

# Spectral Ray Tracing

Dawson Do

daws@berkeley.edu

Jeremy Chen

jcmchen@berkeley.edu

Joshua Fernandes

jbfern@berkeley.edu

Jingyi Guo

guojingyi518@berkeley.edu



**Figure 1:** Renders showcasing spectral rendering of light dispersion, thin-film iridescence, and microfacet materials.

## ABSTRACT

In this project, we write a ray tracer that includes spectral information about rays within the light field. A traditional ray tracer tracks RGB intensities along the ray and propagates the ray using RGB information about scene elements. In reality, light is distributed by wavelength in a continuous space, with visible light occupying the range approximately from

400 nm to 700 nm. Optical phenomena arise from wavelength-dependent interactions, including dispersion, iridescence, and reflection probability. In this project, we refactor the ray tracer we wrote in Homework 3 to incorporate wavelength dependence by adding a layer of wavelength sampling. We also implement scene elements that leverage this feature, including glass materials, mirror surfaces, microfacet surfaces, thin films, and black body emitters. With these, we can create striking visible scenes accessible only through spectral information.

---

Permission to make digital or hard copies of all or part 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 components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.  
*CS 284A: Final Project, Spring 2024, UC Berkeley, CA, USA*

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

## CCS CONCEPTS

- Computing methodologies → Ray tracing; Reflectance modeling; Visibility.

## KEYWORDS

Spectral ray tracing, Dispersion, Iridescence, Thin-Film Interference

## 1 INTRODUCTION

In an RGB ray tracer, the spectrum of light is tracked as a three-vector corresponding to red, green, and blue intensity. In reality, light is an infinite-dimensional spectrum of wavelengths. A more physical rendering equation would consider the entire spectrum of light emitted and consider absorption, reflection, and refraction of the spectrum. A spectral ray tracer keeps track of the wavelength and the intensity of each. This is an important problem to capture dispersion of light, the phenomenon that the index of refraction is wavelength-dependent. This is what leads to a prism splitting light into the constitutive components, and rendering this example image is the ultimate goal of our project. It is a challenging problem since we change a three-dimensional intensity into an infinite-dimensional one. The solution for addressing this challenge is to sample from the spectrum of wavelengths. Then, after collecting all the rays, we then can convert the distribution of wavelengths to red, green, and blue intensities for rendering on a computer screen.

Past projects implement some of the following features, such as iridescence and black body radiation [3, 4]. Our work is inspired by these previous works and seeks to extend these features to a spectral ray tracer, and with additional utility.

## 2 FROM RGB TO SPECTRAL RAY TRACER

### 2.1 Spectral Rays

To refactor the ray tracer for spectral information, we added an inner loop of sampling in `raytrace_pixel()`. For each ray, we also sample wavelengths and store this information in the ray object, which is passed into `BSDF::sample_f()`. The rendering equation is unchanged, but now includes a wavelength dependence, as

$$L_o(p, \lambda, \omega_o) = L_e(p, \lambda, \omega_o) + \int_{H^2} f_r(p, \lambda, \omega_i \rightarrow \omega_o) L_o(\text{tr}(p, \omega_i), \lambda, -\omega_i) \cos \theta_i d\omega_i, \quad (1)$$

where the light intensities are now scalar intensities (rather than RGB vectors). In the implementation, each BSDF now returns a scalar quantity representing spectral intensity, which is summed to estimate the global illumination. We will discuss the changes made to the old non-spectral BSDFs in Section 2.2 and the implementation of spectral BSDFs in Section 3.

We can think of the intensity parameterized by wavelength as an infinite-dimensional vector, where each wavelength has an associated intensity. We can downsample to RGB values using the spectral response functions as

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} \int S_R(\lambda) L(\lambda) d\lambda \\ \int S_G(\lambda) L(\lambda) d\lambda \\ \int S_B(\lambda) L(\lambda) d\lambda \end{bmatrix}, \quad (2)$$

where  $S_i$  is the spectral response function for color  $i$  and  $L(\lambda)$  is the intensity as a function of wavelength. Here, we use the spectrum under D65 illumination to convert from wavelength and intensity to each RGB value.

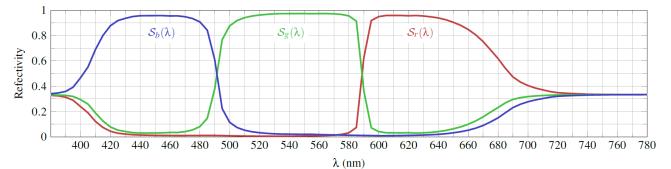
We compute this integral by sampling uniformly in the wavelength space. One option is to take independent random samples, but we instead use *hero sampling*, which chooses the first sample uniformly at random and places the remaining  $n - 1$  samples at equispaced positions relative to the first sample. This method is unbiased and is empirically shown to reduce the color noise [6]. In our implementation, we find that using 5 samples sufficient for convergence.

### 2.2 Spectral Upsampling

The RGB BSDFs from HW 3-1 and 3-2 use a stored RGB albedo to return the color of the surface, but after the above modifications, they need to return the spectral intensity of the corresponding wavelength that hit the surface. To accomplish this, we implement spectral upsampling using the spectral primaries basis computed by Mallet and Yuksel [5], which we store as 81 values corresponding to equidistant wavelengths between 380 and 780 nm. Given an RGB albedo, we compute the spectral intensity at the ray's wavelength using linear interpolation of the stored basis values.

$$S(\lambda) = R \cdot S_r(\lambda) + G \cdot S_g(\lambda) + B \cdot S_b(\lambda) \quad (3)$$

For every BSDF that originally returns an RGB, we convert that to an intensity using Eq. 3.

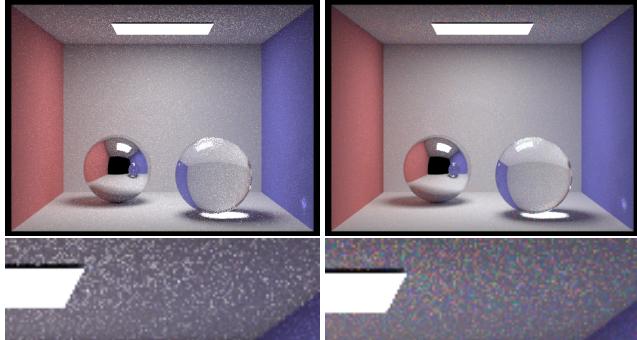


**Figure 2: Spectral Primaries Basis from [5].**

Following these modifications, we can already see differences in our renders, even though we haven't introduced wavelength-dependent interactions. Figure 3 is an example render using the existing scene files with no wavelength-dependent light transport, shown using the RGB ray tracer and the spectral sampling method. Both should be the same in expectation, but we see that the nature of the noise changes. Colorful noise is introduced in the spectral ray tracer due to randomness in sampling wavelengths.

## 3 SPECTRAL BSDF

To prepare the code for the glass prism and other spectral materials, reflective, refractive, and microfacet materials were implemented according to the specifications of HW 3-2. The



**Figure 3:** On the left, RGB ray tracer render. On the right, spectral sampling ray tracer render. Below shows magnifications of the observed noise (grayscale in RGB rendering, colorful in spectral sampling).

next steps were to make wavelength dependent versions of these functions.

### 3.1 Dispersive Glass

We make a simple modification to GlassBSDF so that the index of refraction is a function of the wavelength, given by Cauchy's approximation:

$$\eta(\lambda) = A + \frac{B}{\lambda^2},$$

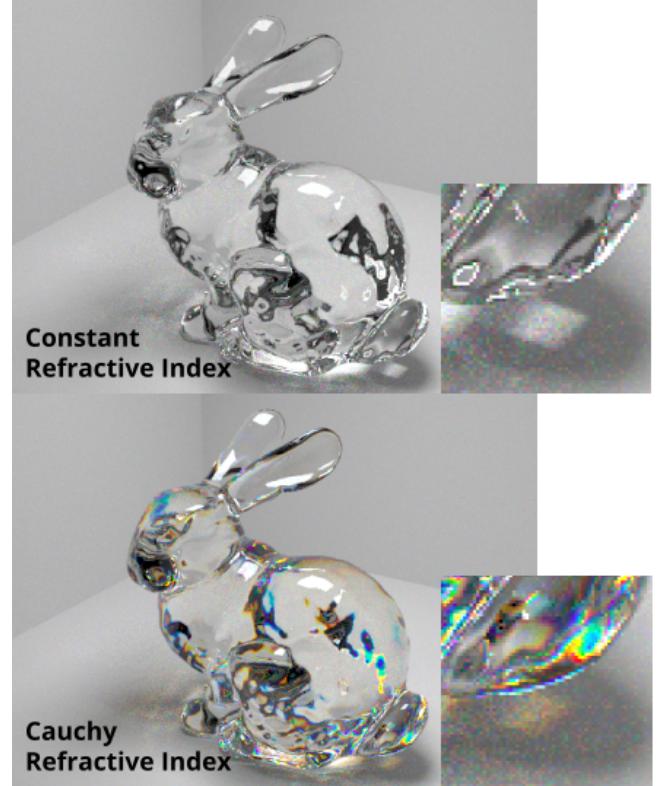
where  $A$  and  $B$  are parameters. In addition, we use the dielectric Fresnel reflectance instead of Schlick's approximation. With this BSDF, we can see chromatic aberrations on the surface and dispersion effects in the shadows, as shown in the renders of the bunny in Figure 4.

### 3.2 Spectral Microfacet

The modification to microfacet materials is similar, and solely inside the computation of the Fresnel reflectance,  $F$ , in the following:

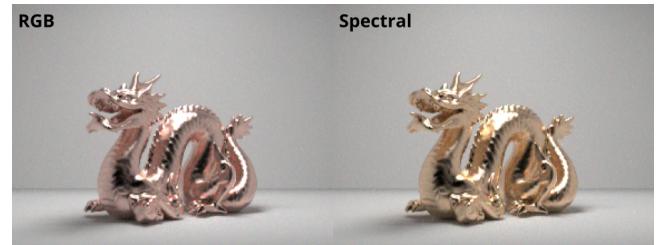
$$f(\lambda, \omega_i \rightarrow \omega_o) = \frac{F(\lambda, \omega_i \cdot h) \cdot G(\omega_i, \omega_o) \cdot D(h)}{4 \cdot (n \cdot \omega_i) \cdot (n \cdot \omega_o)}. \quad (4)$$

In the RGB version, the BSDF stores the refractive index  $\eta$  and extinction coefficient  $\kappa$  at 614 nm (red), 549 nm (green) and 466 nm (blue) and returns the corresponding reflective intensity for these particular wavelengths. The spectral version stores  $n$  arbitrarily spaced wavelength samples ( $\lambda_i \forall i = 1 \dots n$ ) between 380 nm and 780 nm (inclusive) and the corresponding  $\eta_i$  and  $\kappa_i$  at those wavelengths. Given the incoming ray's wavelength, we use a linear interpolation to compute  $\eta(\lambda)$  and  $\kappa(\lambda)$ , which is used to compute the Fresnel reflectance.  $f$  now returns the reflective intensity corresponding to the individually sampled wavelength. In



**Figure 4:** Top, bunny rendered with non-dispersive glass ( $A = 1.45, B = 0$ ). Bottom, bunny rendered with dispersive glass ( $A = 1.45, B = 50,000$ ).

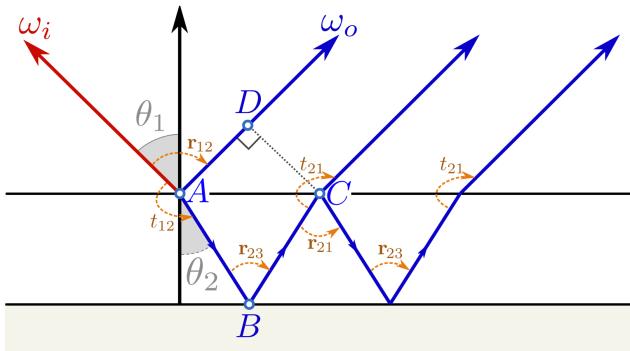
Figure 5, we can see that this subtly affects the color of the material, as expected since we sample more of the visible spectrum.



**Figure 5:** CBdragon with copper surface,  $\alpha = 0.15$ . On the left, RGB up-sampled microfacet BSDF. On the right, spectral microfacet BSDF.

### 3.3 Dielectric Thin-film Interference

We extend the spectral microfacet BSDF to additionally model a thin dielectric film on top of a dielectric or conductor base.



**Figure 6:** Airy reflectance (Eq. 6) is the summation of the reflectance of all paths that bounce inside the thin film. Figure originally from [1].

In this BSDF, we again modify the Fresnel reflectance to account for the bounces between the dielectric film and base (Figure 6). For clarity, we will call this thin-film reflectance the Airy reflectance,  $R$ , which we use to replace  $F$  in Eq. 4. The following formula are based from those reproduced in Belcour and Barla [1], which were originally given in Born and Wolf (Section 14.4) [2]. Assuming equal distribution of polarized to non-polarized light, the reflectance  $R$  is defined as:

$$R = \frac{1}{2} (|\mathbf{r}^\perp|^2 + |\mathbf{r}^\parallel|^2), \quad (5)$$

where  $\mathbf{r}$  is a *complex* reflection coefficient.  $\mathbf{r}$  is *complex* because the conductor base has a complex index of refraction  $\eta + ik$ .  $\mathbf{r}$  for either polarization sums the contribution of all rays reflected inside the film-base interface:

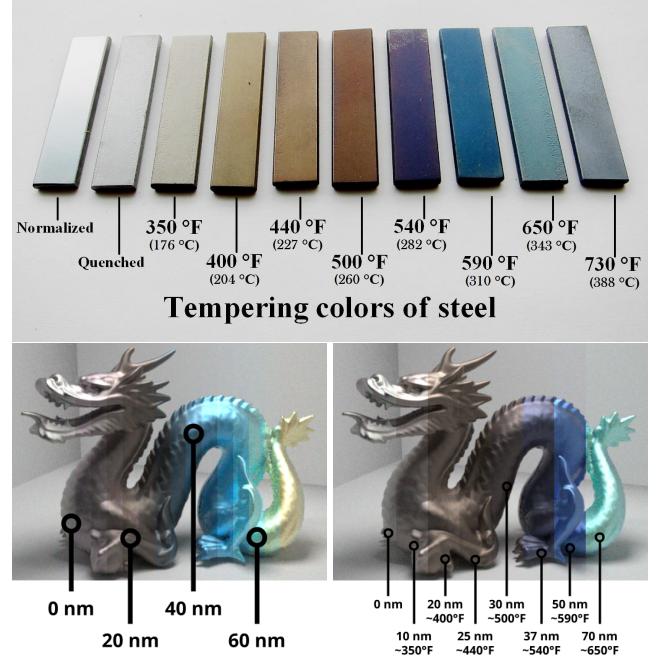
$$\begin{aligned} \mathbf{r} &= \mathbf{r}_{12} + t_{12}\mathbf{r}_{23}t_{21}e^{i\phi} + t_{12}\mathbf{r}_{23}\mathbf{r}_{21}\mathbf{r}_{23}t_{21}e^{2i\phi} + \dots \\ &= \mathbf{r}_{12} + \sum_{k=1}^{\infty} t_{12}\mathbf{r}_{23} [\mathbf{r}_{21}\mathbf{r}_{23}]^{k-1} t_{21}e^{ik\phi} \\ &= \mathbf{r}_{12} + \frac{t_{12}\mathbf{r}_{23}t_{21}e^{i\phi}}{1 - \mathbf{r}_{21}\mathbf{r}_{23}e^{i\phi}} \end{aligned} \quad (6)$$

where  $\mathbf{r}_{ab} = r_{ab}e^{i\phi_{ab}}$  is a complex reflection coefficient going from medium  $a$  to medium  $b$ . Similarly  $t_{ab}$  is a real transmission coefficient from  $a$  to  $b$ .  $\phi$  is the phase shift due to the optical path difference, defined as  $\phi = \frac{4\pi}{\lambda} \eta_2 d \cos \theta_2$ , where  $d$  is the film thickness in nanometers and  $\cos \theta_b = \sqrt{1 - \frac{\eta_1^2}{\eta_b^2}(1 - \cos^2 \theta_1)}$  according to Snell's law. The formula for  $r_{ab}$ ,  $t_{ab}$ , and  $\phi_{ab}$  for each polarization are included in Appendix A.

Using this BSDF, we can reproduce iridescent films on top of both dielectric and conductive bases (Figure 7). In addition, we can also model the effects of tempering on the color of steel alloys. Tempering steel creates a thin surface



**Figure 7:** Soap ( $\eta_2 \approx 1.33$ ) film on PVC ( $\eta_3 \approx 1.54$ ) and Copper.



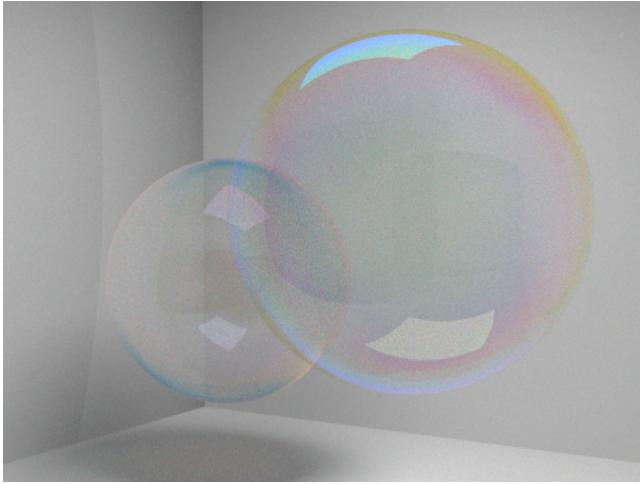
**Figure 8:** Iron oxide film on iron ( $\eta_3 \approx 2.8$ ,  $\kappa_3 \approx 3$ ) dragons, compared to a real reference of tempered steel [wikipedia]. The left dragon uses varying refraction index from [refractionindex.info], the right dragon uses constant  $\eta_2 = 2.91$ .

film of oxide on the base during the heating process. As this film thickens, the appearance changes due to thin-film interference. In Figure 8, we use a base layer of iron with progressively thicker layers of iron oxide,  $\text{Fe}_2\text{O}_3$ , to recreate the phenomenon. On the left, we use a varying refractive index and test thickness between 0 nm and 70 nm at 5 nm

increments. On the right, we reproduce the results from the supplemental materials of [1], with a constant refractive index at the thicknesses they used. This approximation has a few potential sources of error. For one, the oxide on the surface may not just be iron oxide, but a combination which includes chromium oxide. Additionally, the extinction coefficient of oxides are not truly negligible, and thus it cannot be treated as a dielectric material. Our current implementation does not support a conductor as the film, which may account for the inconsistencies.

### 3.4 Films on Transparent Objects

We can directly use the Airy reflectance (Eq. 6) as our probability of reflection in transmissive materials. With this, we can model thin-films on transparent materials such as glass and air. The latter is how we create soap bubbles. However, we are unable to sufficiently model translucency as the soap bubbles still cast shadows (Figure 9). In addition, films in the real world will have varying thickness due to the affects of air movement and gravity. We did not get to implement these two features.



**Figure 9: Soap bubbles modeled using a film ( $\eta_2 = 1.33$ ) on a vacuum ( $\eta_3 = 1.0, \kappa_3 = 0$ ). The foreground and background bubbles have a thickness of 500 nm and 350 nm, respectively.**

## 4 BLACK BODY RADIATION

The final elements of the scene to be converted to a spectral version are lights. A common type of light source is a *black body*, which emits a spectrum of light based on its temperature. The black body model can be used to approximate the spectral power distributions for incandescent lighting, the sun, and even the infrared radiation of living things! The

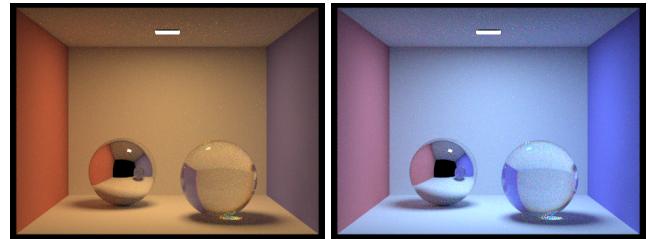
SPD of a black body at temperature  $T$  is given by

$$s(\lambda) = \frac{2hc^2}{\lambda^5} \left[ \exp\left(\frac{hc}{\lambda k_B T}\right) - 1 \right]^{-1}, \quad (7)$$

where  $h$  is Planck's constant,  $c$  is the speed of light,  $k_B$  is Boltzmann's constant, and  $\lambda$  is the wavelength. We allow for black body light emitters (point and area) that we parameterize by two variables, a temperature and brightness. Brightness serves as a normalization factor in front of the SPD that can account for rescaling of the black body.

When we sample the black body BSDF, we simply return the appropriate intensity at that wavelength and similarly when we sample the light in direct lighting sampling.

Colder objects have black body spectra that appear orange-reddish, while warmer objects have black body spectra that appear blueish. At  $T \sim 6500$  K, we get white light, which is the effective temperature of the sun and the basis of the D65 illuminant standard. Figure 10 shows the CBspheres scene for two more extreme temperatures to highlight this difference.



**Figure 10: CBspheres with a black body area light at temperature 3000 K (left) and temperature 12000 K (right).**

## 5 DISCUSSION

Lessons learned – lot more refactoring of existing code than expected, better understanding of optics/physics, manipulating a significant coding project

## 6 CONCLUSIONS

Significance of entire project

Things we didn't do but thought of (speed, novel approaches we could try – sampling, photon mapping)

## REFERENCES

- [1] Laurent Belcour and Pascal Barla. 2017. A practical extension to microfacet theory for the modeling of varying iridescence. *ACM Trans. Graph.* 36, 4, Article 65 (jul 2017), 14 pages. <https://doi.org/10.1145/3072959.3073620>
- [2] Max Born and Emil Wolf. 2019. *Principles of Optics: 60th Anniversary Edition* (7 ed.). Cambridge University Press.

- [3] Sydnie Shea Cohen, Ada Hu, Eugene Lee, and Daniel Tseng. 2021. Iridescence. *CS 184/284A: Computer Graphics and Imaging* (2021). <https://sydnie-shea.github.io/iridescentFinal/>
- [4] Jason Li, Katherine Song, and Alex Yang. 2020. Rendering Iridescence. *CS 184/284A: Computer Graphics and Imaging* (2020). [https://kwsong.github.io/cs184\\_final/final.html](https://kwsong.github.io/cs184_final/final.html)
- [5] Ian Mallett and Cem Yuksel. 2019. Spectral Primary Decomposition for Rendering with sRGB Reflectance.. In *EGSR (DL/I)*. 9–15.
- [6] A. Wilkie, S. Nawaz, M. Droske, A. Weidlich, and J. Hanika. 2014. Hero Wavelength Spectral Sampling. *Computer Graphics Forum* 33, 4 (2014), 123–131. <https://doi.org/10.1111/cgf.12419> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12419>

## A THIN-FILM FORMULA

We reproduce these formula here as we discovered a typo in one of our references. We keep similar notation, with minor changes for clarity. These formula are used for the interference between a dielectric outer medium (air/vacuum), dielectric film, and either dielectric or conductor base (i.e.  $\kappa_1 = \kappa_2 = 0$  and  $\kappa_3 \geq 0$ ). Each medium  $a$  is defined to have index  $\eta_a + i\kappa_a$ , and the complex reflection coefficient is  $r_{ab} = r_{ab}e^{i\phi_{ab}}$ , with:

$$|r_{ab}^\perp|^2 = \frac{(\eta_a \cos \theta_a - u_{ab})^2 + v_{ab}}{(\eta_a \cos \theta_a + u_{ab})^2 + v_{ab}}$$

$$\tan \phi_{ab}^\perp = \frac{2v_{ab}\eta_a \cos \theta_a}{u_{ab}^2 + v_{ab}^2 - \eta_a^2 \cos^2 \theta_a}$$

for perpendicular polarization; and

$$|r_{ab}^\parallel|^2 = \frac{[(\eta_b^2 - \kappa_b^2) \cos \theta_a - \eta_a u_{ab}]^2 + [2\eta_b \kappa_b \cos \theta_a - \eta_a v_{ab}]^2}{[(\eta_b^2 - \kappa_b^2) \cos \theta_a + \eta_a u_{ab}]^2 + [2\eta_b \kappa_b \cos \theta_a + \eta_a v_{ab}]^2}$$

$$\tan \phi_{ab}^\parallel = 2\eta_a \cos \theta_a \frac{2\eta_b \kappa_b u_{ab} - (\eta_b^2 - \kappa_b^2)v_{ab}}{(\eta_b^2 + \kappa_b^2)^2 \cos^2 \theta_a - \eta_a^2(u_{ab}^2 + v_{ab}^2)}$$

for parallel polarization, where

$$u_{ab} = \frac{\sqrt{U_{ab} + V_{ab}}}{2}, \quad v_{ab} = \frac{\sqrt{V_{ab} - U_{ab}}}{2}$$

with

$$U_{ab} = \eta_b^2 - \kappa_b^2 - \eta_a^2 \sin^2 \theta_a, \quad V_{ab} = \sqrt{U_{ab}^2 + 4\kappa_b^2}.$$

Lastly, the Fresnel transmission coefficients are:

$$t_{ab}^\perp = \frac{2\eta_a \cos \theta_a}{\eta_a \cos \theta_a + \eta_b \cos \theta_b}$$

$$t_{ab}^\parallel = \frac{2\eta_a \cos \theta_a}{\eta_a \cos \theta_b + \eta_b \cos \theta_a}.$$

## B CONTRIBUTIONS

- JC:
- DD: Implemented spectral upsampling, RGB down-sampling, spectral microfacet & thin-film BSDFs, composed scenes for final render and figures

- JF: Refactored the code for wavelength dependence, implemented the dispersive glass BSDF, implemented the black body BSDF, and handled parsing of modified COLLADA files for new inputs.

- JG:

Received 30 April 2024