

CHEOPS Mission Reduction Pipeline Development

Simulation of WASP-18 b - Visit 2

João P. S. Rodrigues

Departamento de Física e Astronomia da Universidade do Porto
e-mail: up201405201@fc.up.pt

ABSTRACT

Context. In this work, a possible reduction pipeline for the CHEOPS mission is presented and characterized.

Aims. We hope to achieve satisfactory results on the computation of the light flux in the target star, by finding a transit and keeping the data dispersion bounded to $\lesssim 20$ ppm for six hours of observation.

Methods. The bias subtraction is done by performing an average over all the points in the four columns. The images are divided by extra flatfield ones. The centroid estimation is done by computing the center of mass of a square area of side 20 px around the center of the image. Canny edge detection is applied on a square area around the centroid of the target and the target points are thus computed. The background values are estimated by computing the mode and the mean of all points outside a circle of radius 25 px around the centroid of the target. This values are subtracted from the image before computing the flux on the target area.

Results. We observed the expected transit of a planet in front of the target under observation. The dispersion was proved to be < 20 ppm when using only the background mode subtraction and ~ 20 ppm when using both the mode and mean subtraction. The pipeline is shown to run in ~ 3.6 seconds with a data sample of 406 images, with the production of flux and background estimation graphs.

Conclusions. We conclude that this is a viable method to perform the pipeline since it is able to deliver results with the expected dispersion values in a small computation time.

1. Introduction

The reduction pipeline is one of the most important steps in producing science ready data from star observations. This software should be able to take the raw image data captured by the telescope and compute the light flux of the target star as a function of time. From this function, it is then possible to compute physical parameters from the planet or planets transiting in front of the star at the moment of observation e.g. the planet radius, the orbit period, etc.

It is thus critical to perform a well characterized data reduction. In this report we explain the approach taken to achieve this by applying the classical steps of bias subtraction, flatfield division and centroid estimation and by computing the area of the target using the canny edge detection algorithm.

2. Overscan

The first step in treating the data is to subtract the overscan values from the image in order to remove the so-called Bias - a systematic error such as electronic noise introduced while reading the CCD.

Since each image is provided with four extra columns of overscan which contain no data apart from the reading noise, we can use these columns to compute the overscan. By performing an average over each line of the columns and computing the histogram of these average values, we can approximate the noise with a gaussian curve (see Fig. 1). The fit was optimized using the least squares method and the final parameters are $[A, \mu, \sigma] = [9.23, 5260.86, 7.07]$ with one standard deviation errors of $[0.515, 0.455, 0.457]$ respectively.

This allows us to perform an average over all of the four columns and subtract that value from the image data, without introducing significant error.

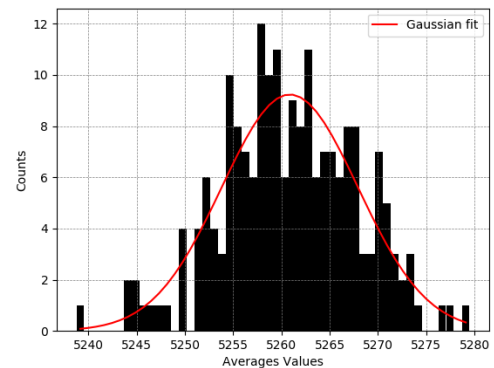


Fig. 1. Histogram of averages for each line of the 4 columns.

3. Flat Field

The next critical step of the reduction pipeline is to account for the imperfections of the CCD screen when acquiring data. This is done by capturing what is called a Flat Field (FF) image which is the data acquired by the sensor when pointed at a uniform screen.

In order to remove this inconsistent data, the captured images must be divided by the data in the FF image, resulting in an image in which all the CCD pixels have roughly the same sensitivity.

After this step is performed, the pre-processing of the images is complete and one can begin computing the area of the image wherein the target lies, with the objective of performing a photometric measurement.

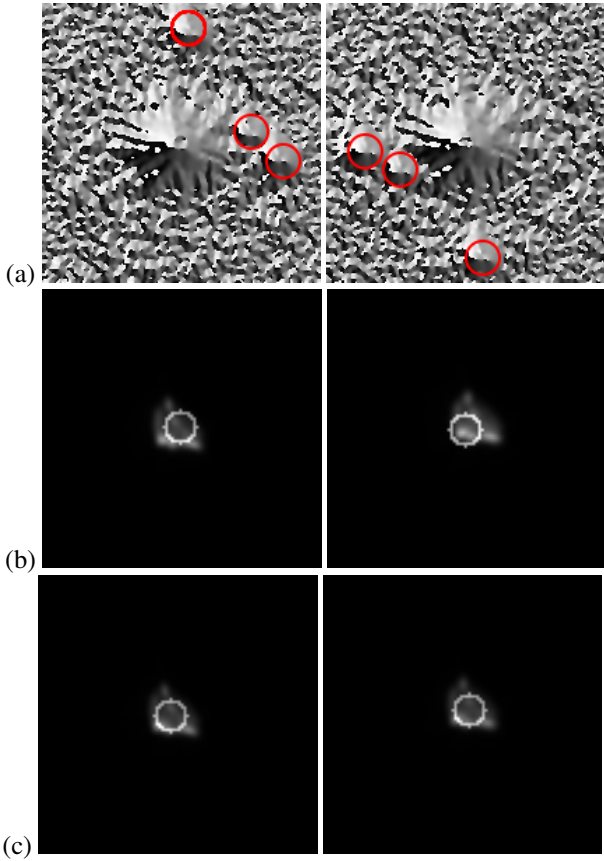


Fig. 2. From top to bottom: (a) Direction of the images' gradient. (b) Center of mass calculated using the whole image. (c) Center of mass calculated using a square of size 20 pixels centered on the images' center.

4. Center of Mass Calculation

Taking into account that the order of the counts belonging to the target is much larger than in the rest of the image, one can assume that a rough approximation of the target's position is a computation of the image's center of mass.

However, by computing the center of mass of the whole image, we find that its position varies in a circular way. This is due to the rotation of the satellite together with the other objects which very faintly appear in the images. These objects are better observed by computing a convolution of the image with a Scharr Kernel of the type:

$$\begin{bmatrix} -3 - 3j & 0 - 10j & +3 - 3j \\ -10 + 0j & 0 + 0j & +10 + 0j \\ -3 + 3j & 0 + 10j & +3 + 3j \end{bmatrix} \quad (1)$$

which computes the gradient of the image both in magnitude and direction. The direction of the image's gradient is shown in Fig. 2 (a) where the existence of at least three other objects, besides the target, is made obvious. We can also notice that these objects are rotating around the target in a clockwise manner which explains the circular movement of the images' center of mass, seen in Fig. 2 (b).

As a means to fix this problem, the center of mass was calculated using a square centered on the images' center and of a radius r_{CM} to be empirically computed in Sec. 8. This way, the calculation only involves the target and not the other objects, thus improving the stability of the center of mass, which can be seen in Fig. 2 (c).

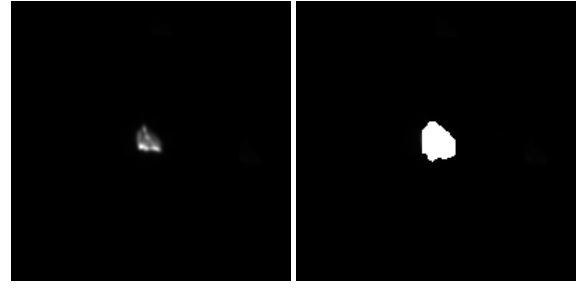


Fig. 3. Comparison of image 0 target with detected area.

5. Canny Edge Detection

From observing the images in Fig. 2 (a), an idea arised to use the gradient of the images to find the region on which to perform the photometry. An obvious candidate for an algorithm popped into mind: Canny Edge Detection (Canny 1986). The algorithm has the following steps:

1. Apply Gaussian filter to remove noise.
2. Compute intensity gradient of the image.
3. Apply non-maximum suppression.
4. Apply double threshold to determine edges.
5. Edge tracking by hysteresis.

Even though this is the standard procedure to perform the canny edge algorithm, we can disregard the third step since applying the suppression is just an edge-thinning technique which is of no use for us since we want to find the area inside the outer edges, anyway.

The same principle (used in the calculation of the center of mass) of applying the algorithm to a box centered on the image was utilized here in order to find only the edges of the target. As such, this algorithm, if correctly applied - using optimized parameters of filter σ and double thresholds, discussed in Sec. 8 - should return the area of the target in a specific image.

As can be seen in Fig. 3, the algorithm seems to be a good candidate for detecting the target's area, for as much as can be seen with the naked eye. This figure was obtained with $\sigma = 0.6$, higher threshold $T = 4000$ and lower threshold $t = 0.3T$.

6. Background Estimation

In order to obtain an accurate result for the photometry, it is essential that we account for the background counts present in the whole image. This value has to be estimated and subtracted from the image.

Firstly, since the image contains other light sources as seen above, the mode of the image should represent better the value of the background as opposed to the mean which should have a contribution from these other sources. Besides that, the target should not be considered in the calculation of the background and thus only points outside a square of side r_{bkg} will be taken into account.

By calculating the mode for each of the images, we can see that it has an approximately sinusoidal variation (see Fig. 4) which can be fitted. The fit parameters and respective one standard deviation error are shown in Tab. 1.

From the sinusoidal fit, we can estimate the period of the mode oscillation to be $1/f = 97.03 \text{ images}$ which approximately corresponds to the period of rotation of the satellite, 100 minutes (since the integration time of the instrument is 1 min).

Table 1. Mode sin fit parameters.

Parameter	Value	1 σ error
A	8.14 e+0	2.98 e−1
f	1.03 e−2	4.64 e−5
ϕ	4.97 e+1	6.89 e−2
DC	6.01 e+1	2.07 e−1

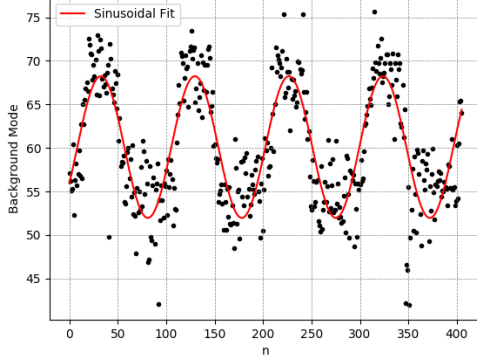
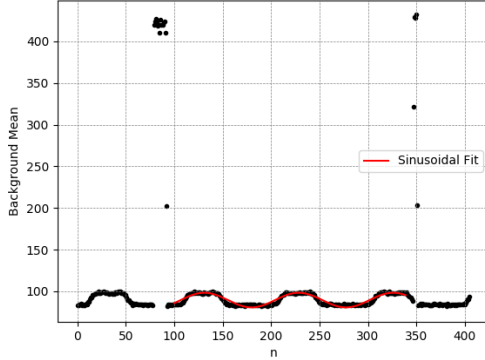

Fig. 4. Background mode throughout the images.

Table 2. Mean sin fit parameters.

Parameter	Value	1 σ error
A	8.78 e+0	2.30 e−1
f	1.02 e−2	6.03 e−5
ϕ	4.96 e+1	8.47 e−2
DC	8.94 e+1	1.64 e−1

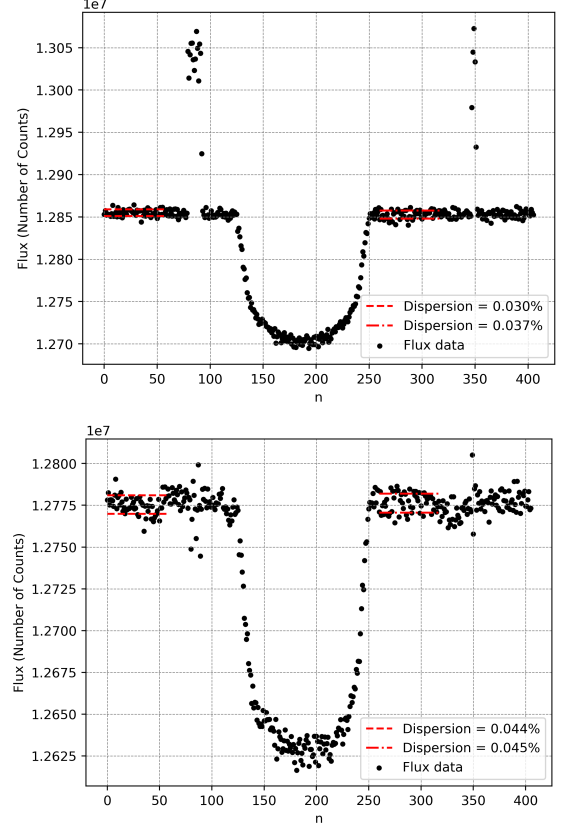

Fig. 5. Background average throughout the images.

On the other hand, the mean of the background can also be calculated in order to account for random oscillations in the image, which are not, in general, captured by the mode. As can be seen in Fig. 2, the mean detects the passage of the telescope through the South Atlantic Anomaly, characterized by random peaks of noise around images 90 and 350.

The same sinusoidal fit can be calculated for the mean data, using images in the range [100 – 340]. Tab. 2 shows the parameters obtained in this calculation. The period of the sine wave is now $1/f = 97.43$ which is in close agreement with the result obtained from the mode.

7. Photometry

The photometry measurement can then be performed, by subtracting the background mode and, optionally, mean from the images and then sum the values of the points belonging to the target. The results obtained for the flux over time are shown in Fig. 6.


Fig. 6. Flux of the target over time. Top: with background mode subtraction; Bottom: both mode and mean subtraction.

The results clearly show a transit happening around the target start by the prominent well in the flux data. We can also see the effect of the South Atlantic Anomaly on the flux, which raises it as found in the background mean data. By subtracting this mean from the data we can see that this effect is greatly reduced. However, by computing the dispersion,

$$\Delta F = \frac{\sigma_F}{\langle F \rangle} \quad (2)$$

of the data in images [0 – 60] and [260 – 320], we can see that this mean subtraction introduces dispersion in the data.

This dispersion can be used as a figure of merit for the effectiveness of the algorithm. We would ideally want $\lesssim 20$ ppm dispersion in 6 hours of observation to fulfill the CHEOPS mission requirements (CHEOPS Status). Since in this case the observation is not that long, we can only estimate the dispersion of the algorithm by computing,

$$\Delta F_{6h} = \frac{\Delta F}{\sqrt{6 * 60}} \quad (3)$$

since ΔF is for a single minute of observation. By observing Tab. 3, we can see that the results are similar to those expected,

Table 3. Estimation of the dispersion for 6 hours of observation using the developed method.

Images	Mode		Mode + Mean	
	[0 - 60]	[260 - 320]	[0 - 60]	[260 - 320]
Dispersion 6h (ppm)	15.894	19.752	20.957	20.543

being smaller than 20 when only subtracting the mode and being less than 1 ppm higher than 20 when subtracting the mean.

8. Parameter Optimization

The results in Fig. 6 were obtained by optimizing the algorithm's parameters in regards to minimizing the dispersion as will be explained in the current section.

We started by optimizing the size of the square for the center of mass computation by setting all other parameters to mere approximations of their optimal value. We can conclude that any value between 20 and 40 would be optimal. We thus chose 20 as the optimal value in order to minimize the number of points used in the calculation and improve computation efficiency.

We proceeded to optimizing the radius of the background calculation. Since this radius limits the area inside which the target is, we would expect that a better background calculation would be the result of a higher value for the radius parameter, thus reducing the dispersion. As expected, we showed that the dispersion tends to reduce for higher values of this parameter. A radius of 25 was chosen for the same reason as before, to reduce the number of points in the calculation while still remaining optimal.

The figures for the optimization of these two parameters are not shown here, but made available as an attachment.

Furthermore, there remained three parameters to optimize, all belonging to the canny edge detection algorithm: σ , the higher threshold (T) and the lower threshold (t).

The σ parameter controls the radius of the Gaussian filter applied to the image to begin the algorithm. By observing Fig. 7 we can see that the dispersion decreases as sigma increases since by applying a larger Gaussian filter we are smoothing the image even more. We choose a value of 1.85 for the optimal σ to balance the minimal dispersion with the computed area of the target - too great of sigma and the algorithm finds the wrong edges and the area greatly increases.

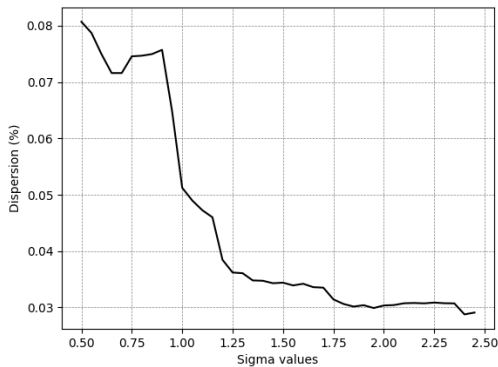


Fig. 7. Dispersion as a function of σ for the canny edge detection.

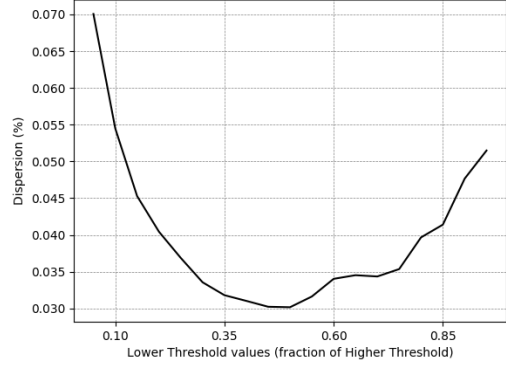


Fig. 9. Dispersion as a function of t for the canny edge detection.

Table 4. Optimal algorithm parameters.

CM	Background	σ	T	t
Square Side	Radius			
20	25	1.85	3500	0.5T

Using this value of σ , we optimized T. In Fig. 8 we can see a minimum in the region of 2000 to 4000, with the dispersion greatly increasing for lower and higher values of T. We thus choose $T = 3500$ by observing the area of the target captured, as for σ .

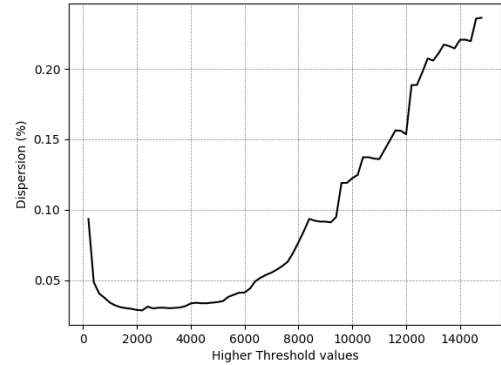


Fig. 8. Dispersion as a function of T for the canny edge detection.

Finally, we computed the lower threshold using all the optimal parameters. The canny edge detection algorithm is expected to be more effective for $t = 0.5T$ thus we are expecting to find a minimum of dispersion for this value. We can see that our expectations are met, in Fig. 9. We thus choose $t = 0.5T$ as the optimal parameter.

The optimal parameters are presented in Tab. 4 for convenience.

9. Timing Optimization

Even though no timing optimization was performed for this algorithm, we can see that it behaves quite well for the studied data.

The pipeline takes approximately 3 secs to produce the flux data with less than a second more to export flux, background mean and background average results.

10. Conclusions

1. We conclude that the approach used to perform the photometry on the sample images is well behaved.
2. Firstly, it is seen that the overscan of the images successfully removes the Bias by performing an average over the four overscan columns and subtracting it from the images.
3. Secondly, the target images are divided by the flatfield ones in order to remove bad pixels.
4. We then proceeded to finding the target in the image by first computing the center of mass. Instead of calculating it for the whole image, we compute it for a small squared area with a 20 px side, around the center of the image. This center of mass calculation is shown to remain valid across multiple images.
5. Afterwards the canny edge detection is performed in a different square area centered on the center of mass of the image and the points belonging to the target are found. This computation is proved to be in accordance with image observations. The parameters used in the algorithm are: $\sigma = 1.85$, $T = 3500$ and $t = 0.5T$.
6. Furthermore, the background calculation is done by computing the mode and the mean of the image area outside 25 px radius centered on the center of mass. This value is then subtracted from the image in order to perform a valid photometry measurement. It is shown that the mode subtraction is effective at removing the background in most images and the mean subtraction should be used to remove the effect of the South Atlantic Anomaly.
7. The photometry is then calculated on the area of the target. This computation is found to be in accordance with the expected values since a clear transit is present in the flux graph. Furthermore, the dispersion is found to be < 20 ppm when using only the mode in the background subtraction and ~ 20 ppm when also using the background mean. This results are what is expected from the CHEOPS satellite.
8. Finally, all the parameters in the pipeline are optimized to reduce the aforementioned dispersion and found to behave as expected.
9. This pipeline is shown to take ~ 3.6 seconds to compute and deliver the results for a data sample of 406 images (corresponding to 406 minutes of observation).
10. Future work should be done by adding a transit analysis to the pipeline to compute e.g. the planet radius and the planet transit period.

References

- Canny, J. 1986, in IEEE Transactions on Pattern Analysis and Machine Intelligence
- Benz, W. , Review of CHEOPS mission and status update