Bringing an Accurate Fresnel to Real-Time Rendering: a Preintegrable Decomposition
Laurent Belcour, Mégane Bati, Pascal Barla

To cite this version:
Laurent Belcour, Mégane Bati, Pascal Barla. Bringing an Accurate Fresnel to Real-Time Rendering: a Preintegrable Decomposition. SIGGRAPH’20 Courses, Aug 2020, Washington, D.C., United States. 10.1145/3388767.3407325 . hal-02883680v2

HAL Id: hal-02883680
https://inria.hal.science/hal-02883680v2
Submitted on 17 Aug 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publications ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
ABSTRACT
We introduce a new approximate Fresnel reflectance model that enables the accurate reproduction of ground-truth reflectance in real-time rendering engines. Our method is based on an empirical decomposition of the space of possible Fresnel curves. It is compatible with the preintegration of image-based lighting and area lights used in real-time engines. Our work permits to use a reflectance parametrization [Gulbrandsen 2014] that was previously restricted to offline rendering.

CCS CONCEPTS
- Computing methodologies → Reflectance modeling.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).


1 INTRODUCTION
VFX productions are increasingly relying on video-game engines during production instead of offline engines. To ensure coherency between the real-time and offline pipeline, coherency between their respective shading models is mandatory. Real-time and offline pipelines have converged to use microfacet reflectance models, yet the treatment of the Fresnel reflectance that describes the interaction of light with a planar surface is not the same and leads to noticeable differences. Offline engines can afford to use the accurate Fresnel equations and currently rely on the Gulbransen parametrization [Gulbransen 2014] (see Arnold’s Uber Shader for example [Georgiev et al. 2019]). Real-time engines on the other hand cannot afford the cost of the analytical Fresnel equation and rely on approximate models such as the one of Schlick [1994], the current state-of-the-art model for video-games engines.

In this work, we introduce an alternative to Schlick’s model [1994] that more accurately reproduces the ground truth Fresnel reflectances for a wide variety of surface parameters. Our solution decomposes Fresnel reflectances using four basis elements that are compatible with state-of-the-art pre-integration methods and incur a small computational overhead.

2 FRESNEL PRE-INTEGRATION
Real-Time Rendering Formulation. To achieve realism, modern real-time engines solve the direct illumination equation:

\[ L_o(\omega_o) = \int_{\Omega} \rho(\omega_i, \omega_o) L(\omega_i) \langle \omega_i \cdot n \rangle \, d\omega_i, \]

where \( L_o \) is the outgoing radiance defined for outgoing directions \( \omega_o \), \( L \) the incident radiance defined over incoming directions \( \omega_i \), and \( \rho(\omega_i, \omega_o) = \frac{L(\omega_i \cdot h) G(\omega_i, \omega_o) D_o(\omega_i, h)}{\omega_i \cdot h} \) is the microfacet-based BRDF model with \( h = \frac{\omega_o \times \omega_i}{||\omega_o \times \omega_i||} \) with a visible normal distribution \( D_o(h) \), shadowing term \( G(\omega_i, \omega_o) \), and Fresnel term \( F(\omega_i \cdot h) \). To achieve real-time frame rates, this equation is approximated by splitting the integral into two terms \( L_o(\omega_o) \approx \bar{F}(\omega_o) \times \bar{L}(\omega_o) \) [Lagarde and De Roussiers 2014]. \( \bar{F} \) includes the average material response with respect to the distribution of visible normals, and \( \bar{L} \) the average
light response:
\[
\bar{L}(\omega_o) = \int_\Omega \bar{L}(\omega_i) D_7(\omega_i, h) \, d\omega_i, \tag{1}
\]
\[
\bar{F}(\omega_o) = \int_\Omega \bar{F}(\omega_i \cdot h) \bar{G}_i(\omega_i, \omega_o) D_7(\omega_i, h) \, d\omega_i. \tag{2}
\]
Both \(\bar{L}\) and \(\bar{F}\) are tabulated with respect to their parameters: \(\bar{L}\) depends on the roughness of the \(vndf\) and the mean reflected direction; \(\bar{F}\) depends on \(\omega_o \cdot n\), the roughness and the complex Index of Refraction (IOR) \(\eta\). The latter is thus four dimensional.

**Split-Sum Approximation.** To reduce the storage cost of the material integral, Karis introduced the Split-Sum Integral [Karis 2013] that uses Schlick’s decomposition of the Fresnel term:
\[
\bar{F} \simeq F_0 \bar{S}_0 + (1 - F_0) \bar{S}_1, \tag{3}
\]
where \(\bar{S}_0\) and \(\bar{S}_1\) are the average of Schlick’s base functions: \(s_0(\omega_i \cdot n) = 1\), and \(s_1(\omega_i \cdot n) = (1 - \omega_i \cdot n)^2\) [Schlick 1994].

However, Schlick’s equation is a poor approximation, especially for metals. Also, it is only parametrized by the reflectance at normal incidence and cannot accommodate for hue variation at grazing angles. This prevents the use of artistic control over metal reflectance [Gulbrandsen 2014] in modern real-time rendering engines (see Figure 2 right).

### 3 AN EMPIRICAL DECOMPOSITION

**Fresnel Decomposition.** Our work builds on decomposing of the Fresnel reflectance as a weighted sum of basis functions \(b_i(\theta)\):
\[
\bar{F}(\theta; \eta) = \sum c_i(\eta) b_i(\theta). \tag{4}
\]
By separating degrees of freedom of the Fresnel reflectance between the base functions and the coefficients, we provide a form amendable to the Split-Sum Integral trick. Indeed:
\[
\bar{F} = \sum c_i(\eta) \bar{b}_i, \tag{5}
\]
where \(\bar{b}_i\) is the integral of \(b_i(\theta)\) times the microfacet response.

**Basis Functions.** We determine \(b_i(\theta)\) using an iterative Eigenvectors extraction (using SVD) over a large set of Fresnel reflectances to generate the different base functions. We forced \(b_0(\theta) = 1\) as the first element and shifted and rescaled the Fresnel reflectances so that the decomposed curves start at 0 and ends at 1. At each iteration, a new basis is obtained by removing the current approximation from the ground truth reflectance, and taking the first eigenvector of its SVD:
\[
b_k = \text{svd} \left[ F - \sum_{i=0}^{k-1} c_i \bar{b}_i \right]_0, \tag{6}
\]
where svd \([-]\) is the first eigenvector. Figure 1 (a) shows the first four elements of our decomposition. In practice, we only kept 4 elements in our basis: the Split-Sum preintegration requires a texture channel per basis element. Four basis elements fits a RGBA texture, requiring a single fetch at runtime.

**Coefficients.** We evaluate the coefficients of this decomposition in the same iterative way by evaluating the Fresnel curve at specific positions. When using 4 basis elements, at most 4 evaluations of the target Fresnel curve are required to constrain the system of unknowns. For that, we used \(\theta_0 = 0, \theta_1 = \frac{\pi}{2}\) to fully determine \(c_0\) and \(c_1\), and we use \(\theta_2 \simeq 1.45, \theta_3 \simeq 1.31\) to determine \(c_2\) and \(c_3\). See our supplemental materials for more details on fitting.

**Gulbrandsen’s Parametrization.** Since the artistic parametrization of metals of Ole Gulbrandsen [2014] enables to evaluate the exact Fresnel curve using the reflectivity \(r\) and edgetint \(g\), it is possible to extract \(c_0, c_1, c_2, c_3\) from \(r\) and \(g\) using the method already described.

### 4 RESULTS
We implemented our Fresnel decomposition in an offline path tracer, a real-time prototype using OpenGL and in the Unity engine. See our supplemental materials and video for more results.

Figure 1 (b), show the integration of our decomposition for a Beryllium conductor in Unity. In Figure 2, we show in our OpenGL prototype that our method correctly reproduces the edgetint of Gulbrandsen. There, we measured that our method was at most 0.05 ms slower than Schlick’s decomposition in full screen at 720p on an Nvidia RTX 2070.

![Figure 2: Our method reproduces better the edgetint than Schlick’s decomposition compared to a Monte-Carlo reference. We used \(r = [0.44, 0.44, 0.44]\) and \(g = [0.1, 0]\) and \(\alpha = 0.08\).](image-url)

### ACKNOWLEDGMENTS
The authors thank Thomas Deliot for implementing the method in the Unity engine and Kenneth Vanhoey, Jonathan Dupuy and Eric Heitz for proof-reading.

### REFERENCES