NPGR026 Practical Exercise Assignment 2 Specular Light-Surface Interactions in a Polarisation-Aware Renderer

Alexander Wilkie

April 16, 2020

1 Overall Goal

Implement the functions needed to

1. calculate light-surface interactions for perfectly smooth surfaces in a polarisation-aware renderer, to
2. maintain the correct reference co-ordinate system during a ray traversal in a scene, and to
3. implement the ability to insert a polarising filter into an optical path.

2 Required Results

A working computer program that allows the user to change the specified angles δ, ρ and ϕ of the inter-reflection geometry shown in figure 1, to optionally activate the polarisation filter P in the indicated position, and to change the material of the surfaces that the light is being reflected off at points X_1 and X_2. None of this requires that you explicitly implement any geometrical primitives in the program; all you have to do is to call the appropriate reflectance functions with the correct parameters!

The program should then output the Stokes vector for the resulting light that reaches the eye-point E via the optical path that starts at the non-polarised light-source L with an intensity of 1. The resulting Stokes vector should be given in the reference co-ordinate system R_e of the optical path segment next to the eye-point.

3 Specification of the Experiment

3.1 Experiment Geometry and Materials

An sketch of the geometry in which the optical path propagates is shown in figure 1. The material at the two points X_1 and X_2 can be either a perfectly smooth dielectric, or a similar conductor - this is something the user can choose. In the first case, the user has to specify the index of refraction (IOR), and the simpler version of the Fresnel terms can be used. If the material is a conductor, the user also has to specify the extinction

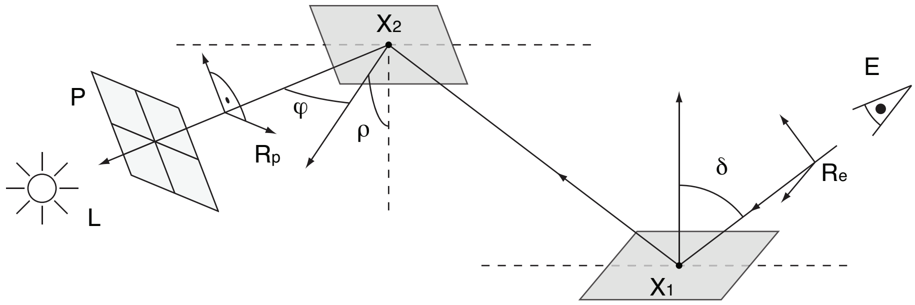

Figure 1: Geometry of the optical path simulated in this assigment. Note that for the sake of simplicity, this sketch assumes that the angles δ and ϕ are identical – which in practice they do not have to be! If they are not, the eye-point and the light source have to change their relative position, and the surface at point X_2 has to tilt for the optical path to follow the same route.

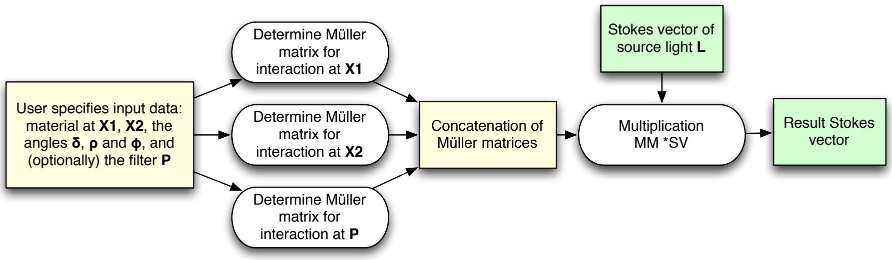

Figure 2: Overview of the program workflow.

coefficient, and the more complex form of the Fresnel terms has to be used. In both cases, only the reflection is of interest, and no refraction is taken into account.

The computations only have to be done for a single wavelength, i.e. only a single Stokes vector, oriented in the reference system P_e, has to be computed as end result (as opposed to a full spectral representation, where each spectrum would contain n individual Stokes vectors – one for each sample). The polarising filter P is assumed to be oriented in the same reference system R_p that the light has after the second reflection at point X_2.

3.2 Focus of this Assigment

The focus of this exercise are the following functions and data structures within the resulting program:

1. The data structures you use for light (a Stokes vector) and reflectance (a Mueller matrix). Note that while light only needs a single reference frame, a Mueller matrix needs two - for those cases where light is being reflected, and its direction changed. If no directional change occurs, the two are identical.

2. The function that, given a material (defined via its - possibly complex - IOR) and

an incident angle, returns the corresponding reflectance Mueller matrix.

3. The function that concatenates two Mueller matrices to a single matrix (this can make a re-alignment of the reference frames necessary)

4. The function that attenuates a given light ray with a Mueller matrix

Approximate expected results for the program are listed in section 5 for your reference, so you can verify that your program works correctly prior to handing it in.

3.3 Formulas needed for this Assigment

3.3.1 Fresnel Reflectance

As mentioned in the lecture, there are two forms of the Fresnel equations. The first, simpler one is used for dielectric materials, while the second, more complicated form is needed for materials with a complex IOR (conductors). The geometry of such a specular reflection is shown in figure 3. Both forms of the equations yield four values, $F_{\|}, F_{\perp}, \delta_{\|}$ and $\delta_{\perp}$. Two of these – the retardance values δ – are normally not even computed for graphics purposes. And the two reflectance values $F_{\|}$ and $F_{\perp}$ are normally just averaged in a non-polarising renderer.

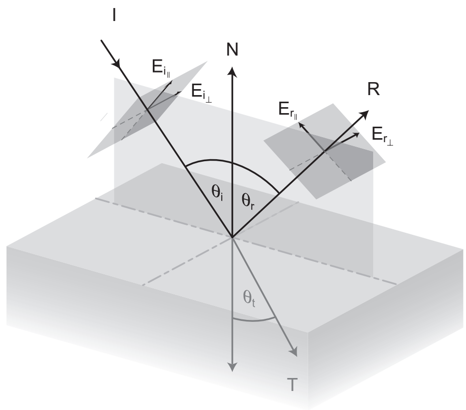

Figure 3: Geometry of a ray–surface interaction at an optically smooth phase boundary between two substances, as described by the the Fresnel equations.

However, for a polarisation-aware renderer, we ultimately need a full Mueller matrix for each reflection, and for a Fresnel reflection off a perfectly smooth phase boundary, this matrix is given by

$$\mathrm{T}_{Fresnel} = \begin{bmatrix} A & B & 0 & 0 \\ B & A & 0 & 0 \\ 0 & 0 & C & S \\ 0 & 0 & -S & C \end{bmatrix}, \tag{1}$$

with

$$\begin{aligned}
A &= \frac{F_\perp + F_{||}}{2} \\
B &= \frac{F_\perp - F_{||}}{2} \\
C &= \cos(\delta_\perp - \delta_{||}) \cdot \sqrt{F_\perp \cdot F_{||}} \\
S &= \sin(\delta_\perp - \delta_{||}) \cdot \sqrt{F_\perp \cdot F_{||}}
\end{aligned} \tag{2}$$

$\delta_\perp - \delta_{||}$ is the total retardance the incident wavetrain is subjected to. The only difference between dielectrics and conductors is which formulas are used for $F_{||}, F_\perp, \delta_{||}$ and $\delta_\perp$. Since computational efficiency is not a concern here, we recommend to use the full form of the Fresnel equations for both cases in this assignment to reduce the number of special cases in the code:

$$\begin{aligned}
F_\perp(\theta,\eta) &= \frac{a^2+b^2-2a\cos\theta+\cos^2\theta}{a^2+b^2+2a\cos\theta+\cos^2\theta} \\
F_{||}(\theta,\eta) &= \frac{a^2+b^2-2a\sin\theta\tan\theta+\sin^2\theta\tan^2\theta}{a^2+b^2+2a\sin\theta\tan\theta+\sin^2\theta\tan^2\theta} F_\perp(\theta,\eta) \\
\tan\delta_\perp &= \frac{2b\cos\theta}{\cos^2\theta - a^2 - b^2} \\
\tan\delta_{||} &= \frac{2\cos\theta[(n^2-k^2)b - 2nka]}{(n^2+k^2)^2\cos^2\theta - a^2 - b^2} \\
&\text{with} \\
\eta &= n + ik \quad \text{(the complex IOR)} \\
2a^2 &= \sqrt{(n^2-k^2-\sin^2\theta)^2 + 4n^2k^2} + n^2 - k^2 - \sin^2\theta \\
2b^2 &= \sqrt{(n^2-k^2-\sin^2\theta)^2 + 4n^2k^2} - n^2 + k^2 + \sin^2\theta
\end{aligned} \tag{3}$$

For dielectrics, one simply sets the extinction coefficient k of the complex IOR to zero. In a real renderer, one would of course use the simplified form of the equations for performance reasons instead.

3.3.2 Polarisation Filters

The Mueller matrix for an ideal polarisation filter is given as formula 6.15 in the NIST manual, the PDF of which is provided as supplementary material along with this assignment text. The PDFs of all parts of this self-study manual can be downloaded from the NIST website, and make for interesting reading, despite their age, and the fact that the PDFs only contain scanned pages of typewritten text. However, only the provided section is directly relevant to polarisation-aware renderers.

3.3.3 Rotation of Collinear Reference Systems

The remaining problem is that of rotating two collinear reference system to match. The formula needed for this is given as equation 6.39 in the NIST manual.

4 Work Details

4.1 Environment for this Assigment

Any programming language of your choice, that can be run or compiled on the University computers. In the opinion of the lecturer, good candidates for such a simple test environment that just has to handle comparatively few numerical operations would be the Python language, or perhaps Java. But within reason, do feel free to use any other system that you are familiar with instead.

As with the first assignment, this computer program does not have to have a fancy user interface – a command-line program is completely sufficient, although you are again of course free to do this in a GUI environment if you feel like it.

4.2 Assignment Examination Details

You are required to hand in the results of this assignment until the end of the semester. Before handing in your solution, make sure to compare the results your program computes with the figures listed in section 5! You are expected to mail the source code of your program to wilkie@cgg.ms.mff.cuni.cz until the set date; receipt of the source will be acknowledged via e-mail. The functionality of this source code will later be tested in your presence on a university computer, and you are expected to exactly know the inner workings of your code, i.e. there will be a short practical examination and discussion about your results on a one-to-one basis.

5 Expected Results

All the results are for an unpolarised lightsource with an intensity of 100, i.e. a Stokes vector of $(100.0, 0.0, 0.0, 0.0)$. The third set of complex IORs are for copper and gold, respectively, at a wavelength of 516 nm. The S_n columns in the table are the Stokes vector components of the result.

5.1 Input Parameters

- **Cases 1-3** correspond to light bouncing off two panes of glass (of different types). Adding the a linear filter between the input light and the first interface causes a significant change in overall reflected intensity.
- **Cases 4-6** are a Fresnel Rhomb (see e.g. the Wikipedia article on it for a sketch), an optical device in which total internal reflection is used to generate circularly polarised light from diagonally polarised input light. Because we are looking at an air-glass interface from the other direction, the real part of the IOR is smaller than 1: $\frac{1}{1.5105}$ is approximately 0.666.
- **Cases 7-9** are two bounces off metal surfaces that are rotated relative to each other. The input filters do have an effect, but as expected for reflections off metal, not as significant as with the sequence of dielectric interfaces.

Test Case	n_1	k_1	n_2	k_2	δ	ρ	ϕ	filter
1	1.330	0.000	1.500	0.000	53.0	0.0	56.0	none
2	1.330	0.000	1.500	0.000	53.0	0.0	56.0	horizontal (0°)
3	1.330	0.000	1.500	0.000	53.0	0.0	56.0	vertical (90°)
4	0.666	0.000	0.666	0.000	48.0	0.0	54.6	horizontal (0°)
5	0.666	0.000	0.666	0.000	48.0	0.0	54.6	vertical (90°)
6	0.666	0.000	0.666	0.000	48.0	0.0	54.6	diagonal (45°)
7	1.120	2.160	0.608	2.120	48.0	34.0	20.0	none
8	1.120	2.160	0.608	2.120	48.0	34.0	20.0	horizontal (0°)
9	1.120	2.160	0.608	2.120	48.0	34.0	20.0	vertical (90°)

5.2 Results

Test Case	S_0	S_1	S_2	S_3
1	0.560409	0.560409	0.0	0.0
2	0.560409	0.560409	0.0	0.0
3	$1.2E-10$	$1.2E-10$	0.0	0.0
4	50.0	50.0	0.0	0.0
5	50.0	-50.0	0.0	0.0
6	50.0	0.0	0.408533	-49.9983
7	33.9959	9.01647	-0.75875	-0.507946
8	19.1245	10.9873	-13.0076	-8.70795
9	14.8714	-1.97081	12.2488	8.2