LTC Fresnel Approximation

Stephen Hill

To explain our Fresnel approximation, I will start first of all with a simpler case that we will then build upon. Let's imagine that we have a microfacet BRDF, $\rho(\boldsymbol{\omega_v}, \boldsymbol{\omega_l})$, with shadowing-masking but without Fresnel. Due to the presence of shadowing, the integral of the cosine-weighted BRDF over the sphere can be less than 1 (i.e. some 'energy' has been lost due to shadowing):

$$\int_\Omega \rho(\boldsymbol{\omega_v}, \boldsymbol{\omega_l}) \cos\theta_l \,\mathrm{d}\boldsymbol{\omega_l} \leq 1. \tag{1}$$

In contrast and by design, LTCs always integrate to 1. Therefore, in order to achieve an accurate fit of the BRDF using LTCs, we store the norm[1] n_D (the *magnitude* of the BRDF) as a scale factor, in addition to M^{-1} (which captures the *shape* of the BRDF):

$$n_D = \int_\Omega \rho(\boldsymbol{\omega_v}, \boldsymbol{\omega_l}) \cos\theta_l \,\mathrm{d}\boldsymbol{\omega_l}. \tag{2}$$

As with M^{-1}, we store n_D in a 2D texture, parameterized by incident direction and roughness.

Now let's turn our attention to Fresnel. Rather than attempting to fit LTCs to a BRDF that directly incorporates Fresnel, we instead choose to treat the Fresnel term separately, as an additional influence on the magnitude of the BRDF. Thus the norm becomes

$$\int_\Omega F(\boldsymbol{\omega_v}, \boldsymbol{\omega_l})\, \rho(\boldsymbol{\omega_v}, \boldsymbol{\omega_l}) \cos\theta_l \,\mathrm{d}\boldsymbol{\omega_l}. \tag{3}$$

Using Schlick's approximation [Sch94], this expands to

$$\int_\Omega [R_0 + (1 - R_0)(1 - \langle \boldsymbol{\omega_v}, \boldsymbol{\omega_h} \rangle)^5]\, \rho(\boldsymbol{\omega_v}, \boldsymbol{\omega_l}) \cos\theta_l \,\mathrm{d}\boldsymbol{\omega_l}, \tag{4}$$

which we can rearrange to the following:

$$\begin{aligned} &= R_0 \int_\Omega \rho(\boldsymbol{\omega_v}, \boldsymbol{\omega_l}) \cos\theta_l \,\mathrm{d}\boldsymbol{\omega_l} + (1 - R_0) \int_\Omega (1 - \langle \boldsymbol{\omega_v}, \boldsymbol{\omega_h} \rangle)^5 \rho(\boldsymbol{\omega_v}, \boldsymbol{\omega_l}) \cos\theta_l \,\mathrm{d}\boldsymbol{\omega_l} \\ &= R_0\, n_D + (1 - R_0)\, f_D, \quad \text{where } f_D = \int_\Omega (1 - \langle \boldsymbol{\omega_v}, \boldsymbol{\omega_h} \rangle)^5 \rho(\boldsymbol{\omega_v}, \boldsymbol{\omega_l}) \cos\theta_l \,\mathrm{d}\boldsymbol{\omega_l}. \end{aligned} \tag{5}$$

Now, in addition to n_D that we had before, we store a second term f_D[2].

While this may seem like a coarse approximation, we found it to work very well in practice from a visual standpoint, and it's an approach that has already proven to be effective in the context of environmental illumination [Kar13; Laz13]. Furthermore, this solution avoids complicating the fitting process or increasing the dimensionality of the tabulated data.

[1] This is mentioned very briefly in our paper at the start of the Representation and Storage section: "Furthermore, we use an additional parameter for the norm . . . ".

[2] We could alternatively store $n'_D = n_D - f_D$, and calculate $R_0\, n'_D + f_D$ at runtime, which saves an ALU instruction.

References

[Kar13] B. Karis. “Real Shading in Unreal Engine 4”. In: *Physically Based Shading in Theory and Practice, ACM SIGGRAPH Courses*. 2013. URL: http://selfshadow.com/publications/s2013-shading-course/.

[Laz13] D. Lazarov. “Getting More Physical in Call of Duty: Black Ops II”. In: *Physically Based Shading in Theory and Practice, ACM SIGGRAPH Courses*. 2013. URL: http://selfshadow.com/publications/s2013-shading-course/.

[Sch94] C. Schlick. “An Inexpensive BRDF Model for Physically-based Rendering”. In: *Computer Graphics Forum* 13.3 (Aug. 1994), pp. 233–246.