Midterm Two: Project Report

Joaquin Stawsky

March 12, 2022

1 Introduction

In this project, we were assigned to choose an extraterrestrial target and fully process its raw data taken from the Charge Coupled Device (CCD) to produce a colorful image. To produce the best image possible we must take both the floating and static biases, and the pixel-to-pixel differences into account. The data used was taken from the 1.0 meter Telescope at the Mount Laguna Observatory (MLO).

2 Background

2.1 Important Terminology

- Charge Coupled Device: A CCD is an electronic detector that is used to measure a relative number of photons during some exposure, to be viewed at optical bands. It identifies when, and where, a photon hits one of its pixels and stores the data as an photo-electron. To get the overall digital count of photons hitting the pixel, we use an Analog-Digital-Converter (ADC) to convert the signal received into a digital number, an Analog-Digital-Unit (ADU).
- Band-Pass Filter: A band-pass filter only allows photons travelling at a certain frequency/wavelength pass through to the detector. The bands that we will use are B, V, and R.

2.2 Understanding the Math

There is one central equation present behind this entire process that must be understood to get the desired image. For each individual pixel:

$$C = \frac{(B_f + B_s + \epsilon N_\gamma)}{g}$$

where...

- C is the count detected (in ADU)
- B_f is the time-dependent, floating bias
- B_s is the time-independent, static bias
  - Both types of bias are measured in e^-
- ϵ is the photon conversion efficiency
  - The actual ϵ is too difficult to compute so we use the relative efficiency as a proportion to another star.
- N_γ is the number of incident photons
- g is known as the gain
  - measured in e^-/ADU

Although we will not directly address the photon conversion efficiency (ϵ), the number of incident photons (N_γ), or the gain (g), they are still being calculated and used in this process.

3 The Process

3.1 Set the Directory Path to Raw Data

In this step, we define the base directory that the raw data is in, the instrument used, the date the observations were made, and the specific (.fit) files that hold the data for the target. To find which files correspond to the target chosen, one must got to the 'log.txt' file and identify the file name corresponding to each band-pass filter. The target chosen is the Wolf-Rayet Star (WR 3), which was observed at the Mount Laguna Observatory's 1.0 meter telescope on September 28, 2017.

3.2 Removing Overscan

When referring to an image's overscan, one is addressing the extra columns of data that are read from the pixels once the shutter is closed. The importance of this region lies in measuring the image's floating bias, which is time dependent, to remove it from the image. To make this happen we define a function to take a two-dimensional array containing the raw data and split the active region from the overscan region. The raw image has the shape (2048, 2200) but since we want the active region to be square, (2048, 2048), the dimensions of the overscan region must be (2048, 152) since Column Number $= 2200 - 2048 = 152$. We then find the average value in the overscan region and subtract it from the active region's values to get a trimmed image without a floating bias.
Additionally, in this step we identify all the static bias image paths and create a datacube to contain them. To start, we append the 21 static bias image paths (25 images is ideal) to a list. Then we iterate through the list, using each path to get the data through the astropy.io.fits package, remove each image's overscan using the function defined before, and add each bias image to a three-dimensional datacube by means of vertical stacking. The resulting datacube is of the shape (21, 2048, 2048).

3.3 Creating the Master Bias Calibration Image

Once the datacube of the static bias images is created, we must find the typical value within each pixel, often known as the median, to create the master bias. An important detail to note is that the axis in which one computes the median must be that in which the files are stored into the data structure. For example, since the datacube created is of the shape (21, 2048, 2048), the axis must equal zero so that the master bias' shape is of the form (2048, 2048). Finally, we show the image using a predefined function that displays the image in grey-scale and specifies the range of data that should be viewed. An example of a master bias image is Figure 1.

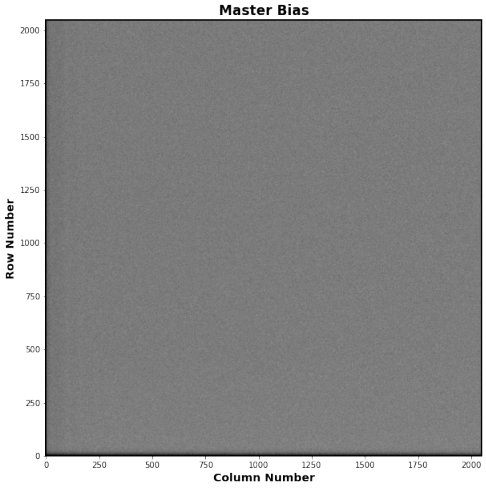

Figure 1: This Master Bias image might look plain grey, but it contains extremely important data to properly visualize one's target object.

3.4 Creating the Master Flat Field Calibration Image

By creating the master flat field image we address the innate differences from pixel-to-pixel. The process begins the same way as in the creation of the master bias image: we must create a list of all the flat field image paths for each filter. In the original directory, they are named twilight images because the uniformity of the sky at twilight is exactly what is needed to account for anomalies in the optical path. A simple example of such anomalies are specks of dust on the filters or the camera lens. After identifying all the paths, we create a function that first gets the data for each flat field image path, then removes the overscan region, subtracts the master bias image from each flat field, divides it by the mean so that the median is close to the 1.0, creates three total three-dimensional data structures containing all the data for each filter, and finally find its median across the zeroth axis to get the typical values for each band-pass. Examples of the final three flat field images are displayed in Figures 2, 3, and 4. This step is essential because if not conducted, the scuffs and circles seen in the flat images would distort the final image of WR 3.

3.5 Fully Processing Images

In this step, we finally start using the data of the star we would like to observe. Using a function that takes the bias image, the band's flat image, and the target object's path (specific to the band-pass filter) as input, we are able to get the data from the target's path and remove the overscan to finally subtract the bias and flat image from them and divide by the mean of the image that had the bias and flat subtracted. Completing this process three times, for each band-pass filter, results in Figures 5, 6, and 7. Although the images may look very similar, they have slight differences that can only be detected with each filter. Combined, these images will produce a colorful display of the WR 3 star.

3.6 Combining the 3 Images

We must first scale the values of the pixels to be within 0 to 255, these scaled value are called 8-bit unsigned integers. This formatting change allows for the matplotlib.pyplot.imshow method to read the data in colors (R, G, B); the respective band-pass filters' colors are R, V, and B. In order to scale the values we must find the appropriate cmin and cmax values for each image to define the data range allowed in the image output rather than the entire data range being used. We use the sigma_clip class from astropy.stats to find a sample for each of the data groups. Then we find the mean and standard deviation for each of the filters. Finally, cmin is calculated by subtracting a factor of the standard deviation from the mean, and cmax is calculated by adding another factor of the standard deviation to the mean. Now we need to clarify the shape and create the new image data structure to be (2048, 2048, 3), which is representative of the amount of rows by the amount of columns by the number of band-pass filters. The final step entails finding the last pair of vmin and vmax values using the mean and standard deviation of the final image data to plot the pixel values. In color, Figure 8 shows the observed target, Wolf-Rayet Star 3, as the brightest object in that portion of the sky.

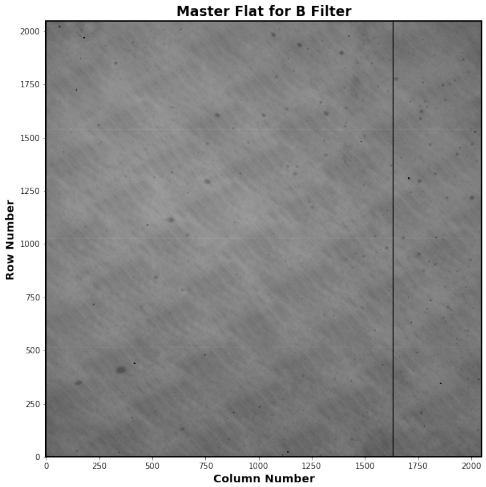

Figure 2: This master flat field image for the B filter shows prominent scuff marks.

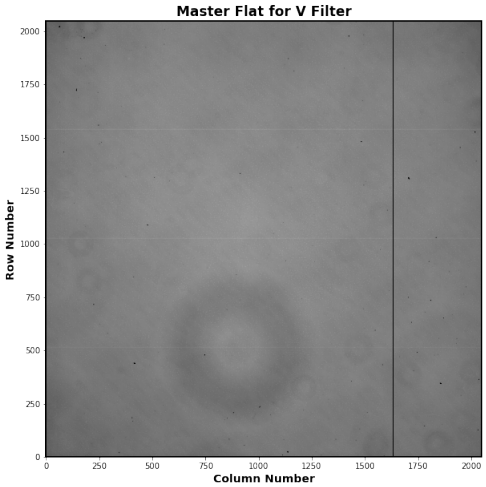

Figure 3: This master flat field image for the V filter shows a very large circle in the foreground, bottom half and other scuffed aspects as well in the back.

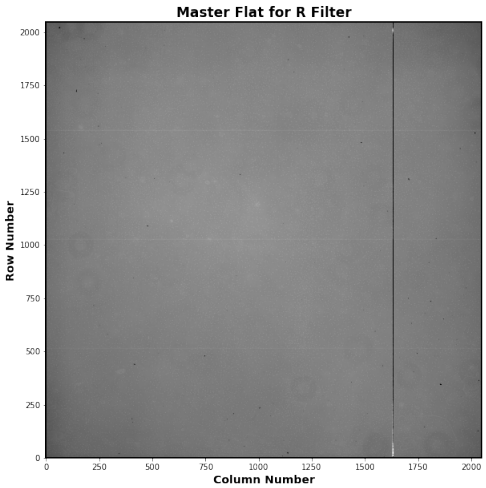

Figure 4: This master flat field image for the R filter shows many small circles in the background.

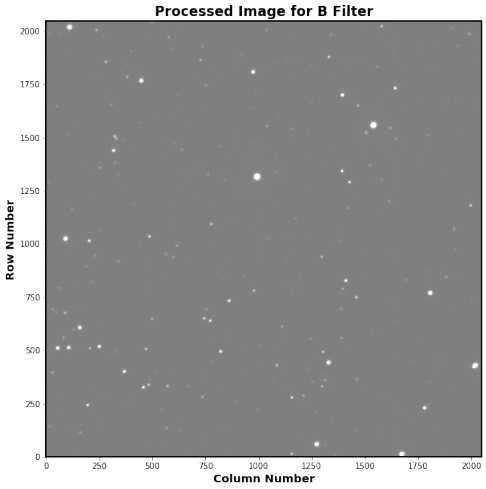

Figure 5: This is the grey-scale version of the processed final image for the B filter.

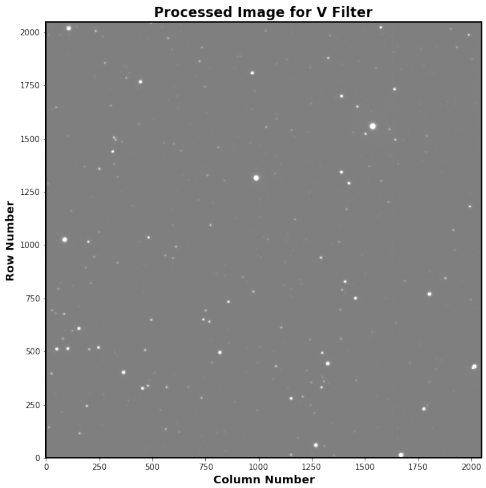

Figure 6: This is the grey-scale version of the processed final image for the V filter.

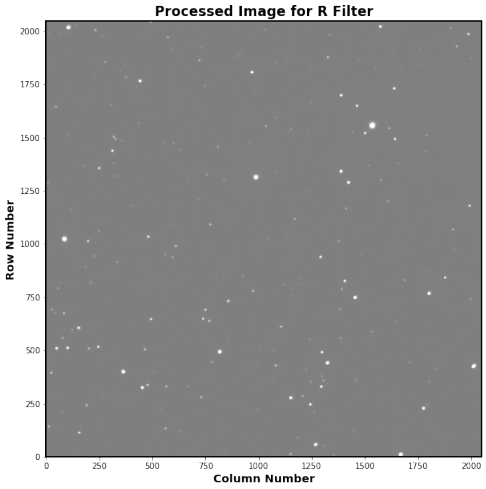

Figure 7: This is the grey-scale version of the processed final image for the R filter.

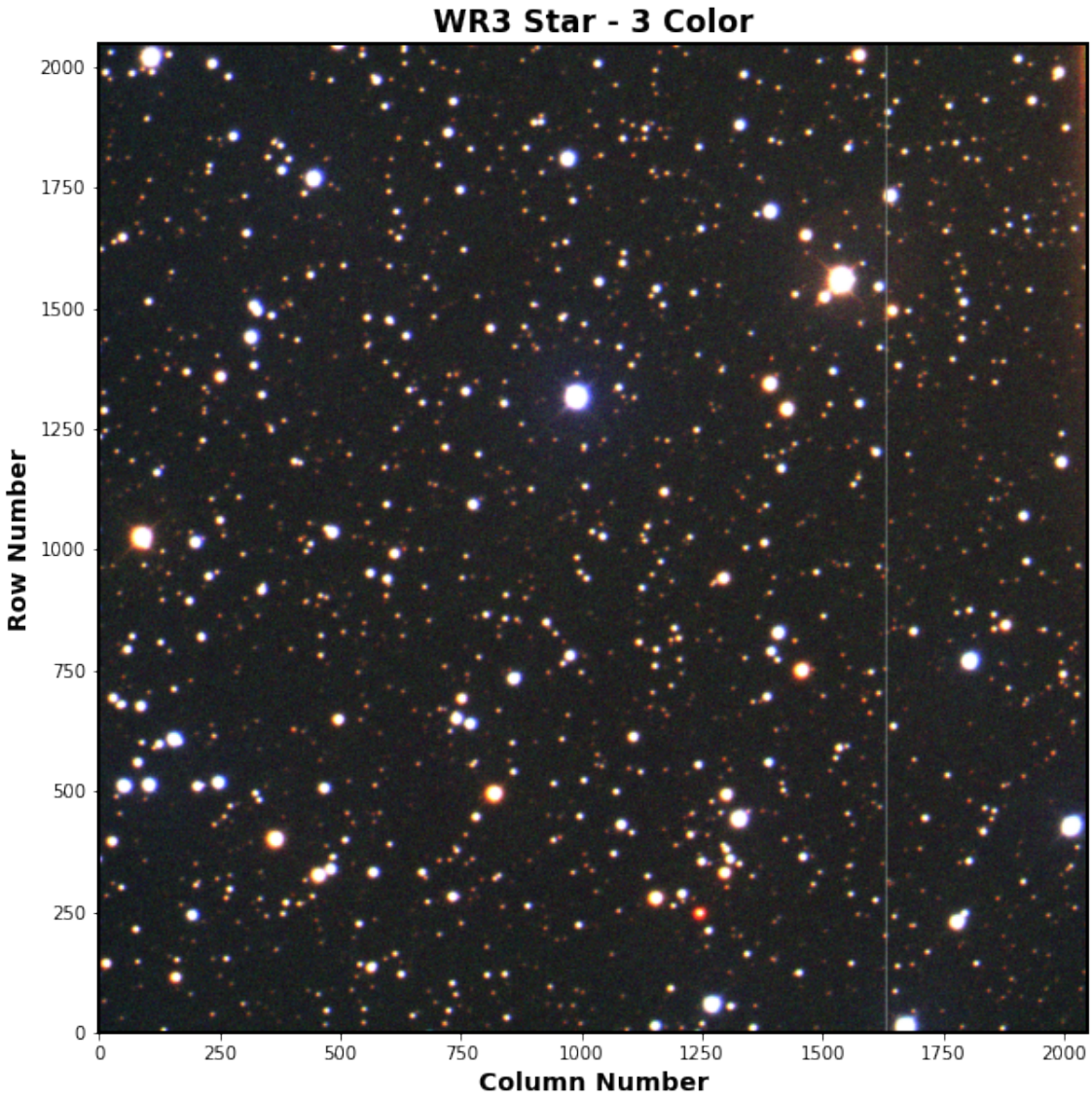

Figure 8: This image has little-to-no static bias, floating bias, and pixel-to-pixel variations.