October 19, 2018
Preprint typeset using LaTeX style emulateapj v. 11/26/04

APPLICATION OF CHARGE-COUPLED DEVICES IN ASTRONOMICAL IMAGING AND DATA PREPROCESSING

Jowita Borowska
Institute of Theoretical Astrophysics, University of Oslo, P.O. Box 1029 Blindern, N-0315 Oslo, Norway; jowitab@student.matnat.uio.no
Laboratory exercise conducted September 27, 2018; Report submitted October 19, 2018

ABSTRACT

Experiments involving a digital camera are conducted in order to study the principles of operation of a Charge-Coupled Device (CCD). First laboratory set-up involves a white light source, aligned with a singlet lens, which enables an observation of the chromatic aberration effect. Camera properties are altered to examine their impact on the image - it appears that increase of the pixel clock causes the image to get darker, similarly a frame rate increase. However, increasing the exposure time produces a brighter image. The source of light in the second experimental set-up is a monochromatic laser. By placing a narrow, 100 μm, slit on the optical path, we are able to observe a Fraunhofer diffraction pattern in the live view of the camera and record it. The image is used to determine the size of pixels of the sensor, equal $\Delta x = 6.08 \pm 0.34$ μm. Afterwards, bias, dark current and flatfield frames are recorded. Acquired data is subsequently processed in Python. For instance, the mean and extrema of pixel counts are computed, as well as the noise distribution of the images and composite of images. The conversion factor, g, relating analog-to-digital units (ADU) to the number of electrons is also measured, $g = 80.13$ electrons/ ADU. Finally, the diffraction pattern image is corrected for all the mentioned effects, which result can be seen on the Figure 6.

Subject headings: CCD detectors, diffraction - Fraunhofer pattern, image processing - bias, dark current, black field

1. INTRODUCTION

The invention of Charge-Coupled Devices (CCDs) has brought an immense improvement of astronomical imaging. Various kinds of detectors have been in use for astronomical observations throught the history, for instance the human eye (being the first one) or photographic plates. Nevertheless, CCDs are undoubtedly the most efficient image sensors among them, used not only for scientific purposes, but also in everyday life (digital cameras).

The CCD is a semiconductor device. Esentially, it consists of an array of pixels formed on the silicon surface of an integrated circuit (Serway & Jewett 2014). While the detector is being illuminated by the electromagnetic radiation, valence band electrons from the p-type silicon layer are excited by photons into the conduction band. These charges are stored over the desired exposure time and subsequently transfered from pixel to pixel for the read-out procedure (AST2210 2015). The number of photo-electrons generated in each pixel, associated with the intensity of light striking the surface, must finally be counted and converted into the digital representation, possible to display on the screen (Serway & Jewett 2014).

Some of the great advantages of CCDs over the other types of detectors include their high *quantum efficiency (QE)*, which gives the fraction of incident photons that can be transfered into the measurable signal of photo-electrons. The QE of CCDs can reach up to 90% at visible light range (to compare, quantum efficiency of photographic film is around 10 %), PHY217 (2014).

However, data obtained from CCD detectors must be processed before a proper analysis, due to some undesirable effects. This pre-processing includes removal of bias level and dark current, as well as flat-fielding.

Bias level is the constant voltage applied to the capacitor measuring the charge before analog-to-digital conversion to account for the readout noise. Therefore, taking the shortest exposure time-image in total darkness, instead of zero counts, a small value is returned, which must then be subtracted.

Another difficulty occurs due to the fact that electrons may be excited into the conduction band not only by photons, but also thermally. These charges are indistinguishable from photo-electrons and the effect of thermal excitation itself is called dark current.

There are more effects, in addition to the ones mentioned above, that account for non-identical responsivities of pixels, even during an illumination by a perfectly uniform source of light. Sensitivity differences are caused, for instance, by pixel-to-pixel variations of QE or their size, dust particles on the optical surfaces, or simply the construction of a telescope, making off-axis sources appear dimmer (*vignetting*). Flatfield correction is meant to compensate for all these effects on the acquired images (whole passage PHY217 (2014)).

In the following sections we aim to study the camera usage and its settings, determine the size of pixels with the aid of diffraction pattern, as well as correct the "raw" image of the said diffraction pattern for bias, dark current and flat field.

2. METHODS

The first experimental set-up, we use, consists of a white light source and a thin singlet lens, aligned with a microcope objective and the color Edmund Optics USB camera, which is connected to the computer. We study the impact of pixel clock, frame rate and exposure time on the live view of the image by changing these camera settings. Afterwards, the focus position of the microscope objective is being altered, in order to reach maxi-

mum pixel values for each (RGB) color, keeping the exposure time fixed.

Next set-up includes a laser, a 100 μm slit and the monochromatic Edmund Optics USB camera. Since we do not use a dampening filter, not exposing the camera by laser directly (without the slit on the light path) is crucial to keep the sensor undamaged and avoid its non-linear behavior.

By aiming the laser on the slit, we are able to observe a light pattern in the live view of the camera, called Frauhofer diffraction pattern. Clearly, following Huygens' principle, the light intensity distribution, that is displayed on the screen, is angle-dependent (Serway & Jewett 2014) and the flux minima occur for

$$\sin(\theta) = \frac{m\lambda}{a}, \tag{1}$$

where a is the slit width, m, the order of the minimum and λ, the wavelength of the monochromatic laser light. Since the distance between the camera and the slit is much larger than the pattern sizes, θ is a very small angle and can be approximated by (result of truncation of the Taylor series for $\sin(\theta)$ and $\tan(\theta)$)

$$\theta \approx \sin(\theta) \approx \tan(\theta) = \frac{x}{s}, \tag{2}$$

where x is the distance between a reference center of the symmetrical diffraction pattern and a chosen minimum, and s is the one between the slit and the CCD detector inside the camera. Since x is measured in the number of pixels displayed on the screen, we can denote it as $x = N\Delta x$, where N is the number of counted pixels and Δx is the width of one pixel. All the formulas above, combined and rearranged, yield the expression for the size of the pixel:

$$\Delta x = \frac{m\lambda s}{aN}. \tag{3}$$

Later, the laser is switched off and the slit removed. Our purpose is from now to record a number of exposures for further processing and correction of the previously obtained diffraction pattern image. We put the dust cover on the camera and record two bias frames (B1 and B2), setting the exposure time to the minimal possible value. Then, five dark current frames are recorded, keeping the same exposure time as the diffraction pattern image has been taken with, as well as one dark frame at the maximum exposure time. Afterwards, the dust cover is removed from the camera. We use a sheet of white paper to reflect the light from the ceiling into the camera and record sixteen flatfield frames. Finally, we close the dust cover again and take five dark frames at the same exposure time as the flats.

Data acquired during the laboratory session are subsequently processed and analyzed using Python (see Appendix B for the code).

First, one of the bias frames is being compared to the dark frame with maximum exposure time, by computing average pixel value and finding its extrema. We also plot histograms of the pixel counts and locate the pixel with maximum value in the 752x480-pixel array of each image.

From now on, we slice the pixel arrays representing the images, and work on a central square region - 300x300 pixels. Both bias frames are loaded, added together and the mean value, $\overline{B}_1 + \overline{B}_2$, is calculated. Next, one bias frame is subtracted from another and the standard deviation, $\sigma_{B_1-B_2}$, is measured. The same procedure is performed on the first two flatfield frames.

All the quantities above are measured in analog-to-digital units (ADU), or simply, pixel counts. As already touched upon in the introduction section, during the readout process, the amount of charge from each pixel is being measured. In principle, the charge is being moved to the capacitor, which causes a voltage change, $V = Q/C$ (where Q is the charge and C is the capacitance). This analoge voltage is measured and didgitalized with use of the analogue-to-digital converter (ADC). Hence, the values obtained in this process are represented in analogue-to-digital units (PHY217 2014). For a 8-bit ADC, pixel counts in this representation can yield $2^8 = 256$ different values, where 0 ADU refers to the completely black pixel and 255 ADU to the brightest, white pixel. We can then determine the conversion factor, g, relating ADU to the actually measured charge (in number of electrons). The more intensive illumination of the CCD, the bigger impact of *shot noise* (noise associated with photons striking the CCD surface in a random manner) on the entire signal's noise. Following Poisson's statistics, the standard deviation of the collected charge (in number of electrons) is proportional to the square root of the number of incident photons (Janesick 2001). As CCDs display linear response - basically, at a fixed energy of light: 1 photon $\Rightarrow$ 1 electron ("CCD theory" 2006) - we can equally write the number of generated electrons under the square root, $\sigma_{\text{electrons}} = \sqrt{F_{\text{electrons}}}$. Here, F indicates that we use flatfield frames, since these are the only properly illuminated images (the other frames were taken with the dust cover over the camera). What actually has been measured while taking the mean value of the signal and the noise level (standard deviation), are both quantities multiplied by the conversion factor, $g\sigma_{\text{electrons}} = \sqrt{gF_{\text{electrons}}}$, thus

$$g = \frac{F_{electrons}}{\sigma^2_{\text{electrons}}} \left[\frac{\text{electrons}}{\text{ADU}}\right]. \tag{4}$$

Implementing the correction for the bias and the noise related to the bias frames, we get the final expression for the conversion factor:

$$g = \frac{(\overline{F}_1 + \overline{F}_2) - (\overline{B}_1 + \overline{B}_2)}{(\sigma^2_{F_1-F_2}) - (\sigma^2_{B_1-B_2})} \left[\frac{\text{electrons}}{\text{ADU}}\right], \tag{5}$$

where $\overline{F}_1$, $\overline{F}_2$, $\overline{B}_1$ and $\overline{B}_2$ are the mean pixel values of consecutively two flatfield and two bias frames, and $\sigma^2_{F_1-F_2}$, $\sigma^2_{B_1-B_2}$ are variances of these frames subtracted from each other.

As already explained, bias level accounts for the digitalization noise (during the readout procedure). The readout noise, being the only source of noise in the bias frame is

$$\text{R.O.N.} = g\sigma_{B_1}\ [\text{electrons}], \tag{6}$$

where the standard deviation of the first bias frame, B_1, is chosen (ideally, both frames should have the same noise, so it is unimportant which one we choose).

We can now combine flatfield images, pair by pair (up to eighth pair/ sixteenth frame taken). Successively, we add one of the frames from a given pair, subtracting the other one, and compute the normalized noise:

$$\sigma_{F_1-F_2}/1,$$
$$\sigma_{(F_1+F_3)-(F_2+F_4)}/2,$$
$$\sigma_{(F_1+F_3+F_5)-(F_2+F_4+F_6)}/3,$$
$$...,$$
$$\sigma_{\sum_{i=1}^{16}(-1)^{(i+1)}F_i}/8.$$

Normalization of the noise is then carried out by the division of standard deviation of a given combination of images by the number of pairs used. This procedure yields the more accurate noise evolution in the course of combining the images than just the standard deviation.

The highlight of all of the actions performed until this point is correction of the diffraction pattern image. In order to do that, a *master* flat image is constucted as the difference between the average of all flatfield frames, F_i, and average of all dark current frames, D_i, taken with the same exposure time as the flats:

$$F_{\text{master}} = \frac{\sum_{i=1}^{16} F_i}{16} - \frac{\sum_{i=1}^{5} D_i}{5} = F_{\text{average}} - D_{\text{average}}. \quad (7)$$

Then, we normalize the master flat image, dividing by its scalar mean (average pixel value):

$$F_{\text{master,normalized}} = \frac{F_{\text{master}}}{\overline{F}_{\text{master}}}. \quad (8)$$

The final correction of the diffraction pattern image is performed by the following operation:

$$I_{\text{corrected}} = \frac{I_{\text{raw}} - d_{\text{average}}}{F_{\text{master,normalized}}}, \quad (9)$$

where I_{raw} is the image to be corrected and d_{average} is the average of all five dark current frames taken with the same axposure time as the diffraction pattern image ($d_{\text{average}} = \sum_{i=1}^{5}(d_i)/5$).

3. RESULTS

Figure 1. shows the window of camera properties in the *camera viewer*. We observe that increasing pixel clock value results in the image getting dimmer. Frame rate-setting follows the same relation. However, increase in the exposure time produces a brighter image.

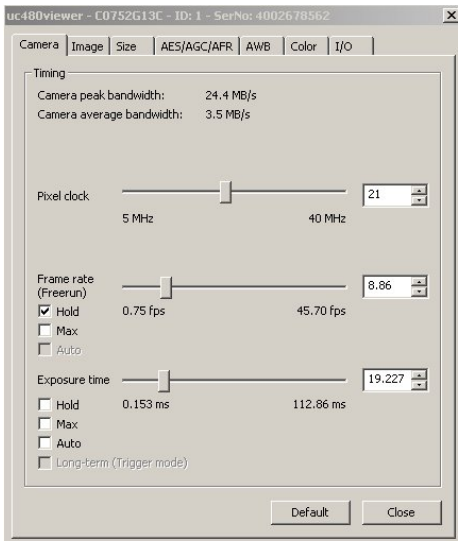

FIG. 1.— Camera properties in the *camera viewer*, which have been altered: pixel clock, frame rate and exposure time.

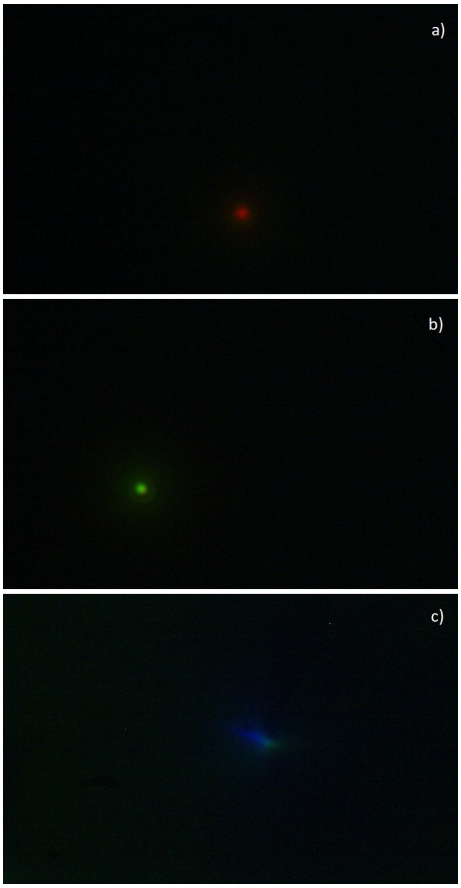

FIG. 2.— Images with maximum counts for each RGB color: a) red, b) green, c) blue, obtained by changing the distance between the microscope objective and the lens.

In the first experimental set-up, the white light source is used. By changing the focus position of the microscope objective, placed after the lens on the optical path, we reach maximum intensity of light for each RGB color. By moving the objective toward the lens, maximum red light intensity is reached, further toward the lens - green, and finally, position closest to the lens gives the maximal blue-intensity (shown in succession on the Figure 2 a), b) and c)). The maximal pixel values of each frame, that have been taken, are as follows: 125 ADU for red maximum, 172 ADU for green maximum and 122 ADU for blue maximum (although in this case, the focus is not entirely on the blue light, so that the actual maximal pixel value falls for the green color and is equal 189 ADU).

Afterwards, we use the second experimental set-up with the monochromatic laser as a source of light. We study the diffraction pattern, being a result of placing a narrow, 100μm, slit on the optical path. The pattern consists of the intense broad band of light (central maximum) and a series of less intense and narrower fringes, occuring alternately with dark fringes (minima). Image of the pattern itself, shown on the Figure 3 (on the next page), is taken with the following camera settings: pixel clock - 12 MHz, frame rate - 15.76 fps, exposure time - 48.314 ms.

The size of pixels of the sensor can be determined with use of the diffraction pattern image. As can be seen on the Figure 3, the minimum of $m = 3$ order occurs the distance $N = 210$ pixels away from the reference center

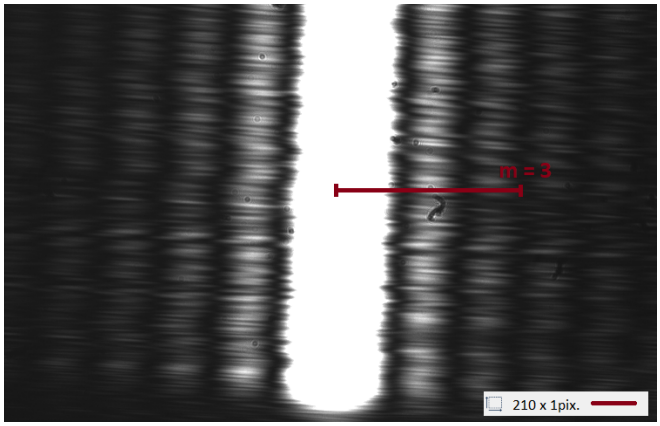

FIG. 3.— Fraunhofer diffraction pattern from a single narrow slit (third order of minimum, $m = 3$, is marked, as it is used for pixel-width calculation).

of the symmetrical pattern. Due to the blurriness of the image, we acknowledge ± 10 pix. uncertainty of this assessment. Furthermore, the nominal wavelength of the laser light, that is being used, is $\lambda = 635$ nm and the distance between the slit and the core of the camera is measured to be $s = 6.7 \pm 0.2$ cm. Inserting these values into the formula (3) and calculating the uncertainty (based on the description in the Appendix A) yields the width of pixels of the sensor, $\Delta x = 6.08 \pm 0.34$ μm (assuming that all pixels are identical).

Subsequently, the bias, dark current and flatfield frames are recorded, as described in the method section. Data pre-processing in Python (see the code in Appendix B) starts with the comparison between one of the bias frames (taken at minimal exposure time, 0.268 ms) and the dark frame with maximal exposure time (set to 373.03 ms). All the following values are given with decimals just for the accuracy reference, ADU are in fact natural numbers (so is the quantity of electrons). Average pixel value is computed to be 7.46 ADU for the bias frame and 7.65 ADU for the dark frame. Whereas, minimum and maximum counts are consecutively 5 and 22 ADU for the bias frame; 5 and 26 ADU for the dark one. We find as well the localization of the pixel with maximal value in the 752x480-pixels array of each frame. This position appears to be the same, and its coordinates are $(x, y) = (427, 439)$. Moreover, histograms of the pixel values are plotted and shown on the Figure 4.

Next, both bias frames are loaded. We add them together and compute the mean value of the central square region, as stated in the method section, $\overline{B}_1 + \overline{B}_2 = 14.99$ ADU. Then, the standard deviation of the frames subtracted from each other is determined to be $\sigma_{B_1 - B_2} = 0.65$ ADU (while the noise of just one bias frame is 0.57 ADU). The same procedure conducted for two flatfield frames yieds $\overline{F}_1 + \overline{F}_2 = 161.35$ ADU and $\sigma_{F_1 - F_2} = 1.50$ ADU.

The conversion factor, g, relating analog-to-digital units (ADU) to the quantity of measured electrons is computed, $g = 80.13$ electrons/ADU . Furthermore, having found g, we can use equation (6) in order to determine the readout noise in electrons for one of the bias frames (B_1), $R.O.N. = 45.80$ electrons.

Successively combining flatfield images, we compute the normalized noise of composite-flats and plot it as a function of number of pairs of frames used, n (see Figure 5). The theoretical prediction that the normalized noise should decrease by the square root of n (using the frame-composite of the first pair as a starting reference) is chosen to be plotted together with the actual results obtained from the data-processing. Theory of the noise composition is more explicitly described in the discussion section.

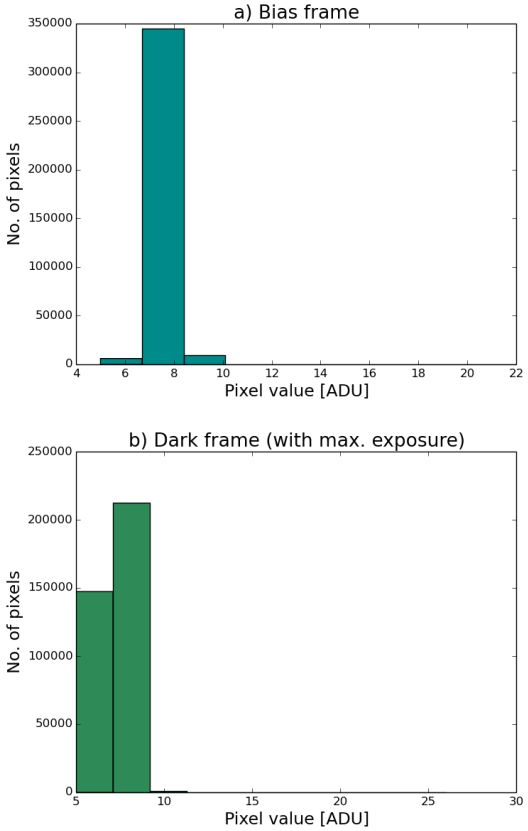

FIG. 4.— Histograms of the pixel values for a) the bias frame, b) the dark current frame with the maximal exposure time possible.

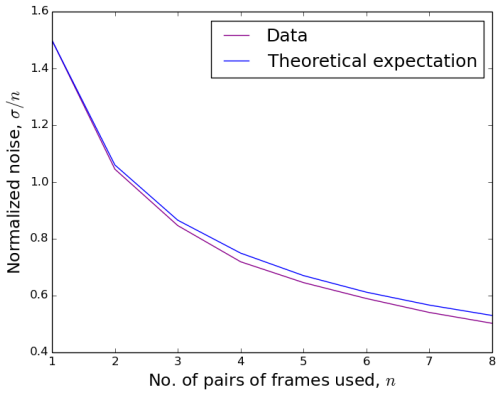

FIG. 5.— Normalized noise, σ/n, of a composite of flat frames as a function of number of pairs of flatfield frames being used in the composition, n. Theoretically predicted decrease by $1/\sqrt{(n)}$ is plotted together.

Following the final procedure, leading to the diffraction pattern image correction, we arrive at the results shown in Figure 6, where the central square region of the raw image is shown (a), together with the corresponding region of the corrected image (b).

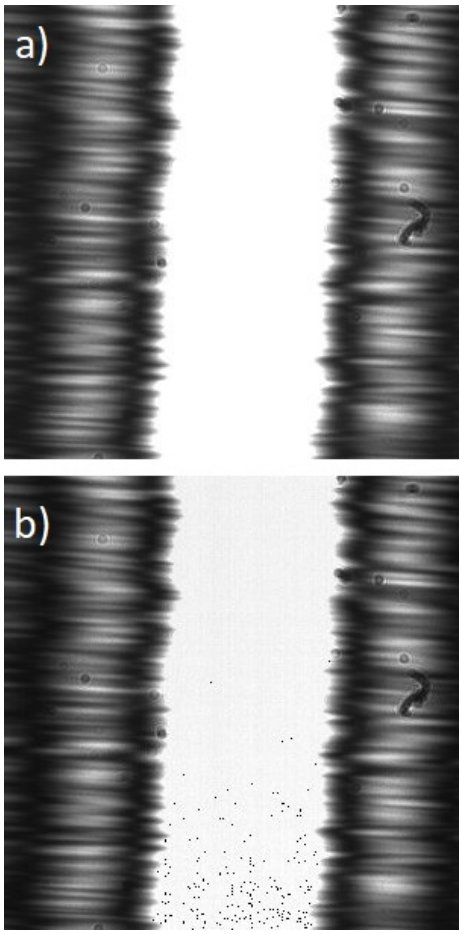

FIG. 6.— The central square region (300x300 pix.) of the diffraction pattern a) without the flatfield correction, b) after the correction.

4. DISCUSSION AND CONCLUSION

Every action performed in the laboratory is under the controll of camera behavior. Adjusting camera properties has a clear impact on the image. Pixel clock frecuency determines the rate at which the charge from every pixel is transmitted within a cycle. We see that increasing its value causes the image to be darker. That seems understandable, as the higher frequency of electron transmission from pixels, the smaller quantity of them can be measured per cycle. This corresponds to the lower pixel values (darker image). Moreover, increasing the frame rate (frequency of the mentioned cycles - number of frames per second) has the same impact on the image and similar explanation - the faster charge is emptied and transmitted from pixels, the less charge is acquired per frame. Contrarily, longer exposure time causes the image to appear brighter. During the longer exposure, more photons strike the CCD surface and thus, more electrons are excited and can be transmitted further, which results in the higher pixel values (brighter image).

The first experimental set-up uses a white light lamp as the light source. White light, consisting of the range of wavelengths, is split while passing through a singlet lens due to variation in refractive indices of different light colors (*dispersion*). The fact that rays of individual light-colors do not focus at the same point after passing through the lens is called *chromatic aberration* and is illustrated on the Figure 7 (Petersen 2016).
Chromatic aberration is the cause of the variation of the distance between the lens and the microscope objective, while placing it in order to reach the maximal pixel counts for each color (results shown in Figure 2). Microscope objective is placed on the right side in reference to the Figure 7, so moving it futher and further toward the lens shifts the focus-overlay from red to green to blue, as observed.

Proceeding to the data processing, the first interesting observation is that the bias frame and the dark frame with maximum exposure time display many similarities. Average pixel values are almost identical (7.46 ADU for the bias and 7.65 ADU for the dark frame), as well as the pixel values extrema. Moreover, the position of the pixel with maximum counts is exactly the same for each frame. However, histograms on the Figure 4 show that the pixel value-distribution is diffferent. This difference is still not so crucial, since for both frames the majority of pixels take the values between 5 and 10 ADU. The reason behind this resemblance is that noise of a dark frame with such a long exposure time is mostly the readout noise (so the same origin as in bias frames). The digitazlization noise has simply a bigger impact on the image than the dark current in the dark frame, so that both images appear very similar. The bias frame is, after all, a dark frame with an extremely short exposure time, so that thermal electrons do not have enough time to accumulate.

We consider noise as the standard deviation, σ, of an image or a composite of images. If the two variables to be combined are independent, the variance (standard deviation squared) of this combination is the sum of variances of each variable. Combining both bias frames, which ideally should have identical noise values, we have

$$\sigma^2_{B_1 \pm B_2} = \sigma^2_{B_1} + \sigma^2_{B_2} = 2\sigma^2_{B_1} \Rightarrow \sigma_{B_1 \pm B_2} = \sqrt{2}\sigma_{B_1}. \quad (10)$$

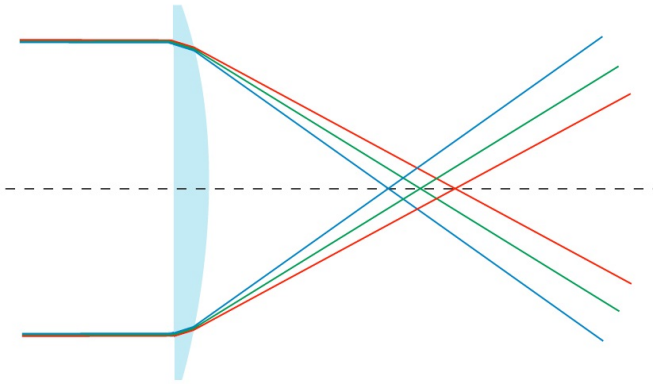

FIG. 7.— Chromatic aberration - red, green and blue rays of light do not converge at the same point after passing through the singlet lens, altered figure from Petersen (2016).

The noise of two bias frames subtracted from each other should be equal to the noise of one bias frame multiplied by $\sqrt{2}$. Results of computation yield $\sigma_{B_1-B_2}= 0.65$ ADU and $\sigma_{B_1}= 0.57$ ADU, hence the formula (10) not exactly holds. The reason is that by subtracting two bias frames, any fixed pattern is removed, so that 0.65 ADU corresponds purely to the noise of the bias frame-composite, while 0.57 ADU is the standard deviation of the whole single bias image.

Subsequently, while combining more images (this time flatfield), addition of variances follows the same rule. Hence, the standard deviation will increase by a factor of $\sqrt{n_{\text{pairs}}}$, with every pair of frames added to the composite of n_{pairs}. Nevertheless, it is more relevant to measure normalized noise, since combining many frames will not only smoothen the noise, but also amplify some noise-peaks. The normalized noise (σ/n_{pairs}) will decrease by a factor of $\sqrt{n_{\text{pairs}}}/n_{\text{pairs}} = 1/\sqrt{n_{\text{pairs}}}$. This theoretical prediction is in a very good agreement with the data-processing results (see the Figure 4).

Analizing the raw diffraction pattern image (Figure 6 a)), we are able to observe some dust particles on the optical surface, creating their own diffraction patterns. These features are obviously not eliminated in the course of image correction. In fact, barely any improvement is noticeable (see Figure 6 b)). Some pixel values have been clearly destructively subtracted, resulting in the black dots visible on the processed image.

Construction of our flatfield images involves reflecting the light from the ceiling into the camera, using a sheet of paper. It is very difficult to achieve an even illumination this way, which introduces a big problem for the accurate image correction. Additionally, the light intensity distribution of the actual diffraction pattern, being much different than the one of a flatfield frame, is yet another flaw.

APPENDIX

A: UNCERTAINTY CALCULATION

Uncertainties of the quantities in the formula (3) propagate to the uncertainty of the result. Since the formula involves only multiplication and division, fractional uncertainties add in quadrature, following "Error Propagation" (2007), and the error in Δx-measurment is

$$\frac{\delta(\Delta x)}{|\Delta x|} = \sqrt{\left(\frac{\delta s}{s}\right)^2 + \left(\frac{\delta N}{N}\right)^2}, \tag{A1}$$

since m, λ and a are certain quantities.

B: CODE

All the programs are followed by the import statement:

```
from PIL import Image
from numpy import *
import matplotlib.pyplot as plt
```

B1) RGB pixel counts

```
red = Image.open('calibrating_red.bmp')
green = Image.open('calibrating_green.bmp')
blue = Image.open('blue_max.bmp')

min_max_red = red.getextrema()
min_max_green = green.getextrema()
min_max_blue = blue.getextrema()

print min_max_red, min_max_green, min_max_blue
#((0, 125), (0, 31), (0, 20)) ((0, 73), (0, 172), (0, 20)) ((0, 19), (0, 189), (0, 122))
#consecutively: (red extrema), (green extrema), (blue extrema)
```

B2) Bias and dark frame with max. exposure time analysis

```
bias = Image.open('bias_1.bmp')
df_max = Image.open('dark_frame_max.bmp')

bias_data = array(bias.getdata())
df_max_data = array(df_max.getdata())

#AVERAGE
average_bias = average(bias_data) #7.45917553191
average_df_max = average(df_max_data) #7.64765070922
```

```
#MIN and MAX
min_max_bias = bias.getextrema() #(5, 22)
min_max_df_max = df_max.getextrema() #(5, 26)

#HISTOGRAMS
plt.figure()
plt.hist(bias_data, color='darkcyan')
plt.title('a) Bias frame')
plt.xlabel('Pixel value [ADU]')
plt.ylabel('No. of pixels')
plt.savefig('histogram_bias.png')

plt.figure()
plt.hist(df_max_data, color='seagreen')
plt.title('b) Dark frame (with max. exposure)')
plt.xlabel('Pixel value [ADU]')
plt.ylabel('No. of pixels')
plt.savefig('histogram_df_max.png')

#VIEW ON THE SCREEN
bias.show()
df_max.show()

#PIXEL WITH MAX. COUNTS
bias_pixels = asarray(bias)
df_max_pixels = asarray(df_max)

pixel_max_bias = unravel_index(bias_pixels.argmax(), bias_pixels.shape)
#(x,y) = (427, 439), defined according to PIL dimensions

pixel_max_df_max = unravel_index(df_max_pixels.argmax(), df_max_pixels.shape)
#(x,y) = (427, 439)
```

B3) Bias frames, Flat frames, g, R.O.N.

```
#EXERCISE 7
B1 = Image.open('bias_1.bmp')
B2 = Image.open('bias_2.bmp')

B1 = asarray(B1)
B2 = asarray(B2)
B1 = B1.astype(float32)
B2 = B2.astype(float32)

#slicing to the central square region
B1 = B1[90:390, 226:526]
B2 = B2[90:390, 226:526]

meanB1 = mean(B1)
meanB2 = mean(B2)
mean_sum_B = meanB1 + meanB2 #14.9919
mean_B = mean(B1+B2) #14.9919

standard_deviation_B = std(B1 - B2) #0.64837

#EXERCISE 9

F1 = Image.open('flat1.bmp')
F2 = Image.open('flat2.bmp')

F1 = asarray(F1)
F2 = asarray(F2)
F1 = F1.astype(float32)
```

```
F2 = F2.astype(float32)

F1 = F1[90:390, 226:526]
F2 = F2[90:390, 226:526]

mean_sum_F = mean(F1) + mean(F2) #161.353
mean_F = mean(F1+F2) #161.353

standard_deviation_F = std(F1 - F2) #1.49904

#EXERCISE 10

g = (mean_sum_F - mean_sum_B)/(standard_deviation_F**2 - standard_deviation_B**2) #80.121934529

#EXERCISE 11

RON1 = g*std(B1) #45.7996048539, std(B1) = 0.571624
RON2 = g*std(B2) #45.7093595953, std(B2) = 0.570497
```

B4) Normalized noise composition

```
F1 = Image.open('flat1.bmp')
F2 = Image.open('flat2.bmp')
F3 = Image.open('flat3.bmp')
F4 = Image.open('flat4.bmp')
F5 = Image.open('flat5.bmp')
F6 = Image.open('flat6.bmp')
F7 = Image.open('flat7.bmp')
F8 = Image.open('flat8.bmp')
F9 = Image.open('flat9.bmp')
F10 = Image.open('flat10.bmp')
F11 = Image.open('flat11.bmp')
F12 = Image.open('flat12.bmp')
F13 = Image.open('flat13.bmp')
F14 = Image.open('flat14.bmp')
F15 = Image.open('flat15.bmp')
F16 = Image.open('flat16.bmp')

F1 = asarray(F1, dtype=float32)
F2 = asarray(F2, dtype=float32)
F3 = asarray(F3, dtype=float32)
F4 = asarray(F4, dtype=float32)
F5 = asarray(F5, dtype=float32)
F6 = asarray(F6, dtype=float32)
F7 = asarray(F7, dtype=float32)
F8 = asarray(F8, dtype=float32)
F9 = asarray(F9, dtype=float32)
F10 = asarray(F10, dtype=float32)
F11 = asarray(F11, dtype=float32)
F12 = asarray(F12, dtype=float32)
F13 = asarray(F13, dtype=float32)
F14 = asarray(F14, dtype=float32)
F15 = asarray(F15, dtype=float32)
F16 = asarray(F16, dtype=float32)

F1 = F1[90:390, 226:526]
F2 = F2[90:390, 226:526]
F3 = F3[90:390, 226:526]
F4 = F4[90:390, 226:526]
F5 = F5[90:390, 226:526]
F6 = F6[90:390, 226:526]
F7 = F7[90:390, 226:526]
F8 = F8[90:390, 226:526]
F9 = F9[90:390, 226:526]
F10 = F10[90:390, 226:526]
```

```
F11 = F11[90:390, 226:526]
F12 = F12[90:390, 226:526]
F13 = F13[90:390, 226:526]
F14 = F14[90:390, 226:526]
F15 = F15[90:390, 226:526]
F16 = F16[90:390, 226:526]


std1 = std(F1-F2)
std2 = std(F1+F3-F2-F4)
std3 = std(F1+F3+F5-F2-F4-F6)
std4 = std(F1+F3+F5+F7-F2-F4-F6-F8)
std5 = std(F1+F3+F5+F7+F9-F2-F4-F6-F8-F10)
std6 = std(F1+F3+F5+F7+F9+F11-F2-F4-F6-F8-F10-F12)
std7 = std(F1+F3+F5+F7+F9+F11+F13-F2-F4-F6-F8-F10-F12-F14)
std8 = std(F1+F3+F5+F7+F9+F11+F13+F15-F2-F4-F6-F8-F10-F12-F14-F16)

std1_normalized = std1/1.
std2_normalized = std2/2.
std3_normalized = std3/3.
std4_normalized = std4/4.
std5_normalized = std5/5.
std6_normalized = std6/6.
std7_normalized = std7/7.
std8_normalized = std8/8.


std_normalized = array([std1_normalized, std2_normalized, std3_normalized, std4_normalized, \
std5_normalized, std6_normalized, std7_normalized, std8_normalized])
#[ 1.4990375   1.04515481  0.84652829  0.7190721   0.64571095  0.58968504  0.54064965  0.50244117]

frames = array([1.,2.,3.,4.,5.,6.,7.,8.]) #no. of pairs of frames used successively

std_model = std1_normalized/sqrt(frames)
#[ 1.4990375   1.05997958  0.86546971  0.74951875  0.67038995  0.6119795  0.56658292  0.52998979]


plt.figure()
plt.plot(frames, std_normalized, color='darkmagenta')
plt.plot(frames, std_model)
plt.xlabel('No. of pairs of frames used, $n$')
plt.ylabel('Normalized noise, $\sigma /n$')
plt.legend(['Data', 'Theoretical expectation'])
plt.savefig('exercise12.png')
plt.show()
```

B5) Image correction

```
F1 = Image.open('flat1.bmp') #flat fields
F2 = Image.open('flat2.bmp')
F3 = Image.open('flat3.bmp')
F4 = Image.open('flat4.bmp')
F5 = Image.open('flat5.bmp')
F6 = Image.open('flat6.bmp')
F7 = Image.open('flat7.bmp')
F8 = Image.open('flat8.bmp')
F9 = Image.open('flat9.bmp')
F10 = Image.open('flat10.bmp')
F11 = Image.open('flat11.bmp')
F12 = Image.open('flat12.bmp')
F13 = Image.open('flat13.bmp')
F14 = Image.open('flat14.bmp')
F15 = Image.open('flat15.bmp')
F16 = Image.open('flat16.bmp')
D1 = Image.open('df1.bmp') #dark frames, exposure time same as the flats
```

```
D2 = Image.open('df2.bmp')
D3 = Image.open('df3.bmp')
D4 = Image.open('df4.bmp')
D5 = Image.open('df5.bmp')
Diff = Image.open('diff.bmp')
d1 = Image.open('dark_1.bmp') #dark frames, exposure time same as diffraction image
d2 = Image.open('dark_2.bmp')
d3 = Image.open('dark_3.bmp')
d4 = Image.open('dark_4.bmp')
d5 = Image.open('dark_5.bmp')

F1 = asarray(F1, dtype=float32)
F2 = asarray(F2, dtype=float32)
F3 = asarray(F3, dtype=float32)
F4 = asarray(F4, dtype=float32)
F5 = asarray(F5, dtype=float32)
F6 = asarray(F6, dtype=float32)
F7 = asarray(F7, dtype=float32)
F8 = asarray(F8, dtype=float32)
F9 = asarray(F9, dtype=float32)
F10 = asarray(F10, dtype=float32)
F11 = asarray(F11, dtype=float32)
F12 = asarray(F12, dtype=float32)
F13 = asarray(F13, dtype=float32)
F14 = asarray(F14, dtype=float32)
F15 = asarray(F15, dtype=float32)
F16 = asarray(F16, dtype=float32)
D1 = asarray(D1, dtype=float32)
D2 = asarray(D2, dtype=float32)
D3 = asarray(D3, dtype=float32)
D4 = asarray(D4, dtype=float32)
D5 = asarray(D5, dtype=float32)
Diff = asarray(Diff, dtype=float32)
d1 = asarray(d1, dtype=float32)
d2 = asarray(d2, dtype=float32)
d3 = asarray(d3, dtype=float32)
d4 = asarray(d4, dtype=float32)
d5 = asarray(d5, dtype=float32)

F1 = F1[90:390, 226:526]
F2 = F2[90:390, 226:526]
F3 = F3[90:390, 226:526]
F4 = F4[90:390, 226:526]
F5 = F5[90:390, 226:526]
F6 = F6[90:390, 226:526]
F7 = F7[90:390, 226:526]
F8 = F8[90:390, 226:526]
F9 = F9[90:390, 226:526]
F10 = F10[90:390, 226:526]
F11 = F11[90:390, 226:526]
F12 = F12[90:390, 226:526]
F13 = F13[90:390, 226:526]
F14 = F14[90:390, 226:526]
F15 = F15[90:390, 226:526]
F16 = F16[90:390, 226:526]
D1 = D1[90:390, 226:526]
D2 = D2[90:390, 226:526]
D3 = D3[90:390, 226:526]
D4 = D4[90:390, 226:526]
D5 = D5[90:390, 226:526]
Diff = Diff[90:390, 226:526]
Diff_central = Image.fromarray(Diff.astype(uint8))
Diff_central.save('diffraction_central.bmp')
```

```
d1 = d1[90:390, 226:526]
d2 = d2[90:390, 226:526]
d3 = d3[90:390, 226:526]
d4 = d4[90:390, 226:526]
d5 = d5[90:390, 226:526]

F_average = (F1+F2+F3+F4+F5+F6+F7+F8+F9+F10+F11+F12+F13+F14+F15+F16)/16.
D_average = (D1+D2+D3+D4+D5)/5.

F_master = F_average - D_average
F_normalized_master = F_master/mean(F_master)

D_raw_average = (d1+d2+d3+d4+d5)/5.

I_corrected = (Diff - D_raw_average)/F_normalized_master
Image_corrected = Image.fromarray(I_corrected.astype(uint8))
Image_corrected.save('image_corrected.bmp')
```

REFERENCES

A summary of Error Propagation. (2007). Retrieved from: http : //ipl.physics.harvard.edu/wp − uploads/2013/03/ PS3_Error_Propagation_sp13.pdf

CCDs. In *Lecture notes*, AST2210. (2015). Retrieved from: https : //www.uio.no/studier/emner/ matnat/astro/AST2210/h15/materials/main.pdf

CCD theory. (2006). Retrieved from: https : //www.astro.ufl.edu/ lee/ast325/handouts/ccd.pdf

Observational Techniques for Astronomers (PHY217), Lectures: Detectors, Working with CCD data. (2014). Retrieved from: http : //slittlefair.staff.shef.ac.uk/teaching/phy217/

Janesick, J. R. (2001). Scientific Charge-Coupled Devices (pp. 101-102) Washington: SPIE

Petersen, B. (2016). Optical Anomalies and Lens Corrections Explained. Retrieved from: https : //www.bhphotovideo.com/explora/photography/ tips − and − solutions/optical − anomalies − and− lens − corrections − explained

Serway, R. A. and Jewett, J. W. (2014). Diffraction Patterns and Polarization, Introduction to Qunatum Physics. In *Physics for Scientists and Engineers with Modern Physics* (pp.1160-1165, 1245). Boston: Brooks/Cole