**Monday**

Mapping the dark Aurora lights using MATLAB

Today was mainly spent on me trying to figure out how to send the info from the final picture to a file and then loop through that giving the illusion of video moving with all the dark points highlighted on it.



(Trying to auto save these parts of the code and loop later)

A few problems are around with he time and the fact the longer it goes the slower the code seems to run. Also, there are to man frames for a 20 minute video. One second of the video produced almost 40 frames which also creates a storage problem and an even greater time problem. The goal is to try to get the frames to about 1200 and map all of those.

**Tuesday**

Mapping the dark Aurora lights using MATLAB

**Wednesdays**

Mapping the dark Aurora lights using MATLAB

July 4th was a lovely day of it was down in the capital

Recursive Particle Tracking

**Thursday**

Mapping the dark Aurora lights using MATLAB

**Friday**

Mapping the dark Aurora lights using MATLAB

Surprisingly I have had a lot of difficulty with this last finishing part. The code is able to take a video and crat frames of that video then code pretty good at determining the darkest points and finding the dark patches. The only problem is still using the figures or the finishing image to so I can stitching them together. I tried saving as a jpg, png and a fig itself but its weird because I not 100% sure where that part of the code is actually referenced.

MATLAB has some protocols that allows it to just assume some actions, which is where the confusion can come in at some time. It’s different from Java or C++ which require you to define any all information used in order to do a request. This is slightly easier because you can definitively find and recall where everything is. For example an array in MATLAB will assume the type based on the first element which is quicker but requires more memory on my part where as in Java you initialize the array first then use it accordingly.

