

## CHARGE-COUPLED DEVICE DETECTORS

JAKOB BORG

*Draft version October 20, 2018*

### ABSTRACT

State problem. Briefly describe method and data. Summarize main results.

*Subject headings:* cosmic microwave background — cosmology: observations — methods: statistical

### 1. INTRODUCTION

Historically many different mediums for capturing exposures in astronomical photography have been used, including film and different kinds of photographic plates. Whenever an exposure is captured some of the information from the incoming photons is lost in the process. Also noise and other limitations of the capturing method is introduced. This results in the exposure not representing the object we observe accurately. Understanding these limitations and sources of noise is crucial in astronomy, where we often observe very distant and faint objects, so that few of the initially emitted photons reach our telescopes and detectors.

Ever since Willard Boyle and George E. Smith invented the Charge-Coupled Device (CCD) in 1969 (Boyle and Smith 1973) they have become increasingly more popular in astronomy as our technology has improved. CCDs have an array of sensitive detectors that converts the energy of photons hitting them into electric charges. These charges can then be manipulated by potential wells which moves the charges to a register to be read. In addition to high accuracy, the CCD registers the exposures digitally. Each detector acts as a pixel in our image, where the amount of charge registered is proportional to the light intensity and may be converted to a pixel value using analog-to-digital units (ADU). This way it's easier to correct the raw image for noise and other limitations as well as perform statistical analysis and editing.

Some of the most important features in a detector is it's resolution, quantum efficiency, amount of thermal noise, spectral response, dynamical range and bit depth. With the steady improvement in technology over the years the CCDs have become extremely accurate for their cost and size compared to prior analog methods. We have made tinier and more versatile detectors, cheaper and cheaper. With smaller and more numerous pixels the resolution has increased. They also have a quantum efficiency approaching the maximum possible. Usually in the range of 90% to 100%, so that as many of the photons as possible is registered. This efficiency is reached for a broad range of wavelengths as the spectral response of the detectors has improved. With a high linear dynamical range we can resolve details in both dim and bright parts of the signal at the same time. And with a high bit depth we can differentiate tiny deviations between neighboring pixels. All of these features makes the CCD cameras particularly useful in astronomy. They are in use in most of

the modern space telescopes and observatories.

In this paper our goal is first to explore some of the properties of a standard color CCD camera. Then we go through the process of taking a well exposed image using a monochromatic CCD camera, record calibration data and process the raw image to remove noise. The main focus will be oriented around the second part. As we will see, monochromatic cameras are most used in astronomy.

### 2. DATA AND EXPERIMENTAL SETUP

For the two cameras we used similar experimental setups, where we switch out a few key features. The main components are an externally powered light source projecting light through a filter piece and made parallel in a collimator tube. The parallel light then passes an aperture and is recorded in the camera. The cameras are controlled and output directly into a graphical user interface (GUI) on a computer.

#### 2.1. Exploring color CCD properties

We used a white light lamp to produce light of different wavelengths in the visible spectrum. As filter piece we used a dampening filter to reduce the light intensity to not over expose the image or damage the camera. We used a singlet lens as aperture  $7.3 \pm 0.2$  cm from the tube to focus the light onto a microscopic objective with a  $M = 20x$  magnifying effect. We recorded the exposure with a 8 bit color Edmund Optics USB camera. The camera properties is tuned to:

- Pixel clock: 24 MHz
- frame rate: 8.86 fps
- Exposure time: 4.794 ms

Schematic of the setup in fig. 1.

#### 2.2. Taking well exposed monochromatic exposure

We used a red Helium-Neon laser with a wavelength of  $\lambda = 635$  nm. To produce an image with interesting features to analyze we used a single slit with an opening  $a = 100 \mu\text{m}$ . The slit reduces the light intensity, so we also switched out the filter by an extension piece without filtering of the same length as the dampening filter to minimize the change in the light path through the collimator tube. We use a 8 bit monochromatic Edmund Optics USB camera with an extension tube of length  $L = 7.5 \pm 0.2$  cm attached to the lens. The slit is then placed in the opening of the tube projecting a diffraction pattern for the camera to record. The camera properties is tuned to :

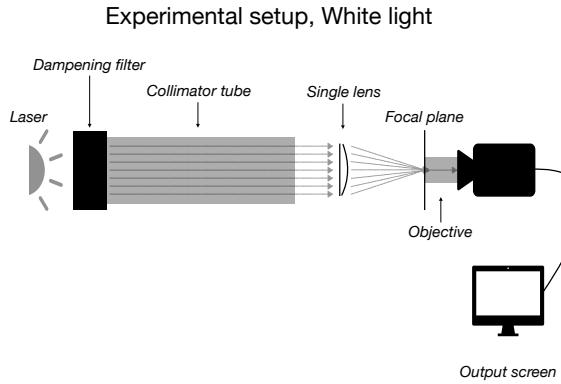


FIG. 1.— Schematic of experimental setup for exploring camera properties of a RGB color CCD camera. The light intensity is reduced in the dampening filter so that the camera is able to differentiate the signal. The light is then made parallel in a collimator tube, and focused by the singlet lens producing an airy disc. The objective is placed in the focal plane of the lens to magnify the exposure for the camera. We can then observe the magnified airy disc on the computer screen.

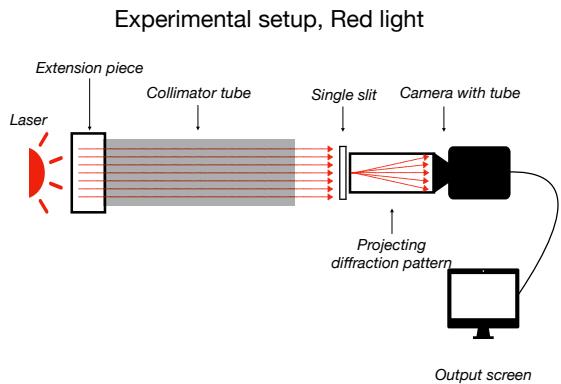


FIG. 2.— Schematic of experimental setup for taking well exposed image with a monochromatic CCD camera. The light is made parallel in a collimator tube, projected through a single slit to produce a diffraction pattern. The light is undisturbed through the extension tube of the camera and recorded. We can then observe the diffraction pattern on the computer screen.

- Pixel clock: 19 MHz
- frame rate: 3.92 fps
- Exposure time: 95.265 ms

Schematic of the setup in fig. 2.

### 3. METHOD

#### 3.1. Color CCD experiment, singlet lens

We calibrated the setup using a red laser to find the focal point of the thin singlet lens. Parallel light focused by a circular aperture produces an Airy disc (Borg 2018), which we used to control the alignment of the lens in the optical axis. By requiring an even and circular Airy disc we minimized the effect of comatic aberrations, which

occur if the plane of the lens is not normal to the optical axis.

After calibration we changed the light source to the mentioned white lamp and viewed the exposure on the graphical user interface (GUI) connected to the camera. Here we experimented with adjusting the cameras pixel clock, frame rate and exposure time to see how they effected the exposure.

The GUI allowed us to look at the registered pixel count in a slice of the image for each color red green and blue (RGB) through a histogram. Due to chromatic aberrations, different wavelengths of light have different focal lengths, which is not accounted for by a singlet lens. Therefore we adjusted the position of the objective slightly back and forth to find the focal length of the different colors, defined by the point where we reached the highest registered intensity in RGB pixels. The maximum found pixel count in each color was measured using a ruler directly on the output screen and measuring the relative height of the peak in each color. Then the pixel count is calculated with uncertainties from the lengths. An example of how the histogram looked for the green maximum is included in appendix B fig. 15.

#### 3.2. Monochromatic CCD experiment, single slit

##### 3.2.1. Well exposed image

To make sure the image of the diffraction pattern was well exposed we tuned the camera properties to record the signal with as high intensity as possible to minimize the effect of noise. The optimal signal level is limited by the dynamical range and bit depth of the CCD, so that we don't over-expose the image. When the light intensity registered by a detector is exceeding the maximum pixel value in the bit depth, the information in the over-exposed pixel will be lost as it is truncated down to said maximum value. In addition, when very close to the maximum exposure level, the sensors might display a non-linear behavior in the quantum efficiency.

In our exposure of the diffraction pattern we are not concerned by the intensity in the center maximum, but rather the distance from the center to each minima and maxima of higher order. Therefor we calibrated the camera such that the pixel value registered for the first order maxima where about one half of the maximum. This way the center maximum of the image is over-exposed, but the features we are interested in are well exposed.

##### 3.2.2. Recording calibration data

We turned off the light source and removed the camera from the optical path of the setup. Then recorded multiple frames to be used for correcting the raw image for bias, dark current and flat field. For the correction frames we used the same pixel clock and frame rate as for the raw image given in subsection 2.2, while adjusting the exposure time.

*Flat field frames*— Flat field correction compensates for variations in quantum efficiency from pixel to pixel through the hole field of view, as well as imperfections in the optical path. We recorded 16 basic flat field frames. Here we used a white sheet of paper held at an angle to reflect a uniform field of light from the ceiling into the camera. To do this we had to move the CCD camera out

of the optical path of the system to face the white paper. We adjusted the exposure time to 0.762 ms so that the average pixel value in the recording is about one-third of the maximum value.

*Bias and dark current frames*— The bias and dark current frames are recorded in total darkness. The bias gives a measure of the lowest level in a signal. Ideally this level should correspond to a pixel value of 0, but is set to a low positive value in order to account for the digitization noise. During the integration time of the exposure in the CCD a dark current is present from generation of extra thermal electrons in the system not due to detection of photons.

Both the bias and dark current exposures were recorded with the dust cover on the camera to reduce the signal as much as possible. We recorded

- five dark frames with the same exposure as the flat fields, 0.762 ms
- five dark frames with the same exposure as the raw image, 95.265 ms
- one dark frame with maximum exposure time, 255.02 ms
- two bias frames with the minimum exposure time 0.169 ms

### 3.2.3. Statistical analysis of correction frames

All of the statistical analysis are written in Python 2.7.14. The scripts are included in appendix C. With the images saved as bitmap files we read the files into two dimensional arrays, where each element in the array corresponds to one pixel with a given value  $\in [0, 255]$ . Let  $B_i$ ,  $F_i$  and  $D_i$  correspond to the arrays obtained from bias, flat field and dark current frame number  $i$  respectively. Using the array representation of the images we can add and subtract them and perform numerical calculations by operating on the arrays.

*Comparing bias and dark frame*— First we compared one of the two bias frames  $B_1$  with the dark current frame captured with maximum exposure  $D_{max}$ . We found the mean, minimum and maximum pixel values of the two frames. In addition we compared the coordinates of the maximum pixel in the arrays, and the distribution of the pixel values.

*Mean value in composite frames*— The mean value of a frame is given by its expectation value. As the expectation value is linear it's trivial to find the mean of composite frames.

$$\mu = E[B_1 + B_2] = E[B_1] + E[B_2] = \bar{B}_1 + \bar{B}_2 \quad (1)$$

*Noise in composite frames*— To model the noise in an image we used the standard deviation of the pixel values. By subtracting one correction frame from another, we find the change in pixel values due to random noise not constant in time. To avoid unwanted effects near the edges we extract the square central part of the image, with 300 pixels on a side.

From the addition rules of variance for  $n$  uncorrelated variables (Devore and Berk 2006) we know the standard

deviation should increase by a factor of  $\sqrt{n}$ . This relation is shown in eq. (17) in appendix A.1.

Using eq. (1) we found the mean of the pixel values in the composite bias frame,  $\bar{B}_1 + \bar{B}_2$ . The non-static noise,  $\sigma_{(B_1 - B_2)}$ , is found from the standard deviation. The same procedure on two of the flat frames gives  $\bar{F}_1 + \bar{F}_2$  and  $\sigma_{(F_1 - F_2)}$ .

*Read out noise in electrons*— As the signal is converted to ADU, there is a conversion factor  $g$  which relates the pixel counts to the actual number of measured electrons. The number of detected electrons are a positive discrete value, so we expect the signal to follow Poisson statistics (Devore and Berk 2006) measured in electrons, such that  $\sigma_{electrons} = \sqrt{F_{electrons}}$ . As the signal is converted using  $g$  our actual measurements are  $g\sigma_{electrons} = \sqrt{g}F_{electrons}$ . Solving for  $g$  and correcting for bias and bias noise we found the value of the conversion factor by

$$g = \frac{F_{electrons}}{\sigma_{electrons}^2} = \frac{(\bar{F}_1 + \bar{F}_2) - (\bar{B}_1 + \bar{B}_2)}{\sigma_{(F_1 - F_2)} - \sigma_{(B_1 - B_2)}} \quad (2)$$

Ideally, the only source of noise in a single bias frame is the read out noise (R.O.N.). Having found the noise for the combined frame we used the relation in eq. (17) to find the noise in a single frame

$$\sigma_{B_1} = \frac{\sigma_{(B_1 - B_2)}}{\sqrt{2}} \quad (3)$$

The R.O.N. in electrons was computed using the conversion factor and the noise from a single bias frame

$$R.O.N. = g\sigma_{B_1} \quad (4)$$

### 3.2.4. Normalized noise from composite frames

By adding multiple pairs of correction frames we even out the random noise from each frame, but in the process amplify the static noise that overlap in multiple (or all) frames. To account for this we normalized the noise by the number of pairs used. The normalization of multiple frames results in a net reduction in the noise by a factor of  $\sqrt{n}$ , as shown in appendix A.1.1. We calculated the normalized noise for successively more pairs of flat field frames. Then the normalized noise together with the theoretical expectation were plotted against the number of pairs and compared.

### 3.2.5. Correcting raw image

In the process of correcting the raw image we used a combination of the same statistical methods used on the calibration frames. First we found the average of all the flat field frames and the dark current frames with the same exposure.

$$F_{avg} = \frac{\sum_{i=1}^1 6F_i}{16}$$

$$D_{avg,F} = \frac{\sum_{i=1}^5 D_{i,F}}{5}.$$

We constructed a normalized master flat frame by correcting the average of the flat frames for dark current,

and divide by the mean value of the master flat.

$$F_{\text{master}} = F_{\text{avg}} - D_{\text{avg},F}$$

$$F_{\text{normed, master}} = \frac{F_{\text{master}}}{\bar{F}_{\text{master}}}$$

To correct the raw image for dark current we found the average dark frame taken with the same exposure as the raw image. Correction for flat field was done using the normalized master flat.

$$D_{\text{avg},I} = \frac{\sum_{i=1}^5 D_{i,I}}{5}$$

$$I_{\text{corrected}} = \frac{I_{\text{raw}} - D_{\text{avg},I}}{F_{\text{normed, master}}} \quad (5)$$

where  $I$  denotes the proper image captured of the diffraction pattern.

### 3.2.6. Calculating the pixel width

We used the corrected image to calculate the size of the pixels in the CCD camera. The diffraction pattern is characterized using the equation for a single slit (Borg 2018)

$$m\lambda = a \sin(\theta) \quad (6)$$

where  $\lambda$  is the wavelength,  $a$  is the slit width and  $m$  is the order of the minimum at angle  $\theta$  from the center maximum. A sketch of the system is displayed in fig. 3. The number of pixels  $n_p$  with width  $w_p$  to minimum  $m = 4$  was found numerically by the intensity in the corrected image. To do this we found a row along the image array crossing the trenches in the pattern with high mean value. To minimize the uncertainty in  $n_p$  we found the number of pixels between the fourth order minima on both sides of the center and divided by 2. Then the angle  $\theta$  was found by  $\tan(\theta) = \frac{n_p w_p}{L}$ . We know the diffraction pattern is in the Fraunhofer regime (Born and Wolf 1999), so we used the small angle approximation

$$\tan(\theta) \approx \sin(\theta) \approx \theta \approx \frac{n_p w_p}{L}. \quad (7)$$

The width of the pixels was then calculated as

$$w_p = \frac{L m \lambda}{a n_p} \quad (8)$$

$$\text{with uncertainty } \frac{\delta w_p}{w_p} = \sqrt{\left(\frac{\delta L}{L}\right)^2 + \left(\frac{\delta n_p}{n_p}\right)^2} \quad (9)$$

where we neglect the uncertainties in  $a$  and  $\lambda$  as they are negligible.

## 4. RESULTS

### 4.1. Color CCD experiment

#### 4.1.1. Experimenting with camera properties

*Pixel clock*— A low pixel clock value resulted in a bright high intensity signal, and vice versa.

*Frame rate*— A low frame rate delayed how changes in the image where displayed, and also seemed to effect the intensity in the exposure.

*Exposure time*— A high exposure time results in high intensity signal, and vice versa.

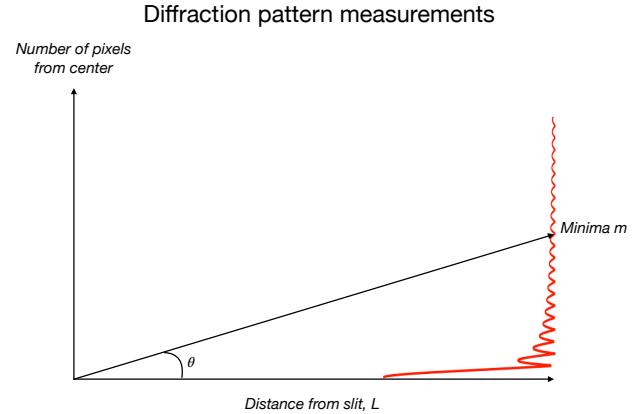


FIG. 3.— Sketch of the geometry in the procedure for finding the pixel size.  $L$  is the distance from the slit to the camera.  $m$  is the order of the minimum an angle  $\theta$  from the center maximum. The number of pixels, and hence the pixel size, can be found using eq. (6).

#### 4.1.2. Chromatic aberrations

We found the red light to have focal point furthest from the camera, blue light closest and the green light in the middle. The maximum count for each color are found to

- Red:  $125 \pm 27$
- Green:  $172 \pm 37$
- Blue:  $27 \pm 6$

The captured images of the Airy disc and and the images taken at the focal point of each color is included in appendix B.

### 4.2. Monochromatic experiment

*Well exposed raw image*— Pixel count across the diffraction pattern in two slices of the raw image displayed in fig. 4. Compared with the mean value of all the rows, row number 37 is a slice with maximum mean value and row number 116 has average mean value. The central square of the captured raw image is included in fig. 8. The full image is shown in appendix B fig. 13.

*Analysis of calibrating frames*— Results from the statistical analysis of the calibration data presented in table 1. Histogram of the distribution of pixels in fig. 5 and fig. 6. The images of the first bias frame and the dark current with max exposure are included in appendix B.

*Read out noise*— Noise from a single bias frame in ADU is found to

$$\sigma_{B_1} = 0.43365 \approx 0.434 \quad (10)$$

The conversion factor is found as

$$g = 79.22801 \approx 79.228 \text{ [electrons/ADU]} \quad (11)$$

The R.O.N. in electrons from a single bias frame is found to

$$\text{R.O.N.} = 34.35707 \approx 34.357 \quad (12)$$

*Normalized noise*— Normalized noise of successively more pairs of frames together with the expected values are included in fig. 7.

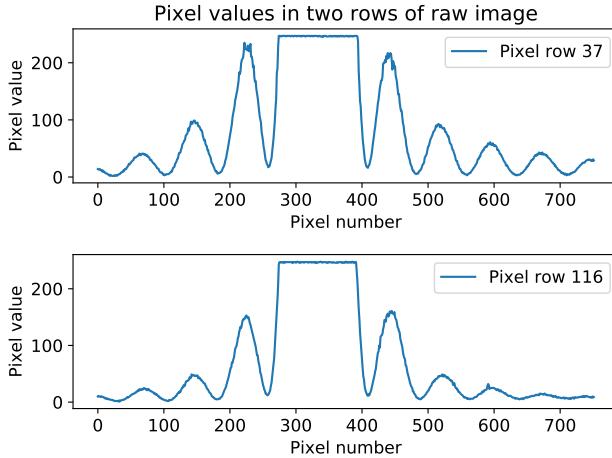


FIG. 4.— Pixel count in two slices across the diffraction pattern. The slices are picked based on the mean pixel count across the row. In the upper plot one of the rows with the maximum mean is displayed, and the lower plot shows one of the rows with average mean.

TABLE 1

Frame	$\mu$	Min	Max	Index	$\sigma$
$B_1$	8.535	6	19	[439, 427]	
$D_{max}$	8.978	6	47	[214, 478]	
$(B_1 + B_2)_c$	17.127	13	25	[247, 244]	0.613
$(F_1 + F_2)_c$	178.648	159	198	[290, 175]	1.554

NOTE. — Statistical values found for calibration data. For frames noted with subscript  $c$  only the central square region of the image is evaluated. The Index-column gives the coordinates  $(x, y)$  for the maximum value pixel. For hole frames:  $x \in [0, 480]$  and  $x \in [0, 752]$ . For central regions:  $x, y \in [0, 300]$ .

*Corrected image*— The central square final corrected image is shown in fig. 9. The corrections done on the full image is included in appendix B fig. 14. We also found the mean pixel value in the two frames to be

- Raw image mean = 76.822
- Corrected image mean = 68.492

*Pixel size*— Figure 10 shows the number of pixels between the two minima of 4th order, giving number of pixels from central maximum

$$n_p = 302 \pm 5 \quad (13)$$

where we add a fair but substantial uncertainty to account for errors. Eq. (8) gives a pixel width

$$w_p = 6.308 \pm 0.2 \mu\text{m} \quad (14)$$

## 5. CONCLUSION AND DISCUSSION

### 5.1. Monochromatic CCD

#### 5.1.1. Comparing calibration frames

*Comparing bias with max exposure dark current*— The dark current and bias frames look similar, as seen in appendix B. Both frames are taken in total darkness, making them look black, but from the statistical analysis we note some differences.

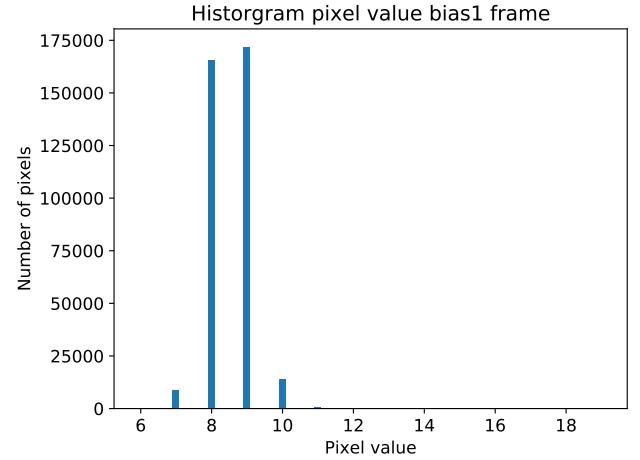


FIG. 5.— The distribution of pixel values for bias frame 1 displayed in appendix B fig. 12.

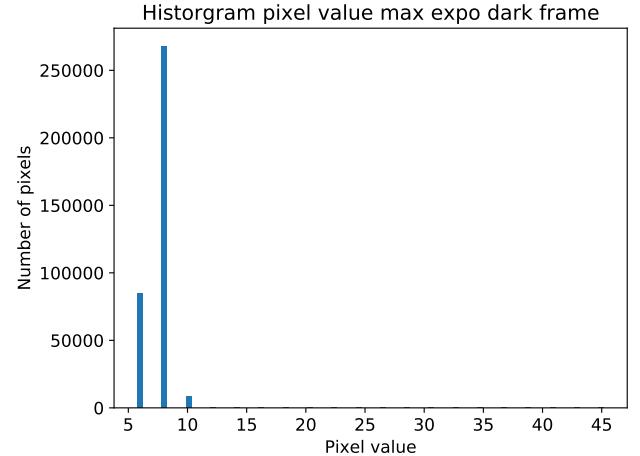


FIG. 6.— The distribution of pixel values for the dark current frame with maximum exposure displayed in appendix B fig. 11. Note the high max value along the x-axis. There are pixels counted here, but the scaling on the y-axis makes them not visible.

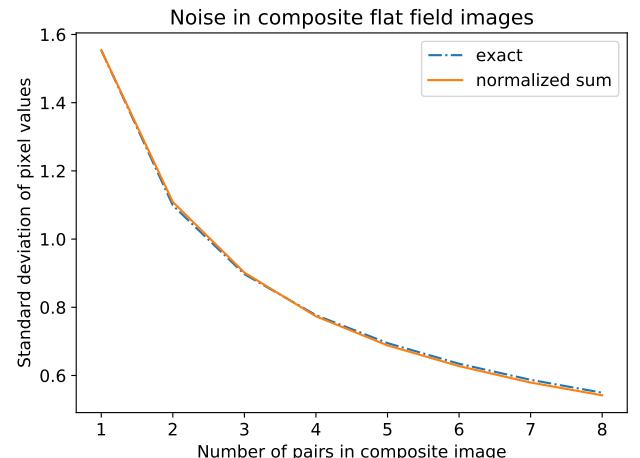


FIG. 7.— The scaling of the normalized error in successively higher number of pairs of flat frames. The expected error is included.

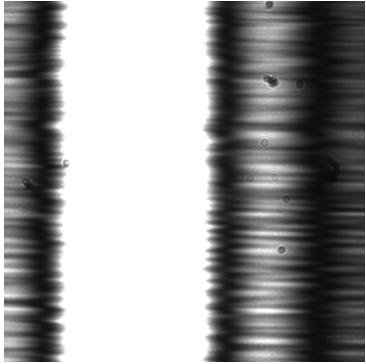


FIG. 8.— Central square part of raw image recorded of the diffraction pattern from subsection 3.2. Notice the imperfections in the optical path. There are dust obstructing the light which results in dark spots. The spots have light edges from the diffraction pattern formed by the shape of the particles. The corrected image is shown in fig. 9.

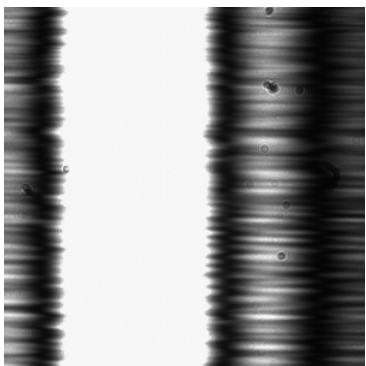


FIG. 9.— Central square part of the exposure of the diffraction pattern from subsection 3.2 corrected with the dark current average and normalized master flat field. The raw image is shown in fig. 8.

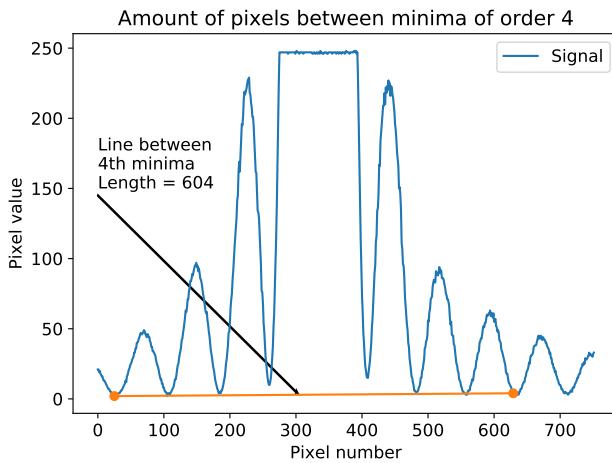


FIG. 10.— The amount of pixels separating the two minima of 4th order. The slice of the image used has a high mean value.

The minimum pixel values are identical, as this is determined by the lowest signal level, but the maxima differ by a value of 28. This is expected as the dark current frame is taken with maximum exposure time so the sensitive detectors records a higher pixel count due to more noise being generated. This is also evident in the mean, where the dark current have slightly higher value than the bias.

The maxima are also located differently in the two frames. We know the noise generated by dark current is random over the array of pixels, so the difference in maxima location and value are to be expected.

These differences can also be deduced by the histograms of the pixel values, although the details are illusive. Both frames have the majority of pixels with value 8, so the higher value pixels is not shown as well.

*Composite frames*— For the composite bias frame we see as expected the mean and min values approximately double the value of a single frame. This is not the case for the max value, indicating that the max values in the two bias frames is not in the same pixel. This is not ideal as the bias frames should be close to identical, but the non-static noise in the frames are quite low so the standard deviation is low as to be expected.

The values for the composite flat frames are as expected much higher, as these frames is not captured in darkness but in a uniform, or close to uniform, light. We see the noise in the composite frame is quite larger than in the bias frames, which is probably due to the primitive way we captured the flat frames. Holding a white sheet of paper in hand is not stable enough from frame to frame to produce the exact same uniform field in each frame. The distance from the field to the camera may change, as well as the plane of the paper shifting. Still, the normalized noise in composite flat frames decrease exactly as expected by the factor  $\frac{1}{\sqrt{n}}$ .

*Conversion factor and read out noise*— From our value for the conversion factor we see that of approximately 80 electrons the CCD register 1 ADU. This is quite poor compared to the detectors used in the wide field cameras aboard the Hubble Space Telescope (STScI 2018), which has a conversion factor in the range [1.52, 1.56]. This comparison is not so fair as we are comparing a standard cheap CCD with high end equipment with a much deeper potential well. The size of the potential well is important as a too high conversion factor over floods the potential wells so that registered electrons leaks out of the well. Our camera would not handle the conversion factor used in the wide field cameras of the Hubble Space telescope, as the potential wells are far to shallow.

The read out noise gives us a measurement of noise generated in the integration time of the exposure. The R.O.N. in electrons are calculated using the conversion factor and noise from a single bias frame, which ideally is only the read out noise in ADU. The standard deviation in a single frame is found from the relation of variance from multiple variables, which should behave accurately as we saw for the noise in composite flat frames. This means that 68% (within one  $\sigma$ ) of the exposures taken with the same setup have registered electrons deviating from the true mean of electrons by 34.357. With a conversion factor of 79.228 [electrons/ADU] this R.O.N. is very good, contributing with approximately 0.5 ADU. This value seems low, and would be interesting to control in further detail in the future.

### 5.1.2. Corrected raw image

The correction process does not change the image significantly, but we can see a slight reduction in the intensity due to the random thermal noise from dark current

in the raw image. This is corrected using the dark current frames, but as we have seen in the statistical analysis of the calibration data, the dark current generated is quite small. The correction is evident in the difference of 8.33 in the mean value of the pixel count from the two images. In most of astronomical observations the raw exposures are prone to a much higher dark current being generated than in our simple setup, such that the correction for these effects are more important.

The most noticeable impurities in our raw image are the diffraction patterns and shadows resulting from the dust particles in the optical path. These obstructions should be corrected using the flat field frames. This is not the case, and is probably due to the poor quality of our flat fields as discussed under section 5.1.1. In addition to the flat fields not being uniform and constant over time, we changed the optical path of the system to capture the flat fields. By doing this we removed all the dust particles in the other instruments used along the path of the laser, so that the flat fields did not obtain the required information to correct for these impurities.

For future improvements we should take flat field correction frames in as close to the same optical path used for the proper image as possible. We should also use a

much more stable source for the uniform field. In addition we could take more than 16 frames, as we have seen the noise in the flat fields decrease dramatically as we increase the number of pairs used.

To conclude, the CCD is of huge importance in observational astronomy due to it's efficiency and applicability to multiple type of exposures. With increasing technology the CCDs approaches the theoretical maximum in most of the key features we need in a detector, as well as being versatile, small, cheap and easy to operate. They also provide easy methods for recording correction data, without having to change the setup too much. In addition to cameras, CCDs are in frequent use in many other instruments including spectrometers and interferometers.

Thanks to my colleagues and co-workers in the execution of the experiments: Nils-Ole Stutzer, Jon Dahl and Barathy Pirabahar.

Thanks to our guide and supervisor in the lab: Ranghild Aurlien.

#### REFERENCES

- Jakob Borg. Effects of diffraction. submitted, September 2018.  
 Max Born and E Wolf. *Principles of Optics*. Cambridge University Press, 7 edition, 1999.  
 Willard Sterling Boyle and George Elwood Smith. Buried channel charge coupled devices, april 1973.

- Jay L. Devore and Kenneth N. Berk. *Modern Mathematical Statistics with Applications*. Springer International Publishing AG, second edition edition, January 2006.  
 Space Telescope Science Institute STScI. Hubble space telescope wfc3 detectors, October 2018. URL [http://www.stsci.edu/hst/wfc3/ins\\_performance/detectors](http://www.stsci.edu/hst/wfc3/ins_performance/detectors).

# Appendices

## A. CALCULATIONS

### A.1. Variance and standard deviation of uncorrelated variables

For  $n$  variables with the same variance  $V[X_i] = \sigma^2$ , the variance of the sum of the variables (Devore and Berk 2006) is given by

$$V[a_1X_1 \pm a_2X_2 \pm \dots \pm a_nX_n] = a_1^2V[X_1] + a_2^2V[X_2] + \dots + a_n^2V[X_n] \quad (15)$$

if the weights  $a_i = \pm 1$  for all  $i$  this simplifies to

$$\sigma_{(a_1X_1 \pm \dots \pm a_nX_n)}^2 = \sum_{i=1}^n V[X_i] = n\sigma^2 \quad (16)$$

We model the noise by the standard deviation of the difference of two frames, found by taking the square root of the variance of the difference. This means our weights are indeed  $a_1 = 1$  and  $a_2 = -1$  so we can use the simplified expression

$$\sigma_{(a_1X_1 \pm \dots \pm a_nX_n)} = \sqrt{n\sigma^2} = \sqrt{n}\sigma \quad (17)$$

#### A.1.1. Normalized standard deviation of pairs of frames

If we combine  $n$  pairs of frames we know from eq. (17) that the standard deviation should increase by a factor of  $\sqrt{n}$ . Normalizing by the number of pairs used

$$\sigma_{composite,normed} = \frac{\sqrt{n}}{n}\sigma = \frac{\sigma}{\sqrt{n}} \quad (18)$$

results in a reduction in the noise a a factor  $\sqrt{n}$ .

## B. ADDITIONAL FIGURES



FIG. 11.— Image of dark current frame with max exposure.



FIG. 12.— Image of bias frame number 1.

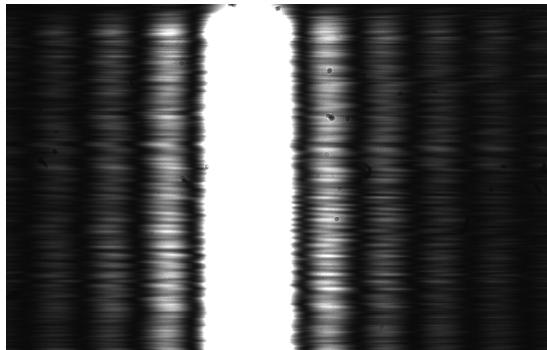


FIG. 13.— The hole raw image captured.

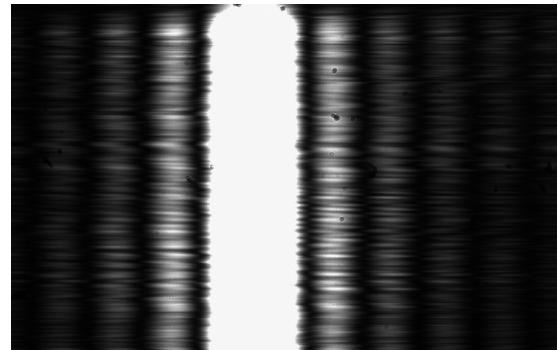


FIG. 14.— Corrections on the hole raw image.

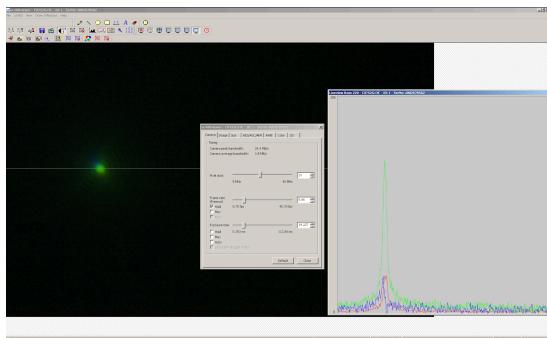


FIG. 15.— Example of finding max pixel count for RGB camera, here viewing the for green.

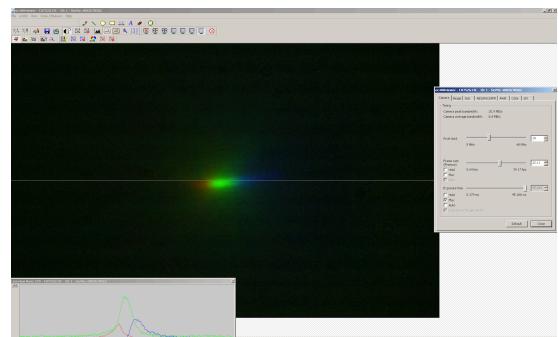


FIG. 16.— Displaying the GUI and how the chromatic aberration separate each color, shown using max exposure time. In the lower right the histogram of pixel count for each color is included.



FIG. 17.— Image captured when finding focal point for red light.



FIG. 18.— Image captured when finding focal point for green light.

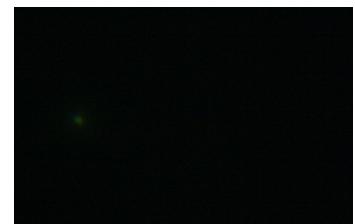


FIG. 19.— Image captured when finding focal point for blue light.

## C. PYTHON SCRIPTS

All codes are written in Python 2.7.14 and ran on macOS 10.14.

C.1. *Calculating pixel count for RGB camera*

```
import numpy as np
from math import sqrt

topp = 18.1
top_unc = 0.1
max_c = 255.
RGBmax = np.asarray([8.9, 12.2, 1.9])
RGBunc = np.asarray([0.3, 0.2, 0.4])

RGB_c = RGBmax*max_c/topp
RGB_unc = sqrt(np.sum((RGBunc/RGBmax)**2))*RGB_c

print 'Max pixel value RGB colors'
print 'Red: {:.0f} pm {:.0f}'.format(round(RGB_c[0]), round(RGB_unc[0]))
print 'Green: {:.0f} pm {:.0f}'.format(round(RGB_c[1]), round(RGB_unc[1]))
print 'Blue: {:.0f} pm {:.0f}'.format(round(RGB_c[2]), round(RGB_unc[2]))
```

C.2. *Analyzing first bias and dark frame*

```
import numpy as np
# may be installed with: pip install imageio
import matplotlib.pyplot as plt
from matplotlib import rcParams

def reading(path, hole = False, sides = 300):
    """
    Reads image file and return as 2D array
    @ path - path to filename
    @ hole - return array of the hole image if True, default to False
    @ sides - gives length of sides of central region if hole = False
    """
    im1 = np.asarray(imageio.imread(path)).astype(int)
    if hole:
        return im1
    else: # return using indices from top left corner of central region
        t1 = (np.asarray(im1.shape)/2.0-1-sides/2).astype(int)
        return im1[t1[0]:t1[0]+sides, t1[1]:t1[1]+sides]

def info(im1):
    """
    Returns mean, min and max of pixel values, as well as index of max value
    @ im1 - 2D array image
    """
    return [im1.mean(), im1.min(), im1.max(), \
            np.unravel_index(im1.argmax(), im1.shape)]

def print_info(im, name):
    print ('For '+name)
    infom = info(im)
    print ('Mean, minimum, maximum pixel values: '\
           +'{:3f}, {:.1f}, {:.1f}'.format(*infom[:-1]))
    print ('Index of maximum: {:s}'.format(str(infom[-1])+'\n'))

if __name__ == '__main__':
    # Path extension
    path = '../images/ex3_4_5/'

    # Read images
    im_bias = reading(path+'bias_min_expo1.bmp', hole=True)
    im_dark_max = reading(path+'dark_max_expo.bmp', hole=True)
    print im_bias.shape
    print_info(im_bias, 'First bias')
    print_info(im_dark_max, 'Max exposure dark')
```

```

# Plotting
font = {'size' : 12}
plt.matplotlib.rcParams['font', **font]
rcParams.update({'figure.autolayout': True})

hist_bias, bins_bias = np.histogram(im_bias, bins = 'auto')
hist_dark, bins_dark = np.histogram(im_dark_max, bins = 'auto')

plt.figure(1)
plt.title('Histogram pixel value bias1 frame')
plt.bar(bins_bias[:-1], hist_bias, width=bins_bias[-1]/100)
plt.xlabel('Pixel value')
plt.ylabel('Number of pixels')
plt.savefig('hist_pixel_bias1.pdf')
plt.show()

plt.figure(2)
plt.title('Histogram pixel value max expo dark frame')
plt.bar(bins_dark[:-1], hist_dark, width=bins_dark[-1]/100)
plt.xlabel('Pixel value')
plt.ylabel('Number of pixels')
plt.savefig('hist_pixel_darkmax.pdf')
plt.show()

```

### C.3. Mean and noise from combining two calibrating frames

```

import numpy as np
from math import sqrt
import imageio # may be installed with: pip install imageio
from histogram_BIAS_max_dark import reading, print_info

def meantwo(im1, im2):
    """
    Returns mean value of the sum of two images
    @ im1 - 2D array image1
    @ im2 - 2D array image2
    """
    added = im1+im2
    return added.mean()

def noisetwo(im1, im2):
    """
    Returns noise from two images by standard deviation of the differance
    @ im1 - 2D array image1
    @ im2 - 2D array image2
    """
    subbed = im1-im2
    return subbed.std()

if __name__ == '__main__':
    # Path extension
    path = '../images/ex3_4_5/'

    # Read images
    im_bias1 = reading(path+'bias_min_expo1.bmp')
    im_bias2 = reading(path+'bias_min_expo2.bmp')
    im_flat1 = reading(path+'flatfield_1.bmp')
    im_flat2 = reading(path+'flatfield_2.bmp')

    # Computing
    mean_bias = meantwo(im_bias1, im_bias2)
    noise_bias = noisetwo(im_bias1, im_bias2)
    mean_dark = meantwo(im_flat2, im_flat1)
    noise_dark = noisetwo(im_flat1, im_flat2)
    g = (mean_dark-mean_bias)/(noise_dark**2-noise_bias**2)
    noise_one_bias = noise_bias/sqrt(2.)
    RON = g*noise_one_bias

    # Printing

```

```

print.info(im_bias1+im_bias2,'Composite bias')
print 'Noise of two bias frames: {:.5f}\n'.format(noise_bias)
print 'Noise of a single bias frame: {:.5f}\n'.format(noise_one_bias)
print.info(im_flat1+im_flat2,'Composite flat')
print 'Noise of two flat frames: {:.5f}\n'.format(noise_dark)
print 'Conversion factor:'
print 'Value in electrons/ADU: {:.5f}\n'.format(g)
print 'Readout noise:'
print 'Noise from one bias frame in electrons: {:.5f}'.format(RON)

```

#### C.4. Calculating and plotting normalized noise in flat fields

```

import numpy as np
import imageio # may be installed with: pip install imageio
import matplotlib.pyplot as plt
from matplotlib import rcParams
from math import sqrt
from histogram_BIAS_max_dark import reading
from mean_noise import noisetwo

def noise_dynamic_normed(images):
    """
    Returns normalized noise of dynamic number of images
    @ images[N,[x,y]] - list with 2D arrays as elements
        @ N - number of images in images, min value 2
        @ x - number of pixels in x-direction
        @ y - number of pixels in y-direction
    """
    N = np.size(images, axis=0)
    if N == 2:      # if only two images
        return noisetwo(images[0],images[1])
    else:
        n = N/2.      # number of pairs
        even = np.zeros_like(images[0])
        odd = np.zeros_like(images[0])
        for i in np.arange(0,N-1,2):
            even += images[i]          #summing images with even index
            odd += images[i+1]         #summing images with odd index
        return noisetwo(odd,even)/n    # returning normalized noise

if __name__ == '__main__':
    # Path extension
    path = '../images/ex3_4_5/flatfield_'

    # Reading
    N_images = 16

    many_im = []
    for k in range(1,N_images+1):
        ending = str(k)+'.bmp'
        many_im.append(reading(path+ending))

    # Calculating and filling arrays
    n = np.arange(1,N_images/2+1)    # pairs array
    noise_pairs = []                 # list with noise from pairs
    for i in range(N_images/2):
        noise_pairs.append(noise_dynamic_normed(many_im[:2*i+2]))
    exact_one_pair = noise_dynamic_normed(many_im[:2])

    # Plotting
    font = {'size' : 12}
    plt.matplotlib.rc('font', **font)
    rcParams.update({'figure.autolayout': True})

    plt.title('Noise in composite flat field images')
    plt.plot(n,exact_one_pair/np.sqrt(n),'-.',label='exact')
    plt.plot(n,noise_pairs,label='normalized sum')
    plt.ylabel('Standard deviation of pixel values')
    plt.xlabel('Number of pairs in composite image')

```

```

plt.tight_layout()
plt.legend()
plt.savefig('noise_composite.pdf')
plt.show()

```

### C.5. Correcting final image and plotting pixel counts

```

import numpy as np
import imageio # may be installed with: pip install imageio
import matplotlib.pyplot as plt
from matplotlib import rcParams
#from math import sqrt
from histogram_BIAS_max_dark import reading, print_info
import sys

if __name__ == '__main__':
    # Path extension
    path = '../images/ex3_4_5/'

    ## Reading
    N_flats = 16
    N_darks = 5
    hole = True

    raw_I = reading(path+'diffraction1_exercise4.bmp', hole)

    many_flats = []
    many_dark_flats = []
    many_dark_raw = []

    for k in range(1, N_flats+1):
        ending = 'flatfield_'+str(k)+'.bmp'
        many_flats.append(reading(path+ending, hole))
    for k in range(1, N_darks+1):
        ending = str(k)+'.bmp'
        many_dark_flats.append(reading(path+'dark_flat'+ending, hole))
        many_dark_raw.append(reading(path+'dark_same_expo_'+ending, hole))
    many_flats = np.asarray(many_flats)
    many_dark_flats = np.asarray(many_dark_flats)
    many_dark_raw = np.asarray(many_dark_raw)

    ## Calculating
    avr_flats = np.sum(many_flats, axis=0)/N_flats
    avr_dark_flats = np.sum(many_dark_flats, axis=0)/N_darks
    avr_dark_raw = np.sum(many_dark_raw, axis=0)/N_darks

    master_flat = avr_flats - avr_dark_flats
    normed_master_flat = master_flat/master_flat.mean()

    corrected_I = ((raw_I-avr_dark_raw)/np.round(normed_master_flat)).astype(np.uint8)
    print_info(corrected_I, 'corrected_I')

    filename = 'corrected_I'
    if hole:
        filename += '_hole'
    filename += '.bmp'
    imageio.imwrite(filename, corrected_I)
    if hole != True:
        sys.exit()

    shine_crit = 82
    mini_crit = 4

    sumrows = np.sum(corrected_I, axis=1)/752      # mean row values
    shiningrows = corrected_I[sumrows>shine_crit]   # rows with high brightness
    focus = shiningrows[1]                           # focus row
    for minimas
    minimas = np.asarray(np.nonzero(focus<=mini_crit))[0]  # minimas indices

```

```

trueminis = [minimas[4]] # picked start ,
for i in range(1, len(minimas)):
    if abs(minimas[i] - truemini[-1]) >= 10:
        truemini.append(minimas[i])

sumrows_raw = (np.sum(raw_I, axis=1)/752).astype(int) # mean row values
shining_raw = corrected_I[sumrows_raw >= sumrows_raw.max()] # rows with high
brightness
mean_raw = corrected_I[sumrows_raw == int(sumrows_raw.mean())] # rows with mean
brightness

print_info(raw_I, 'Raw image')
print_info(shining_raw[0], 'bright row')
print_info(mean_raw[10], 'mean row')

## Plotting
font = {'size': 12}
plt.matplotlib.rcParams['font', **font]
rcParams.update({'figure.autolayout': True})

plt.figure(1)
plt.subplot(211)
plt.title('Pixel values in two rows of raw image')
plt.plot(shining_raw[0], label='Pixel row %d' % (np.nonzero(sumrows_raw >= sumrows_raw.
max())[0][0]))
plt.ylabel('Pixel value')
plt.xlabel('Pixel number')
plt.tight_layout()
plt.legend()
plt.subplot(212)
plt.plot(mean_raw[10], label='Pixel row %d' % (np.nonzero(sumrows_raw == int(sumrows_raw.
mean()))[0][10]))
plt.ylabel('Pixel value')
plt.xlabel('Pixel number')
plt.tight_layout()
plt.legend()
plt.savefig('pixelrows.pdf')

plt.figure(2)
plt.title('Amount of pixels between minima of order 4')
plt.plot(focus, label="Signal")
plt.plot([trueminis[0], truemini[-2]], [focus[trueminis[0]], focus[trueminis[-2]]], '-o')
plt.text(0, 150, "Line between\n4th minima\nLength = %d" % (trueminis[-2] - truemini[0]))
plt.arrow(0, 145, 300, -140, width = 1)
plt.ylabel('Pixel value')
plt.xlabel('Pixel number')
plt.tight_layout()
plt.legend()
plt.savefig('pixellength.pdf')

plt.show()

```