

Physical based rendering cheat sheet

Maksim Jaroslavcevas

April 26, 2024

1 Rendering equation

$$L_o(\mathbf{x}, \omega_o, \lambda, t) = L_e(\mathbf{x}, \omega_o, \lambda, t) + L_r(\mathbf{x}, \omega_o, \lambda, t)$$

where

- L_o is the total spectral radiance of wavelength λ directed outward along direction ω_o at time t , from a particular position x
- \mathbf{x} is the world position
- ω_o is direction of the outgoing light
- λ is a light wavelength
- t is time
- $L_e(\mathbf{x}, \omega_o, \lambda, t)$ is emitted spectral radiance
- $L_r(\mathbf{x}, \omega_o, \lambda, t)$ is reflected spectral radiance

Since we will use this equation in computer graphics scope we can, eliminate irrelevant variables, such as λ and t . Result equation:

$$L_o(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + L_r(\mathbf{x}, \omega_o) \quad (1)$$

As a result we get very neat function. This functions gives a color of a point at position \mathbf{x} from view vector perspective, as we see final color is a sum of a emitted light and reflected light.

First and second functions work with only single light channel, but actually if we calculate each of RGB channel separately and compose into a single three-dimensional vector, result will be the same as just replacing evrething with three-dimensional vector. Next we will consider light as a three-dimensional vector, where each dimension corresponds to red, green blue channels respectively.

Now lets look at reflection function

1.1 Emission function

Todo

1.2 Reflection function

$$L_r(x, \omega_o, \lambda, t) = \int_{\Omega} f_r(x, \omega_i, \omega_o, \lambda, t) L_i(x, \omega_i, \lambda, t) (\omega_i * N) d\omega_i$$

where

- \mathbf{N} is a surface normal
- ω_i light vector
- \dots all other variables are repeated as in General form equation
- $L_i(x, \omega_i, \lambda, t)$ incoming light

Lets simplify this function, eliminating λ, t variables

$$L_r(x, \omega_o) = \int_{\Omega} f_r(x, \omega_i, \omega_o) L_i(x, \omega_i) (\omega_i * N) d\omega_i$$

As we see reflected light is just an \int of all incoming light from an Ω sphere. Since in computer graphics we usually have a finite number of lights, we can replace \int with a Σ .

$$L_r(x, \omega_o) = \sum_{n=1}^N f_r(x, \omega_n, \omega_o) L_i(x, \omega_n) (\omega_n * N) \quad (2)$$

Now we get reflection is a sum of products between incoming lights, f_r functions and $(\omega_n * N)$. At this point our final equation should look something like this

$$L_o(\mathbf{x}, \omega_o) = L_e(x, \omega_o) + \sum_{n=1}^G f_r(x, \omega_n, \omega_o) L_i(x, \omega_n) (\omega_n * N)$$

- \mathbf{G} total number of lights

Now we need to clarify f_r function, but before lets replace some variables, since we will use them quite a lot further, and not all names are quite intuitive.

$$L_o(x, V) = L_e(x, V) + \sum_{n=1}^G f_r(x, L_n, V) L_i(x, L_n) (L_n * N) \quad (3)$$

where

- \mathbf{x} is a world position of a point

- \mathbf{V} view vector
- L_n - n^{th} light vector
- N surface normal
- G total number of lights
- L_i - incoming light function

Note that all vectors are normalized.

1.3 Bidirectional reflectance distribution function (BRDF)

BRDF stands by bidirectional reflectance distribution function, this function defines how light from a source is reflected off an opaque surface. Usually BRDF takes an incoming light direction and outgoing direction, and returns the ration of reflected radiance exiting along to the irradiance incident of the surface from view direction.

Lets take a look at first Fred Nicodemus definition of the BRDF:

$$f_r(\omega_i, \omega_r) = \frac{dL_r(\omega_r)}{dE_i(\omega_i)} = \frac{dL_r(\omega_r)}{dL_i(\omega_i)\cos(\theta_i)d\omega_i} \quad (4)$$

where

- \mathbf{L} is radiance
- \mathbf{E} is irradiance
- θ_i is the angle between ω_i and the surface normal
- i index indicates incident light
- r index indicates reflected light

While the BRDF itself is a general concept that can model any type of reflection, but it is not very suitable for our needs, since in computer graphics we usually interested in diffuse and specular reflections. There are many other BRDF mased light models, such as

- **Lambertian model** used for diffuse reflection
- **Cook-Torrance model (Torrance–Sparrow)** specular-microfacet model
- **Phong** This model adds a specular component to the Lambertian model. Computationally inexpensive but less physically accurate
- **Oren–Nayar** similar to Lambertians model, more physically accurate, but more computationally expensive
- ... and many others

Since different lighting models are used for different reflections, we can use combination of different models, for better result. For example most common use is a **Lambertian model** for diffuse reflections and **Cook-Torrance model** for specular reflections. But we need to consider energy preservation, outgoing energy should not exceed incoming energy.

But in what ratio we need to sum our diffuse and specular reflections? For this purpose we can use **Fresnel** effect. In computer graphics we usually use **Schlick's approximation** for approximating the contribution of the **Fresnel** effect in the specular reflection.

Original **Schlick's approximation** function

$$f_{Schlick}(\theta) = f_0 + (1 - f_0)(1 - \cos \theta)^5$$

where

- f_0 is the reflection coefficient for light incoming parallel to the normal
- θ is the angle between the direction from which the incident light is coming and the normal

Since in all our previous calculations we had no θ angle, lets replace it with vector dot product.

$$f_{Schlick} = f_0 + (1 - f_0)(1 - (V \cdot H))^5$$

where

- V view vector
- H half-way vector

Note that V and H vectors are normalized, and f_0 is a constant specific for a specific material.

Since **Schlick's approximation** gives a ratio of specular reflection component lets denote it as

$$k_s = f_{Schlick}$$

$$k_d = 1 - k_s$$

Note that $k_s + k_d = 1$ (k_s - specular ration, k_d - diffuse ratio).

At this point our *BRDF* (Combination of Lambertian model and Cook-Torrance model for diffuse and specular reflections respectively) function, should look something like this

$$f_{brdf} = k_d f_{lambert} + k_s f_{cook-torrance}$$

2 References

Todo