Smectic Film Thickness Estimator (SFTE) Code Usage and Principles

Keith G Hedlund

April 6th, 2020

University of Colorado Boulder Department of Physics

Soft Materials Research Center

**Principles**

The Smectic Film Thickness Estimator (SFTE) analyzes an image of a smectic film and predicts its thickness by comparing the RGB coordinates provided by the camera with a theoretical model. This code was based off a paper published in September of 2020 by Hiroshi Yokoyama and Wei Chen of Kent State University, titled Rapid Thickness Mapping of Free-Standing Smectic Films Using Colour Information of Reflected Light1. The paper lays out a method for rapidly determining the thickness of a smectic film based off the normalized color information (more about this later). The equations and principles are detailed quite rigorously in this paper, so we’ll largely skip over them here; what is important to know is that the image is naturally split by the camera into red, green, and blue channels. The camera splits a specific wavelength of light into three color channels (red, green, and blue), and each pixel on the screen is assigned its own triplet of these coordinates based on the color of that pixel. The function that the camera uses to split this singular wavelength into three color channels depends entirely on the brand and type of camera and is not at all easy to determine. To get around this, SFTE takes the color channel triplet of a single pixel and compares it to an image taken using the same camera of a piece of black glass, which has a uniform reflectivity of 4% across the visible spectrum. This allows the RGB values from the image of the film to be translated directly into reflectivities of light with a given wavelength, which will be discussed more below. These reflectivities are then compared to an ideal model for the film, which shows how the reflectivity for each color varies with the thickness of the film. This removes entirely the need to determine the camera’s mapping function and allows this method to be applied to any camera provided the appropriate black glass image and color normalization protocols are observed (see below). Instructions for making the plot of normalized reflectivities (Fig. 1b) are available in Hiroshi’s paper. The intersections of the theoretical plot with the measured values of reflectivity are then plotted, and a function finds the set of red, green, and blue points which agree best on thickness; for Fig. 3, this would be the second set of RGB values, as they are the closest to being vertically colinear. This value of thickness is then reported by the program to the user.

Fig. 1: Plots of the predicted RGB values (top) and the normalized RGB values (bottom) as functions of thickness in nanometers.

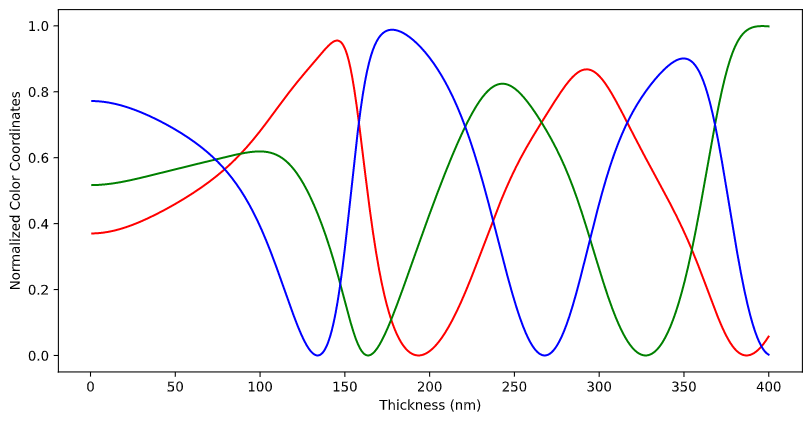
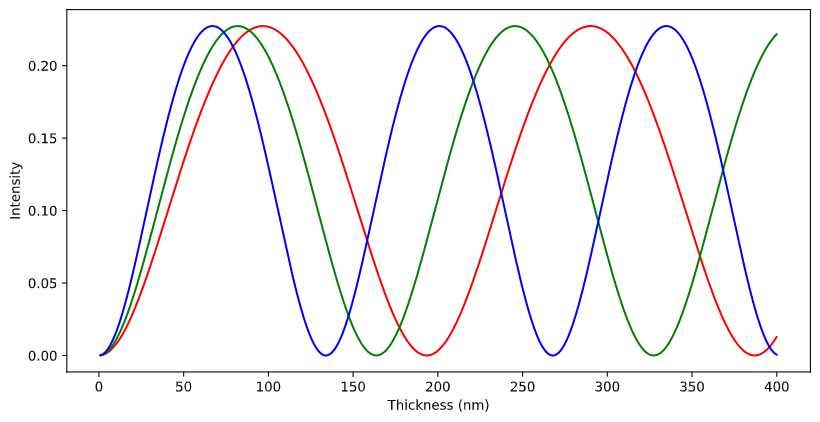
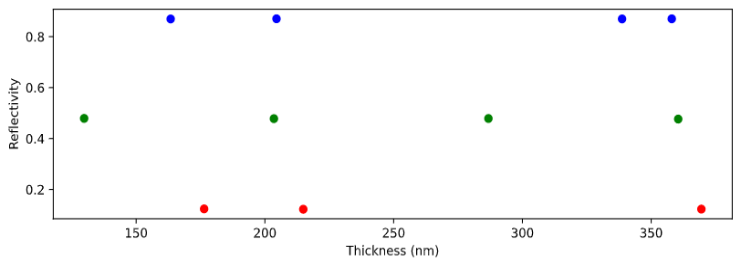


Fig. 2: An image of a piece of black glass. The dark patches are caused by small pits and scratches in the surface which are difficult to avoid; the color information should be read within an area that doesn’t include one of these dark patches.

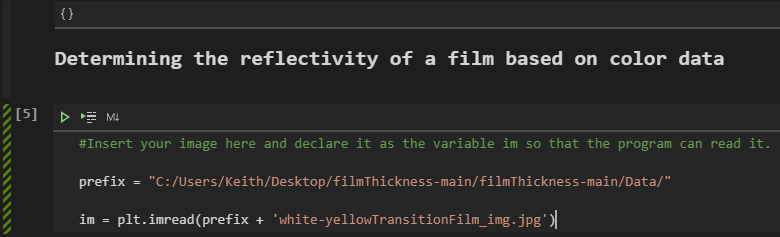
Several assumptions were made in converting Hiroshi’s approach to suit this project, the largest of which being the wavelengths of light which characterize the red, green, and blue channels. These values are initiated in the header of the program and are labelled Lr, Lg, and Lb, respectively. These values are integral in producing the color plots seen above, and physically represent the wavelength to which each color channel of the camera has the highest sensitivity. These values were chosen based on non-camera specific data and should be improved based on the camera being used in a specific experiment. To do this, images of several shades of red should be taken with camera and compared with spectrograph data of the same shades to find which wavelength of light produces the highest pure-red channel response, and likewise for green and blue.

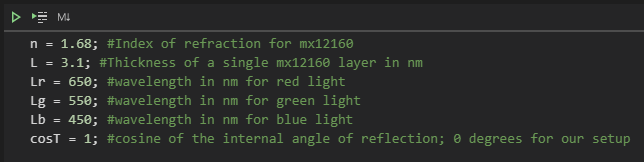
Fig. 3: A plot showing the intersections between the normalized color plot in Fig. 1 and given RGB values extracted from an image of a film.

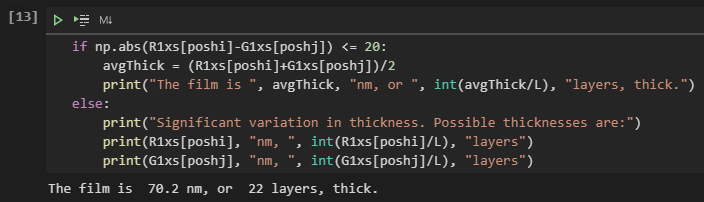


It was also assumed that the index of refraction of the material stays constant as the temperature increases, and that the light passing through the film is non-polarized. If the light passing through the film were polarized, the birefringence of the material would also need to be considered. The error caused by these two assumptions is, in most cases, negligible. However, if the material being used shows large changes in refractive index as the temperature changes or if polarized light is being used, SFTE may introduce much larger errors into the thickness measurement.

**Usage**

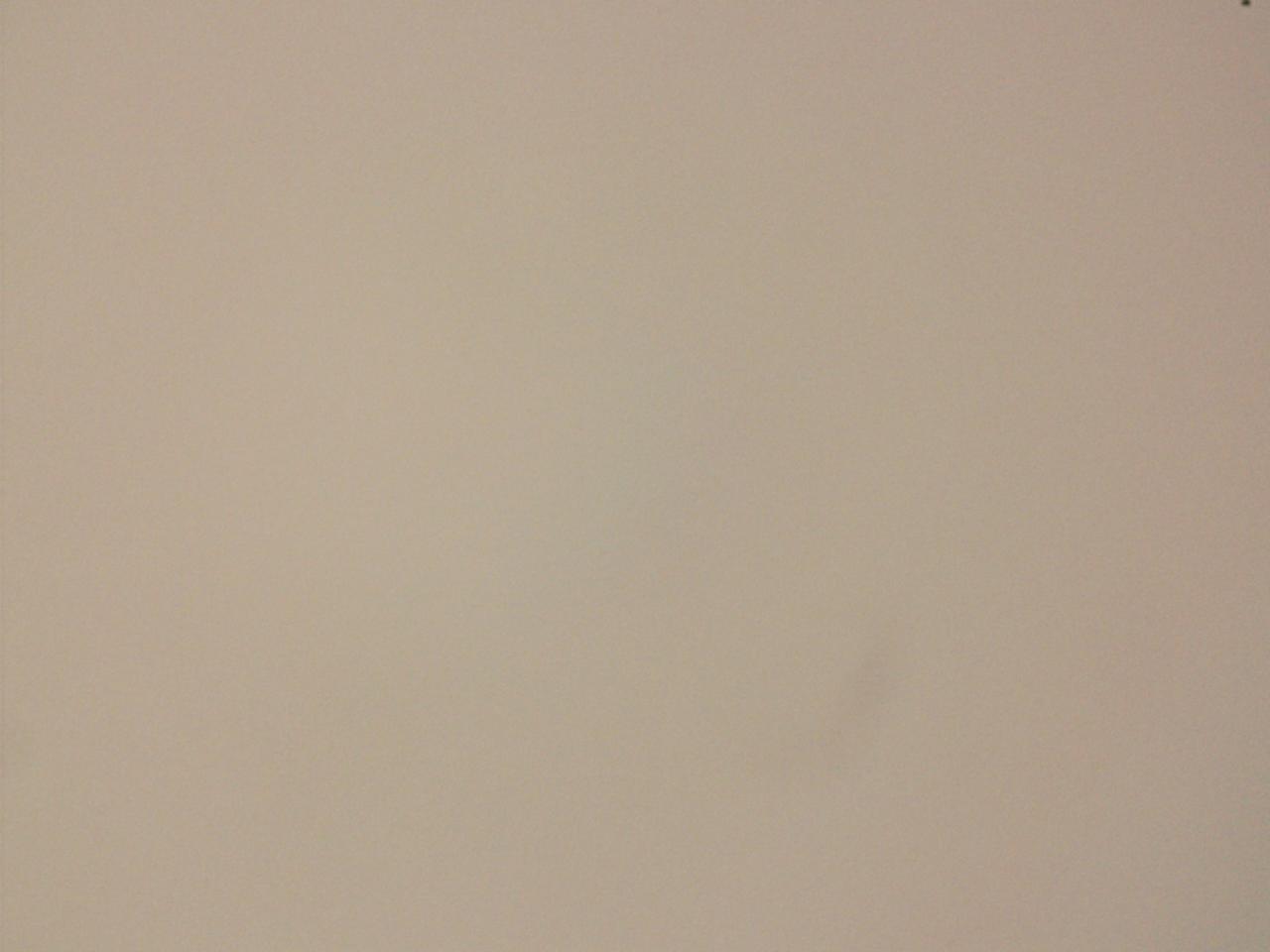
 The code for SFTE is annotated in many places with instructions on how to use it, but this assumes that the coding environment is already set up. The program itself is a Jupyter notebook, so at a bare minimum Anaconda and Jupyter must both be installed; the code itself was written entirely in VSCode, so this is the recommended environment for running and modifying the program. The required packages for the program are numpy, matplotlib, matplotlib.pyplot, math, pandas, colour, and colour.plotting; ensure that all of these are installed in the python coding environment before running the program. Once the environment is properly set up, the variable titled “prefix” should be changed to match the data path of the folder which the image is located in. Right below the “prefix” variable, there will be a variable “im”; this is the variable which represents the image throughout the code, so the part inside the apostrophes should be changed to match the name of the image file.

 After specifying the image to be analyzed, a black glass reference image must also be specified. Scroll down to a little bit below where the image to be analyzed was added, there will be a cell that initializes a variable called “blackGlass” in the same manner as the previous image. Replace the section in apostrophes with the name for the appropriate black glass image. By default, this black glass image will have the same data path prefix as the image being tested, so it’s important for the two images to be in the same folder. Make sure the black glass image being used was taken under identical lighting and microscopy conditions to the image being analyzed, otherwise the color mapping will not work! Once this image is specified, check to make sure all the material properties listed at the top of the program are correct for the liquid crystal being used; by default, they are the material properties for 8CB. If the material properties are not correct, they will need to be modified. The code can now be run, and it will (most likely) produce an accurate estimate for the thickness of the film, visible at the bottom of the code below cell 13.

 If the estimate produced is obviously not accurate, responds with “significant variation in thickness”, or returns an error, the range of possible film thickness for the theoretical plot may need to be increased or decreased; to do this, change the variable “rang” at the top of the program (cell 1) to a greater or lower value. This variable is found at the very top of the code and is the range in nanometers of possible thicknesses for the film. The general thickness of a film can often be determined when working with the film in the lab; islands and droplets on a thinner film will generally move faster, thicker films usually require more material when being drawn, and it will be easier to see layer steps on thinner films especially under 30 layers. These characteristics as well as a general knowledge of the colors that films go through as thickness increases, shown below, is important for determining if the thickness reported by the program is accurate.

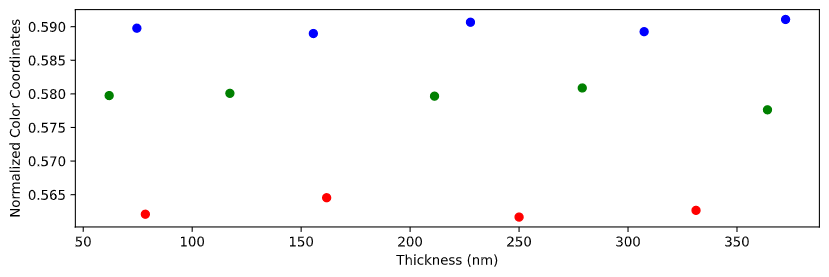
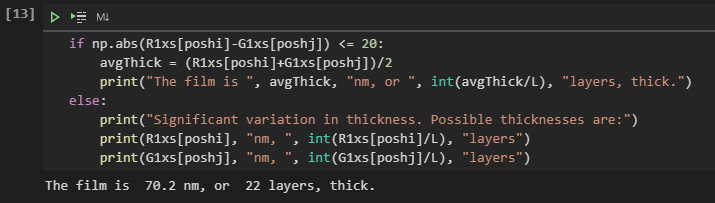


A qualitative plot of color versus thickness for smectic films; the actual film thicknesses which cause these colors depends intimately on the refractive index for the material, hence why it hasn’t been labelled on this image. For 8CB, the first major color shift from gray to yellow occurs at approximately 70nm.

Here is an example of a film image, black glass image, and some of the relevant outputs of the program:

The black glass reference image being used

The film being tested



The reported thickness for this film is 70.2 nm, or 22 layers. This makes sense, as this film is right on the cusp between colorless and yellow which is known to happen around 20 layers.

The plot of intersecting normalized color coordinate values for each channel as a function of thickness.