Workshop on Material Appearance Modeling
MANER Conference - Material Appearance Network for Education and Research (2024)
J. Y. Hardeberg and H. Rushmeier (Editors)

Reflectometer Material Capture

D. Chan and M. Iwanicki
Activision Central Tech

Abstract
Physically-based shaders benefit from physically measured material parameters. The Reflectometer is a simple-to-build, portable device that allows measurement of diffuse albedo, smoothness, and specular F_0 in the field.

1. Introduction

Physically-based shaders model real-world material phenomena using a simplified model that reduces the BRDF dimensionality. To fully exploit the expressivity of this reduced model, it is important to author our material parameters accurately, such that reflected radiance of the surface matches real-world materials as closely as possible.

It is difficult to author diffuse albedo, smoothness (roughness) and specular F_0 values by eye since tonemapping to standard displays compress values and the real-world doesn't always conform to our expectations. During development of *Call of Duty: WWII*, a console and PC game, we created the Reflectometer to help with measuring BRDFs.

2. Reflectometer

The Reflectometer (Figure 1) is a portable, hand-held device that can be used in the field to capture a non-spatially varying BRDF. It measures a small area for directional reflectivity given a distribution of incoming light. We use these measurements to fit diffuse albedo, smoothness and specular F_0 for our shader model. Our intent is to use it to build up a library of physically measured values to inform art asset material authoring.

Internal Design. The Reflectometer is designed as an arc of six LEDs interleaved with six light sensors. It shines light on an approximately 1/8" x 1/8" area from six directions and measures the reflected light from six directions (Figure 2). This is a coarse measurement of BRDF, constrained to the plane defined by the light and sensor directions, which is in turn constrained to be perpendicular to the surface being measured. We used black baffled geometry in the light and sensor tunnels to provide a simple method to direct the light, a technique we borrowed from [Cha15].

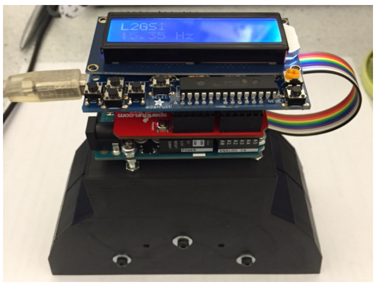

Figure 1: *Reflectometer controlled using Arduino.*

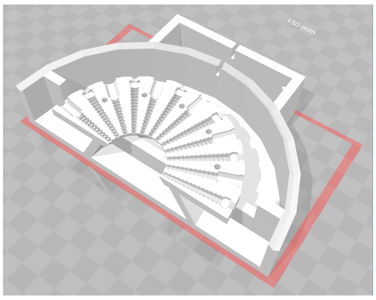

Figure 2: *The inside of the reflectometer, showing baffled geometry for the light and sensor tunnels. Starting from the left, we have light, then sensor, repeating in an arc.*

Visualizing in BRDF Explorer. To visualize our measurements, we modified *BRDF Explorer* [Bur12] to load the CSV files that the Reflectometer outputs. Figure 3 shows how we visualize raw values. Although we only take sensor readings along an arc, we fill outward from the arc and cover the entire hemisphere with these readings when visualizing. This leads to a stepped, cel-shaded look.

3. Calibrating for diffuse albedo and specular F_0

Once we can capture raw reflectivity values, we need to calibrate for diffuse and specular reflectivity. To calibrate for diffuse albedo, we use a measurement of Spectralon as a baseline and assume 100% diffuse reflectance.

It's fairly simple to calibrate diffuse reflectance if we assume a Lambertian surface since our diffuse albedo is just the ratio of our current measurement against the baseline measurement of Spec-

DOI: 10.2312/mam.20241179

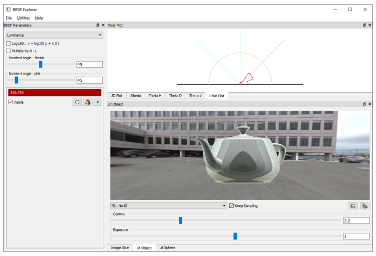

Figure 3: *A measurement of a waterproof invasion bag, visualized in BRDF Explorer. Notice the stepped nature of the lighting and polar plot. This is due to having few measurement samples and visualizing them using nearest-neighbor.*

tralon. The tunnel geometry and the apertures don't affect our Lambertian measurement.

Calibrating for specular reflectance is more involved. We measure reflectance of a first-surface mirror as our baseline and assume a smoothness of 1 (roughness of 0). In order to determine how to map raw reflectivity to specular reflectance, we simulate the bounce of photons off a surface using our GGX-based [WMLT07] specular BRDF given the specific apertures of our LED and sensor tunnels.

We start the photon at a random location on a disk representing the LED. We choose a random point in the aperture to fire our photons through. Then we trace the path of the photon as it hits the surface, sample our distribution of visible normals (vNDF) to find our microfacet normal [Hd14, Hei17], apply masking of the photon after the bounce (we are assuming a single scattering specular model), and reflect a fractional photon, based on F_0.

We record the number of photons that pass through the sensor aperture and reach the sensor. We fire many photons (2^{21}) for each smoothness value (0 to 255), for each F_0 value (0 to 255), from each emitter (LED) to each receiver (sensor), and generate a table where we record how many photons are received by the sensor.

We calculate α_{GGX} for our GGX-based microfacet model from smoothness: $\alpha_{GGX} = (1 - smoothness)^2$. If p_s is the photon count for a specific smoothness s, p_m is the photon count for a perfect mirror, and r_m is the reflectivity measured for a first-interface mirror, we can calculate r_s, the reflectivity of a surface with smoothness s: $r_s = (p_s r_m)/p_m$. We use this large table of r_s values during our fitting process. This is more correct than calculating the specular BRDF analytically during fitting as if we had a point light source and point sensor. It also happens to be faster to execute.

4. Fitting

Once we have diffuse and specular reflectance calibrated, we can fit to our shader model using Differential Evolution [SP97], a nonlinear optimization technique.

We treat Differential Evolution as a black-box optimizer, which passes seven material parameters to our cost function: diffuse albedo RGB, F_0 RGB, and smoothness. Our cost function loops through all lights, then through all sensors and accumulates the difference in log-space between our measurement value and the value calculated using the material parameters passed in. We use analytical Lambertian for our diffuse contribution and the specular table (r_s) generated through photon bouncing for our specular contribution of our microfacet-based BRDF. Differential Evolution returns the material parameters with the lowest cost.

5. Color Correction

Because we are using RGB LEDs with unknown spectral power distributions and a light sensor with an unknown spectral response curve, the captured color values need to be corrected to sRGB for use in our engine.

Even though we are transforming from one three-valued vector to another, we can't use a simple linear transformation for color correction, as we would when transforming between RGB colorspaces that inhabit the same three-dimensional space, e.g. sRGB and XYZ. If we imagine the three-valued vectors as lossy approximations to the full spectral response curves, we can see we are looking for a transformation from one lossy space to another.

The off-the-shelf RGB LEDs that we use are narrow-band LEDs and are not ideal for measuring wide-band spectral distributions, making color correction more involved. We discuss how this design can be improved in Section 7.

Home-made color chart. The standard X-Rite color chart did not have enough colors to color correct accurately. We needed a large sample size of materials to measure for color and these samples needed to be large enough for our Reflectometer to measure. Because of this, we created our own color chart, which is comprised of six panels of paint swatches. We use this, plus the standard X-Rite color checker, plus four colored Spectralon samples, resulting in a total of 208 color samples. Initially, we took cross-polarized RAW photographs of our color chart to collect sRGB values for diffuse albedo of each swatch. Later, we used an internally-developed cross-polarized spectrometer (*CTSpectro*) to validate and re-measure these diffuse albedo values.

Transforming colorspaces with 3D lookup tables. Given a table correlating fit diffuse albedo colors in the reflectometer's unknown color space to cross-polarized capture colors (in sRGB/D65), we use *3D LUT Creator* to create a lookup table to map from our unknown colorspace to sRGB.

It helps to transform the linear reflectometer color values into a more perceptual space before creating the lookup table. We tried several different transformations (log/exp, sqrt/square) and finally settled on cube-root/cubed, based purely on attempting to minimize visual differences.

6. Results

Figure 4 shows results of our measurements and fitting, alongside reference photographs. Notice the smoothness (roughness) and the colors of the materials are portrayed reasonably well. We have used the reflectometer to build a measured material library. We use this to develop art assets, and these measurements help us get a better sense of appropriate smoothness values for a large variety of materials.

7. Reflectometer 2.0

After two years of use by the studios, we gathered significant feedback about the usability of the device and potential improvements.

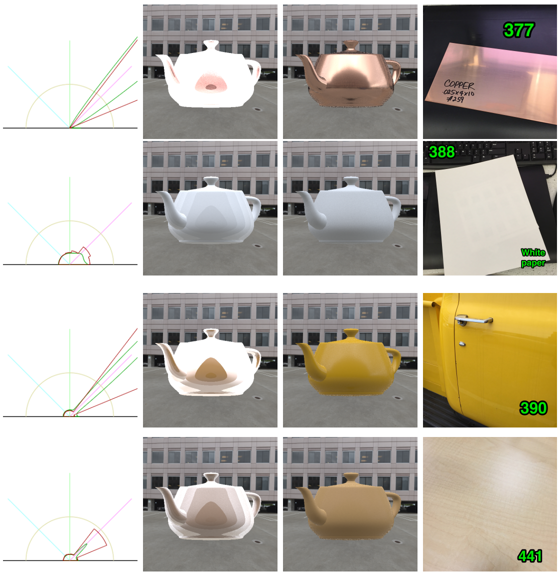

Figure 4: *From left to right: polar plot of BRDF with raw values in red and fit values in green, raw render of teapot, fit render of teapot, and reference photographs of materials measured. From top to bottom: copper, paper, yellow car paint, light wood.*

With more experience in developing in-house hardware we decided to build a second version of the device that would implement the features users asked for. The device is currently in a prototype stage, where we're finalizing the firmware and performing more thorough testing.

Main body. The general shape of the main body remained the same: an arc of six LEDs and six light sensors, placed at the tunnel ends aimed at the center of the sample measured.

While the printed body is easy to manufacture in-house, it inherently limits the minimum device size, due to FDM printing process inaccuracies. We performed a number of experiments using different 3D printing technologies, but none was able to achieve the level of both precision and physical properties (most importantly being impenetrable to light) that we need. We decided to create the main body out of CNCed aluminum that is later anodized to matte black. All optically active surfaces are additionally coated. We experimented with industrial carbon nanotube coatings, but found that a careful application of ultra-matte black acrylic paint, Mussou Black, can achieve comparable results at a fraction of the cost. Since the device is not exposed to extreme conditions or rough handling this was an acceptable trade-off.

To improve efficiency of both lighting and detection, we place a convex lens at the start of each tunnel and a 1mm diameter aperture on the other end. The geometry of the lens is matched to the length of the tunnels, so that the light generated by the LEDs and light reflected off the sample, is collimated onto the sensors. This also allowed us to eliminate the baffles inside the tunnels, which are difficult to manufacture at this scale. The final dome assembly measures 63mm x 56mm, and its cross-section is shown of Figure 5.

Lighting and sensors. Our first reflectometer used addressable RGB LEDs for illumination and light-to-frequency sensors to measure

Figure 5: *The main dome of Reflectometer 2.0, machined in aluminum and anodized in black to reduce stray light. Radial tunnels leading to slots housing sensors and LEDs are visible. Two halves are joined together with screws and positioning pins are used to ensure alignment.*

the amount of reflected light. Upon closer inspection, it turned out that the LEDs used emit light in relatively narrow spectral peaks which is problematic for accurate color measurements, as we might miss some of the intricacies of the color if it doesn't align with these peaks. While the light-to-frequency sensor used has a good spectral response in the visible range, we found more advanced sensors, with greater accuracy that could be used instead.

Version 2.0 of the reflectometer relies on broad-spectrum white LEDs with a high CRI (>95) for illumination. We used 3030 sized, surface-mount LEDs from the Seoul Semiconductor Sunlike series. For light sensing we chose the AMS73211 sensor. It features three separate diffraction gratings that allow for measuring the incoming light with the sensitivities following XYZ matching curves, supports a dynamic range of $3.43E+10$ and provides the measurement results digitally, via I2C bus, eliminating the need for precise analog-to-digital conversion.

The LEDs and the light sensors are mounted on a custom, flexible PCB board that wraps around the perimeter of the dome body, with the individual devices sliding into prepared slots in the dome.

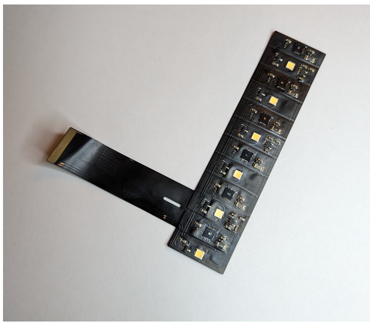

Figure 6: *Flexible PCB with LEDs (whie) and color sensors (black). The PCB wraps around the dome and the elements slide into designated slots.*

Built-in camera. One of the main drawbacks of the initial design was a lack of visibility of the sample under measurement. It made it difficult to accurately position the device, especially over smaller samples, but it also forced the users to take a separate picture of each measured material to provide a visual reference for the recorded values.

Proceedings published by Eurographics - The European Association for Computer Graphics.

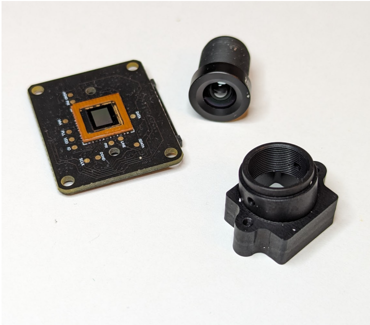

Figure 7: *Custom camera sensor board, custom angled mount and an off-shelf M12 lens allows for alignment of the focus plane with the sample.*

Version 2.0 features a built-in camera that points at the sample. We outsourced a design of a small lens assembly that would be able to focus the image from under the dome, but it turned out that the cost of such custom lens assembly, with acceptable quality, would be prohibitive. We settled on an off-the-shelf 3.8mm, M12 lens. It is however positioned using a custom-made lens mount that orients the lens at a specific angle to both the sample plane as well as the sensor plane. The angles and distances of individual elements were derived using the Scheimpflug principle, so that the focusing plane of the lens coincides with the sample plane and the entire flat sample is in focus, despite not the lens nor the sensor being perpendicular to it, due to spatial constraints.

Due to physical space constraints we were not able to use a pre-made camera sensor board, so we designed one in-house. The board features a low-noise 12-bit, 1.2MP AR0130CS sensor. Images from the camera are displayed on a built-in, 3.2-inch color screen, to allow the user precise positioning of the device. Images from the camera are also saved to an SD-card together with the measured values.

Electronics. The entire system is driven by an STM32H7 microcontroller. It is responsible for controlling the LEDs and the light sensors, readout of the camera sensor and displaying the images on screen. The microcontroller is clocked at 450MHz, and our design includes 16MB of RAM that opens the possibility of efficient calculations of the parameters of the measured BRDFs directly on the device. This functionality is however still under development.

The main device features an SD-card slot, voltage regulators to provide power for both the microcontroller and the LEDs (which require higher volages), a high accuracy constant-current circuit for driving the LEDs to ensure they are generating constant flux, a rechargeable Li-Ion battery (and all the additional electronics for controlling the charging cycles) and a USB-C connection for both communication with the host PC and charging the battery.

Enclosure. The external enclosure of the device is currently 3D-printed, using the FDM process. The complete device measures 118mm x 90mm x 62mm.

Figure 8: *Main PCB of Reflectometer 2.0, with STM32H7 microcontroller in the center, memory chip and SD card slot on the left and various peripherals on the right. Connector on the top is used to attach the screen, the one on the bottom connects to the camera board. Flexible PCB with the sensors and LEDs connects to the one on the left.*

Figure 9: *Complete device in a 3d-printed enclosure.*

Acknowledgments

Thanks to Tom Felker for fabricating the hardware for the first version of the Reflectometer; Peter-Pike Sloan for the many discussions about BRDF capture; Angelo Pesce for writing the Differential Evolution functions used in fitting; Joshua Terry and Thomas Hamilton for driving wider use of measurements in material authoring; Pete Shirley for guidance and help in editing.

References

[Bur12] BURLEY B.: Physically based shading at disney. In *SIGGRAPH Course Notes* (2012). 1

[Cha15] CHAN D. M.: Real-world measurements for call of duty: Advanced warfare. In *SIGGRAPH Course Notes* (2015). 1

[Hd14] HEITZ E., D'EON E.: Importance sampling microfacet-based bsdfs using the distribution of visible normals. In *Computer Graphics Forum* (2014). 2

[Hei17] HEITZ E.: A simpler and exact sampling routine for the ggx distribution of visible normals, 2017. 2

[SP97] STORN R., PRICE K.: Differential evolution- a simple and efficient heuristic for global optimization over continuous spaces. *J. of Global Optimization 11*, 4 (dec 1997), 341–359. 2

[WMLT07] WALTER B., MARSCHNER S. R., LI H., TORRANCE K. E.: Microfacet models for refraction through rough surfaces. In *EGSR* (2007). 2