Weekly Report - Friday, Nov. 10

Grant Saggars

1 Instrument Measurements

I gathered instrument measurements for my diffraction grating this week so that I can design mounts in CAD. I need to disassemble my webcam so that I can also gather measurements for it.

- Short side: 1.41 cm ± 0.05 cm
- Long side: 2.99 cm ± 0.05 cm

2 Design

I have laid out design goals, as this weekend I plan to develop models of the instrument housing.

- I would like the design to be modular: it should be easy to make adjustments to the internal orientation of my components should the need arise. To do this, I will first create mounts that can be installed into place. The end cap may need to fit a collimating tube as well, so I plan to create a threaded end-cap for a pinhole first to determine if the collimator is necessary.

3 Software

I made the most progress in software this week, and I figured it would be good to describe the program:

3.1 Outline & Challenges

A spectrometer produces a plot of intensity vs wavelength, which I will analyze to estimate blackbody temperature. Because intensity can be considered to be the value of a pixel in an image, this plot can be found with my device by taking a line across the image of the diffracted spectra and plotting the value channel for each pixel in that line (after conversion to hsv space).

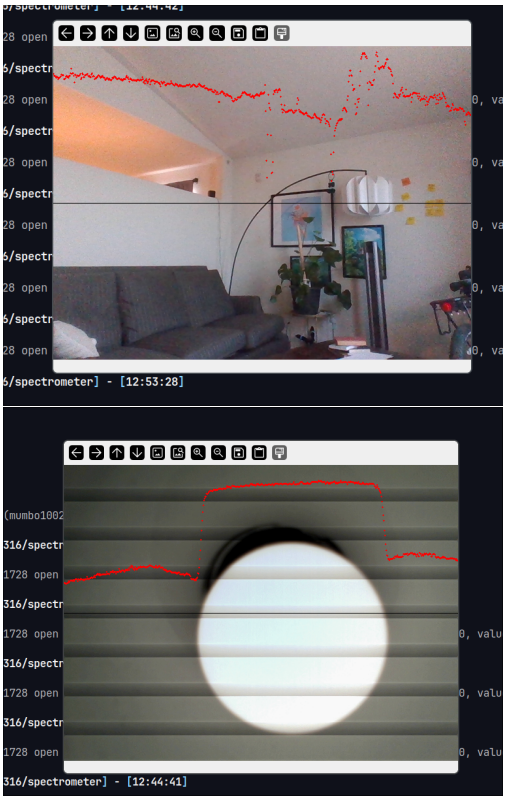
Because I would also like this to be done in real time from a live video feed, an obstacle (and the largest this week) has been speed. While the experiment can be done with a low framerate or even just from a single image, a high polling rate adds polish to the device.

3.2 Implementation

My implementation of the software is handled mostly by OpenCV, a library of functions normally intended for real time computer vision. It provides a means to handle several key aspects of the software: creating a stream for the webcam video, color space conversions, and it provides a solution to the performance issue I faced with matplotlib/pyplot.

OpenCV utilizes the GPU, so the processing of my data is fast enough to be done in real-time. Therefore it is easy to process my image and create a 2D bitmap of pixel values from which analysis can be done. Is it typical for a spectrometer to render a plot of wavelength vs intensity, and this should be easy in principle. The standard libraries for plotting can be made to work with OpenCV, however they are generally a performance bottleneck.

My first implementation used OpenCV to process the frames from the webcam, and matplotlib/pyplot to produce a plot. Performance went from a full 30 frames per second (the framerate of the webcam) to one frame every few seconds because of matplotlib. It was even slower when I tried drawing the image alongside the plot. Initial attempts at fixing performance involved trying to manipulate OpenCV into producing a plot, which I quickly found to be unsuccessful. I solved this performance problem by dropping matplotlib entirely, and instead opting to have OpenCV produce a plot using its ability to draw primitive shapes (currently I am using dots) over the frame. This worked beautifully; the plot is now blazingly fast, and I will continue to develop the visuals throughout the next few weeks.



4 Timeline

I am progressing faster with software than I had anticipated, but a bit slower with construction. I had originally planned to finish CAD this week, and have a prototype of the 3D prints ready. This will have to be done at the start of next week. Software may be (at this rate) completed two weeks faster than I had anticipated.

5 Feedback

My lab group meets on tuesdays, so there was no feedback to be given this week because we had just started. I did, however, learn how to calibrate my sensor with the lab samples from my TA.