

CHEOPS Data Reduction Pipeline

Telescopes and Detectors for Space Sciences - Practical Work

Telmo Monteiro¹

Department of Physics and Astronomy, University of Porto, Rua do Campo Alegre 1021 1055, 4169-007 Porto
e-mail: up202308183@up.pt

April 30, 2024

ABSTRACT

Aims. To partially replicate the official CHEOPS Data Reduction Pipeline (DRP) and extract the light-curve of the planet KELT-11b.

Methods. We calibrated the raw images of KELT-11 using bias, dark and flat-field frames. We also depointed the images by computing the centroid of the star and performed aperture photometry, while subtracting the background flux.

Results. We successfully calibrated and corrected the raw images, obtaining a transit model of the light curve for a radius with a 0.24% error comparing with a tabulated radius.

Key words. methods: data analysis – techniques: image processing – techniques: photometric – space vehicles: instrument – stars: planetary systems – instrumentation: photometers

1. Introduction

CHEOPS (CHAracterizing ExOPlanet Satellite) is an ESA small mission launched in 2019. CHEOPS is designed as a follow-up instrument devoted to ultra high precision photometry, able to detect and/or precisely measure transits of small size exoplanets already known via radial velocity measurements or via transit searches (Hoyer et al (2020)). This mission has as objectives to search for transits, improve mass-radius relation, explore, characterise atmospheres, search for features and perform ancillary science (Benz et al (2021)).

The series of raw images acquired by the instrument were automatically processed into a flux time series, ready for scientific analyses. As part of the Science Operations Centre, the Data Reduction Pipeline (DRP) is in charge of producing these calibrated light curves, with associated intermediate products, which will be delivered to the scientific users (Hoyer et al (2020)).

The objective of this work is to partially reproduce the DRP described in Hoyer et al (2020), exemplified in figure 1, by implementing the steps in the red boxes. The pipeline will be divided in three main modules, similarly to figure 1:

1. Calibration: correction of the instrumental response;
2. Correction: correction of environmental effects;
3. Photometry: transformation of the calibrated and corrected images into a calibrated flux time series or light curve.

In the calibration module, the raw images will be corrected by the bias (or overscan), by the dark current and by the flat-field. In the second module, correction, only the depointing will be made, as the background subtraction is made simultaneously with the aperture photometry in the third module. The depointing and aperture photometry steps will be made in three and two different ways, respectively, which will then be compared with each other. This pipeline will be applied to a data set of a CHEOPS observation of the planet KELT-11 b, as seen in Benz et al (2021).

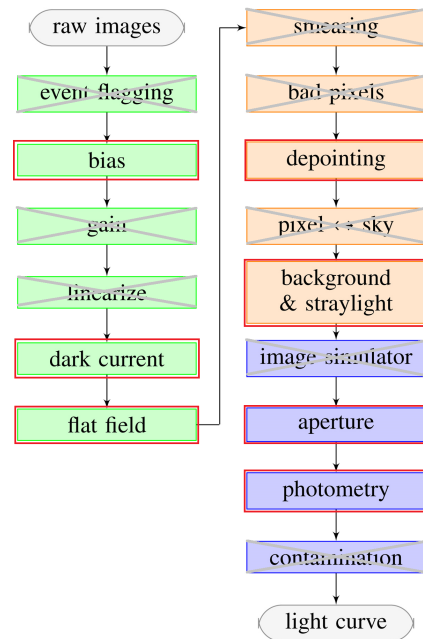


Fig. 1: Data reduction flowchart. Green, orange and blue color are calibration, correction and photometry main modules, respectively. The boxes outlined in red are the ones to implement. Adapted from Hoyer et al (2020).

2. Data

The first case studied was the planet KELT-11b, around KELT-11, a G9V star with a V magnitude in the Gaia band of 7.8311. The data used consist in 1535 observations in the scope of a program led by Andrea Fortier taken during 2 days, 9 and 10 september 2020.

These 1535 observations contain different elements taken from a full-frame image. This full-frame image consist in 1024x1024 pixels plus CCD margins, which is then divided in smaller components:

- subarray: The 200x200 square is the region of interest;
- overscan: or bias, 4x200 in the left, 200x6 in the top;
- dark: 16x200 in left and right, 200x3 in top.

There is also the blank and the imagette, but in this work they will be ignored. Figure 2 shows a schematic view of the photo-sensitive area of the CCD, as described.

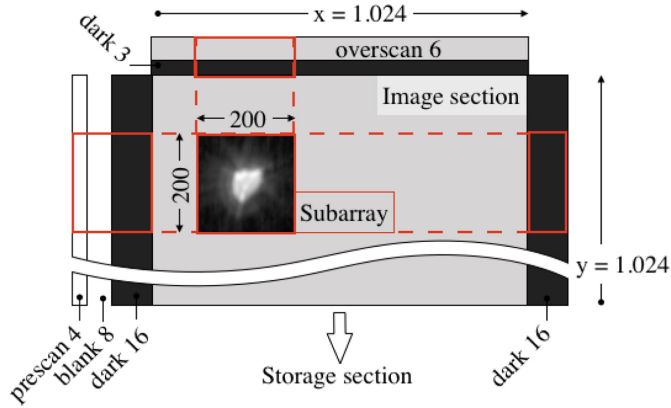


Fig. 2: Schematic view of the photo-sensitive area of the CCD, with the overscan, dark and subarray components. Extracted from Hoyer et al (2020).

3. Calibration

In the calibration module, the raw subarray images will be corrected by subtracting the bias and the dark and dividing it by the flat-field.

3.1. Bias / Overscan

Correcting the raw images by the bias or overscan consist in imposing an electric offset sufficiently large that minimizes problems that might come from small counts and the random fluctuations of the RON (read-out-noise). If there is no significant 2D structure visible in the data, one could subtract the Bias as a constant. If not, then subtract as an image. Visualizing the image was not sufficient to make a decision, so we decided to look at the statistics of each bias image.

The first step was to plot the median flux by row for the left image, as seen in figure 3. Image 944 consisted in an outlier, probably caused by cosmic hits, so in the lowest panel it was removed. As one can see, the median flux by row is not exactly constant over time. The same conclusion can be drawn by looking at figure A.1 in the appendix, which shows the median, mean and standard deviation of the flux in the image as a whole. For this reason, we decided to subtract the bias one image at a time for the respective subarray image, not creating a median master-bias image, a common step. To further analyse if the bias images are as expected, figure 4 shows an histogram of the counts in the image for every observation, except image 944, which was then removed from our data. A histogram of the counts in a bias image should reveal something similar to a Gaussian distribution, which is observed.

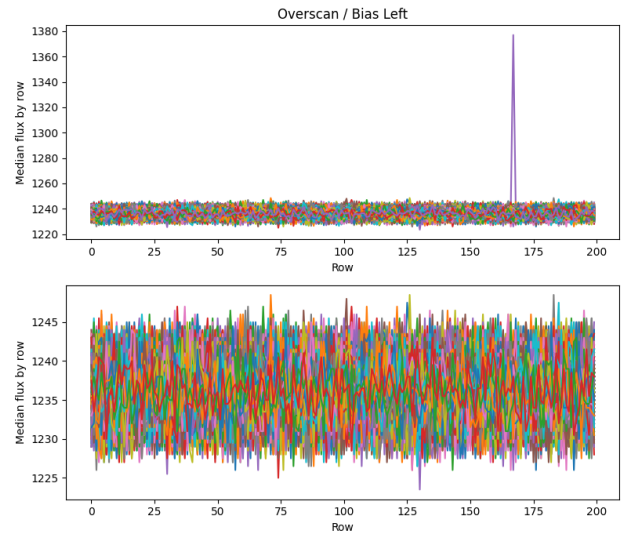


Fig. 3: Median flux by row for every overscan/bias left image. The upper panel shows every observation, while in the lower panel the outlier observation, 944, was removed for better visualization.

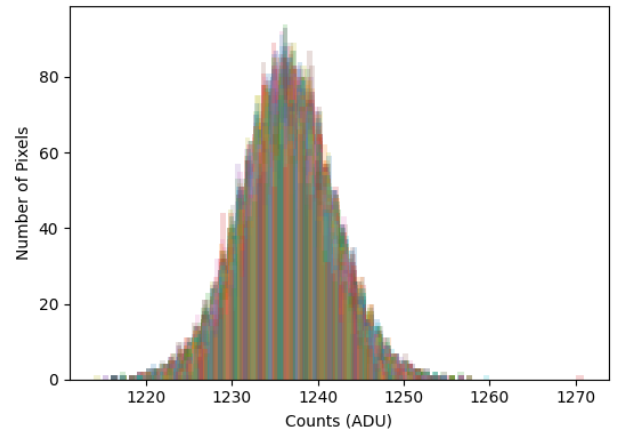


Fig. 4: Histogram of the counts (ADU) for every overscan/bias left image, except 944.

Figure 5 shows the median flux by row for every overscan/bias top image, with several outliers. This can also be seen in figure A.2 in the appendix. For this reason, we decided to discard these images and only use the left bias images.

To perform the subtraction, we decided to compute the median of each row of the bias image, to compress it to a 1D array, and then subtract each row of the subarray image by the correspondent value in the 1D array. We should note that we used median instead of mean, as generally it is more insensitive to outliers.

3.2. Dark current

The dark current consist in charge generated by thermal effects during the exposure, so we need to also subtract it to our subarray images. For the dark current we have left, right and top images. The statistics for these images are shown figures A.3, A.4, A.5, A.6, A.7 and A.8 in the appendix. From these statistics, we took the standard deviation of each dark image and imposed a

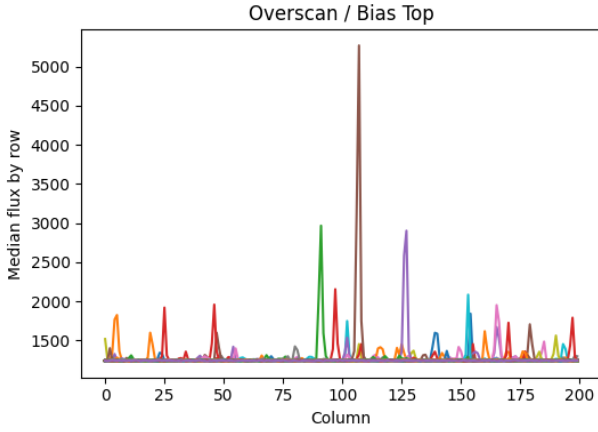


Fig. 5: Median flux by row for every overscan/bias top image.

quality cut of 200 counts: every image with a standard deviation higher than 200 was flagged and excluded from our data. There was also another excluded observation, number 392, which after calibration was left with a big bright trace in the top part of the image, for an unknown reason. This left us with a total of 1496 images from the original 1535.

The measured dark current also contains the bias value, so the very first step we took was to subtract to the dark images the correspondent bias value. After that, we proceeded similarly to the bias, computing the median of each row (or column in the case of the top image) and subtract it to the subarray image.

3.3. Flat-field

The flat-field image consists in a image of an homogeneous light source (dome lamp, twilight sky or spectrum) to detect and correct pixel-to-pixel variations of the CCD response. The calibration is done by simply dividing the scientific image (the subarray) by the flat-field. This step is needed because not all pixels have the same quantum efficiency and gain.

The flat-field image used is shown in figure 6 and was derived from monochromatic images corresponding to the spectral energy distribution of a $T_{\text{eff}} = 5410$ K, similarly to described in Hoyer et al (2020).

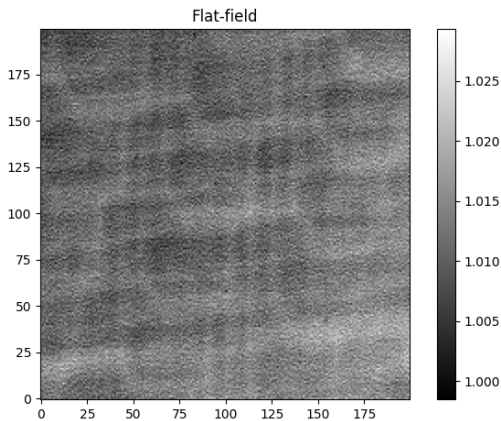


Fig. 6: Flat-field image used for calibration, derived from monochromatic images corresponding to the spectral energy distribution of a $T_{\text{eff}} = 5410$ K.

3.4. Final calibrated images

The gain is what relates the number of electrons with the number of ADU. To compute it, we can use different approaches, being the simplest one to assume that the read-out-noise (noise associated with the electronic process used to read the signal) is much smaller than the photon noise, valid when measuring a bright source. We have that the gain is equal to the number of electrons divided by the number of counts. The gain is computed by comparing shot noise, that is related only with the number of electrons and the observed standard deviation. This implies that

$$\text{Gain} = \left(\frac{\sqrt{\langle N_c \rangle}}{\sigma_{N_c}} \right)^2, \quad (1)$$

where N_c is the number of counts. The gain is in units of electrons per ADU and the distribution of ADU in a bias frame (recall figure 4) is related to the read noise, by $\sigma_{\text{ADU}} = \text{RON}/\text{Gain}$, so by computing the standard deviation of the bias frame we obtain the RON in units of electron per pixel. We tried to obtain a mean value for the RON and the gain, but the values obtained were not realistic (too high, $256 e^-/\text{pixel}$ and $53 e^-/\text{ADU}$, respectively). So instead of trusting our values, we used the ones described in Deline et al (2017): $5.35 e^-/\text{pixel}$ and $2.28 e^-/\text{ADU}$, respectively.

As the pixels in the subarray outside the circle that contains the actual counts are initially padded to zero, when we subtract the bias and the dark current these values become negative. For this reason, for a final touch in the calibration we pad these negative values back to zero and multiply the image by the gain to have the flux in units of electrons. Figure 7 shows an example of a subarray image, the first taken, before and after the calibration steps.

4. Correction

The correction module aims to correct the individual calibrated images from environmental effects, such as background pollution, but also the defocusing of the telescope on the target star, caused by the rotation of CHEOPS and the proper motion of the star. Normally, the star would remain at the center of the subarray image, but this doesn't happen and so we need to know where the center of said star is. Generally, the background correction is also made in this module. Nonetheless, due to the nature of the methods employed in the photometry module, this correction is made simultaneously with the photometry and will then be explained.

4.1. Depointing

The depointing step basically consisted in detecting the position of the star, by computing the center of mass. For this we tested two different methods:

- manual computation of the center of mass;
- using the DAOSStarFinder algorithm from `phot_utils`.

We tested a third different method, using functions from `scipy.ndimage` and `scikit-image` for image processing, but as the results were far off the desired, so ultimately we decided to leave this test out. Additionally, in the early stages of this work, the possibility of applying PSF fitting was considered, but as the star profile is not a Gaussian-like profile due to a blockage of light right in the center of the camera.

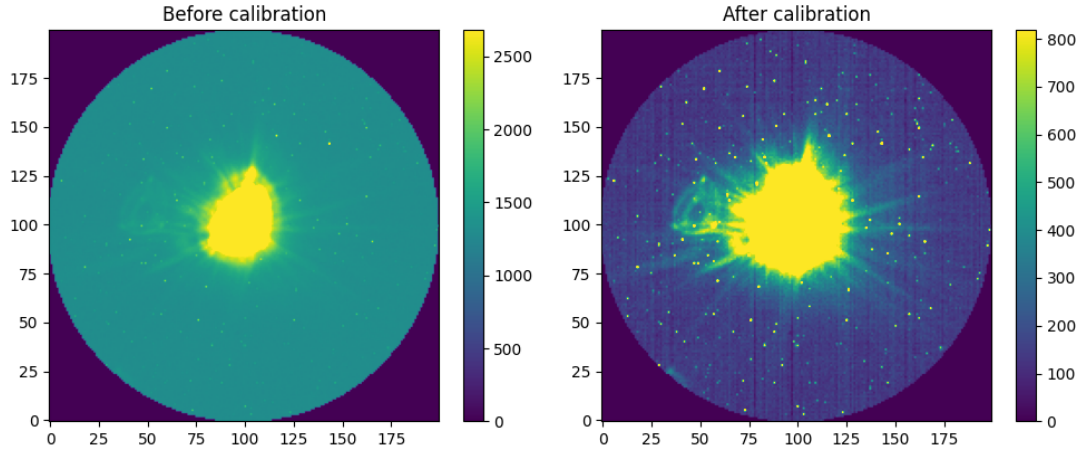


Fig. 7: Subarray image number 1 before calibration (raw) and after. For better visualization, we normalized the color bar with z-scale.

4.1.1. Manual method

The first method employed involved the computation of the center of mass of the image by making use of the physical definition of center of mass. The coordinates of the center of mass are given by

$$\mathbf{R} = \frac{\sum_{i=1}^n m_i \mathbf{r}_i}{\sum_{i=1}^n m_i}, \quad (2)$$

where m_i is the counts/flux in that pixel and \mathbf{r}_i is the pixel's coordinate. Normally this works well, but there could be a background star with an unusual amount of counts. We know by inspection the approximate location of the star, so we included in the algorithm the option of making a circular mask that only considers the flux inside, with a radius such that the background star is left out. Either way, a threshold needs to be defined, in order to neglect background counts.

To estimate a good threshold value, we considered four circular masks with radius of 15 pixels each in the extremities of the box (left, right, top, bottom), as seen in figure 8. This way, even if there is a bright source (be it the star or other things) in that red circle, the other three compensate for it. Then we compute the standard deviation of the flux of each circular mask, σ_i^{bck} , and take the median of the four values. The threshold inserted in the centroid computation function was set as $100 \sigma_{\text{median}}^{\text{bck}}$, because it's a number high enough to be insensitive to background stars and among an interval of values that represent a plateau in the centroid positions (figure 9).

4.1.2. DAOSTarFinder

In order to find stars in a two-dimensional image, the `phot_utils` Python package provides a variant of the star-finding algorithm of the DAOPHOT program, FIND, first described by Stetson (1987), and named DAOSTarFinder. This algorithm leverages a priori information provided by astronomers and the approximate size of an unresolved stellar image and the smallest brightness value expected from legitimate random noise, for example. FIND traverses the image pixel by pixel, estimating the brightness of a hypothetical star centered at each location. Gaussian profile fitting is employed to model the brightness distribution in a surrounding subarray of pixels, aiding in identifying potential star locations. If a star image is present, the

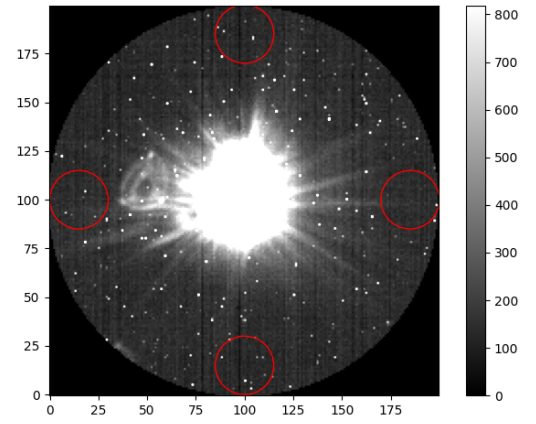


Fig. 8: The four circular masks to compute the background standard deviation, drawn over subarray image number 0 as an example. In this case the median background standard deviation value is around 25.14 counts, so the threshold is 2514 counts.

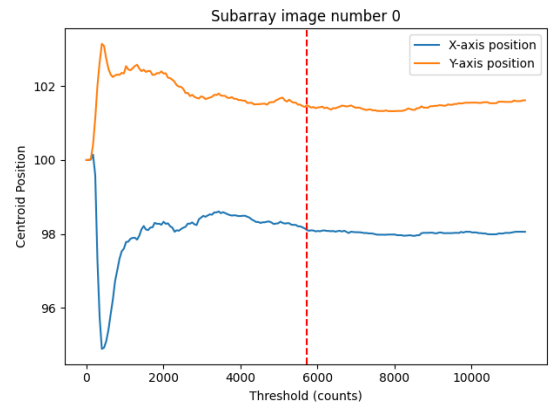


Fig. 9: Example of centroid position (X and Y) as a function of threshold value used for subarray image number 0. The vertical red line represents $100 \sigma_{\text{median}}^{\text{bck}}$.

fitted Gaussian profile's central height will be proportional to the star's brightness, and so indicating the star's position.

DAOSTarFinder follows a similar approach. It searches images for local density maxima exceeding a specified threshold and having size and shape characteristics similar to a predefined 2D Gaussian kernel. The centroid of each identified object is determined by fitting the marginal x and y 1D distributions of the Gaussian kernel to the corresponding distributions in the input image data.

In this method there are basically two parameters to be set: the threshold counts and the FWHM, the major axis of the Gaussian kernel in pixels. To estimate the threshold, the algorithm performs a 3σ -clip to the total image and computes the standard deviation of the left over data, hereby the background flux. To be consistent with the manual method, the threshold is then 100 times the standard deviation of the background. The FWHM estimation was trickier, as the star doesn't form a 2D Gaussian because of the blockage at the center, which impedes any normal Gaussian fitting. To guess the best FWHM to use, figure 10 shows the centroid position as a function of the FWHM used. The FWHM chosen consists in the width before there is an abrupt change in the position of the centroid, 18 pixels.

4.1.3. Comparing the two methods

Figures 11 and 12 compare the centroid calculations of both methods, the first in a 1-1 relationship and the second the position over time. The time used was the converted MJD time to BJD, by adding 2400000.5 and subtracting the day of the observations (2458918). As one can see, the positions retrieved with DAOSTarFinder present an offset on both the x and y positions. For a visual example, figure A.9 in the appendix shows that the position of the centroid for the first subarray image with the manual method is centered better than with DAOSTarFinder.

5. Photometry

After the data has been fully calibrated and corrected, the pipeline performs aperture photometry to retrieve the light curve. The aperture consists in a circular binary mask that is centered in the centroid computed in the previous module. Once again two methods were employed: one using functions from `phot_utils` and the other a manual algorithm. First we perform the aperture photometry, then we compute the SNR of the final images and the flux errors. Finally, we produce the light curve and estimate the planet KELT-11b radius using a simple piece-wise model, whose parameters were obtained with the `curve_fit` function and with a MCMC algorithm.

5.1. Aperture photometry

Aperture photometry consists in integrating light inside a pre-defined, fixed aperture. This aperture was chosen to be circular, due to the intrinsic symmetry of such shape. We define an aperture of radius r and an annulus around it with an inner radius 30 pixels bigger and a width of 5 pixels. As this annulus is far away from the star so that it doesn't encompass stellar flux, is good to obtain the counts coming from the background, by computing the median counts. The aperture flux is subtracted by the background median counts by multiplying it by the area of the aperture (πr^2). The rings described can be found in figure A.10 in the appendix.

In this step of the pipeline, the only parameter to be chosen is the radius of the aperture. To better understand what radius we should choose, figure 13 shows the photometric growth curves

for both methods used, as a fraction of total flux in the image in function of the aperture radius. For ease of use, we decided to choose an aperture radius of 33 pixels, as it seems a good bet and is consistent with the default radius in Hoyer et al (2020).

5.2. SNR and Error calculations

To assess the signal-to-noise ratio of the data and thus the quality of them, we use the CCD equation as described in Mortara and Fowler (1981)

$$\frac{S}{N} = \frac{F}{\Delta F} = \frac{F}{\sqrt{F + N_{\text{pix}}(BS + BD + BR^2)}} \quad (3)$$

where F is the number of electrons from the star, collected within a given aperture of area N_{pix} , BS is number of electrons coming from the sky, per pixel, BD is the number of electrons coming from the dark current, per pixel, and BR is the number of electrons coming from the RON, per pixel. BD was obtained by computing the median of the left, right and top dark images and taking the median of the three. Every flux was converted from ADU to electrons by multiplying by the gain. The SNR for each image is shown in figure A.11 in the appendix, varying between 400 and 1200 SNR.

The calculation of uncertainties in the light curve is made through

$$\text{Err} = \sqrt{F + bg + N_{\text{pix}} \times N_{\text{stack}} \times (\text{gain} \times \text{RON})^2 + \text{dark} \times t_{\text{exposure}} \times N_{\text{pix}}} \quad (4)$$

where $bg = \text{background} \times N_{\text{pix}} \times t_{\text{exposure}}$. The time of exposure was 30 seconds and the number of stacked images is 2. This equation was extracted from Silva et al (2020). The mean uncertainty in the flux is around 0.02% for the totally manual method.

5.3. Light Curve

Finally, in this step we get the final light curve to be analysed. We normalize the flux by its median and 2σ -clip it to leave out any outliers. We are also interested in estimating the radius for KELT-11b, which is estimated to be $R_p = 1.295 \pm 0.025 R_{\text{Jup}}$ by Benz et al (2021), around the star KELT-11 of $R_{\star} = 2.807 \pm 0.036 R_{\odot}$. The radius of a transiting planet can be estimated by

$$\frac{\Delta F}{F} = \frac{R_p^2}{R_{\star}^2} \Leftrightarrow R_p = \left(\frac{\Delta F}{F} \right)^{1/2} R_{\star}, \quad (5)$$

where $\frac{\Delta F}{F}$ is the depth of the transit. Simplifying a lot, one can assume a very simple transit model consisting in a piece-wise rectangular step function:

$$\begin{cases} a & \text{if } x < x_0 \\ b & \text{if } x \geq x_0 \text{ and } x < x_0 + c \\ a & \text{otherwise} \end{cases}$$

where x is the input values (the flux), x_0 is the position of the step (the initial time of transit), a is the value of the function before the step (the flux before transit), b is the value of the function during the step (during transit) and c is the duration of the step (transit). So in this model, the depth $\Delta F/F$ in equation 5 would simply be $b - a$.

For a first approximation, we used the `curve_fit` function of `scipy` to estimate these four parameters, with an initial guess by visual inspection, with an offset from the Benz et al (2021)

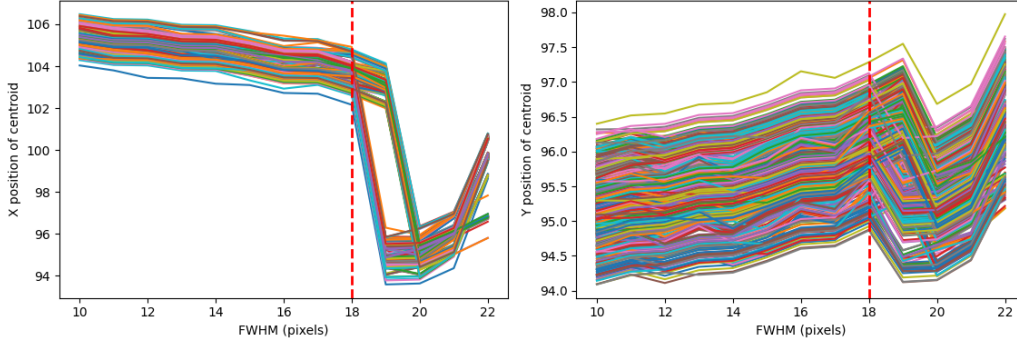


Fig. 10: X and Y positions of the centroids as a function of the FWHM in pixels used. The vertical dashed line is the 18 pixels FWHM.

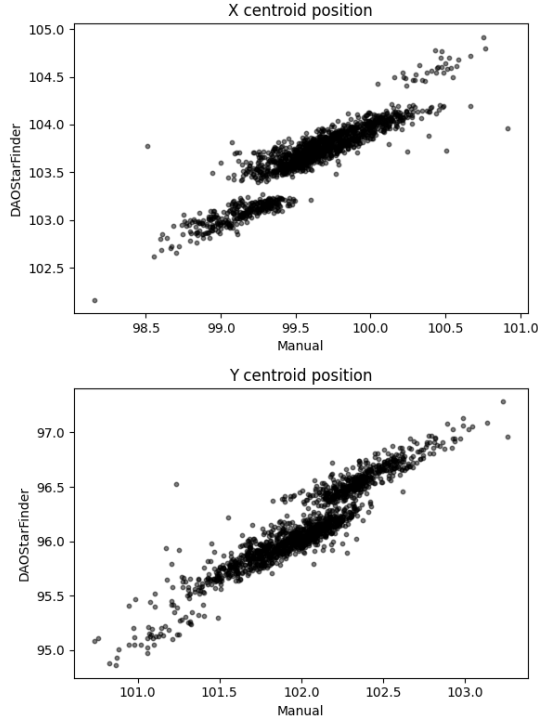


Fig. 11: 1-1 plot of the x and y positions of the centroids obtained with DAOSStarFinder and the manual method.

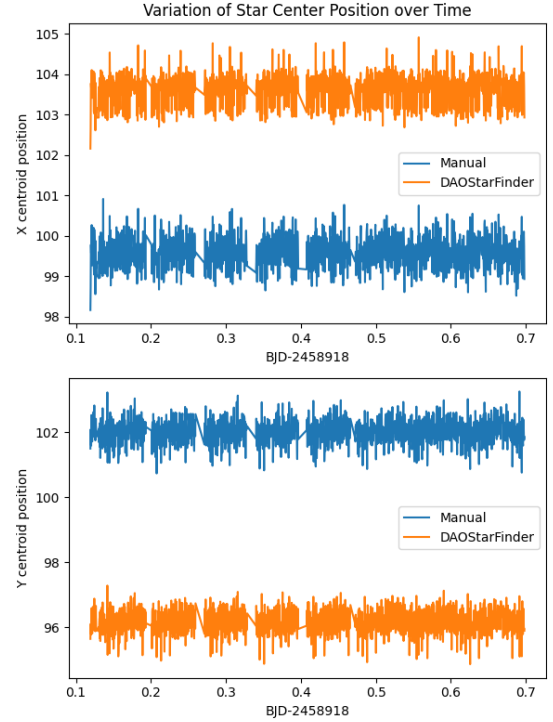


Fig. 12: Variation over time (BJD) of the x and y positions of the centroids obtained with DAOSStarFinder and the manual method.

value of 7%. Next, we use these parameters as initial guesses in the MCMC algorithm that was based in the one described in Pasha (2020). Using 1000 walkers and 500 steps, we achieve an error (comparing with the "tabulated" value) of 0.238% with the fully manual method (centroids and aperture photometry). For a full comparison, we computed the radius error in percentage ($|R_{p,literature} - R_{p,work}|/R_{p,literature} * 100$) for every combination of centroid computing method and aperture photometry method, in a total of four combinations. Table 1 shows the results for each combination and all of them retrieve very close results. Despite this, we must note that they depend on the initial guesses provided, as well as the restriction intervals.

Figure 14 shows the light curve of the data for KELT-11 and the best-fit model using MCMC, as well as the residuals, using the full manual method. The values obtained for the parameters of the rectangular model are shown in table 2. The corner plot of these parameters is shown in figure A.12.

Table 1: Computed radius error comparing with the one in Benz et al (2021), for each combination of methods. The values are in percentage.

Centroid computation	Aperture photometry	
	Manual	phot_utils
Manual	0.239	0.239
DAOSStarFinder	0.236	0.243

6. Conclusion

In this practical work, we successfully partially replicated the CHEOPS data reduction pipeline, employing the bias, dark current and flat-field calibration, depointing the object and subtracting the background flux. We also performed aperture photometry and obtained a final light curve. With this and a simple rectangular model, we obtained good estimates for the radius of KELT-

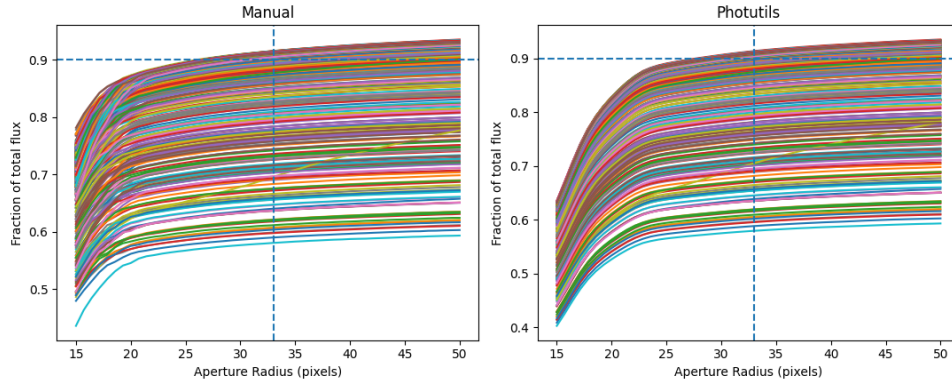


Fig. 13: Photometric growth curves for each method (manual and `phot_utils`) for every image. The vertical dashed line is the 33 pixels radius and the horizontal dashed line is the fraction of total flux equal to 90%.

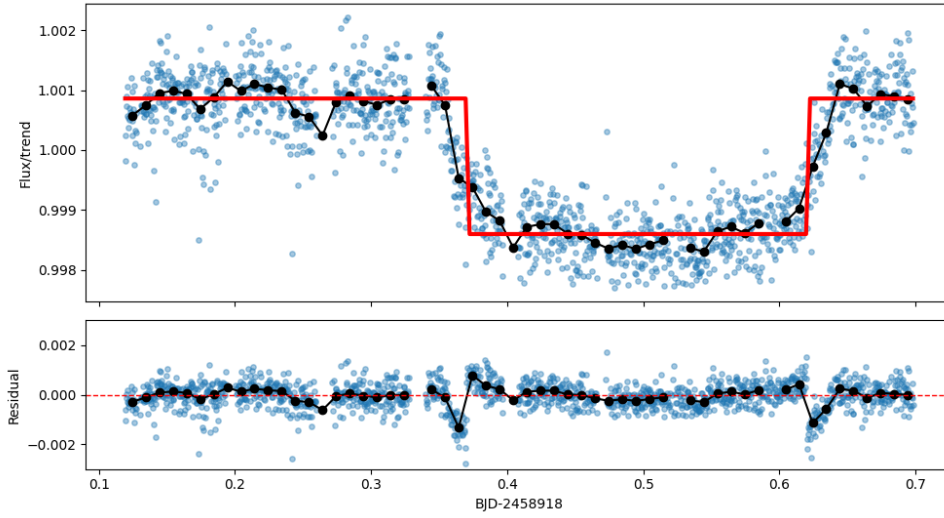


Fig. 14: Observed light curve of KELT-11 and model fit using MCMC. The measured fluxes (light blue points) are also shown binned in time (black points connected by black line). The transit model is shown in the upper panel in red. The lower panel shows residuals from the transit model.

Table 2: Parameters of the model using fully manual methods to the transit of KELT-11 b observed by CHEOPS.

Parameter	Value	Error
x_0	3.7	10^{-7}
a	1.00086	10^{-7}
b	0.9986	10^{-7}
c	0.25	10^{-7}

- include the steps in the pipeline that were not done in this work but done in Hoyer et al (2020);
- simulate images to better calibrate the pipeline;
- fit a physically accurate model to the light curve to retrieve the planetary parameters, as done in Benz et al (2021).

Nevertheless, there is many more considerations that could be taken into account, but we believe that the objectives of this work were attained.

11b. For the depointing, we computed the centroids of the object with two different functions, as well as for the aperture photometry step. The manual functions were successful in doing a comparable job to pre-made functions.

There were several challenges along the work, mainly in finding the optimal values for parameters such as the threshold, the FWHM (for the DAOSTarFinder algorithm) and the aperture radius. In spite of these challenges, we think we managed to obtain solid values. We noted that there was dissimilarities in the centroids computed with each method, but the final radius estimate does not reflect much of a difference.

Some future work to consider would be:

References

- Hoyer, S., Guterman, P., Demangeon, O., Sousa, S. G. et al 2020, A&A, 635, A24
- Benz, W., Broeg, C., Fortier, A. and Rando, N. et al 2021, Experimental Astronomy, 51, 1, 109-151
- Stetson, P. B. 1987, PASP, 99, 191
- Mortara, L. and Fowler, A. 1981, Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, 290, 28
- Silva, A. M., Sousa, S. G., Santos, N., Demangeon, O. D. S. et al 2020, MNRAS, 496, 1, 282-294
- Deline, A., Sordet, M., Wildi, F. and Chazelas, B. 2017, International Conference on Space Optics — ICSO 2016, 10562, 105624E
- Pasha, I. 2020, github, https://prappleizer.github.io/Tutorials/MCMC/MCMC_Tutorial.html

Appendix A: Appendix

Appendix A.1: Bias/Overscan statistics

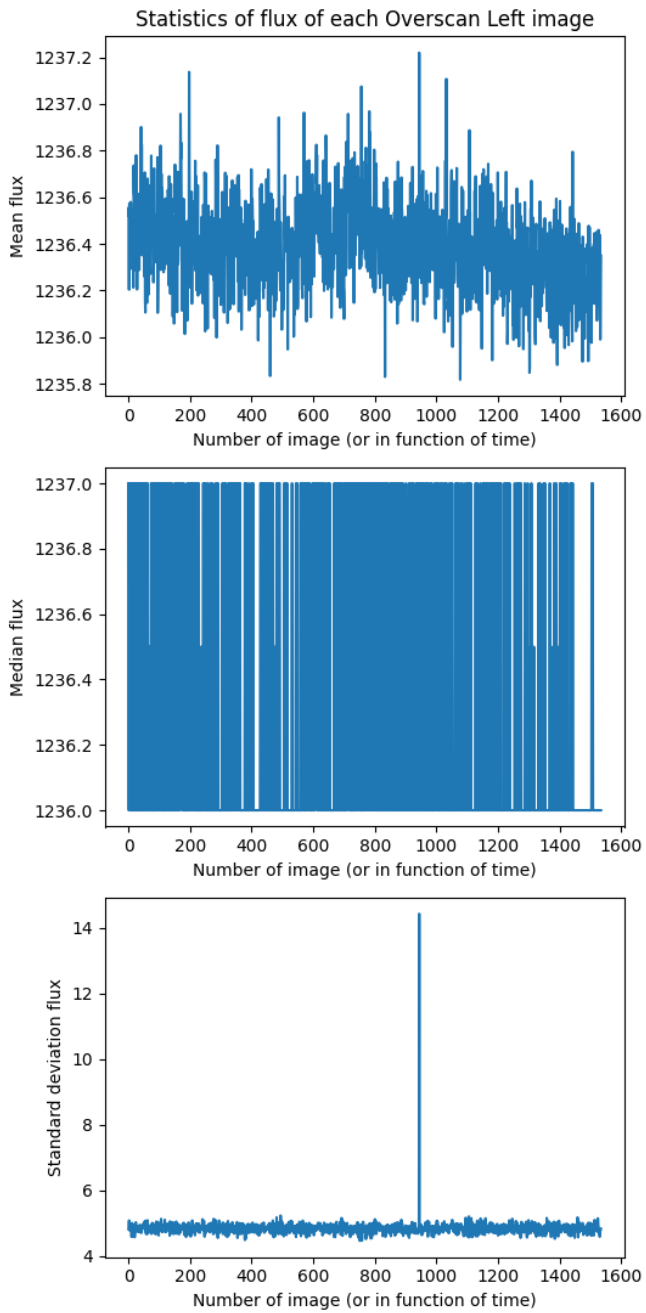


Fig. A.1: Mean, median and standard deviation of flux in bias/overscan left images.

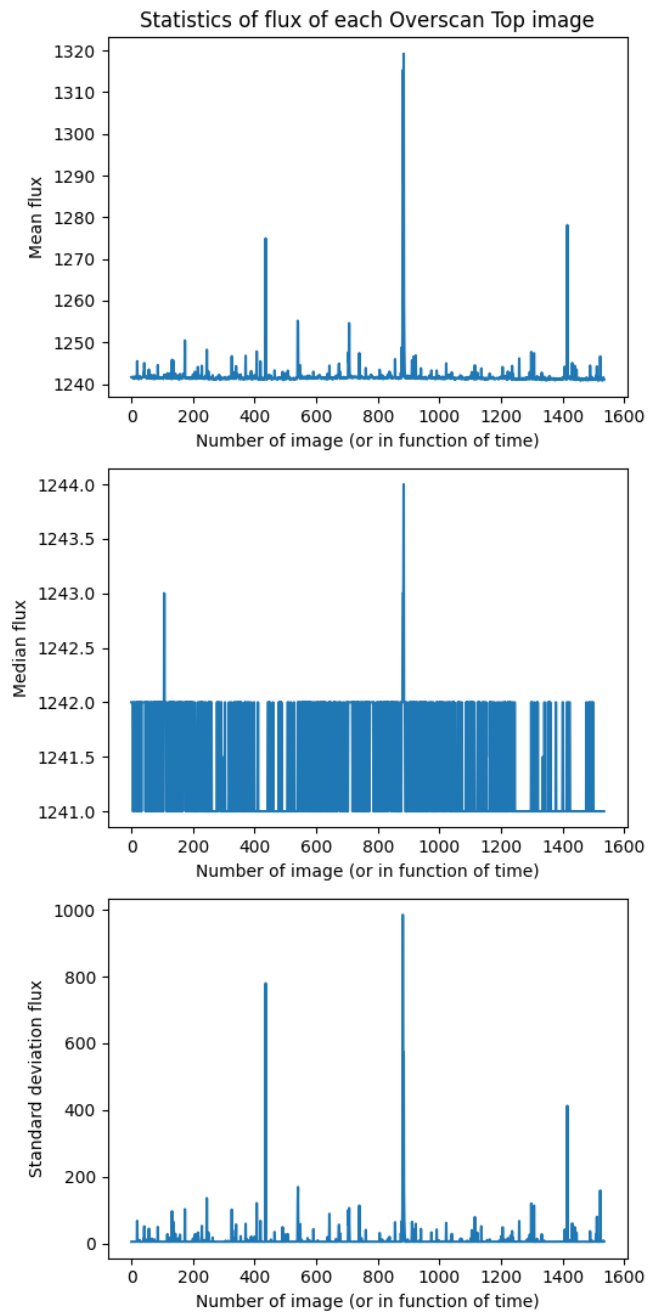


Fig. A.2: Mean, median and standard deviation of flux in bias/overscan top images.

Appendix A.2: Dark current statistics

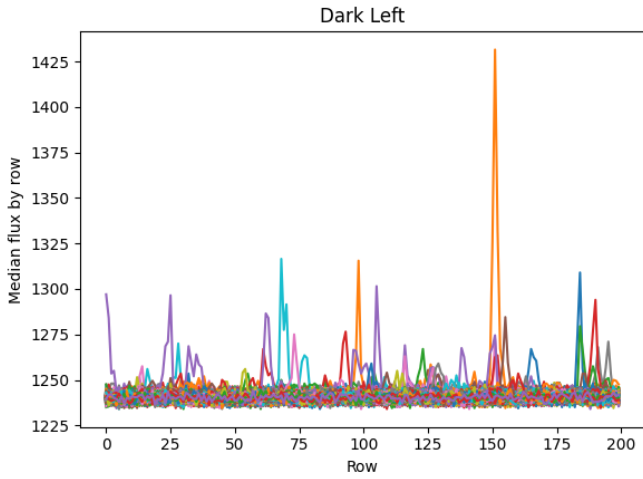


Fig. A.3: Median flux by row for every dark left image.

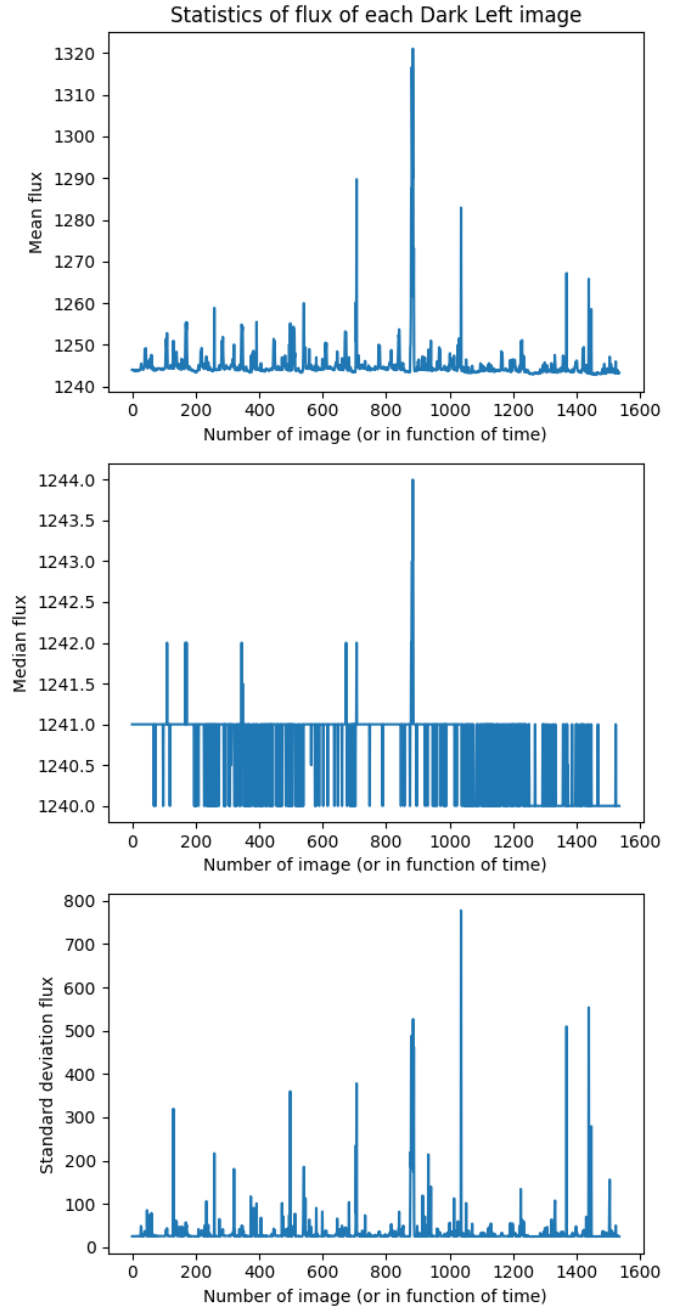


Fig. A.4: Mean, median and standard deviation of flux in dark left images.

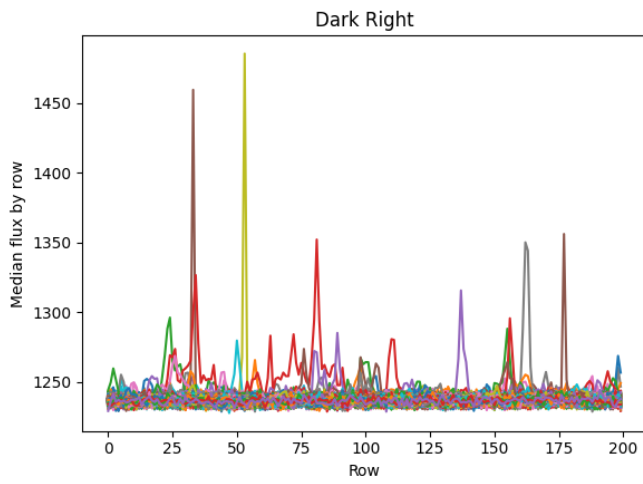


Fig. A.5: Median flux by row for every dark right image.

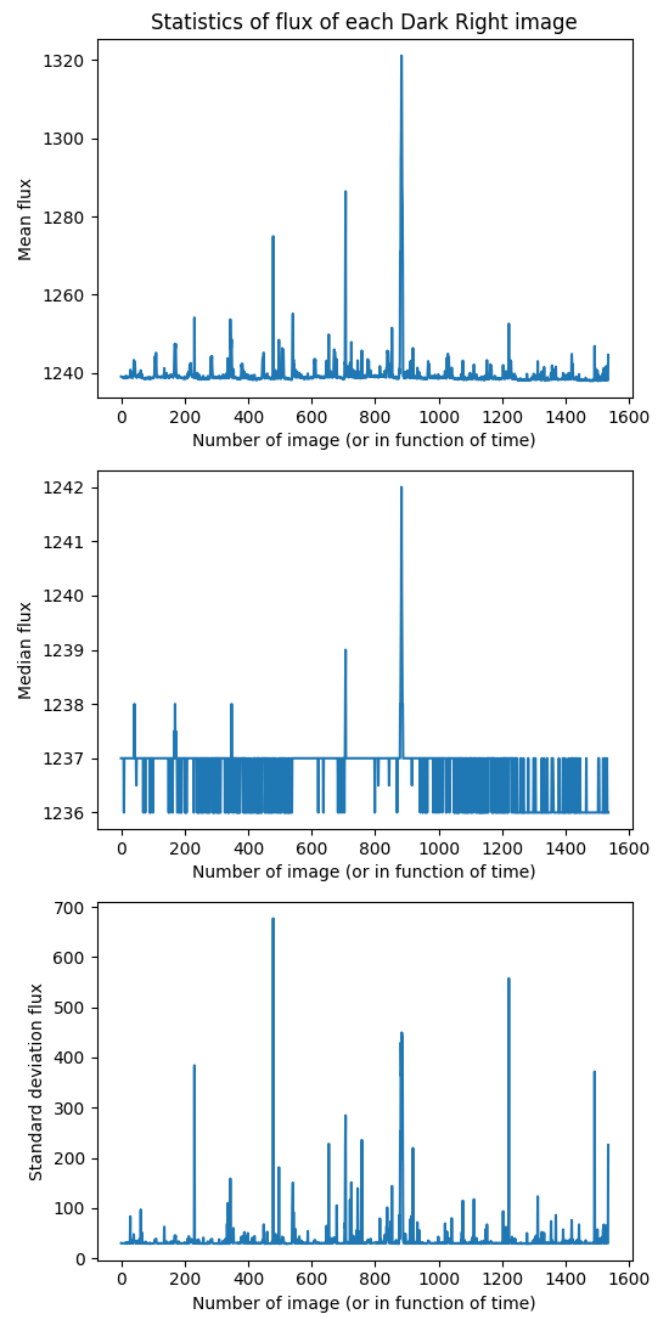


Fig. A.6: Mean, median and standard deviation of flux in dark right images.

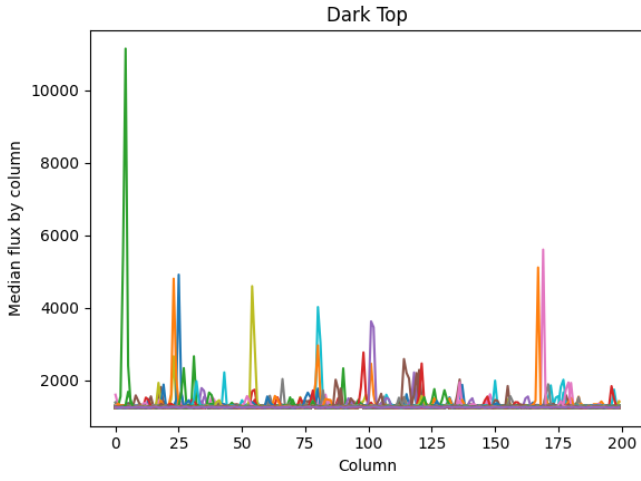


Fig. A.7: Median flux by row for every dark top image.

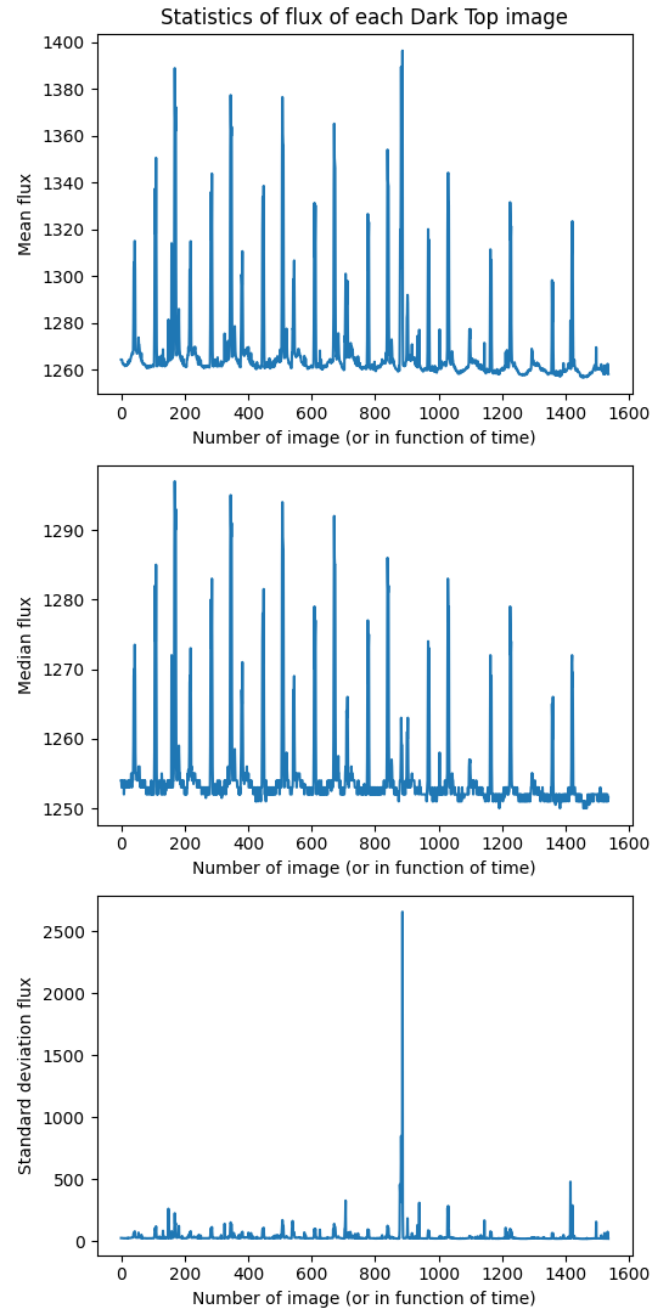


Fig. A.8: Mean, median and standard deviation of flux in dark top images.

Appendix A.3: Correction

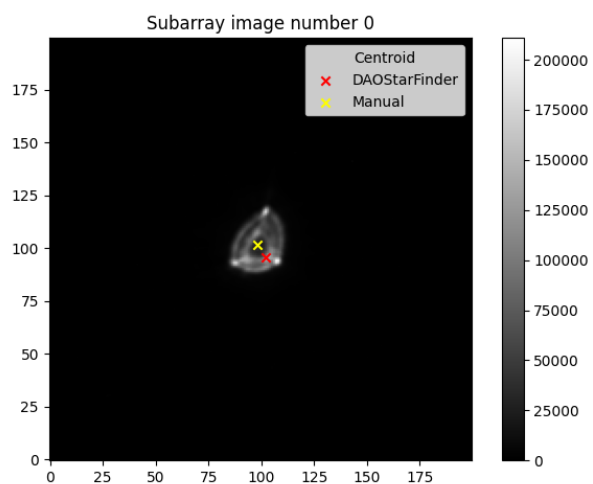


Fig. A.9: Centroid of the subarray image number 0 for each method. No scale was used in the colormap.

Appendix A.4: Photometry

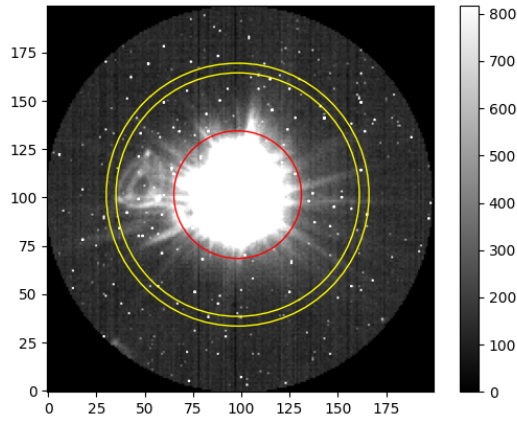


Fig. A.10: Aperture radius (red) and annulus rings (yellow) of 33, 63 and 68 pixels, respectively. Subarray image number 0 with color-map Z-scaling.

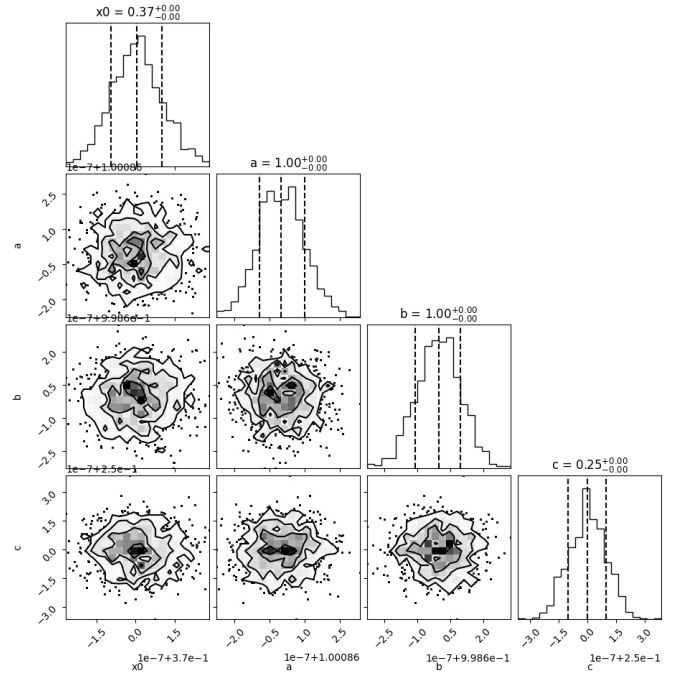


Fig. A.12: Corner plot of the parameters obtained with MCMC for fully manual methods.

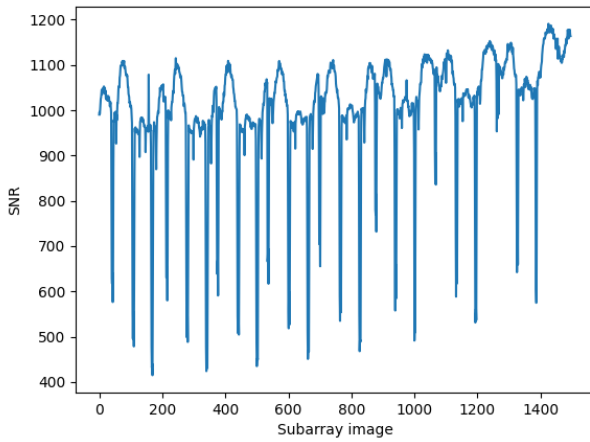


Fig. A.11: Signal-to-noise ratio of each calibrated and corrected subarray image. The centroids were computed with the manual method, as well as the aperture photometry.