Adding Zemax Transformation and Output Support for SOC-210

Ashish Jayamohan
Surface Optics, June 12 - September 8, 2023

Overview

I worked on adding support for the Zemax coordinate system and output format to the SOC-210 bidirectional reflectometer. I specifically added changes that allow users to effectively transform standard measurements of reflectance, centered around the normal vector, to any given set of angles. In combination with the variable resolution I added, this allows for users to measure the reflectance properties of any given specular or diffuse sample with increased resolution around the specular peak, while also saving time by decreasing the resolution for points further from the specular peak. Additionally, I also added support for conversion from the standard BRDF output format to the Zemax BSDF format.

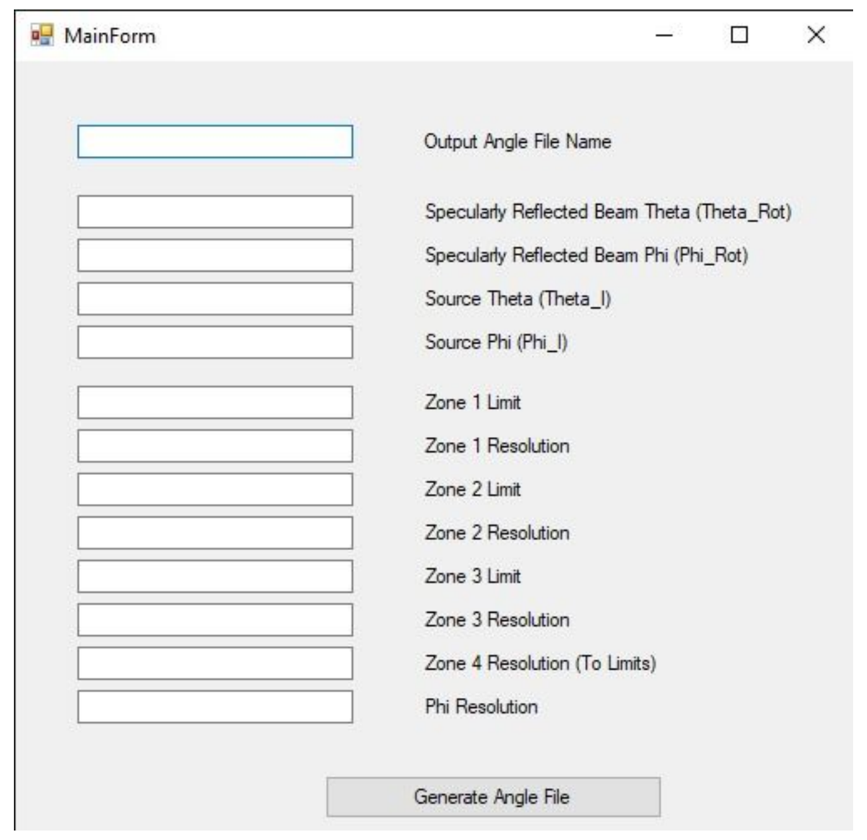
Lessons Learned

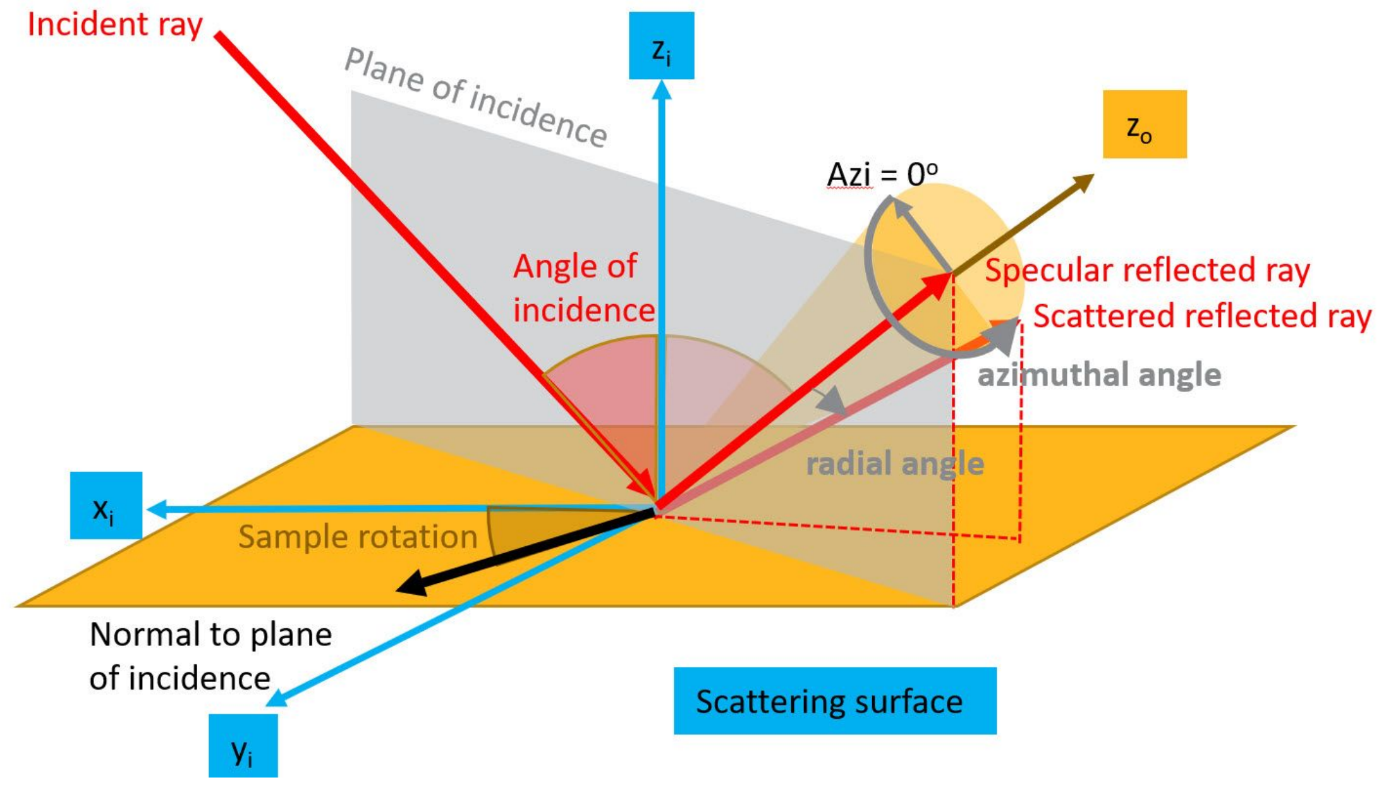
- Good design is often as important, if not more important, than being able to simply code. Having a well thought out program allows you to debug faster and makes it easier for others trying to work on the code base.
- Doing great work can sometimes by stifled by a lack of communication. Communicating effectively, clearly, and concisely allows people to understand why what you did is important.
- Focus on what know you can do first, then everything else. Being focused on what you know you can do allows you to prioritize work that you can get done faster than tasks that may be in uncharted territory.
- Write your thoughts down! It's so easy to forget a great idea or something important in a presentation if you don't have some notes written down. Writing also allows you to have a greater sense of accomplishment as you work.

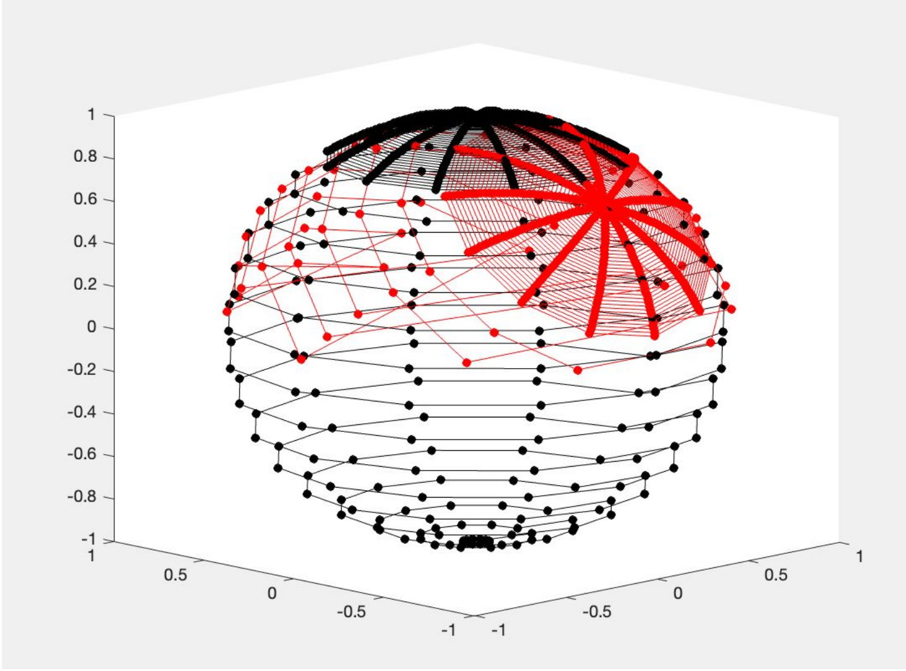
Technical Skills

The hardest technical challenge, by far, was adding support for the transformation between the current coordinate system and the Zemax coordinate system. As an added challenge, the transformation had to run fairly quickly, which created runtime challenges. I wrote a preliminary version of this transformation that allowed users to input necessary parameters into the command line and generate a functional angle file, a file with all of the transformed points of measurement used as input for the reflectometer. I was able to speed up this process and make this transformation run in near real-time through added efficiency in the ordering and generation of new points. I was also able to add several features that allowed for faster and more efficient testing and visualization. In addition, I also created a simple SLL-based Windows application that served as a GUI layer for the core, backend transformation.

Another technical challenge I faced was the conversion between the SOC BRDF output file format and the Zemax BSDF file format. The Zemax output file format stores the measurements in a matrix where the columns represent how far a given point is in the ϕ rotational direction from the specular peak. The rows represent how far a given point is in the Θ rotational direction from the specular peak. I was able to write a Python script that efficiently parsed through the BRDF file format and generated the file dynamically. Rather than storing the entire matrix, I was able to make it more efficient by just storing all the measurements from a slice and writing in batches to the output file. This allowed for the final conversion to be significantly faster than any pre-existing solution out on the market.







Ethics and Society

The research being done with the reflectometer I was working with is truly incredible. This summer, I was able to converse with a few customers and learned about their usage of the reflectometer. Although I cannot disclose the nature of the projects, it was very interesting to hear about the active areas of research in defense, aerospace, agriculture, and so much more. Researchers are using these tools for good, and I am certain that my contributions will make a tangible difference for these researchers, allowing them to take faster, more accurate measurements and benefit society in their own unique ways.

UCSD Connections

- **CSE 100** (Advanced Data Structures) - By far the most useful class for my internship experience. It was in this class that I learned much about C++ and graph and tree theory, both of which came in very handy this summer.
- **CSE 21** (Mathematics for Algorithms and Systems) - Another very useful course for my internship. In particular, the section about loop invariants helped me quite a bit.
- **CSE 30** (Computer Organization and Systems Programming) - This class was the general foundation I needed for learning C and C++. I benefited greatly from learning about memory allocation and cleanup procedures. Writing my own malloc/free functions was extremely helpful in understanding memory spaces and what really happens when you compile and run a program.