Detecting and Measuring Light Using the Photoelectric Effect
1/30/2022
Mikaela Larkin
mmlarkin@ucsd.edu
Lab group B

Abstract

The photoelectric effect, in which photons are absorbed into a material and eject an electron in their place, is a powerful tool in the detection of light from astronomical objects. In this lab we examine the charge-coupled device on the Direct Imaging Camera in the Nickel 1-meter Telescope at the Lick Observatory, which measures the physical phenomenon numerically. Our ultimate goal of this lab is to measure the gain and read noise of the device, which describe the performance and systematic error of the device. In order to reach this result, we analyze data collected pointing the camera at a flat screen to detect its intrinsic properties. We manipulate this data to investigate the statistics of the light compared to Poisson statistics, the effect of number of exposures on standard deviation, and histograms compared to Gaussian and Poisson probability distributions. Our final results for gain indicates that the detector has a significant loss of signal in its circuit before reaching the amplifier, but a low amount of read noise indicating that transmissions may be faint, but are accurate as to what has been detected.

Introduction

The photoelectric effect may be numerically measured by charge-coupled devices (CCDs), a common technology used in telescopes that transport converted photoelectrons to an amplifier which enumerates them according to their original pixel on the device for analysis. In this lab, we explore the precision by which the Direct Imaging Camera in the Nickel 1-meter Telescope at the Lick Observatory uses CCD technology to detect light by testing statistical probability distributions with data collected across several exposure times. The ultimate goal of this lab is to calculate the gain and read noise of the device. The gain is the number of electrons collected per analog-to-digital units (ADU), which should be a constant representing the factor between number of incident photons and number of ADU counted. The read noise is defined as the statistical error in the total number of electrons measured by the circuit.

In a broad sense, the methods used in this lab take large data sets collected on the telescope and test their information against statistical theories. In particular, we will calculate the mean, standard deviation, variance, the mean of means, and the standard deviation of the means for a collection of 23 data sets collected at varying exposure times. The data sets were collected with the telescope pointed at a blank screen in order to specifically examine the abilities of the device itself. The calculated values were then compared to the predictions of Poisson and Gaussian statistics, which approximate counting experiments and large sample sizes, respectively. Our final results for gain and read noise provide insight as to how well the telescope is performing and provide insight in order to maximize data accuracy.

Observations and Data

The Nickel 1-meter telescope is located in the oldest dome of the Lick Observatory in Northern California. The telescope is capable of being accessed and controlled remotely by campuses in the UC system, including UC San Diego. In order to explore and test its capabilities, we accessed the Direct Imaging Camera on the telescope, which uses charge-coupled device (CCD) technology to collect images in various filters. CCDs take advantage of the crystalline structure and nature of silicon as a semiconductor to evaluate incoming light. Silicon consists of an upper conductive band and a lower valence band with an energy gap in between, which is defining for semiconductors. Photons reach the detector's valence band and are converted into electrons through the photoelectric effect, which may be detected by the CCD. These electrons are excited to the conductive band of the silicon, moving freely through the crystalline structure and leaving behind a positive charge in the valence band. Before the electrons are able to recombine with the positive charges, the CCD introduces an electric field to separate the particles.

When an image is captured by the CCD, electrodes transfer each column of electric charge to a serial register, which then transports the charges to an amplifier. These images include an "overscan" region of 32 additional columns that documents pixels that do not physically exist in the CCD as a means of measuring the bias in each image. These overscan regions contain hot pixels, which have a high dark current, meaning there are electron holes created by means other than the photoelectric effect that must be subtracted from the data that is to be analyzed. The final output from the amplifier stores the original image pattern (**Figure 1**) in a Flexible Image Transport System (FITS) file. The FITS file contains all information about the image's collection in binary format with an ASCII aspect, describing details such as exposure time and celestial coordinates.

In order to inspect the Direct Imaging Camera's use in this lab, we collected data on January 6, 2022 in the visible light wavelength spectrum. The objective was to collect several bias frames in which there is no exposure time, only the intrinsic bias in the device, and flat dome frames in which white light is pointed at a blank white screen on the telescope's dome for several different amounts of exposure time. The windows visible in remote observing included a live feed of the telescope and dome, guidance for angling the detector, current weather conditions surrounding the telescope with an all-sky camera feed, and software controls for the Direct Imaging Camera. Before collecting any data, the settings on the camera had to be properly calibrated for our usage. The calibration included setting "Binning" to 2, meaning we are combining pixels on the 2048x2048 detector to 1024x1024 in order to capture cleaner images. We then set the read speed to slow to minimize the amount of noise, set the camera to record, and named the recordings as "Bias" to begin since the first data we collected were bias frames. Before our first image was captured, the physical settings of the observatory needed to be changed. First, the filter had to be set to B (blue) in order to rotate the built in filter wheel. Second, the aperture needed to be set to 0 in order to open the camera.

The first four images we captured were bias frames, which are images captured with no exposure time in order to only measure the CCD's intrinsic signal. To ensure the images are captured properly, we see in real time the images being collected by the detector. To prepare for capturing flat field images, we orient the dome to have the blank screen designed for this specific purpose in view using the telescope guidance window by setting an azimuth of 218 degrees. We then point the telescope itself to zero hour angle to align it with the flat screen. We then removed the cover from the mirror and turned on the flat field lamp. Finally, we opened the shutter, renamed the object to “Flat Field,” and captured 2-3 images each for exposure time 3, 6, 12, 48, 64, 96, 192, 384, and 768 seconds.

The main sources of potential error in observation derive from the systematic operation of the CCD. The first major source of error to be accounted for is bias, which is signal in the detector that exists regardless of illumination and exposure time. We try to account for this error by subtracting the data from bias frames from the flat field frames, but this still may provide error if the bias frames are unable to capture all necessary anomalies in the pixels. Some amount of bias signal tends to be artificially added to the detector in order to prevent negative input as may occur due to the read noise. Read noise is intrinsic error associated with the voltage of the circuit measured, and it is critically important to understanding the functionality of the device since alters the amount of electrons being detected by a constant amount. Electron hole pairs may be created in the CCD by thermal fluctuations and quantum tunneling in addition to those created by the photoelectric effect, producing an error called dark current. Dark current therefore contributes to Poisson noise, meaning they both have to be subtracted from the final result similar to the subtraction of bias. There are several ways these errors become apparent in images collected, such as with hot pixels which are points of high dark current which need to be avoided in data analysis for accurate results. There were also a few bright columns in the detector images, which are areas in which electric charge has become trapped and new data can not be obtained.

Data Reduction and Methods

The 28 FITS files collected were analyzed using Python on JupyterHub to inspect their data, inspect their error, and define the amount of noise in the images. The final image was determined to be oversaturated, which will be further explained in the Calculations and Modeling section, and was therefore removed from the array. This left behind 4 bias frames with no exposure time, and 23 flat dome frames with varying exposure times. The target to be analyzed were the flat dome files, which were prepared by subtracting the mean of bias frame data points from each file, and by narrowing the data to a region excluding overscan pixels. At this point, the data has been prepared for statistical analysis pointing towards a result of gain and read noise.

In order to analyze the data and approach the final desired results, the data first had to be flattened and parsed into an array of bias frames and flat dome frames. The mean, standard deviation, and variance of the values were then calculated in order to compare mean and variance, which should theoretically be equivalent in Poisson statistics. Drawing this comparison illustrates the amount of error in Poisson counting experiments. The collected data points were then converted from ADU to ADU per second to equalize their values across the different exposure times and compared using the mean of means and standard deviation of the mean. These values were then plotted across different numbers of exposures in order to illustrate how many exposures are necessary to get an accurate result. The next comparison was of two randomly selected histograms to both the Poisson distribution and the Gaussian distribution in order to illustrate how well these probability functions approximate the discrete data points collected. The final step was to generate a plot of the mean of values in ADU against the variance in ADU^2 with a best fit line to approximate it. The inverse of the slope of this line gives a numerical value for the gain in electrons per ADU and the intercept of the line gives the read noise of the detector. Each of these steps will be described in greater depth in the following section, Calculations and Modeling.

Calculations and Modeling

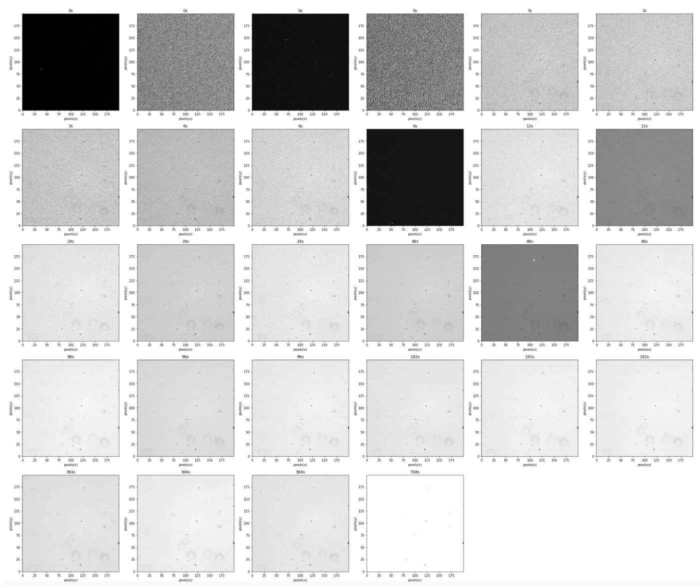

Figure 1:
Original images from CCD detector, zoomed-in to clean regions without overscan. The last image in the bottom right corner is blank

In order to read the data into Python, a loop was created to iterate over each data file and create a single array with 28 subsections containing information about the amount of analog-to-digital units (ADU) in each pixel. Using the ASCII preamble to each FITS file, the images were shown (**Figure 1**) to visualize their different properties. A separate array was also created with the corresponding exposure time for each data set. Upon inspection of the data sets, the final frame with a corresponding exposure time of 784 seconds appears to have been oversaturated, meaning it has reached the upper limit of photons able to be detected. The oversaturation was confirmed by checking the region of the image without the overscan data and finding a maximum value of 65535 ADU, which is much higher than the maximum values of the other images. The oversaturated data file was therefore removed from the array to be analyzed. The remaining files were then split into an array of bias frames and an array of flat dome frames. The bias frames

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad \textbf{(1)}$$

$$\sigma_x = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2} \qquad \textbf{(2)}$$

$$\sigma_x^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2 \qquad \textbf{(3)}$$

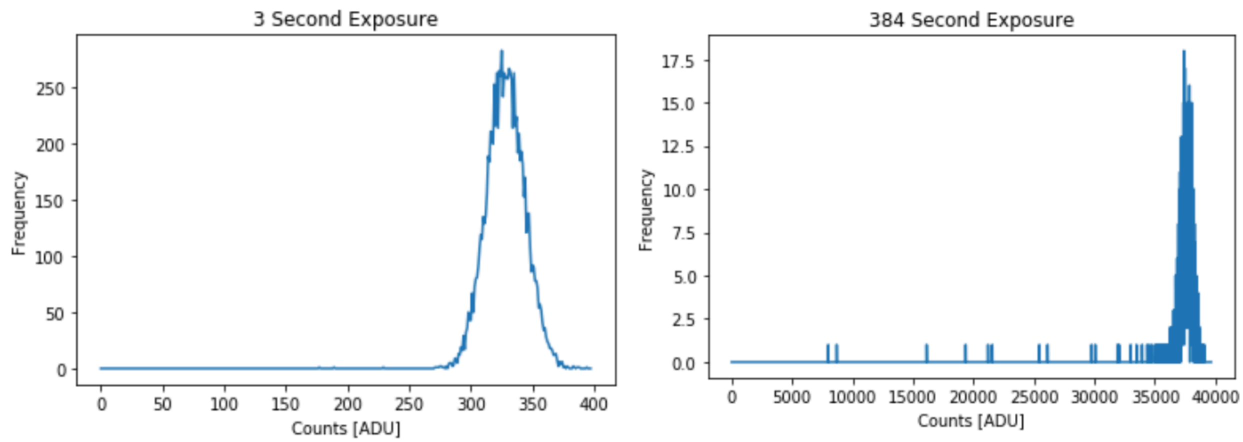

Figure 2: Two sample histograms for images taken at 3 seconds of exposure time (left) and 384 seconds of exposure time (right)

were confirmed to be the first four files by printing the exposure time from each data set's ASCII preamble and checking which ones were zero. The mean of the four bias frames was subtracted from each of the flat dome frames in order to remove the intrinsic bias from the scientific target datasets.

The first step in analyzing the data was to flatten the data into one dimension and to calculate the mean (**Equation 1**) and standard deviation (**Equation 2**) of each data set using self-made functions in Python, the code for which may be found in the Appendix. These functions were also used to further define the variance (**Equation 3**) of each data set, which is the square of the standard deviation. In these three equations, N refers to the total number of data points in the file and x_i refers to the index of data points in each file. Histograms were then plotted for each exposure time, two of which are shown in **Figure 2,** which plot the distribution of data points in ADU against the frequency at which they occur. The mean and variance of each value for each exposure time was also calculated along the way. This information was used to plot the mean vs. variance in the counts of each data set (**Figure 3**) in order to compare to the Poisson statistics theory for counting photons in which these values should equivalent. The plot was done on a logarithmic scale to best examine the difference between the observational experiment and Poisson theory, which will be explored further in the Discussion section.

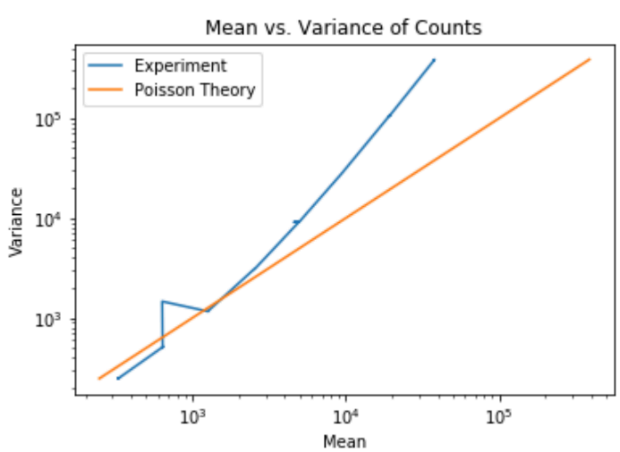

Figure 3: Comparison between mean vs. variance for collected data (blue) line) and Poisson statistics.

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} \bar{x}_i \qquad \textbf{(4)}$$

$$\sigma_{\bar{x}} = \frac{1}{N} \sqrt{\sum_{i=1}^{N} \sigma_i^2} \qquad \textbf{(5)}$$

Before further analysis of the data points, we converted their units from total ADU to ADU per second by dividing by their corresponding exposure time. The conversion allows for comparison between the different exposure time on an equalized measurement scale. The mean of means (**Equation 4**) and standard deviation of the means (**Equation 5**) were calculated for each data set and placed into an array for analysis. The variables used in these equations are the similar to those for **Equations (1)** - (**3**); N is total number of data points, $\bar{x}_i$ is indexed mean values, and σ_i is indexed standard deviation values. The arrays of these values were then plotted against the various, discrete numbers of frames (**Figure 4**) to illustrate the property by which and increased number of samples minimizes the standard deviation from the mean. The plot indicates that for higher means, there is larger standard deviation which will be further discussed.

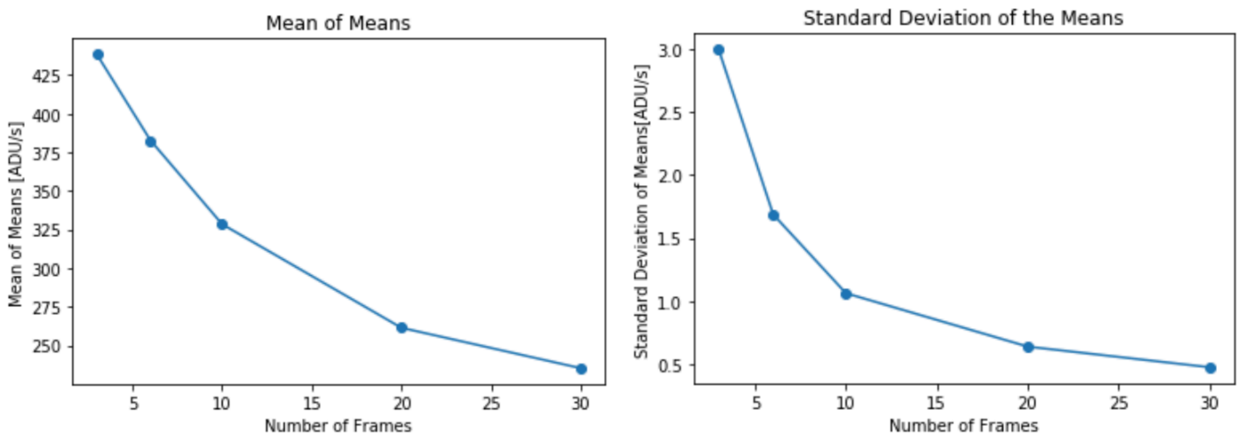

Figure 4: Mean of means (left) and standard deviation of the means (right) for data point compared to various numbers of exposures taken.

The next stage of analysis of the data is comparison to previously defined distributions with two histogram plots shown in **Figure 2**. These two data sets were taken at 3 seconds of exposure time and 384 seconds of exposure time to analyze the difference this makes in alignment with distributions. The first distribution the data was compared to was the Poisson distribution (**Equation 6**), which is typically used for counting experiments such as this one. The variables in this equation are μ for mean, x for data points, and e is Euler's constant. The Stirling approximation for factorials was used on the denominator in this equation since the data points include large values of x. The second distribution the data was compared to is the Gaussian distribution (**Equation 7**), which is typically used for random observations and large sample sizes. The Gaussian distribution equation uses the same variables as the Poisson distribution in

$$P(x;\mu) = \frac{\mu^x}{x!}e^{-\mu} \approx \frac{\mu^x}{\sqrt{2\pi x}}(\frac{x}{e})^{-x}e^{-\mu} \tag{6}$$

$$P(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \tag{7}$$

addition to σ for the standard deviation. The comparison to these two distributions is shown in **Figure 5**, but our lab group was unable to configure Python to work for plotting the defined functions so this was created using SciPy. In the Discussion section, the comparison between histograms and these two probability distributions will be discussed in more depth.

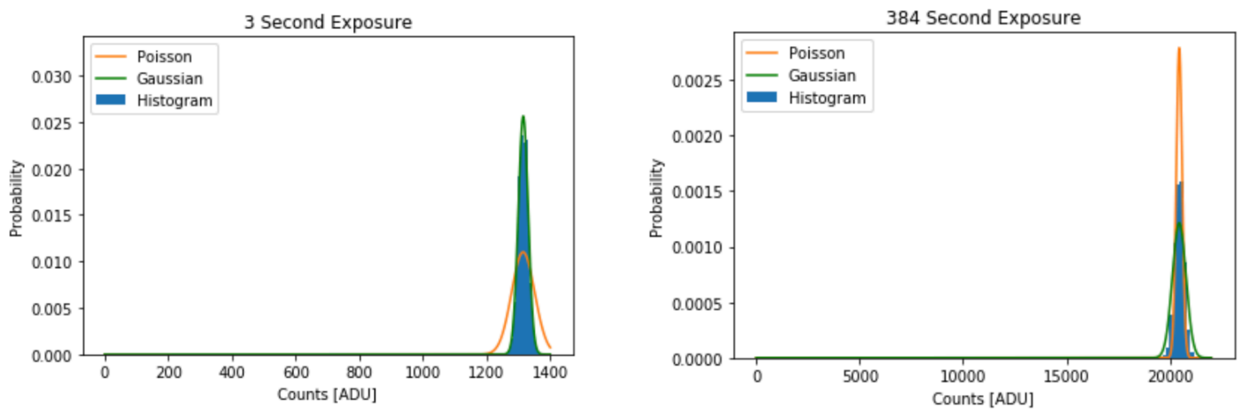

Figure 5: Histogram, Poison Distribution, and Gaussian Distribution for 3 second exposure (left) and 384 second exposure (right)

The final stage of analysis began with plotting the mean vs. variance in a similar fashion to that in **Figure 3**, but this time for the values of each pixel in ADU (**Figure 6**). This plot was then used to approximate our final results of gain and read noise by using a best fit line and analyzing the slope and intercept. The derivation for how these values connect to the plot is shown in **Equations (8) - (13)**, which begin using constants V_{pe} as the voltage from accumulated electric charge from photoelectrons, N as number of photoelectrons per pixel after dark current has been subtracted with bias frames, e as the charge of an electron, and C as the capacitance of the CCD. The next equation in the derivation uses ADU and ADU_0 as analog-to-digital unit signals for data points and bias and g as a general proportionality constant in ADU / volt for the circuit. The remainder of the equations introduce σ_N^2 as variance associated with Poisson noise which is equivalent to N and σ_0^2 as variance associated with read noise σ_0. The plot therefore generates the value of gain in electrons per ADU from the inverse of the slope and read noise as the square root of the intercept. Since the intercept is negative for the best fit line, the value for read noise was approximated by the left end of the plot shown in **Figure 6** as approximately **10 ADU per second** with the intercept appearing to be approximately 100 ADU per second. The full calculated value for gain is **0.10535592740604899 ADU per second.**

$$V_{pe} = \frac{Ne}{C} \quad \textbf{(8)}$$

$$ADU - ADU_0 = \frac{gNe}{C} \quad \textbf{(9)}$$

$$\sigma_f^2 = \sum_i (\frac{\delta f}{\delta i})^2 \sigma_i^2 \quad \textbf{(10)}$$

$$\sigma_{ADU}^2 = (\frac{\delta ADU}{\delta N})^2 \sigma_N^2 + (\frac{\delta ADU}{\delta ADU_0})^2 \sigma_0^2 \quad \textbf{(11)}$$

$$\sigma_{ADU}^2 = (\frac{ge}{C})^2 N + \sigma_0^2 \quad \textbf{(12)}$$

$$\sigma_{ADU}^2 = \frac{ge}{C}(ADU - ADU_0) + \sigma_0^2 \quad \textbf{(13)}$$

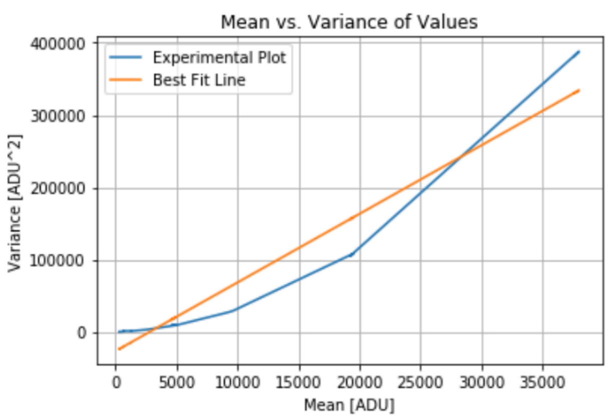

Figure 6: Mean vs variance with orange best-fit line for gain calculation from slope and with grid on to estimate read noise from y-intercept

Discussion

The first major result to assess is displayed in **Figure 3,** showing the relationship between the mean value and variance of each data set. Poisson statistics are used for counting experiments such as the experiment represented here, and by definition must have a mean that is equivalent to the variance as is shown by the orange line in **Figure 3.** The two lines are plotted on a logarithmic scale and there appears to be a spike in variance for the frames around 6 seconds of exposure. This means that there is a spike in standard deviation in these values since standard deviation is the square root of variance. The most likely cause of this discrepancy is a systematic error in these images, such as those caused by hot pixels and dark current. Otherwise, the plot forms a general trend in which for lower mean values, the experimental data closely aligns with the projection of Poisson statistics, while the variance begins to increase above the mean for higher mean values. The increase in variance likely occurs due to Poisson noise, which is intrinsic to the counting of photons in Poisson statistics at large mean values. The shift occurs at approximately 1000 on the logarithmic mean scale, indicating an upper limit at which Poisson statistics may be accurately applied. The Gaussian distribution more accurately depicts populations with higher mean values as the Poisson distribution approaches an upper limit. Further research in this area of study could be conducted to determine exactly which point in the data set indicates that this shift occurs.

The next result to analyze is shown in **Figure 4,** plotting the mean of means and standard deviation of the means from each data set in ADU per second against randomly selected, discrete values for number of frames, or images taken of the data, between 1 and 30. The mean of means corresponds to the amount of data points collected with more frames. Theoretically, this value should increase with the number of frames since there is more information being gathered. However, the plot shows the opposite trend with a lower mean of means for a larger amount of frames. This is because the data has been converted to a rate in ADU per second, thus for a larger number of frames the mean rate decreases since the detector slows its detection as it reaches the maximum amount of data the CCD can interpret. A higher number of frames should theoretically correspond to a smaller standard deviation since more trials should decrease the amount of error in results. This outcome is shown to be correct by the plot, which indicated an exponential decrease in standard deviation with increasing number of frames. The standard deviation goes below 1 ADU per second and begins to flatten at about 10 frames. Our data set has about 3 frames per exposure time, thus, in order to double the accuracy of the measurement I would increase this number of samples by a factor of 10 to 30 frames.

Histograms were created in the early part of the lab code for each of the data sets in order to visualize how the amount of noise and frequency of values changes for increased exposure times. Two examples of such plots, at 3 seconds and at 384 seconds, are shown in **Figure 2**. In order to analyze how well the Poisson and Gaussian distributions approximate the frequency of counts in the experimental data, these two functions were plotted against each of the two selected histograms in **Figure 5**. Our lab group ran into issues with the code in attempting to plot our own functions for the distributions, thus, we used built in functions from the Python package SciPy to achieve these plots. The final results indicate that the Gaussian distribution is much closer to approximating the value of the histogram since the images have such a large number of data points, as is consistent with statistical theory of distributions. For the 3 second exposure, the Poisson distribution greatly underestimates the curve, and for the 384 second exposure it overestimates the curve. It appears that there is a median amount of exposure time at which the Poisson and Gaussian align to best suit a histogram curve, which could be determined in further experimentation to project the exact point at which the distributions converge for data collected by the Direct Imaging Camera.

The final result for the lab is the gain, which is the amount of electrons collected per ADU, and the read noise, which is the intrinsic error in total number of electrons detected by the device. The value of approximately 0.105 ADU for the gain implies that there is a significant decrease in the output signal compared to the amount of input signal. This result is likely due to the old age of the detector, with electrons filling the holes in the valence band of the CCD, saturating the device. Computationally, the amount should be very accurate since it was capable of being precisely defined from the mean vs. variance plot. The value for read noise, however, is merely a best estimate since we were unable to have the Python plot extend to reach the exact intercept. By the appearance of the zoomed-in portion of **Figure 6**, the intercept should lie at about 100 ADU per second and therefore should have read noise of about 10 ADU per second. We chose to use this approximate intercept as the final read noise rather than the exact intercept of the best fit line since this intercept is a negative and would result in an imaginary read noise. The best fit line is much more accurate for approximating the portion of the plot which exists in the data than it is for a projection onto an axis. The value of 10 ADU per second is relatively common for this type of CCD in its intrinsic error due to existing electrons in the device. Further research could be conducted to better estimate the read noise by finding the most accurately modeled exposure time at which the Poisson and Gaussian distributions align, minimizing the variance and allowing the final plot to closer reach the y axis. Knowledge that the output signal is faint but accurate could also allow researchers to better approximate which exposure times are optimal for both Gaussian and Poisson statistical analysis, maximizing the accuracy of data collected by the telescope.

Appendix

Our lab group operated by asking questions as they arose in a group chat after we were unsuccessful in finding a live meeting time that worked for everybody. We therefore mostly worked independently, but some key parts of my final code were inspired by fellow group members. The code for **Figure 1**, which displays all of the images from the CCD, was inspired by Megan after I had searched for a way to format and find a clean region of data to display. I was also initially unsure how to subtract the bias frames from the flat dome frames, and once again received assistance from Megan. Another major component that was discussed was scaling and plotting the Poisson and Gaussian distributions with histograms in Python, which I worked through collaboratively with Megan and Joman. We were unable to configure a properly normalized plot using the functions I defined, so Megan contributed a method using SciPy to create the resultant plot for analysis. My further contributions to the group were by sharing the other functions I derived as shown below as they pertained to helping others as well as my method of plotting a histogram.

For reference in this lab, I utilized information from the lecture material and portions of code from the discussion sections. I also referenced Wikipedia for Stirling Approximation (https://en.wikipedia.org/wiki/Stirling%27s_approximation) and Charge-Coupled Devices (https://en.wikipedia.org/wiki/Charge-coupled_device). Finally, I referenced StackOverflow (https://stackoverflow.com/) for specific mathematical and coding questions as they arose.

Self-defined Python functions used for data analysis:

Mean:

```
def my_mean(arr):          # Function for mean
    total = 0
    count = 0
    for i in arr:
        total += i         # Sum over values in array
        count += 1         # Total count of values
    return total / count
```

Standard Deviation:

```
def mom_sdom(files):                    # Function for mean of means and standard devition of the means
    flat_zip = zip(files, flat_exp_time)
    for i, j in flat_zip:
        data = fits.getdata(i)[400:600, 400:600]
        x = data.flatten()
        x = x / j                       # Conversion from ADU to ADU / s
        mom_list.append(my_mean(x))
        sdom_list.append(my_std(x))
    mom = my_mean(mom_list)
    sdom = np.sqrt(np.sum(np.array(sdom_list)**2)/len(sdom_list)**2)
    return [mom, sdom]
```

Mean of means and standard deviation of the mean:

```
def my_std(arr):                  # Function for standard deviation
    total = 0
    count = 0
    val = 0
    for i in arr:
        total += i                # Sum over values in array
        count += 1                # Total count of data points
        val += (i-(total/count))**2  # Sum of (value - mean)^2
    return (val/count)**0.5
```

Poisson distribution:

```
def P(x, m):                                            # Function for Poisson Distribution
    stir = np.sqrt(2 * np.pi * x) * (x / np.exp(1)) ** x  # Stirling's approximation for factorial
    return (m**x * np.exp(-m))/(stir)
```

Gaussian distribution:

```
def G(x, m, s):  # Function for Gaussian Distribution
    return (s*np.sqrt(2*np.pi))**-1 * np.exp(-(1/2)*((x-m)/s)**2)
```

Code to create arrays for mean, standard deviation, and variance:

```
means_list = []
std_list = []
var_list = []
x_list = []

for data in flat_data:
    x = data.flatten()                # Flattening each data set to 1 dimension
    x_list.append(x)
    mean = my_mean(x)                 # Calculating mean with previously defined function
    means_list.append(mean)           # Creating array of all calculated means
    std = my_std(x)                   # Calculating standard deviation with previously defined function
    std_list.append(std)              # Creating array of all calculated standard deviations
    var = std**2                      # Calculating variance by squaring standard deviation
    var_list.append(var)              # Creating an array of all calulated variances
```

Code to make histogram of single exposure:

```
count_min = 0
count_max_3 = x_list[0].max()
count_range_3 = np.arange(count_min, count_max_3+1)
hist_list_3 = []

for i in count_range_3:  # Creating Histogram for 3 second exposure
    count_freq = len(np.where(x_list[0]==i)[0])
    hist_list_3.append(count_freq)
```

Code for mean vs. variance plot compared to Poisson:

```
plt.plot(means_list, var_list, label='Experiment')  # Plot of means vs. variances for each flat dome data set
lims = [np.min([means_list, var_list]), np.max([means_list, var_list])]
plt.plot(lims, lims, label='Poisson Theory')        # y = x line for equal mean and variance (Poisson)

plt.title('Mean vs. Variance of Counts')
plt.xlabel('Mean')
plt.ylabel('Variance')
plt.xscale('log')
plt.yscale('log')
plt.legend()
```

Code for plot of histogram compared to Poisson and Gaussian with SciPy:

```
short = fits.getdata('d5.fits')[400:600,400:600].flatten()  # SciPy plot for histogram
plt.hist(short, label='Histogram', normed=True, bins=100)

x = np.arange(0,1400,1)  # SciPy plot for Poisson
y = poisson.pmf(x, my_mean(short))
plt.plot(x, y, label='Poisson')

gauss = norm.pdf(x, my_mean(short), my_std(short))  # SciPy plot for Gaussian
plt.plot(x, gauss, 'g',label='Gaussian')

plt.xlabel('Counts [ADU]')
plt.ylabel('Probability')
plt.title('3 Second Exposure')
plt.legend()
```

Code for mean vs. variance with best fit for gain and read noise:

```
x = [int(j) for j in means_list]
y = var_list
plt.plot(x, y, label='Experimental Plot')  # Plot of means vs. variances for each flat dome data set
m, b = np.polyfit(x, y, 1)
plt.plot(x, [m*i for i in x] + b, label='Best Fit Line') # Plot of best fit
Gain = m ** -1 # Final value for gain

plt.draw()
plt.legend()
plt.title('Mean vs. Variance of Values')
plt.xlabel('Mean [ADU]')
plt.ylabel('Variance [ADU^2]')
```