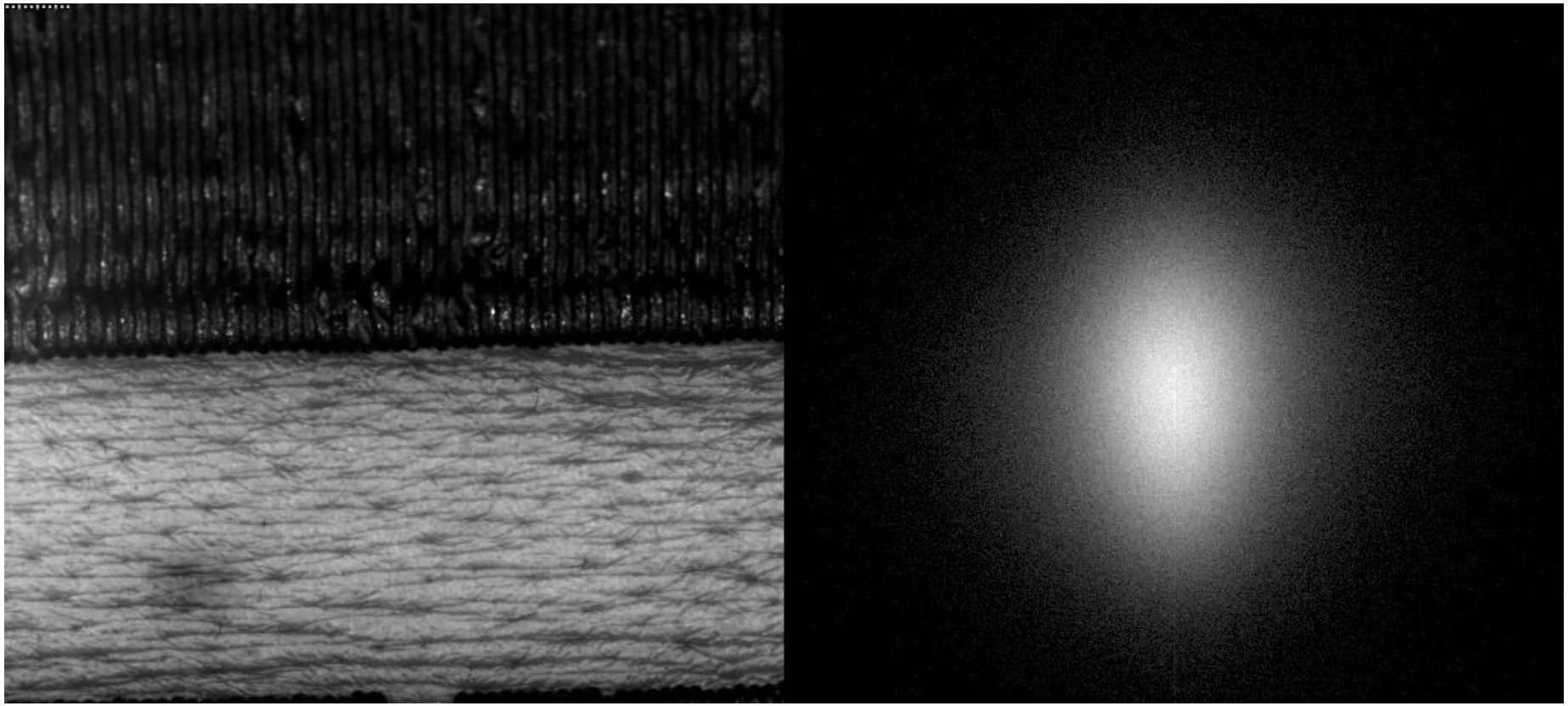


Image from “Skin Microstructure Deformation with Displacement Map Convolution”, Nagano et al., SIGGRAPH 2015

It’s important to account for the effect of surface deformation on NDFs. The following images are from the “Skin Microstructure Deformation” paper by Nagano et al. which will be presented at the “Appearance Capture” session this afternoon. The left side shows a patch of skin under varying amounts of compression & stretch; its NDF is on the right.

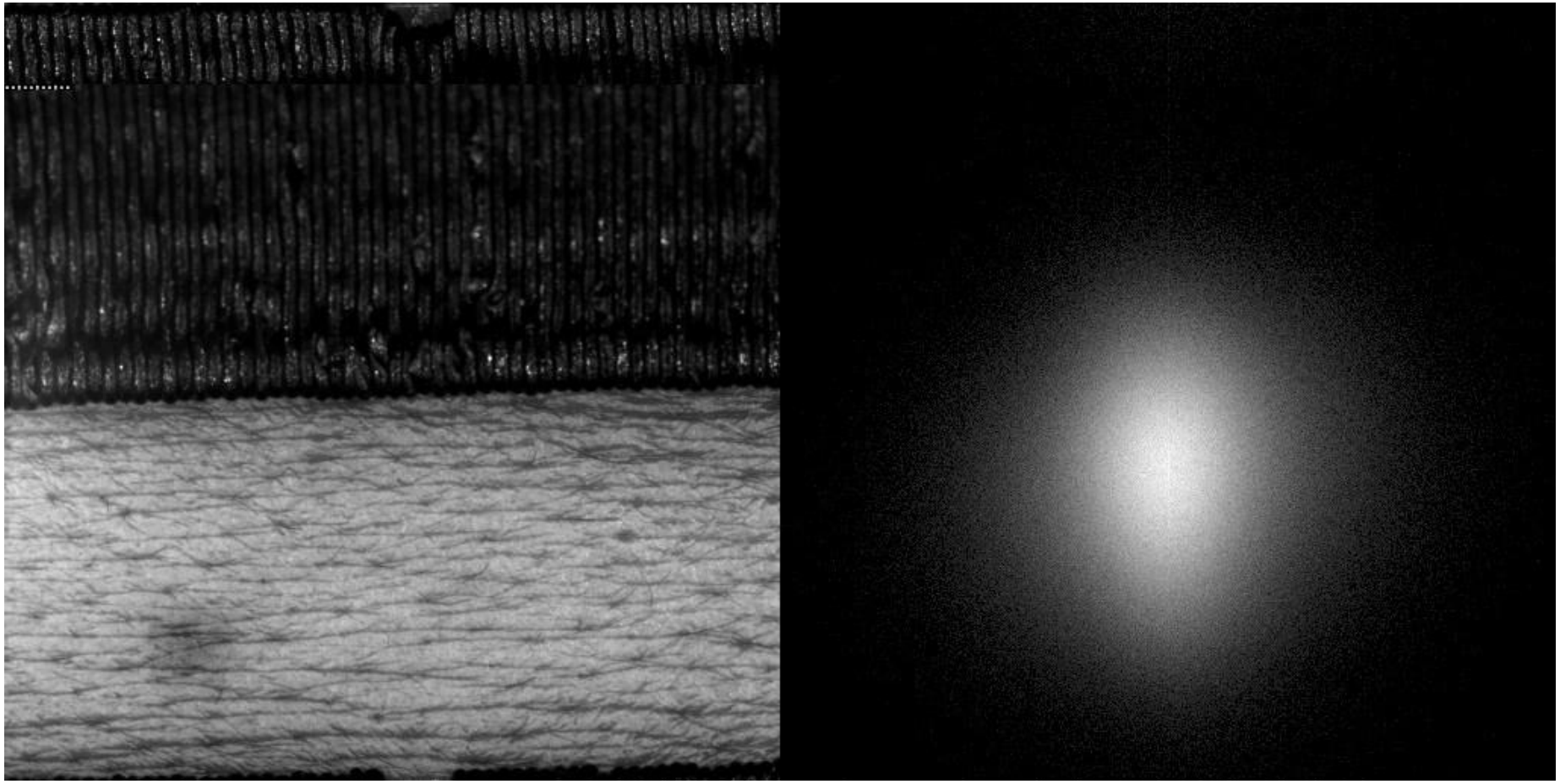


Image from “Skin Microstructure Deformation with Displacement Map Convolution”, Nagano et al., SIGGRAPH 2015

We can see that as the patch of skin changes from being compressed...

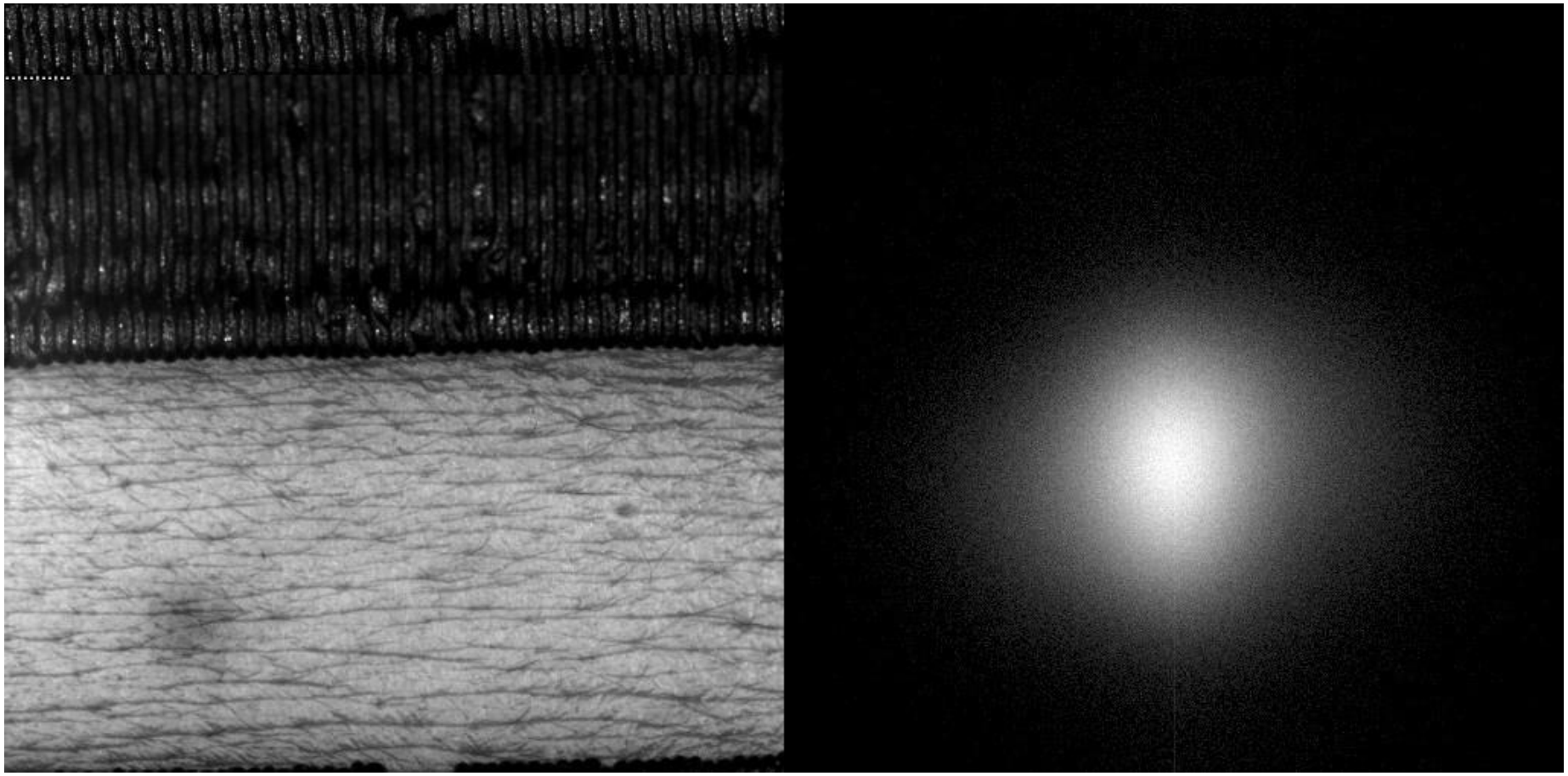


Image from “Skin Microstructure Deformation with Displacement Map Convolution”, Nagano et al., SIGGRAPH 2015

…

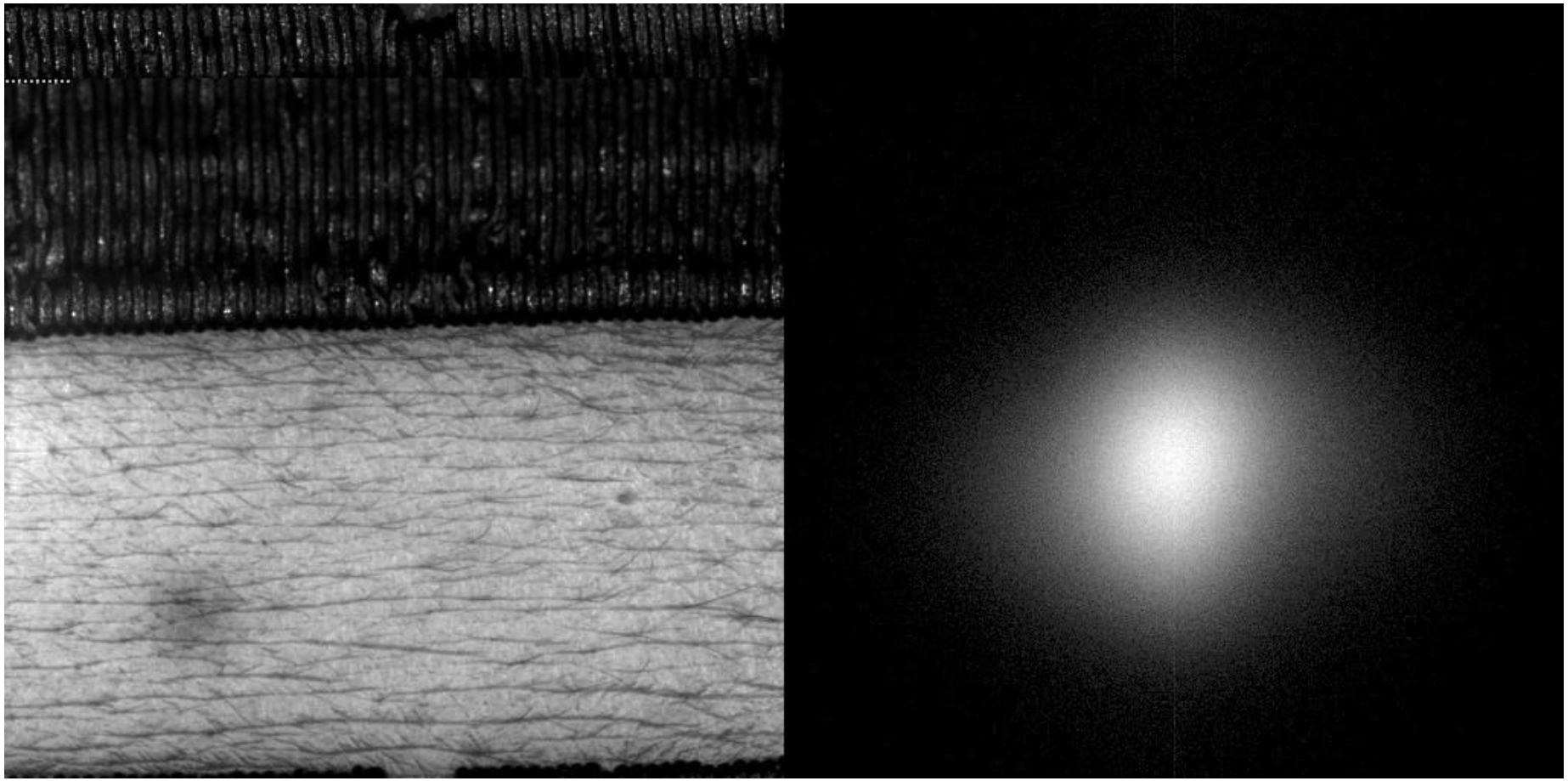


Image from “Skin Microstructure Deformation with Displacement Map Convolution”, Nagano et al., SIGGRAPH 2015

…

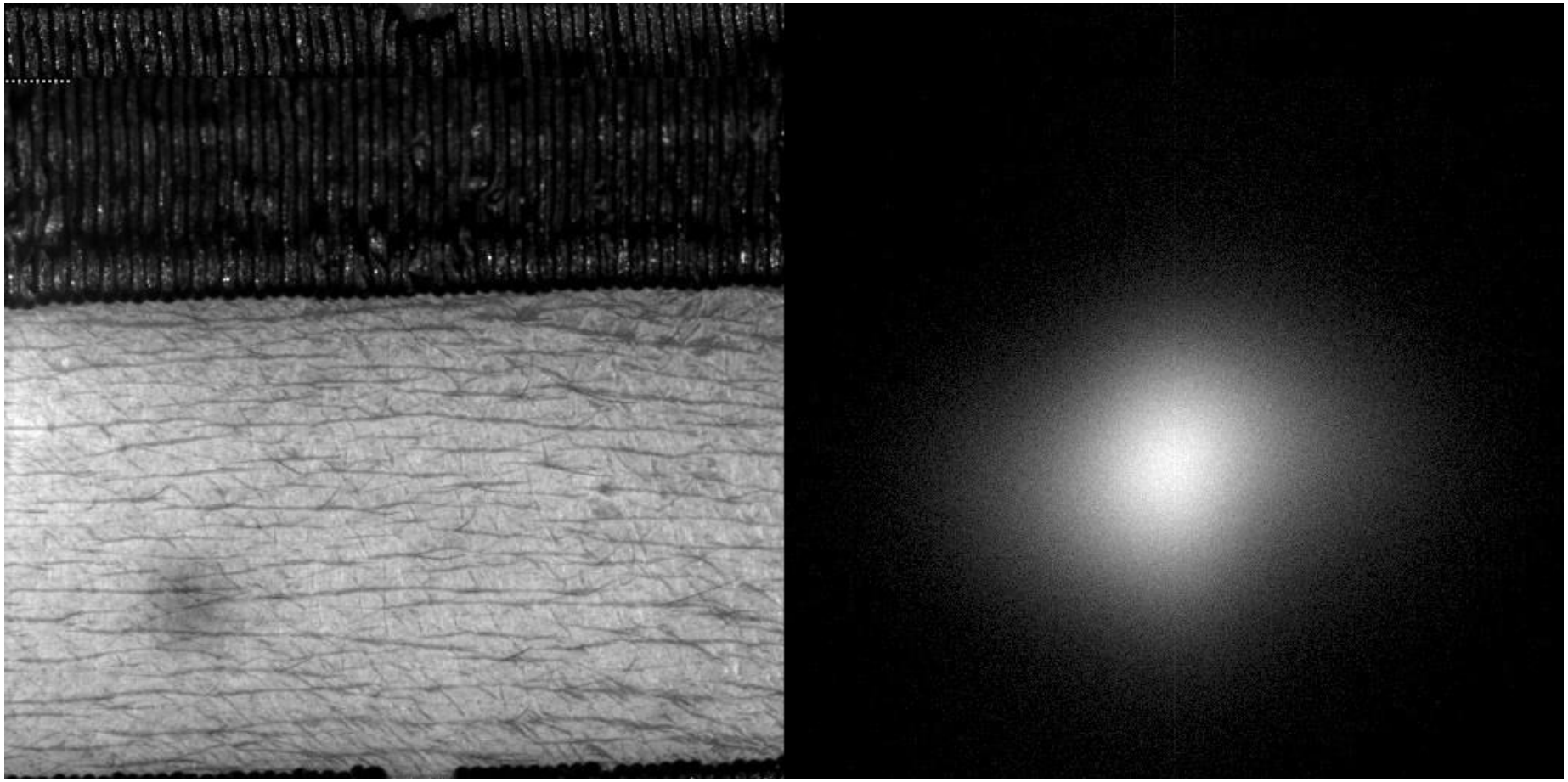


Image from “Skin Microstructure Deformation with Displacement Map Convolution”, Nagano et al., SIGGRAPH 2015

...to being stretched, its NDF...

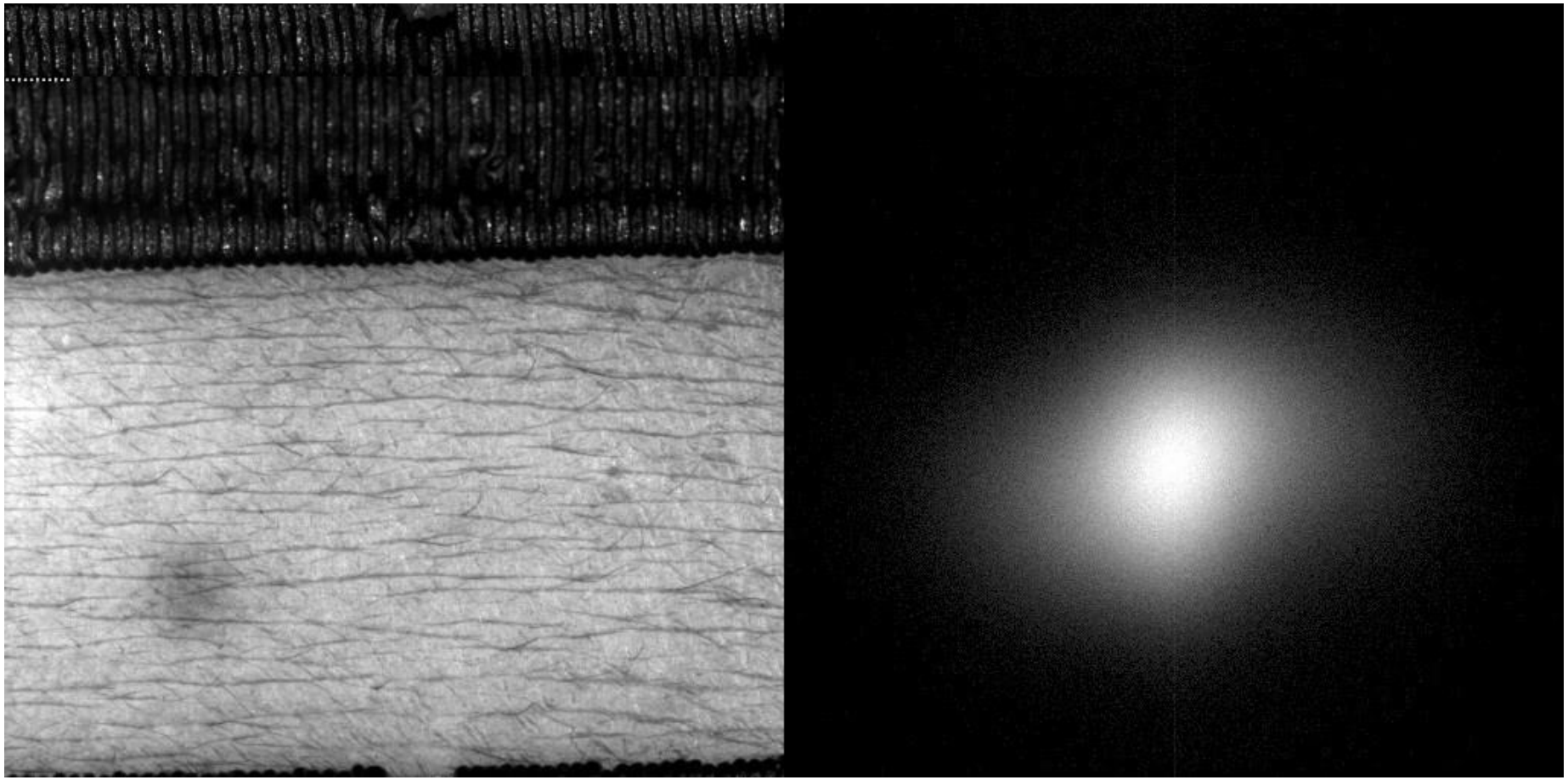


Image from “Skin Microstructure Deformation with Displacement Map Convolution”, Nagano et al., SIGGRAPH 2015

...changes...

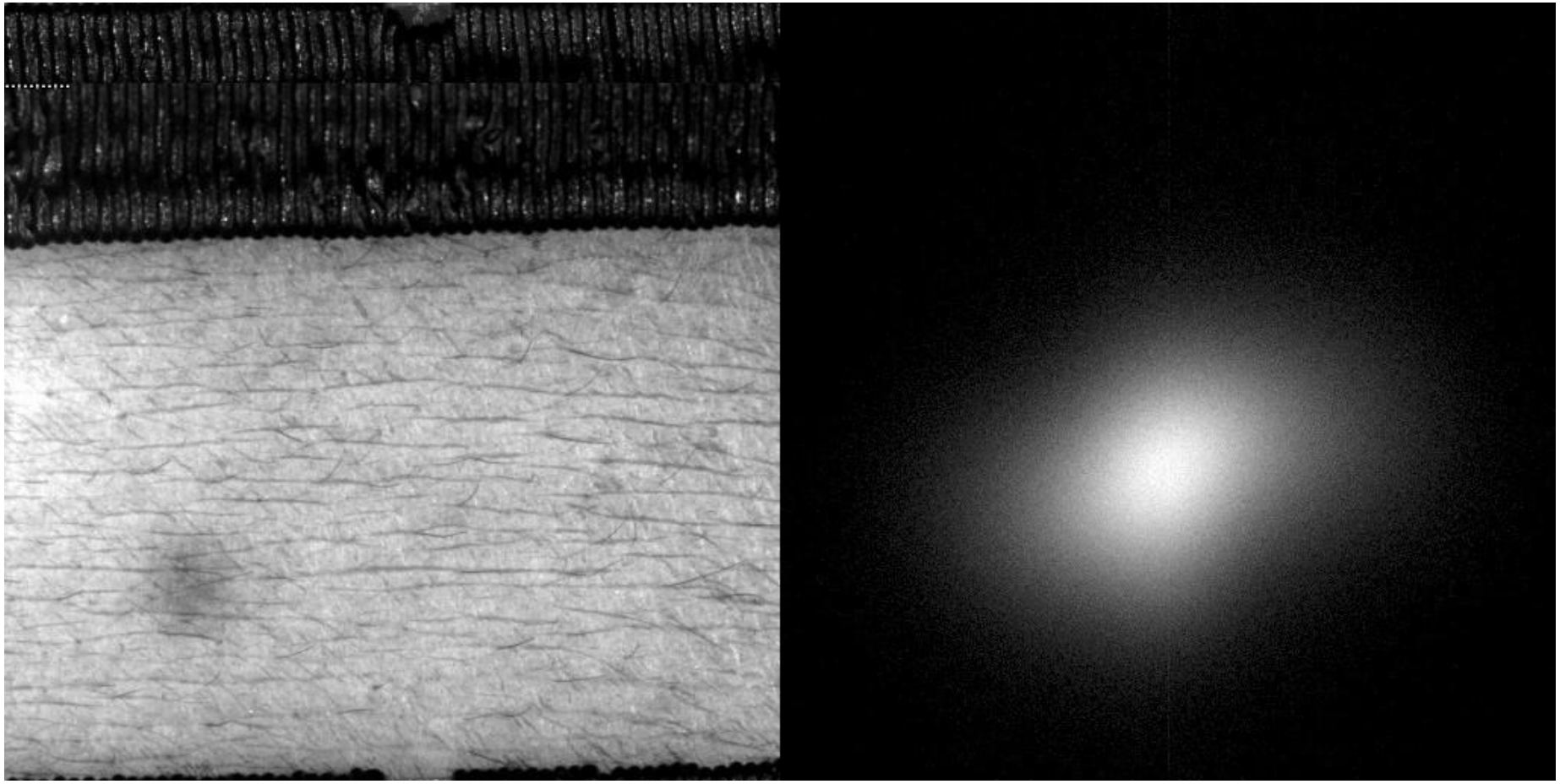


Image from “Skin Microstructure Deformation with Displacement Map Convolution”, Nagano et al., SIGGRAPH 2015

...accordingly.

<one more image>

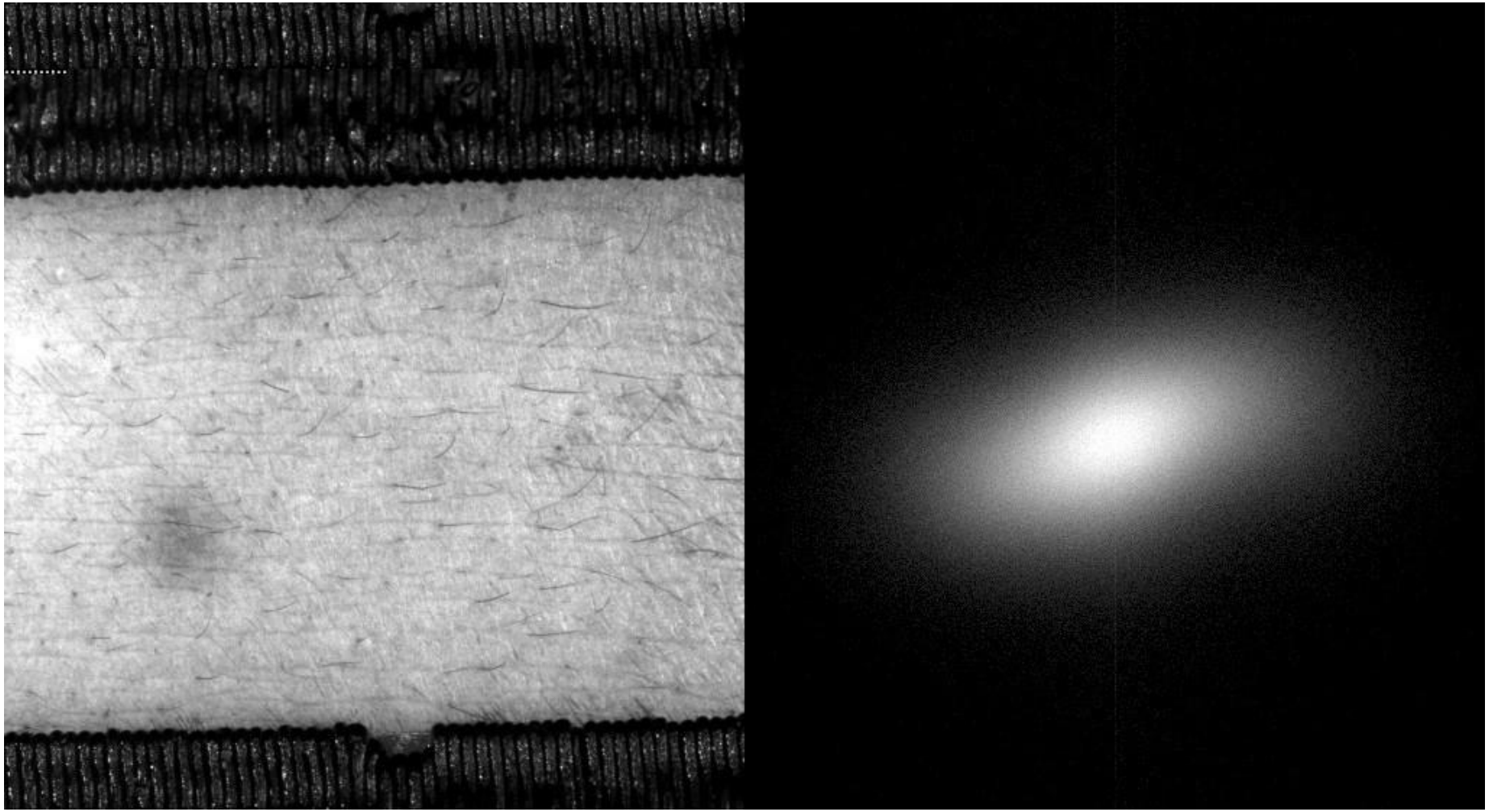
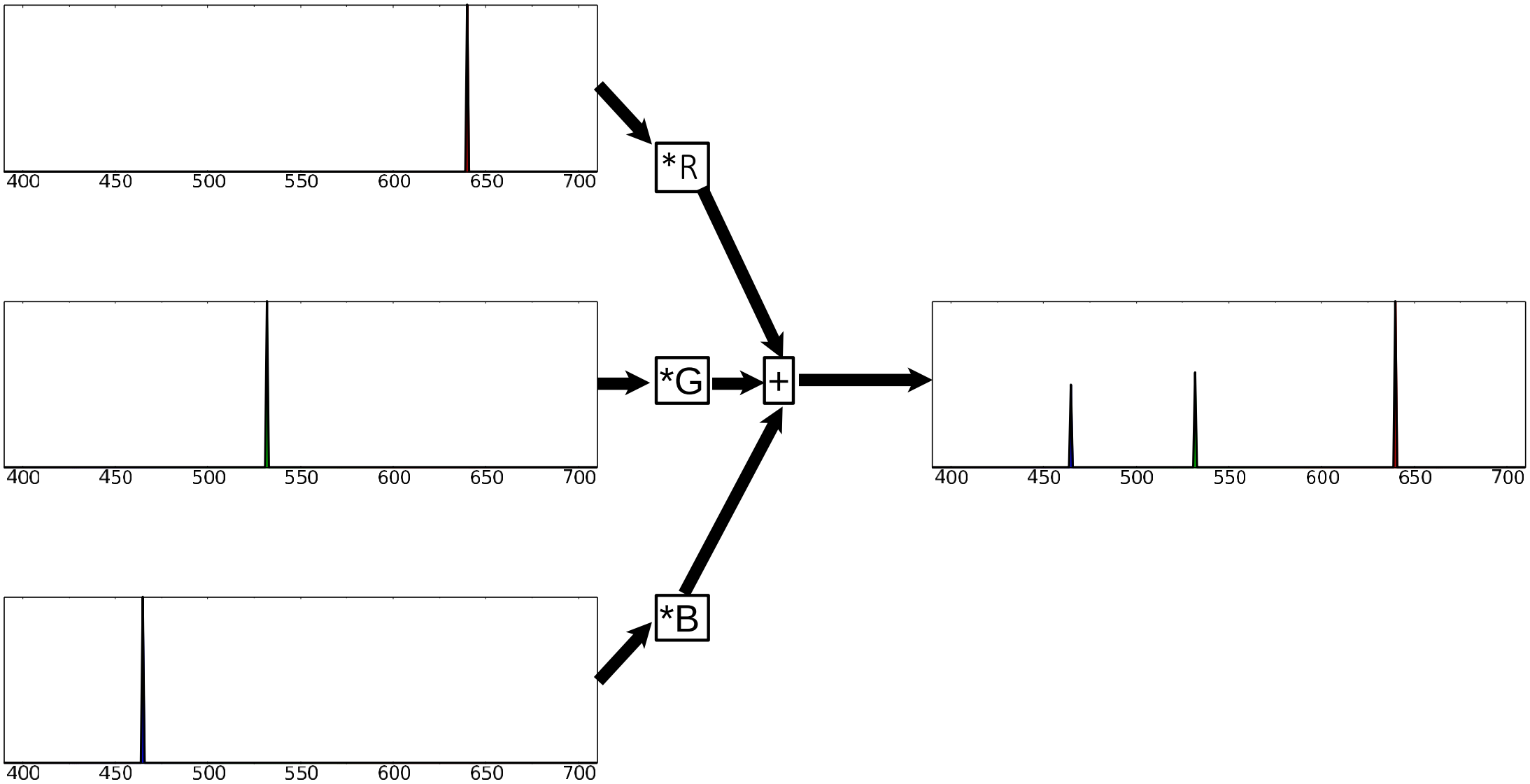
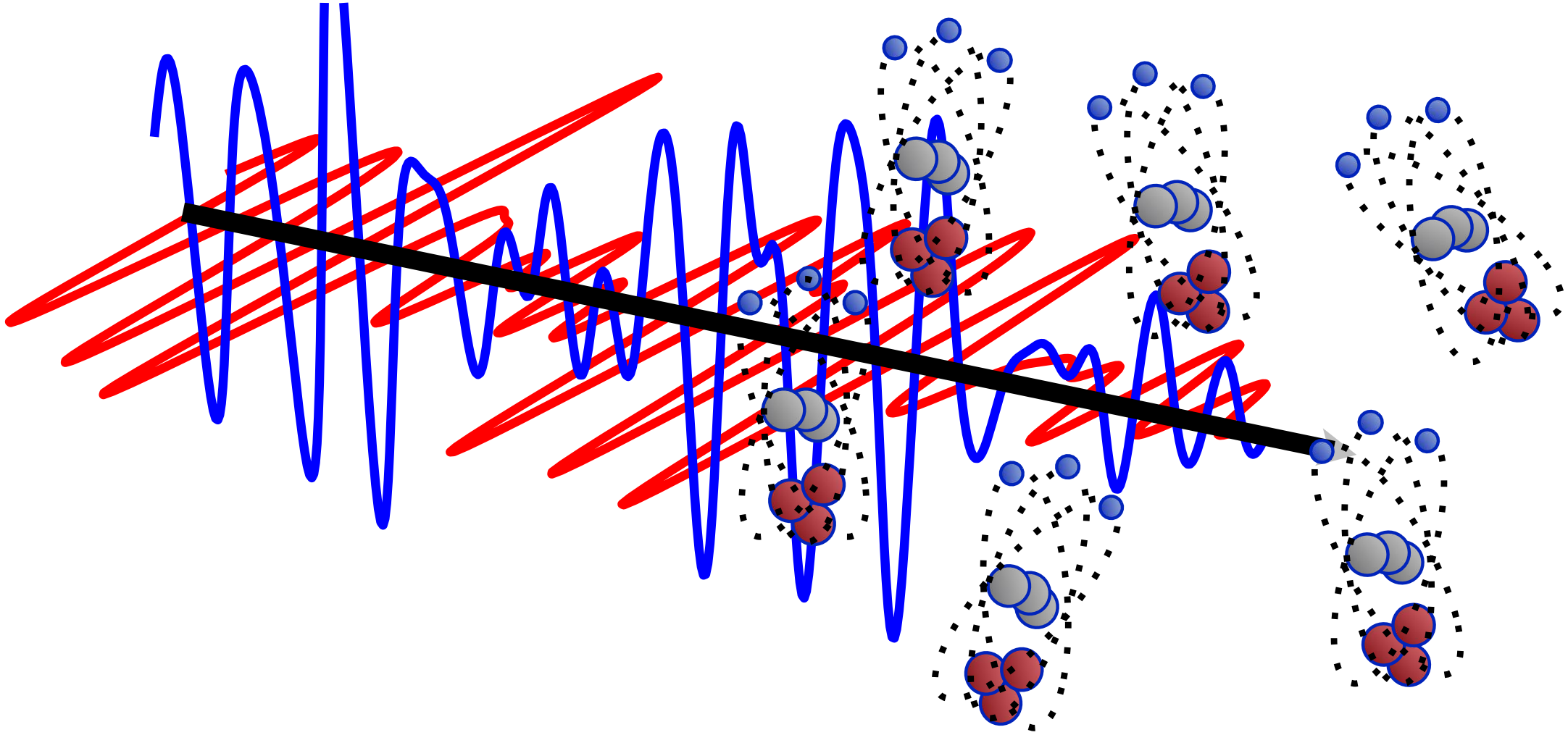


Image from “Skin Microstructure Deformation with Displacement Map Convolution”, Nagano et al., SIGGRAPH 2015

Although this paper was about skin, this type of behavior will occur with any flexible surface material .


400
450
500
550
600
650
700
*R
*G
+
*B



...stretching the molecule's positive and negative charges apart forming *dipoles*. This absorbs energy from the incoming wave.

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	

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

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

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

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

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