

To Measure a Transit with a 0.5m Telescope

Sven Kiefer

Supervisor: Dr. Sascha P. Quanz

Advisor: Luca Tortorelli

Associate: Adrian A. Gheorghe

May 7, 2018

Abstract

This thesis answers the question if it is possible to measure an exoplanetary transit with a 0.5m telescope at Hoenggerberg. It is written in parallel with the thesis of A.Gheorghe who treated the same subject and together with whom the python code was developed. In total, 6 measurements of known transits have been made and compared with the modelled prediction. The program which was developed during this process can evaluate observation data with very little user input. This program was then generalized to be applicable for different telescopes. Additionally, the effects of filters on measurements and the usability of the local linear approximation method were analysed. After the analysis of the observation data with the program, 2 exoplanetary transits have been measured. Furthermore, the limits which transits can be observed have been determined for the HPP 0.5m Telescope.

Contents

1	Introduction	3
2	Preparation	3
2.1	Possible stars	3
2.2	Linearity of the CCD	3
2.3	Dark Frames	4
2.4	Filters	5
2.5	Local Linear approximation	6
2.6	Full Width Half Maximum	8
3	Evaluation Program	9
3.1	Image Correction	9
3.2	Finding and Filtering of Stars	9
3.3	Calculating the Flux	9
3.4	Finding a Transit	10
4	Observations	11
4.1	Observation 1: WASP-93-b (25.10.17)	11
4.1.1	Details of the Transit	11
4.1.2	Transit Calculation	12
4.2	Observation 2: Qatar-2-b (21.3.18)	12
4.2.1	Details of the Transit	12
4.2.2	Transit Calculation	13
4.3	Observation 3: WASP-104-b (10.4.18)	14
4.3.1	Details of the Transit	14
4.3.2	Transit Calculation	14
4.4	Observation 4: Hat-P-36-b (17.4.18)	15
4.4.1	Details of the Transit	15
4.4.2	Transit Calculation	16
4.5	Observation 5: WASP-85A-b (19.4.18)	17
4.5.1	Details of the Transit	17
4.5.2	Transit Calculation	17
4.6	Observation 6: TrEs-3-b (19.4.18)	18
4.6.1	Details of the Transit	18
4.6.2	Transit Calculation	19
5	Discussion	20
5.1	The Telescope and our Evaluation	20
5.2	Qatar-2-b	20
5.3	WASP-104-b	20
5.4	Hat-P-36-b	21
5.5	WASP-85A-b	21
5.6	TrEs-3-b	21
6	Conclusion	21
7	Outlook	22
7.1	Improvement of the Experiment and the Program	22
7.2	Automation of the Program	22
7.3	How small is too small?	22
7.4	Importance for amateur astronomer	22
8	Acknowledgements	23
9	Appendix	24

1 Introduction

There are many known exoplanets found by professional space and ground based telescopes. Most of these transits are confirmed and can be found in various databases. The limitation on which transits can be found is mostly given by the equipment, observation place and weather conditions. With this experiment we wanted to answer the question if it is possible to measure an exoplanetary transit with a 0.5m telescope at Hoenggerberg. The exact specification can be found in the HPP Observatory Quick User Guide [6].

There exists several methods to detect exoplanets. We used the transit method, which uses the slight darkening of a star if an exoplanet crosses the line of sight. How much the flux of a star changes depends on the size of the exoplanet. How often we are able to measure it depends on its orbital period. This means that the ideal objects to observe are Hot Jupiters. These are large planets with a small distance to their host star.

Most of the expensive telescopes are located at far off places like the Chilean desert to reduce the light pollution. Hoenggerberg is between Zurich and Oerlikon which are both brightly lighted during the night. This makes the ETH campus one of the worst possible places in Switzerland to observe an exoplanetary transit. Nevertheless, we assumed it to be possible to perform a measurement in which we confirm an exoplanet. There was already an attempt to measure an exoplanetary transit by A. Bohn. Unfortunately the night was too short, and the transit could not be confirmed [7].

With this experiment, we tried to confirm an exoplanetary transit and thus proof that it is possible to find them even near Zurich. We also wanted to probe the limitations of the equipment necessary to find a transit. Of course, the size of the telescope and the CCD matters, but there are other factors. For example, it is important to have a good auto tracking, cooling of the CCD and a program to evaluate the data accurately.

A second goal of our experiment was to write a code to evaluate our images. The idea was to have a program which can correct, calculate and evaluate the data at once by only setting some initial conditions. The previous method used [7] needed many manual adjustments and thus was time consuming. The program should give us a tool to evaluate our data mostly automatic which speeds up the whole process.

2 Preparation

2.1 Possible stars

Since the success of the Kepler mission, many exoplanets are known. The information about them can be found in several databases for exoplanets. For this experiment we used "Exoplanets.org" [3] and the database of the Czech Astronomical Society [2]. This allowed us to get the best possible transit for each night. It is worth mentioning, that we weren't interested in a specific transit and therefore took the one we could observe the best. In our case that included the following criteria:

1. A transit depth of at least 1%, but preferably over 2%
2. A transit duration shorter than 4 hours
3. The transit should be fully visible during the astro-night (maximum darkness)
4. A declination of the host star over 30 degrees (horizontal coordinate system) during the whole measurement

With these criteria we still had an average of 4 transits per night. In most nights we even had one with a depth of more than 2%. We tried to use the longer nights during the winter but this was not possible because of the weather conditions. The criteria should also be satisfied 1 hour before and after the transit to make it clearly visible. This means a typical measurement time of 3-6 hours. During the shorter nights and because of technical errors, this was not always given.

2.2 Linearity of the CCD

Camera and mirror errors are a common problem and can be corrected with flat fields and dark frames. These corrections are never perfect and thus we still want to use a "good region" on the CCD. In our measurement we

only used relative values of the stars we analysed, not absolute values. A "good region" in our case are pixels on the CCD which behave linearly with the brightness of our stars.

To find the linear regions on the CCD, we took several flats either during twilight or a cloudy night. The CCD is approximately linear in a range between 1000 to 20'000 pixels [7] and thus we took all flats with a mean value in this range.

We then wanted to find the regions of the CCD which have a linear behaviour. To get a good overview of the CCD, we did the calculations in 64x64 pixel bins. We then plotted the bin values to the mean of each picture. If a pixel has a linear dependence to the mean, it also has a linear behaviour respective to other stars.

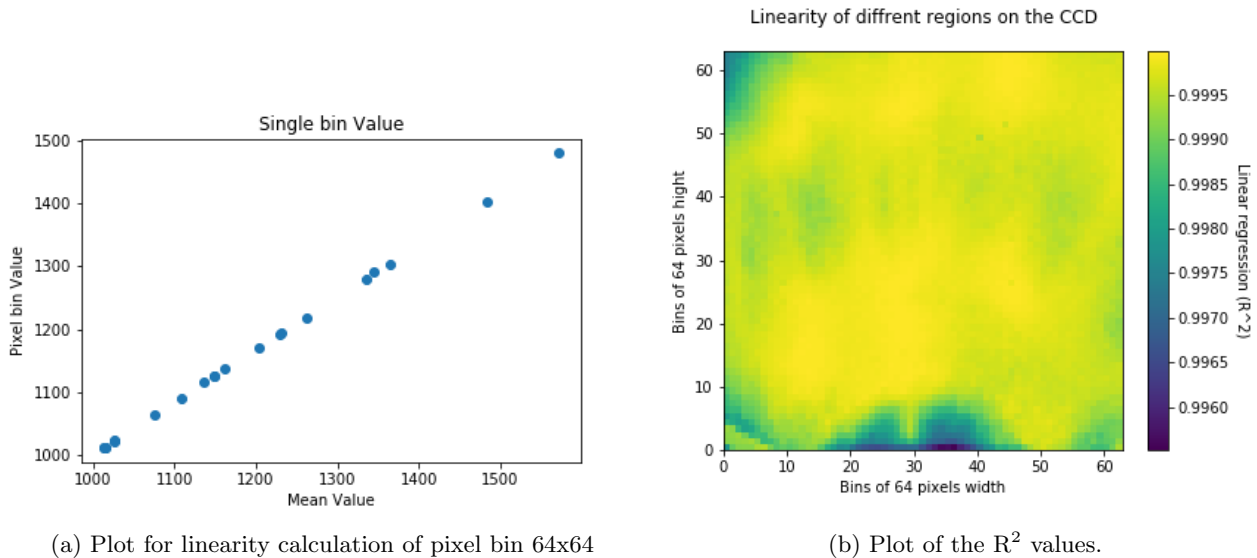


Figure 1: Plots to determine the linearity of the CCD

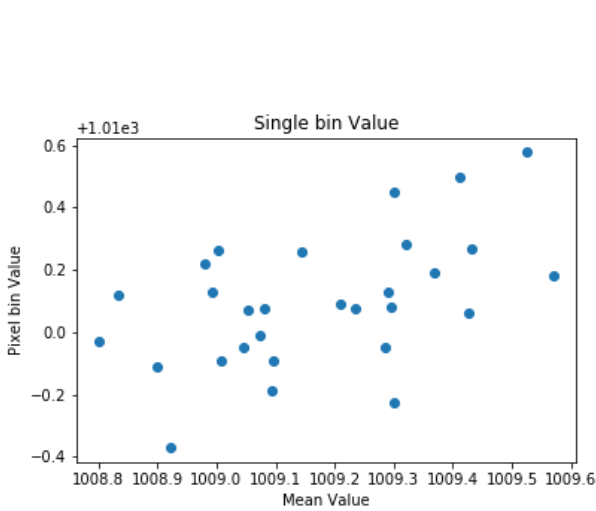
For our analysis we took 21 flat fields with a mean value between 1000 and 2000. This was the typical value range for our transit stars. We also assured that no individual pixel value was above 20'000 to be guaranteed in the linear regime. The example for one bin is shown in figure 1a. Out of this plot, we calculated the linear regression for each bin and plotted the result in a colour map which can be seen in figure 1b. The R^2 values are close to 1 for all pixel bins. There are some pixels at the edge which behave slightly worse than the one in the centre. During our experiment, we always imaged our transit star near the centre of the CCD. Because of imperfections in the alignment of the telescope, the exact position still varied. We didn't position the reference stars because other sources of errors were far more important, like differences in brightness.

2.3 Dark Frames

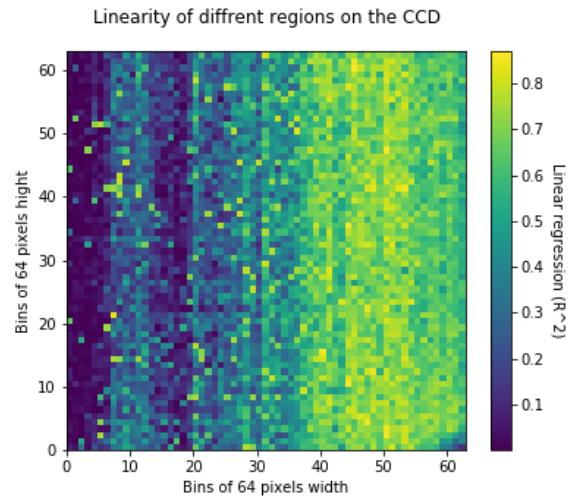
Dark frames are taken to correct the thermal and read out noise. This is done by taking an image with the same exposure time as the measurement image. Afterwards the dark frame can be subtracted from the raw data and flat fields by simple subtraction. To reduce this noise, we cooled the CCD down to -10C. Similar to the flat analysis, we want to check if the CCD behaves linear in respect to the dark frames.

We then wanted to analyse if the dark correction really has a linear behaviour and thus doesn't lead to additional errors in our data. The calculation of the linearity regarding dark frames was done in the exact same way as the calculation in the previous section, using the same program. We also used 64x64 pixel bins to get a good overview of the CCD. The only difference was that we analysed dark frames instead of flats.

We analysed 29 dark frames with exposure times between 0.18 to 20 seconds. The example for one bin is shown in figure 2a. It is important to notice that the mean values are extremely close together and thus the difference between the individual darks is also small. Because all data points are this narrow, we expect the R^2 value to be smaller and to vary more over the different regions of the CCD. The R^2 value over the CCD can be seen in



(a) Plot for linearity calculation of pixel bin 32x32



(b) The R^2 of the CCD regarding dark frames.

figure 2b. The left side of the CCD still has a good linear behaviour were the right side is almost a random distribution. A possible explanation of this difference could be the read-out system of the CCD, because it collects the values of the pixels in lines. This random error will not be important during our measurements because it is very small. The dark frame error scales with integration time (not with the brightness of the stars) and thus is the same for all stars.

2.4 Filters

To calculate the flux of a star, one can use a filter to get more precise value or to reduce influences of nearby stars. We wanted to check if an additional filter on the stars could reduce noise by nearby stars or other sources. It is important to analyse the behaviour of the calculated flux under small filter offsets, because our star finding algorithm has small displacements. Thus we have small differences between the found and the actual position. To check which filters are sensitive to such changes, we simulated a Gaussian distribution and calculated the total "signal" with different filters. We did this for several offsets and three different filters: Gaussian, pyramid and linear (no filter).

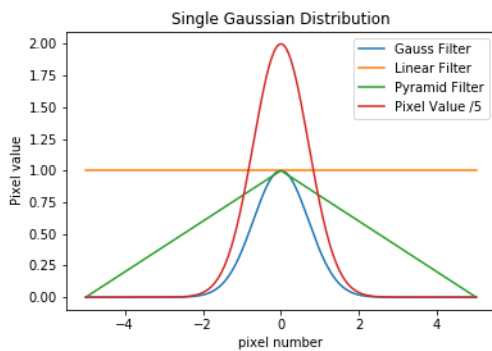


Figure 3: Simulation of a single Gaussian and the different filters placed at the centre.

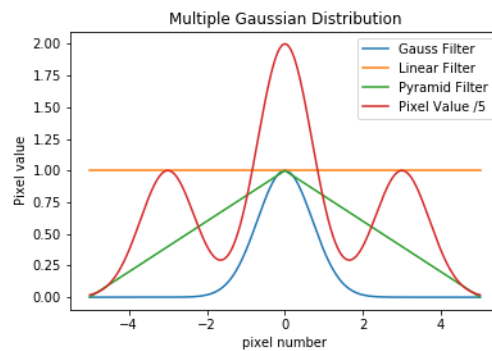


Figure 4: Simulation of three Gaussian and the different filters placed at the centre.

The distribution as well as the filters can be seen in figures 3 and 4. The filter will be multiplied with the data and thus gives us a weighted mean of the distribution which simulates a star. With the Gaussian and the pyramid distribution, values far off the centre get weighted less. The linear distribution weights all values equal. We then set the position of the filters at different offsets to see how the total sum changes. The values at different offsets can be seen in figures 5 and 6.

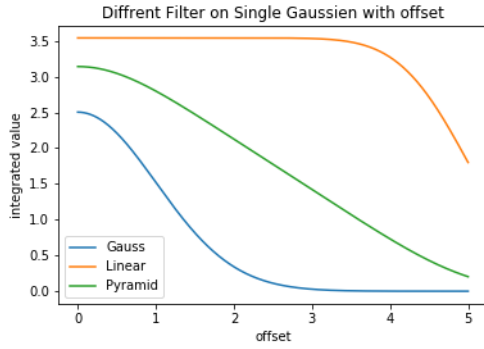


Figure 5: Sum of the single Gaussian under the filters with different offsets.

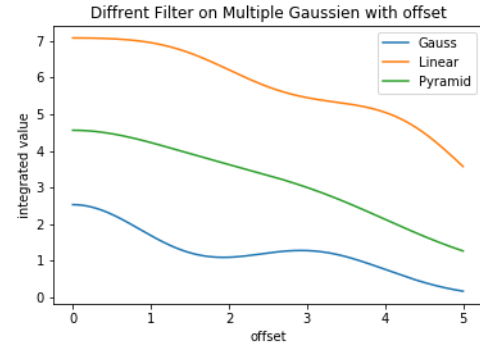


Figure 6: Sum of the three Gaussian under the filters with different offsets.

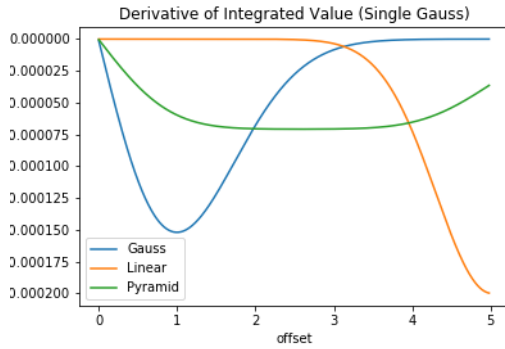


Figure 7: Derivation of the sum of the different filters for the three Gaussian.

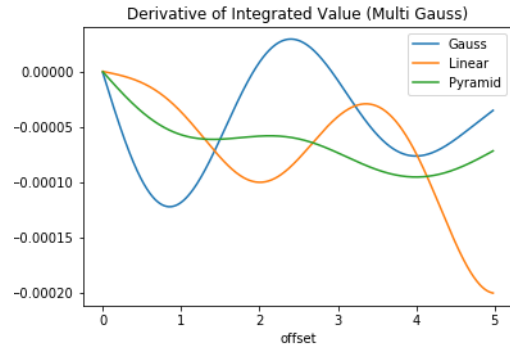


Figure 8: Derivation of the sum of the different filters for the three Gaussian.

Out of this simulations, we already can see a clear tendency for a single star. The linear filter gives far more stable results for small offsets then the other two. For large offsets the linear filter gets worse. For several stars, the difference isn't that large any more. To get a more distinguishable value we took the derivative. There the filters should preferably have a constant dependence. The derivations can be seen in figures 7 and 8.

The plot shows clearly that the linear filter is the best choice for our experiment. The Gaussian or pyramid filter would reduce the influence of nearby stars but also increase the errors on the flux calculations. Because we assume all star beside the transit star as constant, the influence of nearby star will give only a small additional error compared to the one caused by the offset. We also used the Gaussian filter on our observation data and found that the random errors indeed doubled compared to no filter.

2.5 Local Linear approximation

One of the most important criteria, if we can detect a transit or not is the random error on our data. We improved the image rate to its maximum to have as many data points as possible. Because of this, we can process our data to get clearer results. Our approach to achieve this was the local linear approximation (LLA). This method takes the mean value of two data points next to each other as a new data set. This gives us the linear approximation in a small region around each data point. With this calculation, we loose one measurement point at the edge of the graph. For our evaluation this doesn't matter as long as the baseline is long enough.

$$\text{Old Set: } x_i \quad \text{New Set: } y_i = \sum_{i=1}^{N-1} \frac{x_i + x_{i+1}}{2} \quad (1)$$

Overall the data points will now be closer to their local mean value. This gives us a better estimation of the mean value at different parts of our graph. It is important to only linearise locally because of the curve of a transit. If we would linearise the whole graph, the transit depth would become smaller and thus more difficult to detect. The down side of this method is that the values get affected by errors of subsequent data points. We

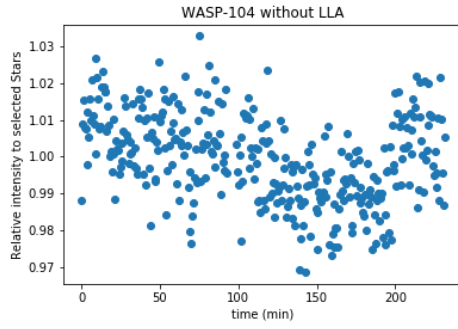


Figure 9: Relative intensity of WASP-104-b without local linear approximation.

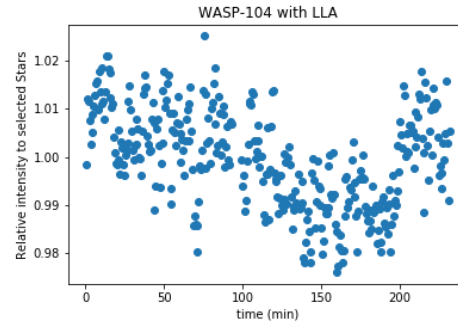


Figure 10: Relative intensity of WASP-104-b with local linear approximation.

tested the effect on our measurement of WASP-104-b to see how it actually changes the graph. The result can be seen in figures 9 and 10. A slight increase in transit depth can already be seen and the propagation of local errors is small.

The LLA can also be applied several times on the same data set. The values get influenced by an additional value each time the LLA gets applied. This leads to flattened edges of the transit. Normally, we have enough values around these edges to use the LLA several times before this influence gets to large. We tested this again on WASP-104-b with different number of LLA repetitions.

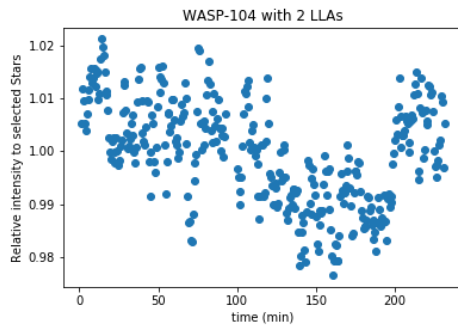


Figure 11: Relative intensity of WASP-104-b with 2 LLAs.

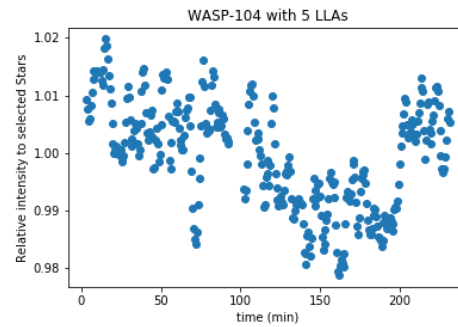


Figure 12: Relative intensity of WASP-104-b with 5 LLAs.

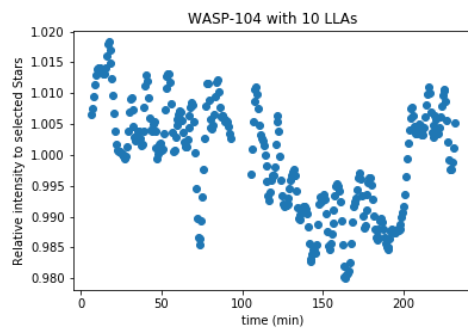


Figure 13: Relative intensity of WASP-104-b with 10 LLAs.

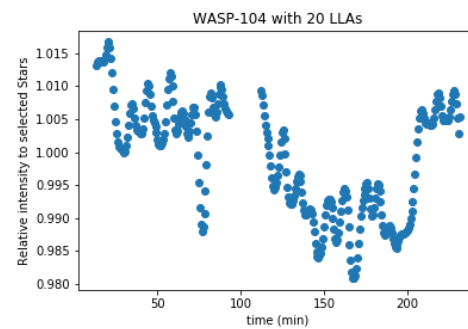


Figure 14: Relative intensity of WASP-104-b with 20 LLAs.

The figures show that the LLA works as predicted. A clear propagation of local errors can be seen after applying the LLA 20 times. The sinus-like behaviour of the values comes from local minima and maxima which get approximated linearly. A good example of a case when the approximation fails can be seen in figure 14 at minute 80. Originally we had some out layer values which no propagated through the others nearby. The transit can now be seen much clearer because most values are approximated fairly well. We also see that the edges of the transit get flattened but only slightly.

We then compared the new data set with our model of WASP-104-b to see if it still fits. The result can be seen in figure 15. The LLA processed data points still fight the model. The flattening of the left edge is now visible but still small compared to the transit length.

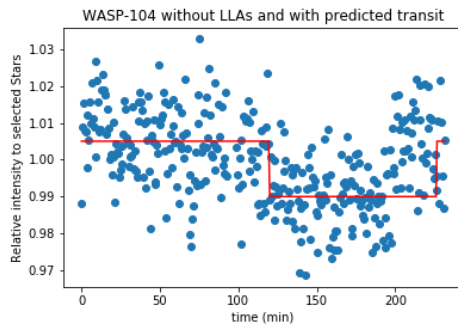


Figure 15: Relative intensity of WASP-104-b without LLAs and with the predicted transit (red line), base height adjusted.

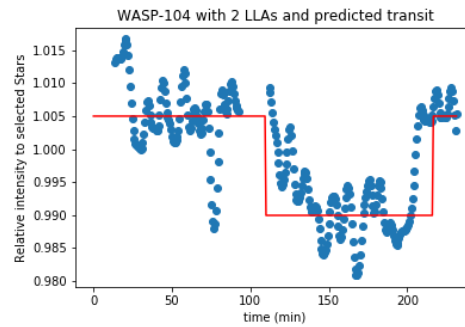


Figure 16: Relative intensity of WASP-104-b with 20 LLAs and with the predicted transit (red line), base height adjusted.

For our experiment, we decided to not apply the local linear approximation. Mostly, because our main problem was not the visibility of the transit depth but rather systematic errors we couldn't correct. The only transit too shallow was WASP-93-b. But because of too few data points, the LLA was not applicable in this case. Nevertheless, it could be a useful tool to observe transits at the limit of the telescopes' transit-resolution.

2.6 Full Width Half Maximum

To distinguish a transit from its baseline, we used the full width half maximum (FWHM) criteria. In most cases this was not necessary because the transit was deep enough to be clearly distinguishable. The calculation of the FWHM was done by calculating the histogram of the baseline (or the transit). An example of the FWHM calculation can be seen in figure 18.

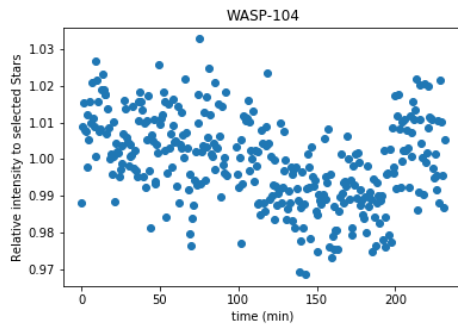


Figure 17: Relative intensity of WASP-104-b.

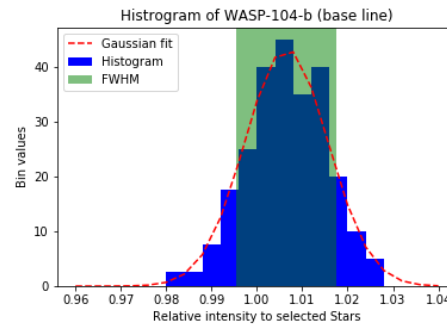


Figure 18: Histogram of WASP-104-b with Gaussian fitting and calculation of the FWHM.

We tried both methods for WASP-104-b. First we estimated the range of the random errors by eye, which yielded a value of at least 2%. With this estimation, we can distinguish transits of at least 2% depth. To get a better result, we then calculated the FWHM, which yielded a value of 2.2%. This means we are able to distinguish transits of at least 1.1%. The FWHM is always better than our first estimation by eye. In most cases the transits are already distinguishable by eye and we don't need to calculate the FWHM.

3 Evaluation Program

We wrote a python program for our evaluation of the data (figures 50 and 51). It works in three parts: flat and dark correction, flux calculation of the stars found and the plotting the data to make transits visible. The program is designed to compute all steps using only the raw data and the filtering condition. We used different libraries to calculate all this which gave us fast and reliable results. We used our observation data to optimize the individual steps and find reasonable start values for the filters. In this section we will give a short overview of the different steps of the python code. A more detailed explanation of the individual steps and algorithms can be found in A.Gheorghe's Thesis with whom the code has been written in parallel [8].

3.1 Image Correction

The raw images are full of errors caused from dust on the mirrors, read out noise on the CCD and bias values on the pixels. These are common sources of errors in astronomical data. They can be corrected by taking flat fields and dark frames before or after the experiment and correcting the data with them. The flats were taken in the twilight before or after the measurement. The darks were taken shortly before the experiment. We took 5 to 10 flats and darks for each measurement and made a master image by taking the median of each pixel. The reduced pictures can be calculated applying the following formula for each pixel:

$$data = (raw_data - master_dark) \frac{\langle master_flat - master_dark \rangle}{(master_flat - master_dark)} \quad (2)$$

We decided to not correct for bias because our integration times are shorter than 30s. This leads to a very small dark current. We assumed the correction for the bias values to be included in the darks. Additionally, we filtered for hot pixels. The flats and darks can remove dead pixels which always yield the same values but not for variable ones. These can be caused by read out errors, cosmic radiation or bad pixels, which sometimes yield wrong values. We corrected those by median filtering the data with the python library `ccdproc` [1]. This filtering is applied after the flat and dark correction.

3.2 Finding and Filtering of Stars

The next step is to find the stars in all pictures and link them to each other. There are always small displacements between the images. They are mostly small rotations and therefore cannot be easily tracked accurately with cross correlation. These lead to stars disappearing or appearing during the measurements. To get a steady reference frame the program only chooses stars visible in all images.

We used the `trackpy` library to find the stars and track them through all subsequent images [5]. These functions were originally written to track bacteria under a microscope. It searches for Gaussian distributions and keeps track of them over all images. Our stars are also Gaussian distributions and thus work with `trackpy`. We can automatically filter out too faint or disappearing stars in all pictures. Our program cannot filter out stars which are too bright or which have an unsteady flux. This can be done manually to get a steady reference frame. The tracking and filtering process we wrote can be used for every telescope as long as they made a subsequent set of the same part of the sky. The filter settings on the other hand will change if the code is used to evaluate data from another telescope. We didn't implement an automatic fitting of the filter settings because the evaluation already takes long. For our data we found the filter settings were very similar in all measurements.

3.3 Calculating the Flux

The flux calculation is an important step and can result in large errors or long run times. We decided to use two different libraries to optimize both. Several tools exist to calculate the flux of a star. For our experiment, we used `photutils` [4] according to the thesis of A. Bohn [7]. The position of the stars is calculated by the `trackpy` algorithm. We know the positions of all stars on all frames from `trackpy`. To start we set a circular aperture around the stars position and calculate the flux with the aperture photometry algorithm from `photutils`. We tested several sizes for the aperture and found that a radius of around the same size of the star gives a good first approximation. This minimizes the influence of small misplacements from `trackpy` and the influence of other stars nearby. A smaller aperture also reduces the area which can be affected by cosmic rays.

Furthermore, we tried a Gaussian filter on the star. The idea was to reduce marginal errors of cosmic rays and stars. In the end the variation of the not perfectly set filter was a lot larger than the reduction we got. We decided to not use a filter. If there is a second star at the age of the aperture it will influence the result but still less than the simple Gaussian filter we tried to use. The simulation can be seen in section 2.4.

3.4 Finding a Transit

Up to now it didn't matter which of the stars is the one with a predicted transit, as long as it doesn't get filtered out. We now search for a transit in all stars. This way, we can find transits even if we didn't know that one will happen and we also know that we didn't alter the filters to give us the result we wanted. We also have the curves of all other stars and they should all be constant. The total error in measurement, preparation or during the processing can be estimated with these stars.

We did the comparison by taking the median of all stars in one image. This way the result will not be influenced by fluctuation of a single star. It also has the advantage that if a second transit happens, it doesn't influence the transit curve of the first one. The disadvantage is that many stars have systematic errors. Most of them are also too bright or too faint and are not ideal for comparison. Overall, we get a good first estimation of which stars have a possible transit. Afterwards, this evaluation can be used to select good reference stars to compare the possible transit star with.

4 Observations

We tried to take measurements between October 2017 and May 2018. We had cloudy weather from the end of October till the beginning of March and thus couldn't take any measurements. During this time, we tried to improve the code but were limited by the little data we had to test it. After March we could observe several possible transits. We then used this data to improve our measurement technique and our program. In the end, we had data of 6 different possible transits to evaluate. Additionally we used the observation data to make the analysis in section 2.2 and 2.3.

We had some additional observation which we couldn't analyse. Mostly because we were interrupted by clouds. One measurement could not be evaluated, because the host star of the transit was too dark. We then took some example pictures of different stars. We found that stars with a magnitude higher than 14 are not distinguishable from the background noise if we use reasonable integration times.

We were also limited in our imaging angle because of the building our telescope is placed on. The view was blocked on the North side. There was also a bright warm light on top of the building, which influenced our measurements if we measured too close to it. The southern part is exactly above Zurich which also led to errors due to light pollution. On the East side was a small forest which blocked a lot of the light coming from cities nearby and thus was our preferred direction.

4.1 Observation 1: WASP-93-b (25.10.17)

Our first observation was of WASP-93-b. We had some clouds and only partly clear sky which got a bit better over the duration. We chose this star even though it is not ideal, because we didn't have a better one. This night was one of very few good observation nights in October. The idea was to get at least some data, which we could use to write our program.

4.1.1 Details of the Transit

WASP-93 is a 12 magnitude star in the constellation of Cassiopeia [2]. The Sky Chart as well as the information about the Transit can be seen below.

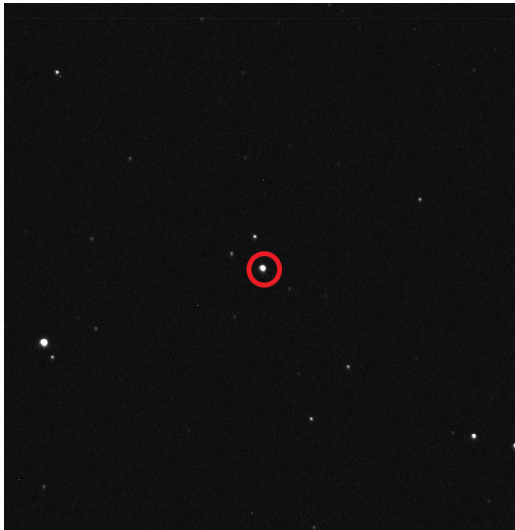


Figure 19: The position of WASP-93

Detailed information about WASP-93:

RA	00h 38' 10"
DEC	-51° 22' 25"
Magnitude	11.97 mag
Transit Depth	0.0119 mag
Transit Start	23:44 UTC (25.10.17)
Transit End	01:58 UTC (26.10.17)

Detailed in-

formation about the Telescope:

Cooling	-10 C
Diameter of Telescope	0.5 m
Focal length	12800 Microns
Exposure time	7 s

Images were taken from 23:00 until 02:30 once every 3 minutes. We had a technical issue from 00:35 to 00:45 because the telescope reached its maximal rotation and couldn't correct it automatically. This resulted in a 15 minute brake of imaging. Afterwards, the camera and therefore the images were shifted and rotated. This means we evaluated the two parts separated and set them together afterwards.

4.1.2 Transit Calculation

We evaluated the data with our program. As mentioned, we did it for both parts separate but used the same filter settings. Because of the rotation, both parts use different reference stars and thus don't have the same relative intensity. As mentioned before, the filters were set to minimize the noise on all stars and not especially on the one with the predicted transit. The result of the program can be seen in figures 20 and 21.

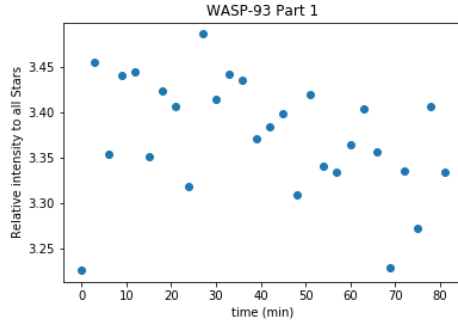


Figure 20: Relative intensity of WASP-93 from 23:00 to 00:35 UTC.

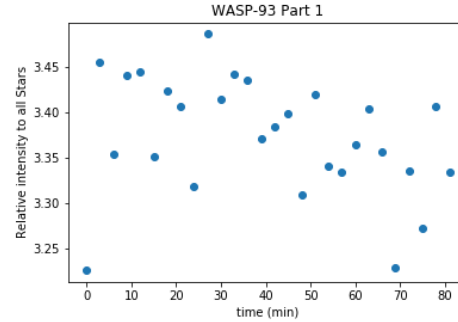


Figure 21: Relative intensity of WASP-93 from 00:45 to 02:30 UTC.

Afterwards, we used the first evaluation to find possible reference stars. Because many stars appear in all images, we could choose good reference stars. We decided on stars with similar brightness and light curve as our transit star. The result of the evaluation and combination of both parts together can be seen in figure 22. The predicted transit can be seen in figure 23 [3]. The errors in our measurements are around 3%. This makes it not possible to see a transit of only 1%. Nevertheless, a tendency can be seen in the picture and thus is a good start. These pictures were evaluated with a later version of our program.

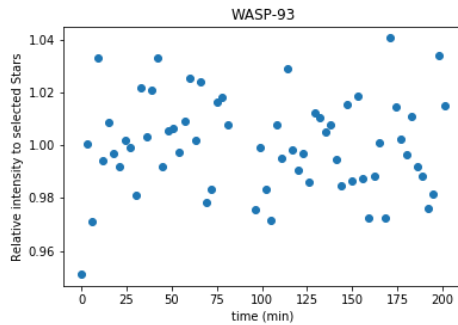


Figure 22: Relative intensity of WASP-93 from 23:00 to 02:30 UTC.

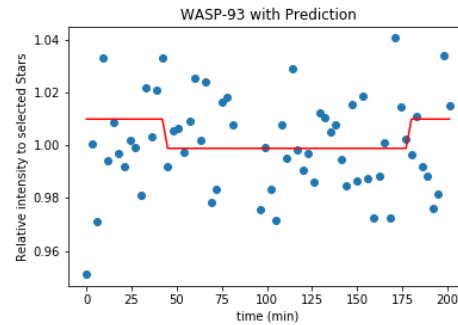


Figure 23: Relative intensity of WASP-93 from 23:00 to 02:30 UTC with the predicted transit (red line), base height manually adjusted.

4.2 Observation 2: Qatar-2-b (21.3.18)

After, we had no chance to observe during the winter months, we could start again in March. Our first possible target was Qatar-2-b. The conditions were better and we could observe without clouds. We chose this transit because it was the only one with a brightness higher than 1.5%. The declination of around 30 degrees wasn't ideal but still high enough to make reasonable measurements. The rotation error of the telescope, which we already had in the first observation, occurred again, exactly at the end of the transit.

4.2.1 Details of the Transit

Qatar-2 is a 13.3 magnitude star in the constellation of Virgo [2]. The Sky Chart as well as the information about the transit can be seen below.

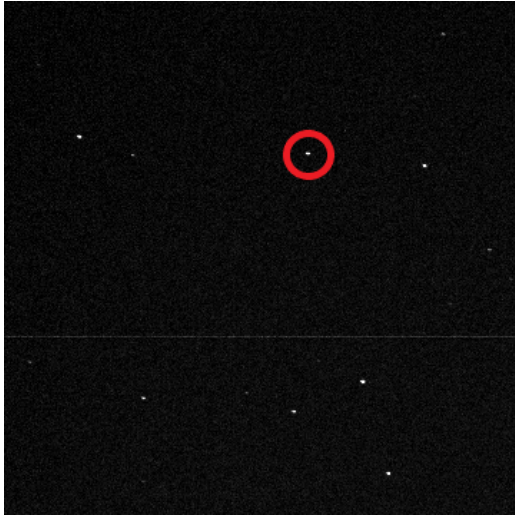


Figure 24: The position of Qatar-2-b

Detailed information about Qatar-2:

RA	13h 50' 37.1"
DEC	-06° 48' 14.4"
Magnitude	13.3 mag
Transit Depth	0.0374 mag
Transit Start	01:26 UTC (22.03.18)
Transit End	03:15 UTC (22.03.18)

Detailed information about the Telescope:

Cooling	-10 C
Diameter of Telescope	0.5 m
Focal length	12800 Microns
Exposure time	25 s

Images were taken from 00:25 until 04:15 once every 40 seconds. We had a rotation error of the telescope from 03:08 to 03:20 during which no images could be taken. We also had problems evaluating the data from 03:00 to 03:08 which we assume to be related to the rotation error. In total we lost 26 consequent images. We were not able to find Qatar-2 before we started the second part of our measurement. Thus, we could not optimize the position nor the integration time and used the same as before the error occurred. We could find the star in the second part during the analysis. The brightness of the transit star in both parts are approximately the same. The star was still visible during the whole second part but it was closer to the edge of the CCD.

4.2.2 Transit Calculation

We analysed the data with our program. As mentioned, we first ran both parts separately but used the same filter settings to reduce the noise of all stars. All found stars were used for this analysis. Because of the shift between part 1 and 2, those parts do not use the same reference stars and thus the same stars have different relative intensities.

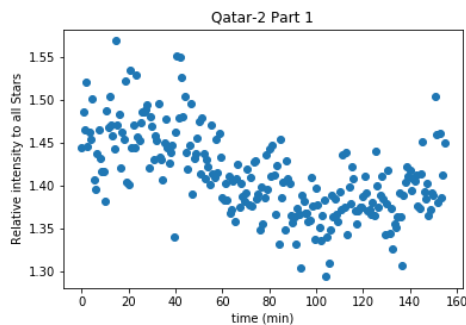


Figure 25: Relative intensity of Qatar-2-b from 00:25 to 03:08 UTC.

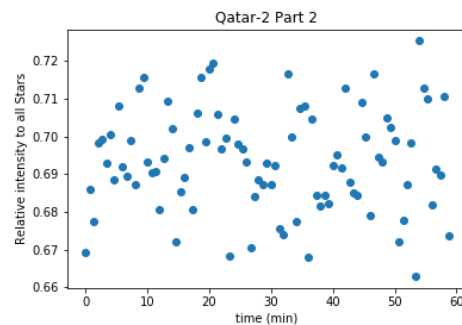


Figure 26: Relative intensity of Qatar-2-b from 03:15 to 04:15 UTC.

Afterwards, we used the first evaluation to find possible reference stars. This was difficult because of the large shift between the two parts. We still found some suitable stars which we then used to evaluate both parts together. In figure 27 a possible transit can be seen. There is a drop in the values of around 3.5% for 110 minutes. The errors on this measurement are approximately 2.5% which is sufficient to confirm a transit of this depth. The red line in figure 28 is the predicted transit according to our source [3].

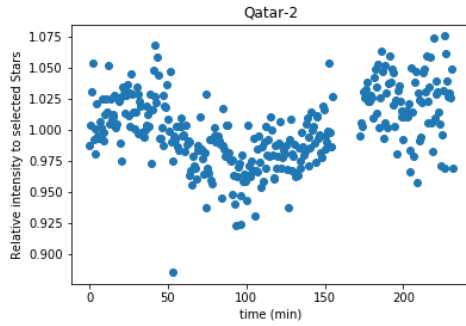


Figure 27: Relative intensity of Qatar-2-b from 00:25 to 04:15 UTC.

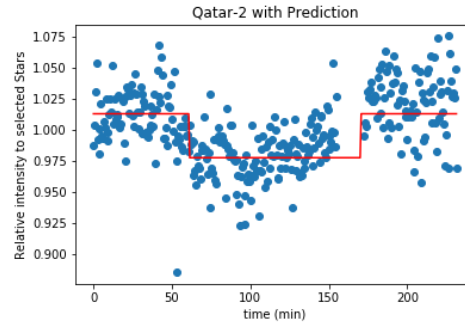


Figure 28: Relative intensity of Qatar-2-b from 00:25 to 04:15 UTC with the predicted transit (red line), base height manually adjusted.

4.3 Observation 3: WASP-104-b (10.4.18)

In April, we had the possibility to make several measurements. The first one was WASP-104-b. We had some clouds but could not see them directly on our images. Even if we cannot see the clouds, they could still have an influence on our measurements. We chose WASP-104-b according to our criteria in section 2.1. Especially the high declination (horizontal coordinate system) and the relatively low magnitude made it a good choice.

4.3.1 Details of the Transit

WASP-104 is a 11.12 magnitude star in the constellation of Leo [2]. The Sky Chart as well as the information about the transit can be seen below.

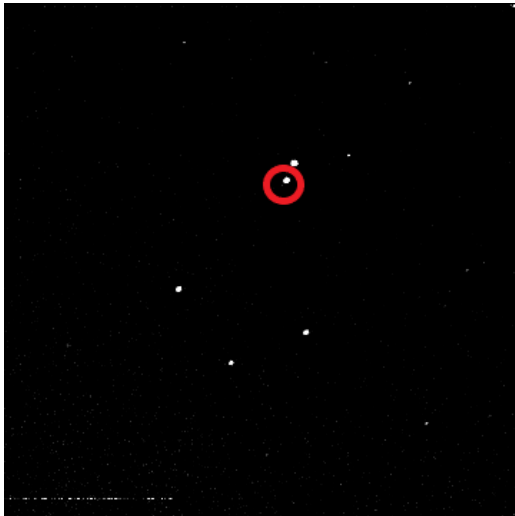


Figure 29: The position of WASP-104

Detailed information about WASP-104:

RA	10h 42' 24.61"
DEC	+07° 26' 06.3"
Magnitude	11.12 mag
Transit Depth	0.0158 mag
Transit Start	00:31 UTC (11.04.18)
Transit End	02:17 UTC (11.04.18)

Detailed information about the Telescope:

Cooling	-10 C
Diameter of Telescope	0.5 m
Focal length	12800 Microns
Exposure time	25 s

Images were taken from 22:00 until 02:30 once every 40 seconds. We had a rotation error of the telescope from 23:34 until 23:38, which resulted in the loss of 8 data points. Before restarting after the error, we found the transit star again and placed it in the middle of the CCD. The brightness of the star was still in the same range as before.

4.3.2 Transit Calculation

Because of the error, we evaluated the different parts separate as a first step. We had several other stars as possible references and many of them were of approximately the same brightness. The first analysis was done automatically by our program and the transit star was compared to all others.

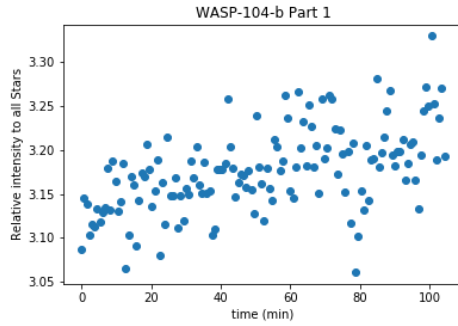


Figure 30: Relative intensity of WASP-104-b from 22:00 to 23:34 UTC.

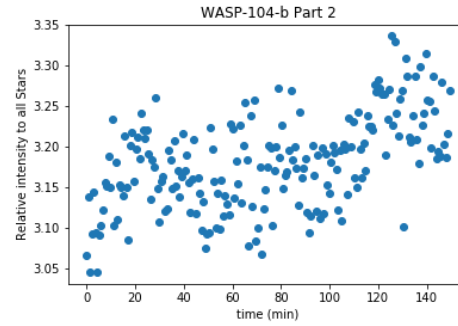


Figure 31: Relative intensity of WASP-104-b from 23:38 to 02:30 UTC.

Afterwards, we used this first evaluation to look for suitable reference stars. Our first analysis with stars of similar brightness did not yield good results because of small systematic errors. To correct those, we chose stars with a similar light curve. The result can be seen in figure 32. Even though we had many reference stars, we still had too few to get rid of the systematic error completely. The model of the transit [2] can be seen in figure 33. A drop in values can be seen at the right time and depth.

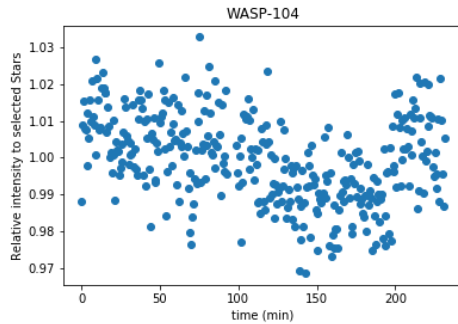


Figure 32: Relative intensity of WASP-104-b from 22:00 to 02:30 UTC.

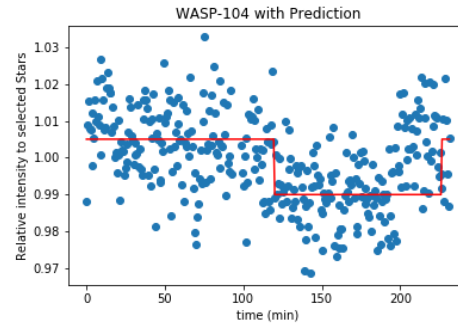


Figure 33: Relative intensity of WASP-104-b from 22:00 to 02:30 UTC with the predicted transit (red line), base height manually adjusted.

Because the transit depth was rather small, we calculated the FWHM (full width half maximum). This allowed us to get a better estimation of the depth that is still distinguishable. We found the FWHM for the first baseline to be approximately 2.2% around the mean value. This allowed us to distinguish a transit of 1.5% from its baseline.

4.4 Observation 4: Hat-P-36-b (17.4.18)

After the long break during winter, we wanted to imagine every possible star to get as much data as we can. The night of the 17 April was partly cloudy but we tried nevertheless. We chose a transit star with the criteria in section 2.1. This made Hat-P-36 the best choice this evening. Especially the declination (horizontal coordinate system) which started at 80 degrees and stayed over 50 degrees made it a good choice for a partly clouded night.

4.4.1 Details of the Transit

Hat-P-36 is a 12 magnitude star in the constellation of Canes Venatici [2]. The Sky Chart as well as the information about the transit can be seen below.

Images were taken from 22:21 until 04:00 once every 40 seconds. This was approximately 3 hours before the Transit started. We did this to get a good baseline to work with. Especially because of the bad weather condition, we expected to need a good baseline for reducing the errors. Once more, we had a rotation error

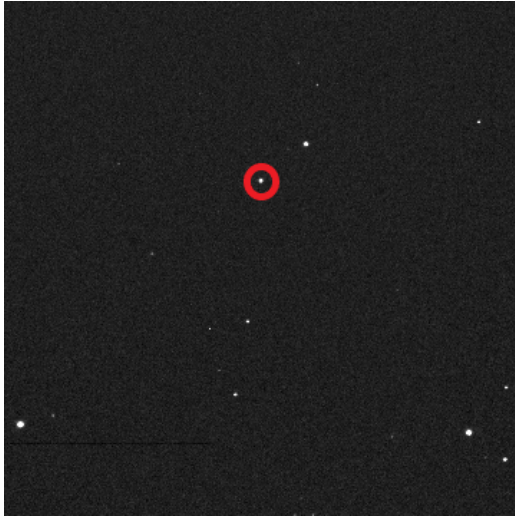


Figure 34: The position of Hat-P-36-b

Detailed information about Hat-P-36:

RA	12h 33' 03.96"
DEC	+44° 54' 55.3"
Magnitude	12.26 mag
Transit Depth	0.0204 mag
Transit Start	01:11 UTC (18.04.18)
Transit End	03:23 UTC (18.04.18)

Detailed information about the Telescope:

Cooling	-10 C
Diameter of Telescope	0.5 m
Focal length	12800 Microns
Exposure time	15 s

from 00:56 to 01:03 which resulted in a loss of 11 data points. Before restarting after the error, we could find the transit star again and placed it in the middle of the CCD. The brightness of the star was still in the same range as before.

4.4.2 Transit Calculation

Because of the error, we evaluated the different parts separate as a first step. We had several other stars as possible references and many of them were of approximately the same brightness. The first analysis was done automatically by our program and the transit star was compared to all others. Because of the bad weather, we had large random errors and some systematic errors presumably from clouds in our line of sight. A good example of such an error can be seen in figure 35 at 80 minutes.

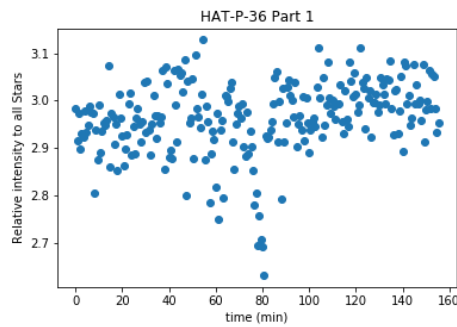


Figure 35: Relative intensity of HAT-P-36-b from 22:21 to 00:56 UTC.

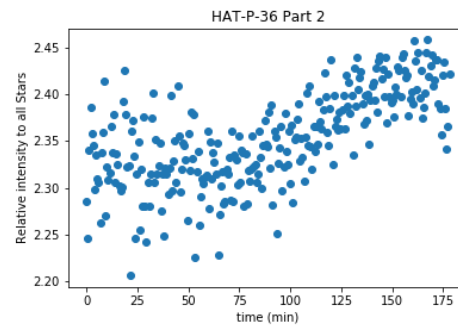


Figure 36: Relative intensity of HAT-P-36-b from 01:03 to 04:00 UTC.

Afterwards we used this first evaluation to look for suitable reference stars. Our first analysis with stars of similar brightness did not yield good results because of small systematic errors. To correct those, we choose stars with a similar light curve. The result can be seen in figure 37. Even though we had many reference stars, we still had too few to get rid of the systematic error completely. There is also a slight shift in height between the two parts. The influence of the weather is visible in figure 37. In the beginning it was cloudier, which yielded in errors of around 3-4%. Towards the end, the weather cleared and the error was around 1-2%. The model of the transit [2] can be seen in figure 38. A drop in values can be seen at the right time and depth.

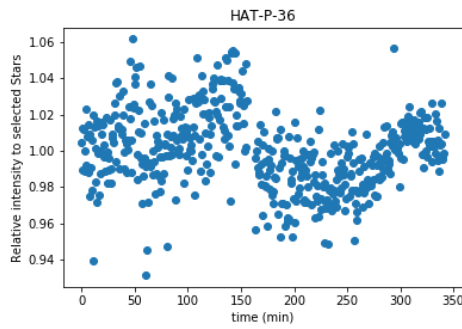


Figure 37: Relative intensity of HAT-P-36-b from 22:21 to 04:00 UTC.

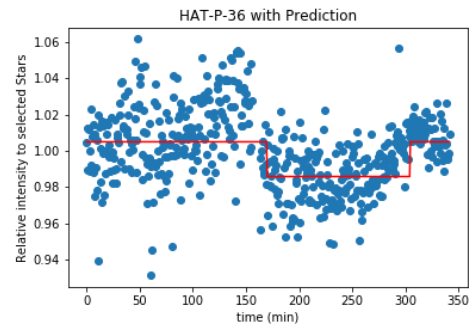


Figure 38: Relative intensity of HAT-P-36-b from 22:21 to 04:00 UTC with the predicted transit (red line), base height manually adjusted.

4.5 Observation 5: WASP-85A-b (19.4.18)

We had the possibility to measure two transits in one night. The first one was WASP-85A-b. The sky was clear and stayed cloudless over the whole measurement. We chose this transit because the timing allowed us to measure it in the same night as TrEs-3-b. The declination (horizontal coordinate system) stayed over 30 degrees during the whole measurement duration and it had a transit depth of over 2%. During the experiment we had the same rotation error as in the previous experiments but it happened mid transit thus we didn't lose important data.

4.5.1 Details of the Transit

WASP-85A is a 11.2 magnitude star in the constellation of Virgo [2]. The Sky chart as well as the information about the transit can be seen below:

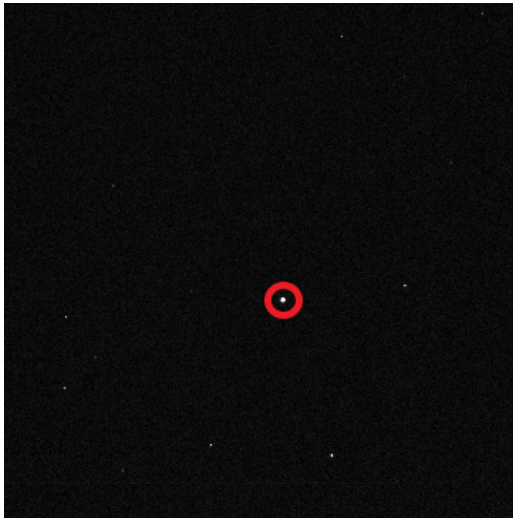


Figure 39: The position of WASP-85A

Detailed information about WASP-85A:

RA	11h 43' 38.01"
DEC	+06° 33' 49.4"
Magnitude	11.2 mag
Transit Depth	0.0284 mag
Transit Start	22:48 UTC (19.04.18)
Transit End	01:29 UTC (20.04.18)

Detailed information about the Telescope:

Cooling	-10 C
Diameter of Telescope	0.5 m
Focal length	12800 Microns
Exposure time	15 s

Images were taken from 22:04 until 02:01 once every 30 seconds. The rotation error occurred at 00:00 and could be solved at 00:04. During this time, we missed 9 Images. Before restarting after the error, we found the transit star again and placed it in the middle of the CCD. The brightness of the star was still in the same range as before.

4.5.2 Transit Calculation

Because of the error, we evaluated the different parts separate. We only had few other stars and thus we had to take stars which weren't optimal. Additionally, we noticed a periodic error in many of these stars of the first

part. We tried to reduce the periodic error by choosing the right reference stars, which didn't work. In the end we choose all stars of similar brightness as reference stars for a first evaluation.

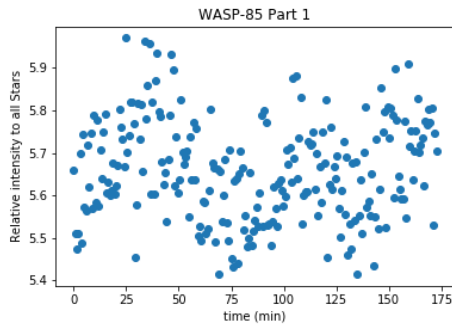


Figure 40: Relative intensity of WASP-85A from 22:04 to 00:00

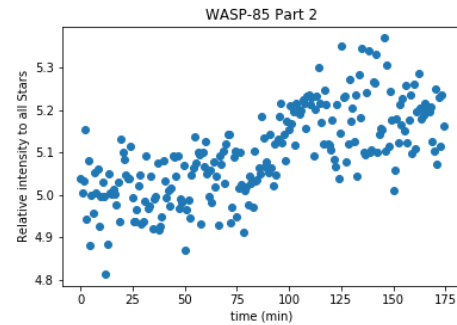


Figure 41: Relative intensity of WASP-85A from 00:04 to 02:01

The periodic error in the first part was not only in the transit star, but also in many of the possible reference stars. We compared stars with similar periodicity in the systematic error and chose them as references. We found that we could either reduce the periodic error or the systematic shift between the parts but not both. This was mainly because of the lack of reference stars. The best approximation can be seen in figure 42. In figure 43, we compare it with our model [2]. There is a small shift in time, but this could result from the periodic error.

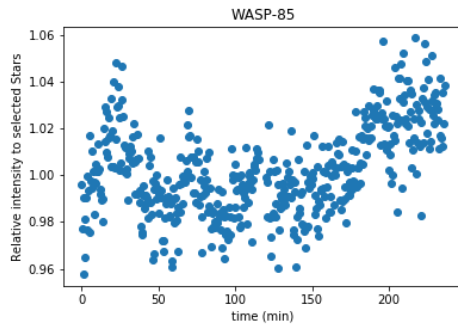


Figure 42: Relative intensity of WASP-85A-b from 22:04 to 02:01 UTC.

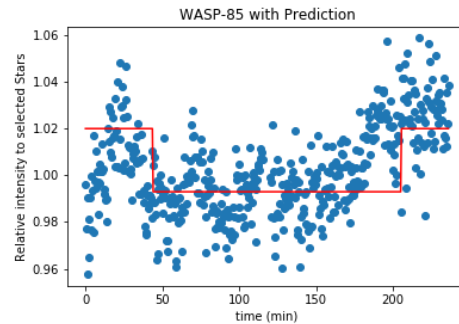


Figure 43: Relative intensity of WASP-85A-b from 00:04 to 02:01 UTC with the predicted transit (red line), base height manually adjusted.

4.6 Observation 6: TrEs-3-b (19.4.18)

This was the second transit of this night shortly after WASP-85A-b. The weather remained good and we could start with the measurement on time. We chose this transit because the timing allowed us to measure it in the same night as WASP-85A-b. We could take all images above 50 degrees declination (horizontal coordinate system) thus we had less errors because of atmospheric influences. Unfortunately, we had 2 different computer errors shortly after each other and lost 30 data points in total. Nevertheless, we finished the measurement and evaluated the data.

4.6.1 Details of the Transit

TrEs-3-b is a 12.4 magnitude star in the constellation of Hercules [2]. The Sky chart as well as the information about the transit can be seen below:

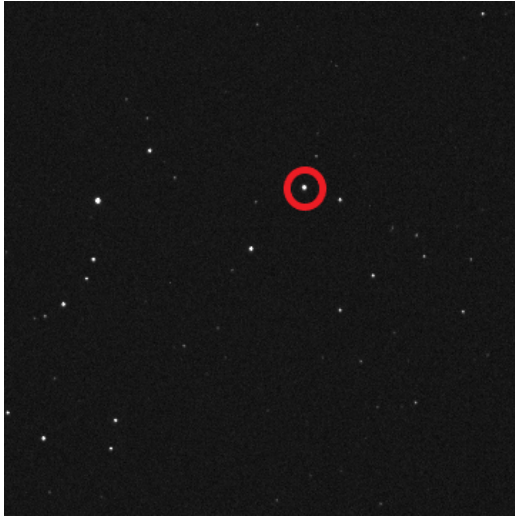


Figure 44: The position of TrEs-3

Detailed information about TrEs-3:

RA	17h 52' 07"
DEC	+37° 32' 46"
Magnitude	12.4 mag
Transit Depth	0.0291 mag
Transit Start	02:42 UTC (20.04.18)
Transit End	03:59 UTC (20.04.18)

Detailed information about the Telescope:

Cooling	-10 C
Diameter of Telescope	0.5 m
Focal length	12800 Microns
Exposure time	30 s

Images were taken from 02:06 until 04:31 once every 45 seconds. As mentioned we had 2 computer errors during the measurement. The first one between 02:58 and 03:14, the second one between 03:20 and 03:24. Both could be solved and didn't influenced the other images but we lost 30 data points.

4.6.2 Transit Calculation

Because of the errors, we evaluated the different parts separate as a first step. We had many stars in our images and thus we had large systematic and random errors in all stars. In the second part we only had 6 images and thus no real tendency could be seen in any of the stars. In the last part, we had more data but because of too many stars we had again large systematic errors. The transit star evaluated with this method can be seen below for each different part.

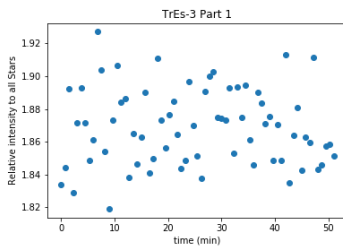


Figure 45: Relative intensity of TrEs-3 from 02:06 to 02:58

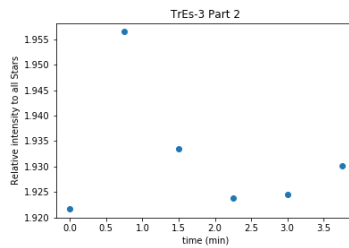


Figure 46: Relative intensity of TrEs-3 from 03:14 to 03:20

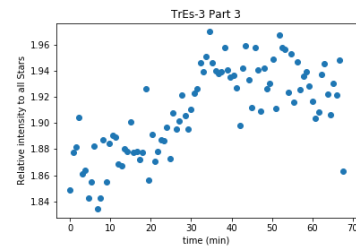


Figure 47: Relative intensity of TrEs-3 from 03:20 to 04:31

Out of this result we selected stars which behaved like our transit star. This included systematic behaviour and brightness. We chose 4 reference stars which satisfied these conditions for a more selected evaluation. A possible transit is visible in figure 48. There is a drop in values of around 2.5% for 1 hour. The errors on this measurement are approximately 2% which is good enough to find a transit of this depth. The red line in figure 49 is the predicted transit according to our source [3].

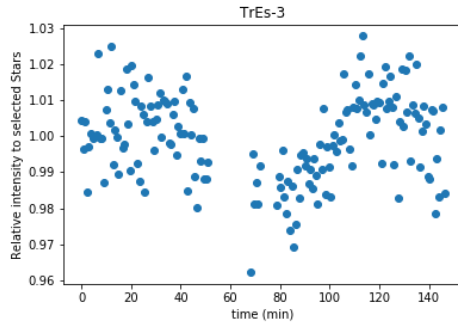


Figure 48: Relative intensity of TrEs-3 from 02:06 to 04:31

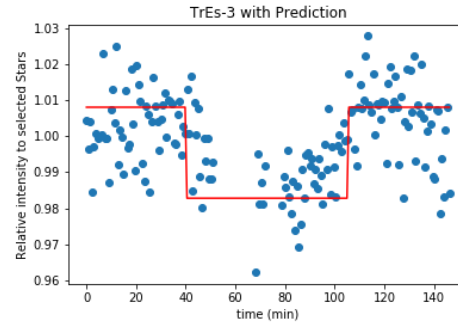


Figure 49: Relative intensity of TrEs-3 from 02:06 to 04:31 with the predicted transit (red line, base height manually adjusted)

5 Discussion

After several months of observing, we gathered data of 5 different possible transits. We always chose easy targets to get a clear result. In the end, we still had stars with different properties. We observed stars with magnitudes in the range between 11.1 to 13.3, transit depths of magnitudes 0.0158 to 0.0374 and we had clear skies but also partly clouds. We can use these different conditions to make predictions on what is possible to measure and what is rather impossible.

5.1 The Telescope and our Evaluation

The limitation on which transits one can measure largely depends on the condition during the observation night and the telescope. During our evaluation, we noticed that our filtering parameters were nearly identical. One reason for this is that we always chose the integration time to set our transit star around the same value. Another is the telescope which gives always similar values for the Gaussian distribution of the stars and thus our filters in the flux calculation were all approximately the same.

The random errors in all our data with a clear sky is approximately 2%, during partly cloudy nights even up to 4%. This error rises from various sources including the telescope, our surrounding and the evaluation. We saw that the errors became larger if we measured parts of the sky close to cities. We always tried to measure away from those light sources, but this wasn't always possible. The next problem was that we were surrounded by a city which is why we assume a large influence of the light pollution in all parts of the sky. The background correction we used to reduce our systematic errors also lead to additional random errors. The telescope is also a source of errors even though we made dark and flat correction. Our evaluation of the data also leads to some random errors, because our positioning and flux calculation algorithms.

To be able to make a clear statement if we observed a transit or not, we need the transit depth to be bigger than our random errors. Our random errors are approximately 2%, so our transit depth should at least be equally deep. Fortunately, this was given in most of our measurements. During partly cloudy nights, the error rises to 4%. There are currently no transits known with a depth of more than 3.5% and thus it is not possible to observe a transit during partly clouded nights.

5.2 Qatar-2-b

Our measurement on Qatar-2-b has a clearly visible baseline and drop in values during the modelled transit time. Even though there is still a small systematic error during the first part of the measurement, the transit values can be clearly distinguished from the baseline values. The time of our predicted transit also fits the modelled one. Hence, we consider this transit confirmed.

5.3 WASP-104-b

This transit had only a depth of 1.5% but we had errors of 2%. Thus we calculated the FWHM, which yielded a value of 2.2%. This is good enough to confirm a transit of 1.5% depth. If we compare our observed data with

the model, we can see a drop in values of approximately the right depth. The beginning also matches the model, but the end doesn't. This could be explained in small imperfections in the modelling and our evaluation. Over all we assume this measurement as a plausible transit, but not as a definite confirmation.

5.4 Hat-P-36-b

We had several difficulties during this measurement. The weather conditions lead to large errors and the rotation error shortly before the transit led to difficulties in the compression. Thus, the start of the transit is not well defined and cannot be confirmed. The end of the transit is clearly visible and matches the model, but because we couldn't make more pictures, the baseline is not well defined. Overall it is not possible to clearly confirm the transit, even though the second part matches the model well.

5.5 WASP-85A-b

The systematic errors in the first part of the measurement make it impossible to confirm this transit. The second part has less systematic errors and the baseline as well as the transit depth can be defined and matches the model. The periodic error in the first part is much larger than the transit depth. Even though the tendency of a transit can be seen, it cannot be confirmed.

5.6 TrEs-3-b

TrEs-3-b has a rather short transit duration and a higher magnitude. This leads to less data points than we had for the other measurements. Additionally, we had two technical difficulties which reduced the amount of data further. Nevertheless, the baseline and the transit can be seen very clearly. Both parts match each other in values. They also fit inside our model. We consider this transit confirmed.

6 Conclusion

It is possible to measure a transit with the 0.5m telescope on the HPP building. The limiting factors are mostly weather conditions and if a suitable transit is visible during the night. For this telescope and our evaluation program a suitable transit is given by the following properties:

1. Fully visible during the night, including at least an hour before and after the transit
2. A depth of at least 1.5%
3. A magnitude of the host star of less than 14
4. A declination (horizontal coordinate system) of more than 30 degrees during the whole measurement time

Our program can find transits if the standard filter settings are given. These settings are defined by the telescope and the exposure time. It is not always possible to detect the transit automatically. To confirm a transit, the right reference stars must be selected. Normally, the stars with similar brightness are a good choice. If there are large systematic errors in the images, the light curve also needs to be considered.

It is best to use no additional filtering masks on the stars. Masks reduce the influence of nearby stars but the errors rising from small offsets are larger. It is better to just use a linear regime on the CCD. It is difficult to set the star to a certain region and even more difficult to hold it there. For our CCD, only the edges had a small non-linear behaviour and thus it is enough to place the transit star approximately in the middle and rule out reference stars at the edge of the CCD. If necessary, the local linear approximation can be used to make the search for transit easier. In the end, the difficult transits have to be distinguished by the full-width-half-maximum criteria.

7 Outlook

The main goal of this experiment was to test if it is possible to find an exoplanet with a relatively small telescope despite the light pollution of a city nearby. The idea was to probe the limits of what is possible and how good our results can be. Unfortunately, we had bad weather conditions over a long period which didn't allow us to realise all our ideas or improvements. If we had more time we would realise the following ideas, ordered by their priority.

7.1 Improvement of the Experiment and the Program

The biggest problem with our experiment was the few observations because of the bad weather condition over several months. With more data, the program could be optimised and improved much more than we managed. During spring the weather got better and we could start testing our program and improve it. We would not recommend repeating this experiment during the fall or winter months. Even if the weather would be better, a completely clear sky is necessary and there will only be few of them.

There are not many things to be improved on the measurement itself. We already took more flats than necessary and the same goes for the darks. We took as many images as technically possible to reduce the statistical errors in our measurements. This rate is given by the processing speed of our camera and cannot be increased much. If the same transit can be measured several times or simultaneously with several telescopes, the results could be improved and one might find improvements on the measurements themselves.

The program we wrote could also be ameliorated. The flux calculation could be improved to reduce the random error. We assume that especially a better background correction could reduce the random error on the flux significantly. We already implemented some ideas but they only resulted in minimal improvement.

7.2 Automation of the Program

A next step would be to make the program fully automatic. For example, the filter values would need to be fitted. We tried to implement this but ended with run times over 8 hours which is not practical for this application. Another improvement could be the automatic selection of suitable reference stars. We manually chose the stars with similar behaviour to our transit star during our first evaluation. There are possible algorithms to rule out stars with a strange light curve or the ones that are simply too faint or bright. A helpful addition to our program would be an algorithm that matches pictures with a large offset. This could be used to fit pictures after the rotation error without doing it by hand. The difficulty is to find an algorithm that can do this efficiently and afterwards matches the data into a format that can be evaluated.

7.3 How small is too small?

Our telescope had a diameter of 0.5m. This is small for professionals but rather big for amateur astronomers. One of our ideas was, to measure the same transit with different sized telescopes to see if we can get the same results. Not only the size matters but also the rest of the equipment. It would be much more difficult to get a good measurement if the tracking wasn't as good as ours and even worse if you must do it by hand. To test this idea would be an important next step in defining the limits on telescopes for exoplanetary research.

7.4 Importance for amateur astronomer

The detection of exoplanets not only depends on good equipment but also on chance of observing the right star during its transit. This chance can be improved if amateur astronomers were able to search for exoplanets. The detection of exoplanets with smaller telescope and cheaper CCDs is still very hard. But the program we wrote could be a first step in the right direction. If it could be generalized more, it could possibly evaluate data of normal cameras and smaller telescopes. If the minimal equipment to measure a transit is also known, we can predict which amateur astronomers would be able to find exoplanetary transits. With a suitable interface our program could then be the tool to evaluate their data. Because our program needs rather small user input, it is a good start for a software for non-physicists. We already started to implement some interface structure but it still needs to be worked out.

8 Acknowledgements

I would like to thank Adrian Gheorghe for doing this project together with me. Especially the development of the python code was difficult and time consuming. His commitment allowed us to create an analysis tool even with only one measurement.

Furthermore, i want to thank Luca Tortorelli for his introduction to the telescope and support during this project. We had several technical difficulties which we could solve quickly because of his help.

Finally, i would like to thank Dr. Sascha Quanz for giving us the opportunity to work on this project, the freedom to develop our own approach and the help to achieve our goal. In the end, his support in our and the previous transit projects allowed us, for the first time, to confirm an exoplanetary transit from the ETH campus Hoengerberg with the HPP 0.5m Telescope.

References

- [1] CCDPROC ccdproc v2.0.dev1352. <https://ccdproc.readthedocs.io/en/latest/#> (Apr. 2018).
- [2] ETD - Exoplanet Transit Database. <http://var2.astro.cz/ETD/predictions.php> (Apr. 2018).
- [3] Exoplanet Orbit Database | Exoplanet Data Explorer. <http://exoplanets.org/> (Apr. 2018).
- [4] Photutils photutils v0.4. <https://photutils.readthedocs.io/en/stable/> (Apr. 2018).
- [5] Trackpy: Fast, Flexible Particle-Tracking Toolkit trackpy 0.3.2 documentation. <https://soft-matter.github.io/trackpy/v0.3.2/> (Apr. 2018).
- [6] Simon Birrer. Hpp observatory quick user guide, 2015.
- [7] Alexander Julian Bohn. Detection of an Exoplanet Transit Using the 0.5m Telescope on HPP, July 2016.
- [8] Adrian Alexandru Gheorghe. Towards measuring exoplanetary transits using the hpp telescope and developing a standalone program for data analysis, May 2018.

9 Appendix

We had different python scripts to process our data. The programs we used to evaluate the data for this report are listed below. They were written together with the scripts in the thesis of A. Gheorghe [8]. Both use the same algorithms and analysis. The first program (figure 50) can automatically evaluate the data and give an estimation if a transit was found or not. This program can only analyse a continuous set of images. If an error occurred (like the rotation error) the parts have to be analysed separately. The program needs to be placed next to 4 folders:

1. Data: containing the raw data gathered by the telescope
2. Flat: containing the flat fields of the observation
3. Dark: containing the dark frames of the observation
4. Reduced: the reduced images will be saved in this folder
5. Result: the intensity graphs of the data will be saved in this folder

Additionally, the evaluation will be saved as a text file. This files can be used for the second program (figure 51) to combine the different parts. To do so, the files have to be named "result_1", "result_2", etc. The program also needs the star-numbers of the reference stars. Unfortunately, these stars must be found in both parts manually. There must also be a folder called "Result" in which the evaluation gets saved. To attune the programs to different telescopes or data, only the variables in the beginning of each code has to be adjusted. This allows a easy and fast use of the program.

Figure 50: Python code for analysing the individual parts of our observations.

```
# -*- coding: utf-8 -*-
"""
Created on Sun Apr 22 12:41:58 2018

@author: Sven Kiefer
"""

#Bibliotheken
import numpy as np
import trackpy as tp
import matplotlib.pyplot as plt
import photutils as pu

from astropy.io import fits
from natsort import natsorted
from astropy.table import hstack

import glob
import os
import ccdproc

#Filter Variables: Need to be adjusted for different Transits and Telescopes
min_m = 5000 #Minimum brightness of the star
mask_radius = 10 #radius of the filter -> not always radius of star
mask_radius_out1 = 30 #for Background correction -> inner radius
mask_radius_out2 = 45 #for Background correction -> outer radius
time_per_frame = 45 #time between pictures (in Seconds)
select = np.array([]) #Add number of stars you want to compare
scaling = False #Set true if tranist should be scaled to 1

#The program works automatically from here

print("Start Part 1")
#-----Data auslesen und korrigieren

#Delete old reduced
redu_n = glob.glob('Reduced/*.FIT')
for redu_i in redu_n:
    os.remove(redu_i)

#get references to all images
data_n = glob.glob('Data/*.FIT')
flat_n = glob.glob('Flat/*.FIT')
dark_n = glob.glob('Dark/*.FIT')
data_n = natsorted(data_n) #Sort files by name

#size of pictures
pic_size_x = np.shape(fits.open(data_n[0])[0].data)[0]
pic_size_y = np.shape(fits.open(data_n[0])[0].data)[1]
```

```

#Create master image by median
def master(name_n):
    length = len(name_n)
    pics = np.zeros([pic_size_x, pic_size_y, length])
    i = 0
    for name_i in name_n:
        pics[:, :, i] = fits.open(name_i)[0].data
        i += 1

    #if no picture given, create a all 1 picture
    if(i != 0): m_pics = np.median(pics, axis=2)
    else: m_pics = np.ones((pic_size_x, pic_size_y))

    return m_pics

#crate master dark, flat
m_dark = master(dark_n)
m_flat = master(flat_n)

#-----correction of data
for data_i in data_n:
    dataa = fits.open(data_i)
    dataa[0].data =
        (dataa[0].data - m_dark)*np.average(m_flat - m_dark)/(m_flat - m_dark)

    #Remove Hot Pixels
    ccddata = ccdproc.CCDData(dataa[0].data, unit='adu')
    dataa[0].data = ccdproc.median_filter(ccddata, 2).data

    #Save files to reduced folder
    outp = 'Reduced/' + str(data_i[5:-4]) + '_reduced.FIT'
    dataa.writeto(outp)
    dataa.close()

print("Start Part 2")
#-----Finde flux of all stars

#Open images
frames = list()
redu_n = glob.glob('Reduced/*.FIT')
redu_n = natsorted(redu_n)
for redu_i in redu_n:
    hdu = fits.open(redu_i)
    m_redu = hdu[0].data
    m_redu = np.array(m_redu, dtype = 'uint16')
    frames.append(m_redu)
    hdu.close()

#get values
number_of_frames = len(frames)
pic_size_x = frames[0].shape[0]
pic_size_y = frames[0].shape[1]

```

```

#find trajectory
f = tp.batch(frames, 7, minmass = min_m) #Locate all stars in all images
t = tp.link_df(f, 10, memory = 0) #find trajectorye
t1 = tp.filter_stubs(t, number_of_frames) #Eliminate partly vanished stars
t1_matrix = t1.as_matrix() #convert to matrix

#find number of stars
number_of_stars = (int)(len(t1_matrix[:,0])/number_of_frames)

#Information about the progress
plt.figure()
tp.plot_traj(t1) #show a plot of the Stars found
print("")
print("You found %i peaks with a minimum mass of %i." %(number_of_stars,min_m))

#-----Flux calculation
#Frames/Stars/(Flux, x, y, particle numbr, signal)
dataa = np.ones((number_of_frames, number_of_stars, 5))
j=0
for i in range(number_of_stars):
    for k in range(number_of_frames):

        # Define circular aperture
        apertures = pu.CircularAperture((int(t1_matrix[j, 0]),
            int(t1_matrix[j, 1])), mask_radius)
        # Define circular sky annulus for background subtraction
        annulus_apertures = pu.CircularAnnulus((int(t1_matrix[j, 0]),
            int(t1_matrix[j,1])),r_in=mask_radius_out1, r_out=mask_radius_out2)
        # Determine raw flux inside the aperture
        rawflux_table = pu.aperture_photometry(frames[k], apertures)
        # Determine flux inside the annulus
        bkgflux_table = pu.aperture_photometry(frames[k], annulus_apertures)

        #Actual Flux calculation, raw subtraction
        phot_table =
            hstack([rawflux_table, bkgflux_table], table_names=['raw', 'bkg'])
        # Determine average background flux per pixel inside the annulus
        bkg_mean = phot_table['aperture_sum_bkg'] / annulus_apertures.area()
        # Multiply with aperture are and subtract it
        bkg_sum = bkg_mean * apertures.area()
        flux = phot_table['aperture_sum_raw'] - bkg_sum

        #fill data
        dataa[k,i,0] = flux
        dataa[k,i,1] = t1_matrix[j, 0]
        dataa[k,i,2] = t1_matrix[j, 1]
        dataa[k,i,3] = t1_matrix[j, 9]
        dataa[k,i,4] = t1_matrix[j, 5]
        j+=1

#Save Dataa
results_file = open("results.txt", "w")
for k in range(number_of_stars):
    for l in range(number_of_frames):
        results_file.write(str(dataa[l,k,0]) + "," + str(dataa[l,k,1]) + ","
            + str(dataa[l,k,2]) + "," + str(dataa[l,k,3]) + ","

```

```

        + str(dataaa[l,k,4]) + "_"
    results_file.write("\n")

results_file.close()
##### #if program has already the results in text files, the program starts here

#Loading Dataa, gather information
data_file = open("results.txt","r")
count1 = 0
count2 = 0
for line in data_file:
    line = line[:-2]
    data = line.split("_")
    count1 = len(data)
    count2 +=1

dataa = np.zeros((count1, count2, 5))
data_file.close

#Loading Dataa, Load data
data_file = open("results.txt","r")
stars_r = 0
frames_r = 0
for line in data_file:
    line = line[:-2]
    data = line.split("_")
    for row in data:
        data_s = row.split(",")
        dataa[frames_r,stars_r,0] = float(data_s[0])
        dataa[frames_r,stars_r,1] = float(data_s[1])
        dataa[frames_r,stars_r,2] = float(data_s[2])
        dataa[frames_r,stars_r,3] = float(data_s[3])
        dataa[frames_r,stars_r,4] = float(data_s[4])
        frames_r += 1
    frames_r = 0
    stars_r += 1
data_file.close()

print("part 3 started")
#-----Transit finder

#Delete old results
end_n = glob.glob('Result/*.png')
for end_i in end_n:
    os.remove(end_i)

#Precalculation
size = dataa.shape

#Automatic tranist finder
for star in range(size[1]):

    #creat 2 arrays and extrac the object info
    ref_data = np.zeros((size[0], size[1]-1, size[2]))
    obj_data = dataa[:, star, :]

```

```

j = 0
for i in range(size[1]-1):
    if(i == star): j += 1
    ref_data[:, i, :] = dataa[:, j, :]
    j += 1

#ref flux calculation
ref_flux = np.zeros((size[0]))
for j in range(size[0]):
    ref_flux[j] = np.nanmedian(ref_data[j, :, 0])

#calculate relativ intenisty
x = obj_data[:,0]/ref_flux
if scaling: x = x/np.nanmedian(x) #scaling

#Delete Strange Data points
for i in range(len(x)):
    if x[i] > 100 or x[i] < 0.01:
        x[i] = np.NaN

#plot preparations
ran = np.linspace(0,size[0]-1,size[0])*time_per_frame/60

#ploting
plt.figure()
plt.plot(ran, x, 'o')
plt.title("Star number %i at (%i x %i) [last position]"
          %(star, obj_data[-1,1], obj_data[-1,2]))
plt.xlabel("time (min)")
plt.ylabel("Relative intensity to all Stars")
plt.savefig("Result/Intensity_Star" + str(star) + ".png")

#Selected Tranist claculation
if len(select) != 0:
    select_data = dataa[:, select, :]
    for star in range(select.shape[0]):
        #Precalculation
        size = select_data.shape

        #creat 2 arrays and extrac the object info
        ref_data = np.zeros((size[0], size[1]-1, size[2]))
        obj_data = select_data[:, star, :]
        j = 0
        for i in range(size[1]-1):
            if(i == star): j += 1
            ref_data[:, i, :] = select_data[:, j, :]
            j += 1

        #ref flux calculation
        ref_flux = np.zeros((size[0]))
        for j in range(size[0]):
            ref_flux[j] = np.nanmedian(ref_data[j, :, 0])

```

```

#calculate relativ intenisty
x = obj_data[:,0]/ref_flux
if scaling: x = x/np.nanmedian(x) #scaling

#Delete Strange Data points
for i in range(len(x)):
    if x[i] > 100 or x[i] < 0.01:
        x[i] = np.NaN

#plot preparations
ran = np.linspace(0,size[0]-1,size[0])*time_per_frame/60

#ploting
plt.figure()
plt.plot(ran, x, 'o')
plt.title("Star number %i at (%i x %i) [last position]"
          %(star, obj_data[-1,1], obj_data[-1,2]))
plt.xlabel("time (min)")
plt.ylabel("Relative intensity to selected Stars")
plt.savefig("Result/Selected_Intensity_Star" + str(star) + ".png")

```

```

#""

```

Figure 51: Python code for combining the individual parts and plotting the predicted transit.

```
# -*- coding: utf-8 -*-
"""
Created on Sun Apr 22 13:57:20 2018

@author: Sven Kiefer
"""

#Bibliotheken
import numpy as np
import matplotlib.pyplot as plt

#Variables
nr_of_files = 1 #Number of files to load
diff_to_next = np.array([0]) #Number of missed pictures between files
selected_stars = np.array([[ ]]) #Selected Stars in each part
time_per_frame = 45 #time between pictures (in Seconds)
scaling = False #Set true if tranist should be scaled to 1

#-----Loading
#preparation
dataa_tot = np.zeros((0, len(selected_stars[0, :]), 5))

#Load all individual files and set them together
for i in range(nr_of_files):
    #read out data
    data_file = open("results_" + str(i+1) + ".txt", "r")

    #collect information
    count1 = 0
    count2 = 0
    for line in data_file:
        line = line[:-2]
        data = line.split("_")
        count1 = len(data)
        count2 += 1
    dataa = np.zeros((count1, count2, 5))
    data_file.close

    #Load files
    data_file = open("results_" + str(i+1) + ".txt", "r")
    stars_r = 0
    frames_r = 0
    for line in data_file:
        line = line[:-2]
        data = line.split("_")
        for row in data:
            data_s = row.split(",")
            dataa[frames_r, stars_r, 0] = float(data_s[0])
            dataa[frames_r, stars_r, 1] = float(data_s[1])
            dataa[frames_r, stars_r, 2] = float(data_s[2])
            dataa[frames_r, stars_r, 3] = float(data_s[3])
            dataa[frames_r, stars_r, 4] = float(data_s[4])
            frames_r += 1
```

```

        frames_r = 0
        stars_r += 1
    data_file.close()

#add files
    dataa_tot = np.append(dataa_tot, dataa[:,selected_stars[i,:],:], axis=0)
    dataa_tot = np.append(dataa_tot,
        np.zeros((diff_to_nest[i],len(selected_stars[i,:]),5)), axis=0)

#-----Transit finder
#Precalculation
size = dataa_tot.shape

#Automatic tranist finder
for star in range(size[1]):

    #creat 2 arrays and extrac the object info
    ref_data = np.zeros((size[0], size[1]-1, size[2]))
    obj_data = dataa_tot[:, star, :]
    j = 0
    for i in range(size[1]-1):
        if(i == star): j += 1
        ref_data[:, i, :] = dataa_tot[:, j, :]
        j += 1

    #ref flux calculation
    ref_flux = np.zeros((size[0]))
    for j in range(size[0]):
        ref_flux[j] = np.nanmedian(ref_data[j, :, 0])

    #calculate relativ intenisty
    x = obj_data[:,0]/ref_flux
    if scaling: x = x/np.nanmedian(x) #scaling

    for i in range(len(x)):
        if x[i] > 100 or x[i] < 0.01:
            x[i] = np.NaN

    #plot preparations
    ran = np.linspace(0,size[0]-1,size[0])*time_per_frame/60

    #plot that shit
    plt.figure()
    plt.plot(ran, x, 'o')
    plt.title("Star number %i at (%i x %i) [last position]"
        %(star, obj_data[-1,1], obj_data[-1,2]))
    plt.xlabel("time (min)")
    plt.ylabel("Relative intensity to selected Stars")
    plt.savefig("Result/Intensity_Star" + str(star) + ".png")

#"""

```