A Multiple-Scattering Microfacet Model for Real-Time Image-based Lighting

Carmelo J. Fdez-Agüera

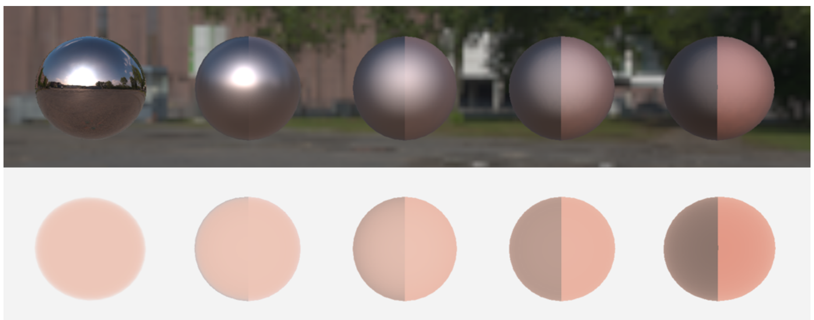

Figure 1. Copper material (right) with and (left) without multiple scattering.

Abstract

This article introduces an extension to real-time image-based lighting models that incorporates multiple scattering; it is suitable for real-time applications due to its low run-time cost. The main insight is that the precomputed integrals used for computing the single scattering term already contain all the information needed to simulate the remaining light bounces. The result is a technique that adds little overhead compared to its single scattering counterpart, but accomplishes perfect energy conservation and preservation. Even though the derivation is presented for a GGX bidirectional reflectance distribution function (BRDF), it can be trivially applied to other models as long as the split-sum approximation can be used to precompute the BRDF integral, or to find an analytical fit to it.

1. Introduction

The technique presented in this article builds on common image-based lighting models for real-time applications. In Section 2, we introduce the basis of the technique by enforcing energy conservation on a perfect reflective surface. In Section 3, we generalize the result to arbitrary conductors, taking Fresnel reflectance into account. We do so by reusing the precomputed data that was already used for the single scattering

 ISSN 2331-7418

term. Finally, in Section 4, we show how to extend the result to dielectric materials, and how the result both preserves energy at high roughness and prevents excess energy at low roughness, yielding complete energy conservation and preservation. As shown in Figure 1, regaining this missing energy can have a strong visual impact on certain surfaces.

1.1. Related Work

Karis [2013] introduced the split-sum approximation as a method to precompute the single scattering BRDF integral for image-based lighting. His method uses prefiltered radiance with a precomputed integral for the BRDF response. He stores this integral as a two-component look-up table and uses it as a scale and bias for the Fresnel term.

Kulla and Conty [2017] use the furnace test to compensate for the missing energy in the single scattering models and add an extra lobe to the BRDF based on that missing energy. The reasoning is analogous to the one presented in this article, but they compute several 2D and 3D look-up tables, and their results are intended for path tracing and not for real-time image-based lighting.

Hill [2018a] refines the technique by Kulla and Conty and explores the expansion of the Fresnel term in multiple scattering events as a geometric series. Further [2018b], he uses a path-traced simulation to compute the energy emitted at every scattering event, which makes it more accurate than the model presented here, but leads him to precompute a look-up table for each event in the multiple scattering. His approach only explores analytical lights and doesn't address the generalizations for image-based lighting.

1.2. Furnace Test

For energy to be conserved on a reflective surface, it must be true that for any incident light direction ω_i, the integral of reflected light along the viewing hemisphere equals the amount of incident light:

$$\int_\Omega \mathrm{BRDF}(F_0, r, \omega_i, \omega_o) \cos\theta_o d\omega_o = 1,$$

and given a reciprocal BRDF, the following must also hold:

$$E(\omega_o, r) = \int_\Omega \mathrm{BRDF}(F_0, r, \omega_o, \omega_i) \cos\theta_i d\omega_i = 1. \tag{1}$$

Equation (1) defines the furnace test, which we'll use several times in this article. The function $E(\omega_o, r)$ is also known as the directional albedo of a BRDF and is equivalent to integrating the BRDF against a uniform white environment.

Figure 2. Precomputed GGX first-bounce energy of a perfect reflector with horizontal axis $\cos(\theta_o)$ and vertical axis roughness.

2. Energy Preservation from Perfect Reflectors

The simplest surface for which we can derive a multiple scattering model is the perfect reflector, which is a surface with reflectance $F = 1$. This eliminates the Fresnel term in the BRDF and simplifies integration. Integrating the single scattering BRDF (f_{ss}) against a solid white environment (Equation (2)), we obtain the energy that escapes the surface after the first bounce of light in a particular direction, the single scattering directional albedo E_{ss}. Figure 2 shows the result of this integral using the same BRDF as Epic's [2013]. In fact, it is equivalent to adding the scale and bias terms they propose:

$$E_{ss}(\omega_o, r) = \int_\Omega \frac{D(\omega_i, \omega_o, r)G(\omega_i, \omega_o, r)}{4 \cos\theta_i \cos\theta_o} \cos\theta_i d\omega_i. \tag{2}$$

If energy is preserved, Figure 2 would be perfectly white, but since it only accounts for the first bounce of light, some areas are darker. By applying energy conservation (Equations (3) and (4)), we can obtain the energy corresponding to the later bounces E_{ms}:

$$1 = E_{ss}(\omega_o, r) + E_{ms}(\omega_o, r), \tag{3}$$

$$E_{ms}(\omega_o, r) = \int_\Omega f_{ms} \cos\theta_i d\omega_i = 1 - E_{ss}(\omega_o, r). \tag{4}$$

We add an extra BRDF lobe to account for that missing energy:

$$L_o = \int_\Omega (f_{ss} + f_{ms}) \cos\theta_i L_i d\omega_i = \int_\Omega f_{ss} \cos\theta_i L_i d\omega_i + \int_\Omega f_{ms} \cos\theta_i L_i d\omega_i. \tag{5}$$

On the right side of Equation (5), the first integral is the traditional single scattering term, which we can compute using Kulla's split sum, and the second integral is the multiple scattering lobe. Similar to what we do for the single scattering term, we can approximate this integral using a split sum (Equation (6)). The first term of the split is exactly E_{ms}, and if we assume that energy participating in secondary scattering events is uniformly diffused, the second term can be approximated with the cosine-weighted irradiance, usually represented either as a low-resolution texture, or baked into spherical harmonics coefficients:

$$\int_{\Omega} f_{ms} L_i \cos\theta_i d\omega_i \approx \int_{\Omega} f_{ms} \cos\theta_i d\omega_i \int_{\Omega} \frac{L_i}{\pi} \cos\theta_i d\omega_i. \tag{6}$$

Notice that this approximation is actually exact for uniform environments, which explains the perfect furnace test in Figure 3. It's still a good approximation for lights that cover a big portion of the hemisphere, where energy is more or less evenly distributed, like sky domes, but not for narrow lights where most energy comes from a single direction (e.g., analytical point or directional lights).

Equation (7) is the approximation for a perfect furnace test (image-based lighting) and Equation (8) provides the approximation for the split-sum calculation for perfect reflectors.

$$\int_{\Omega} f_{ms} \cos\theta_i d\omega_i \int_{\Omega} \frac{L_i}{\pi} \cos\theta_i d\omega_i = (1 - E_{ss}) \int_{\Omega} \frac{L_i}{\pi} \cos\theta_i d\omega_i, \tag{7}$$

$$L_o = E_{ss}\text{radiance} + (1 - E_{ss})\text{irradiance}. \tag{8}$$

Figure 3 shows a comparison between single scattering image-based lighting using the split-sum approximation and the model with multiple scattering for perfect reflectors. The increased albedo is quite noticeable for high roughness values.

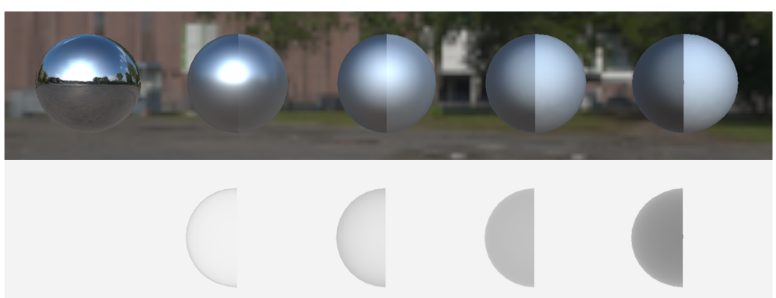

Figure 3. Spheres with perfect reflector material (metallic=1, $F_0 = 1$) with roughness from 0 to 1; (above) an HDR environment and (below) a uniform environment (furnace test). The left half of the spheres uses the split-sum method, while the right half uses the multiple scattering approximation.

3. Generic Metals, Fresnel Term

For generic metals we need to take into account the Fresnel term. In order to reuse the split-sum approximation for perfect reflectors (Equation (8)), we'd like our equation for uniform lighting to be in the form:

$$E = F_{ss}E_{ss} + F_{ms}E_{ms}. \tag{9}$$

After the first bounce, light will be randomly scattered in all directions. We can model the distribution of light in these secondary bounces as having uniform energy in all directions, so it can be represented by an attenuated form of the cosine-weighted irradiance. Under that hypothesis, every secondary scattering event behaves the same way as the first one, except it takes this attenuated irradiance as its light source. This approximation makes sense because we already assume that incoming light is more or less evenly distributed when we use the split sum, and it will only get more diffused as it scatters by reflecting on randomly oriented microfacets.

In much the same way as Hill [2018a], we can develop $F_{ms}E_{ms}$ into a geometric series for the multiple bounces of light. On each light bounce, a fraction E_{avg} escapes the surface, and a fraction of $1 - F_{\text{avg}}$ gets absorbed by the conductor, so only $(1 - E_{\text{avg}})F_{\text{avg}}$ is left for the next bounce, where F_{avg} is the cosine-weighted average of the Fresnel term. Notice we can't approximate F_{avg} with F_{ss} because while the latter depends on the viewing angle, the first represents attenuation for light coming from all directions, and so isn't directional. However, F_{avg} has an analytical solution when using the Schlick approximation (Equation (10) and (11)).

$$F_{\text{avg}} = 2\pi \int_0^{\pi/2} (F_0 + (1 - F_0)(1 - \cos(\theta))^5) \frac{\cos\theta}{\pi} \sin\theta d\theta, \tag{10}$$

$$F_{\text{avg}} = F_0 + \frac{1}{21}(1 - F_0). \tag{11}$$

When $F_0 = 1$, Equation (12)

$$E = F_{ss}E_{ss} + \sum_{k=1}^{\infty} F_{ss}E_{ss}(1 - E_{\text{avg}})^k F_{\text{avg}}^k, \tag{12}$$

and Equation (3) must be equivalent, so,

$$1 = E_{ss} + \sum_{k=1}^{\infty} E_{ss}(1 - E_{avg})^k = \sum_{k=0}^{\infty} E_{ss}(1 - E_{avg})^k,$$

,

$$1 = \sum_{k=0}^{\infty} E_{ss}(1 - E_{\text{avg}})^k = \frac{E_{ss}}{1 - (1 - E_{\text{avg}})},$$

$$E_{\text{avg}} = E_{ss}.$$

This implies that E_{avg} is dependent on the viewing direction, as is E_{ss}, meaning it takes more bounces for light to escape in some directions than others. By substitution of E_{avg} back into Equation (12), we have

$$E = F_{ss}E_{ss} + F_{ss}E_{ss} \frac{(1 - E_{ss})F_{\text{avg}}}{1 - (1 - E_{ss})F_{\text{avg}}}. \tag{13}$$

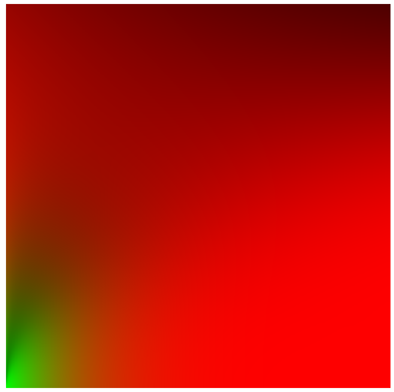

Figure 4. Precomputed GGX first bounce energy, separated into a term that depends on F_0 (red) and an independent term (green).

And combining Equations (9) and (13) we get

$$F_{ms}E_{ms} = F_{ms}(1 - E_{ss}) = F_{ss}E_{ss}\frac{(1 - E_{ss})F_{\text{avg}}}{1 - (1 - E_{ss})F_{\text{avg}}},$$

$$F_{ms} = \frac{F_{ss}F_{\text{avg}}E_{ss}}{1 - F_{\text{avg}}(1 - E_{ss})}. \tag{14}$$

The problem with the previous equation is that we still don't have an expression for F_{ss}, but we can get one by integrating the single scattering BRDF. In Equation (15), we do that using Schlick's approximation to Fresnel:

$$\begin{aligned} F_{ss}E_{ss} = F_0 \int_\Omega \frac{f_{ss}(\omega_i, \omega_o)}{F(\omega_o, \omega_h)}\left(1 - (1 - \omega_o \cdot \omega_h)^5\right) \cos\theta_i d\omega_i \\ + \int_\Omega \frac{f_{ss}(\omega_i, \omega_o)}{F(\omega_o, \omega_h)}(1 - \omega_o \cdot \omega_h)^5 \cos\theta_i d\omega_i. \end{aligned} \tag{15}$$

This is the integral that Epic Games computes into a look-up table for single scattering image-based lighting [2013], consisting of a scale (f_a) and bias (f_b) for the base reflectance F_0 (see Figure 4), and we'll refer to it as

$$F_{ss}E_{ss} = F_0 f_a + f_b. \tag{16}$$

By substitution of Equation (16) into Equation (14), we get

$$F_{ms} = \frac{(F_0 f_a + f_b)F_{\text{avg}}}{1 - F_{\text{avg}}(1 - E_{ss})}. \tag{17}$$

So we can express the complete equation for metals as follows:

$$L_o = (F_0 f_a + f_b)\text{radiance} + \frac{(F_0 f_a + f_b)F_{\text{avg}}}{1 - F_{\text{avg}}(1 - E_{ss})}E_{ms}\text{irradiance}. \tag{18}$$

```
// Common code for single and multiple scattering

// roughness-dependent Fresnel
vec3 Fr = max(vec3(1.0 - roughness), F0) - F0;

vec3 kS = F0 + Fr * pow(1.0-ndv, 5.0);

vec2 f_ab = textureLod(uEnvBRDF, vec2(ndv, roughness), 0).xy;
vec3 FssEss = kS * f_ab.x + f_ab.y;

float lodLevel = roughness * numEnvLevels;
vec3 reflDir = reflect(-eye, normal);

// Prefiltered radiance
vec3 radiance = getRadiance(reflDir, lodLevel);
// Cosine-weighted irradiance
vec3 irradiance = getIrradiance(normal);

// Multiple scattering
float Ess = f_ab.x + f_ab.y;
float Ems = 1-Ess;
vec3 Favg = F0 + (1-F0)/21;
vec3 Fms = FssEss*Favg/(1-(1-Ess)*Favg);

// Composition
return FssEss * radiance + Fms*Ems * irradiance;
```

Listing 1. GLSL code for generic conductors.

Listing 1 shows an example implementation of image-based lighting for metals. Notice the overhead introduced by multiple scattering code is very low. Results can be compared in Figure 5.

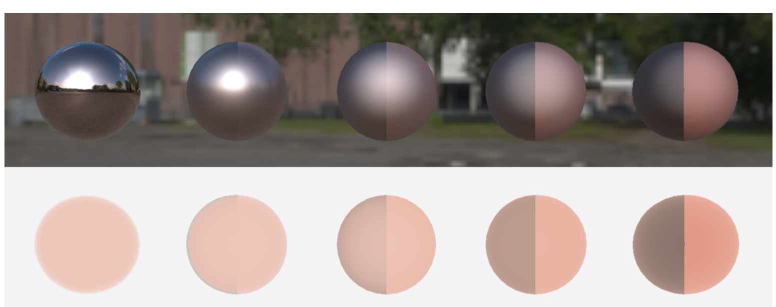

Figure 5. Spheres with copper material with roughness from 0 to 1; (above) in an HDR environment and (below) in a uniform environment (furnace test). The left half of the spheres uses the split-sum method for image-based lighting, while the right half uses the multiple scattering approximation.

4. Dielectrics

With dielectrics there is a further term, corresponding to how the energy not reflected by Fresnel, diffuses under the surface and radiates back. Let K_d be the diffuse albedo of the surface, which models energy absorption under the surface, then

$$E = F_{ss}E_{ss} + F_{ms}E_{ms} + K_d E_d.$$

E_d is often computed as $1 - F_0$. However, for grazing angles of incoming light, the Fresnel term will be higher (and less saturated), so less of that light should end up in the diffuse term. In order to get a better approximation of f_d, we can observe that a perfectly white dielectric ($K_d = 1$) doesn't absorb light, thus must radiate back as much energy as it receives:

$$1 = F_{ss}E_{ss} + F_{ms}E_{ms} + E_d,$$

$$E_d = 1 - (F_{ss}E_{ss} + F_{ms}E_{ms}).$$

This fixes energy conservation for dielectrics, while still taking into account the multiple scattering specular lobe. It doesn't explicitly model the effects of light scattering back and forth between the diffuse and specular interfaces, but since specular reflectance is usually quite low and unsaturated for dielectrics, radiated energy will eventually escape the surface unaltered, so the approximation holds.

Figure 6 shows a furnace test on a grey background of the single-scattering (left) and multiple scattering (right) models for varying roughness. For high roughness, introducing multiple scattering recovers some missing energy, and for smooth surfaces, the excess energy towards the edges is eliminated. This is further illustrated in Figure 7, where both models are compared to a reference photograph. Notice how the darker rim in the multiple scattering sphere (center), matches more closely the distribution of light in the reference. While the effect is certainly more subtle than with metals, removing the excess energy produces a more natural look. Listing 2 shows the code required to compute the proper K_d term for dielectrics. Again, the overhead over traditional diffuse IBL is quite small.

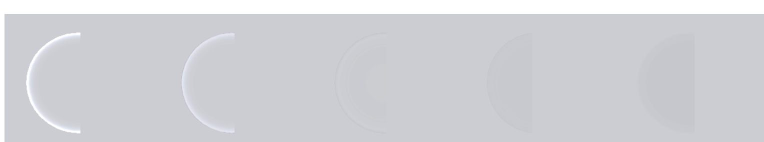

Figure 6. Spheres with white dielectric material with roughness from 0 to 1 on a grey furnace environment. Left half of the spheres uses SS ibl with $E_d = 1 - F_0$, right half uses the MS approximation.

Figure 7. Close up of a smooth plastic ball. Single scattering model (left), multiple scattering model (center), and reference photograph (right).

```
// GLSL code for dielectrics
// Common code for single and multiple scattering

// Roughness dependent fresnel
vec3 Fr = max(vec3(1.0 - roughness), F0) - F0;
vec3 kS = F0 + Fr * pow(1.0-ndv, 5.0);

vec2 f_ab = textureLod(uEnvBRDF, vec2(ndv, roughness), 0).xy;
vec3 FssEss = kS * f_ab.x + f_ab.y;

float lodLevel = roughness * numEnvLevels;
vec3 reflDir = reflect(-eye, normal);

// Prefiltered radiance
 vec3 radiance = getRadiance(reflDir, lodLevel);
// Cosine-weighted irradiance
vec3 irradiance = getIrradiance(normal);

// Multiple scattering
float Ess = f_ab.x + f_ab.y;
float Ems = 1-Ess;
vec3 Favg = F0 + (1-F0)/21;
vec3 Fms = FssEss*Favg/(1-(1-Ess)*Favg);

// Dielectrics
vec3 Edss = 1 - (FssEss + Fms * Ems);
vec3 kD = albedo * Edss;

// Composition
return FssEss * radiance + (Fms*Ems+kD) * irradiance;
```

Listing 2. GLSLcode for generic dielectrics.

5. Conclusions

Introducing the effect of multiple scattering back into image-based lighting can have a big impact on the appearance of materials. Aditionally, ensuring energy conservation and preservation makes materials more robust to changes in environment illumination, which is one of the benefits of physically based rendering in general. This article has shown that it is possible to achieve this effect with very little overhead, and that the visual impact can definitely be worth the cost.

References

HILL, S. 2018. A multi-faceted exploration (part 2). *Self Shadow blog* (June). URL: https://blog.selfshadow.com/2018/06/04/multi-faceted-part-2/. 46, 49

HILL, S. 2018. A multi-faceted exploration (part 3). *Self Shadow blog* (Aug.). URL: https://blog.selfshadow.com/2018/08/05/multi-faceted-part-3/. 46

KARIS, B. 2013. Real shading in unreal engine 4. In *Physically Based Shading in Theory and Practice, ACM SIGGRAPH 2013 Courses*, ACM, New York, NY. URL: https://cdn2.unrealengine.com/Resources/files/2013SiggraphPresentationsNotes-26915738.pdf. 46, 47, 50

KULLA, C., AND CONTY, A. 2017. Revisiting physically based shading at Imageworks. In *Physically Based Shading in Theory and Practice, ACM SIGGRAPH 2017 Courses*, ACM, New York, NY, 7:1–7:8. URL: https://blog.selfshadow.com/publications/s2017-shading-course/imageworks/s2017_pbs_imageworks_slides_v2.pdf. 46

Author Contact Information

Carmelo J. Fdez-Agüera
6 Suffolk St., 3rd Floor
D02 NY39 Dublin, Ireland
cjtech90@gmail.com

Fdez-Agüera, A Multiple-scattering Microfacet Model for Real-time Image-based Lighting, *Journal of Computer Graphics Techniques (JCGT)*, vol. 8, no. 1, 45–55, 2019
http://jcgt.org/published/0008/01/03/

Received:	2018-10-01	Corresponding Editor:	Christophe Hery
Recommended:	2018-11-12	Editor-in-Chief:	Marc Olano
Published:	2019-01-22		
Revised:	2019-02-01		