Multispectral Imaging at Astroplant

Matthias Kortleven
4290615

Industrial Internship
Imaging Physics
TU Delft

June, 2019
Auréle Adam
Sidney Niccolson

Contents

1 Introduction **1**
1.1 Company/Project Overview 1

2 Plan de Campagne **2**
2.1 MoSCoW analysis 2
2.1.1 Must Haves 2
2.1.2 Should Haves 2
2.1.3 Could Haves 2
2.1.4 Won't Haves 2
2.2 The current system 4
2.3 Implementation strategies 7
2.3.1 Camera 7
2.3.2 Lighting/Filters 7
2.3.3 Fulfilling requirements 8
2.3.4 Concrete Plan 9

3 Production **11**
3.1 Building the module 11
3.1.1 NDVI 11
3.1.2 Lighting considerations 11
3.1.3 Other aspects 12
3.2 Refining the module 15
3.2.1 Diffuse light source 17
3.3 Coding 23
3.3.1 Code Structure 23
3.3.2 Code workings 24

4 Results and Discussion **26**
4.1 Linearity Test 26
4.2 Stability Test 27
4.3 Intensity Falloff Test 27
4.4 Memory/CPU Utilization 28

5 Conclusion and Recommendations **29**
5.1 Learning Process 29

References **30**

1 Introduction

1.1 Company/Project Overview

AstroPlant is an educational citizen science project with the European Space Agency (ESA) to engage a new generation of space farmers, collect data and ideas for agriculture on Mars, develop open source research equipment, and create awareness of regenerative and closed-loop life support systems. The Astroplant kit is a semi-closed hydroponic growth system with various sensors measuring the environment. To gain additional insight and automate the plant growth and development characterisation process (phenotyping), we are looking to develop a multispectral camera to be integrated in Astroplant kits. The focus lies on developing a camera able to monitor a small set of variables incorporated in the general plant growth model being developed by ESA, with limited cost in mind. The current Astroplant kit in development has most of the necessary sensors to monitor the most basic variables. The company consists of mainly part timers that handle different aspects of the development of the kit. This internship will fall under the responsibility of the Tech department. Due to this set up, the internship leans heavily on the independence of the intern and the ability to solve problems themselves.

The project consists of prototyping a cheap camera module, attached to a Raspberry Pi, to gather information from the plant in different spectral ranges. Together with the led module developed by the company it should be possible to gather data from the far red spectrum to the ultraviolet range. Just in the visible spectrum the system should be able to for example estimate leaf surface area and other parameters flagged in the ESA model for plant growth. This project will research the feasibility of those cheap camera modules for this type of information gathering. If possible the system should be integrated with what is already there. The feasibiblity of the project is not known exactly beforehand, as some parameters might require more advanced cameras than the low-cost requirement can supply. The desired outcome however is some extra parameters for the system that is already in place. Computer Vision can give important extra information about plant growth and plant health, especially when information of different spectra of light is used.

This research was carried out at Astroplant, commissioned by the Delft University of Technology as part of the industrial internship required for the Master Applied Physics. Sincere gratitude goes to my supervisors, Auréle Adam and Sidney Niccolson and the rest of the people at Astroplant.

2 Plan de Campagne

As mentioned in the descrition of the project as listed by Astroplant, the main goal of the internship is to develop a camera system that monitors a set of parameters regarding plant growth with the intention to use this data to incorporate into the plant growth model developed by ESA. Partly because of the educational aspect of the Astroplant kit, this also has to be done keeping limited cost in mind. Also keeping in mind the limited time available for the internship it is important to initially set a set of goalposts to work towards. To facilitate this a MoSCoW analysis was done to determine a set of delivarables on which the specific implementation could be built.

2.1 MoSCoW analysis

The MoSCoW method is a well known prioritization technique that lists aspects of a project into four categories according to importance. Since it is important to argee on this beforehand, this analysis was done as one of the first things during the internship.

2.1.1 Must Haves

- System automatically returns information about plant health and growth over a period of time. A specific quantity that needs to be monitored is the stable NDVI (850 nm and 660 nm). For testing purposes it's fine that the average NDVI over the entire plant is calculated and returned as a number.
- System must be cheap (money is available for a camera module and some simple other components, but the entire camera system can't cost much more than around 50 euros.
- System must be integrable with the kit already in place, this means processing and calculations need to be done on the Raspberry Pi Zero W, and the camera needs to be compatible with that system.
- System must be able to provide a recent image of the plant inside the kit to allow for visual inspection by the user.
- CPU and memory utilization of the system must be tracked for the various subroutines, so that an accurate estimate can be made of the performance on different Raspberry Pi systems. Goal is insight in the feasibility to run on the Pi's themselves or whether something beefier is necessary.

2.1.2 Should Haves

- The system should be capable of obtaining full NDVI pictures, and at least save them to disk.
- The system should return some information about plant parameters such as leaf area, width and height.
- The current kit has nice monitoring software for all the regular sensors that are already in place. If possible, the outcomes of the camera system should be integrated with this system as well.

2.1.3 Could Haves

- It would be nice to store several pictures of the plant from the past, so that a time lapse of the plant growth can be shown to the user if prompted.

2.1.4 Won't Haves

- There seems to have been some research in the form of fluorescence imaging. The sensor we use in this project however, is probably not sensitive enough to pick up these signals, and will probably suffer too much from light leakage inside the kit.

- There is some research around showing promising results for full-fledged spectrometry with smartphone sensor technology [12], however, this requires some advanced optical components and simulation from a mechanical point of view. This can not realistically be accomplished within a few months of internship.

2.2 The current system

In order to make a fitting plan for the approach of the design of the camera system, it is important to know what is already there. The current kit (V5) consists of a wooden case of dimensions 116 x 42 x 42 cm, with inner dimensions 61 x 41 x 41 cm. The inside of the case is coated with a reflective surface, to effectively use as much of the light from the light source as possible. The plant is placed at the bottom of the case and is fed nutrients by a hydroponics system present in the pot. An overview can be seen in figures 1 and 2.

Figure 1: Overview of the finished Astroplant kit V5.

At the top of the case a set of sensors is implemented:

- BH1750 light sensor
- BME280/DHT22 temperature/humidity sensor
- MH-Z19 CO2 sensor
- Water temperature sensor

The idea of the kit is that these sensors can be added modularly, so that the user can fit the kind and amount of sensors to their needs. This means that the exact types of sensors mentioned here is subject to change, but will be used for the test setup which is currently (February 2019) being built. In order to facilitate this modularity, an extension board for the Raspberry Pi was built and implemented in the current system. Some sensors use an I2C bus to communicate with the brain of the machine: the Raspberry Pi Zero W. The CO2 sensor uses the UART interface and the temperature/humidity sensor makes use of the 1-wire bus. Next to all the sensors, a LCD display is available on the kit to display some statistics or information about the current status of the kit.

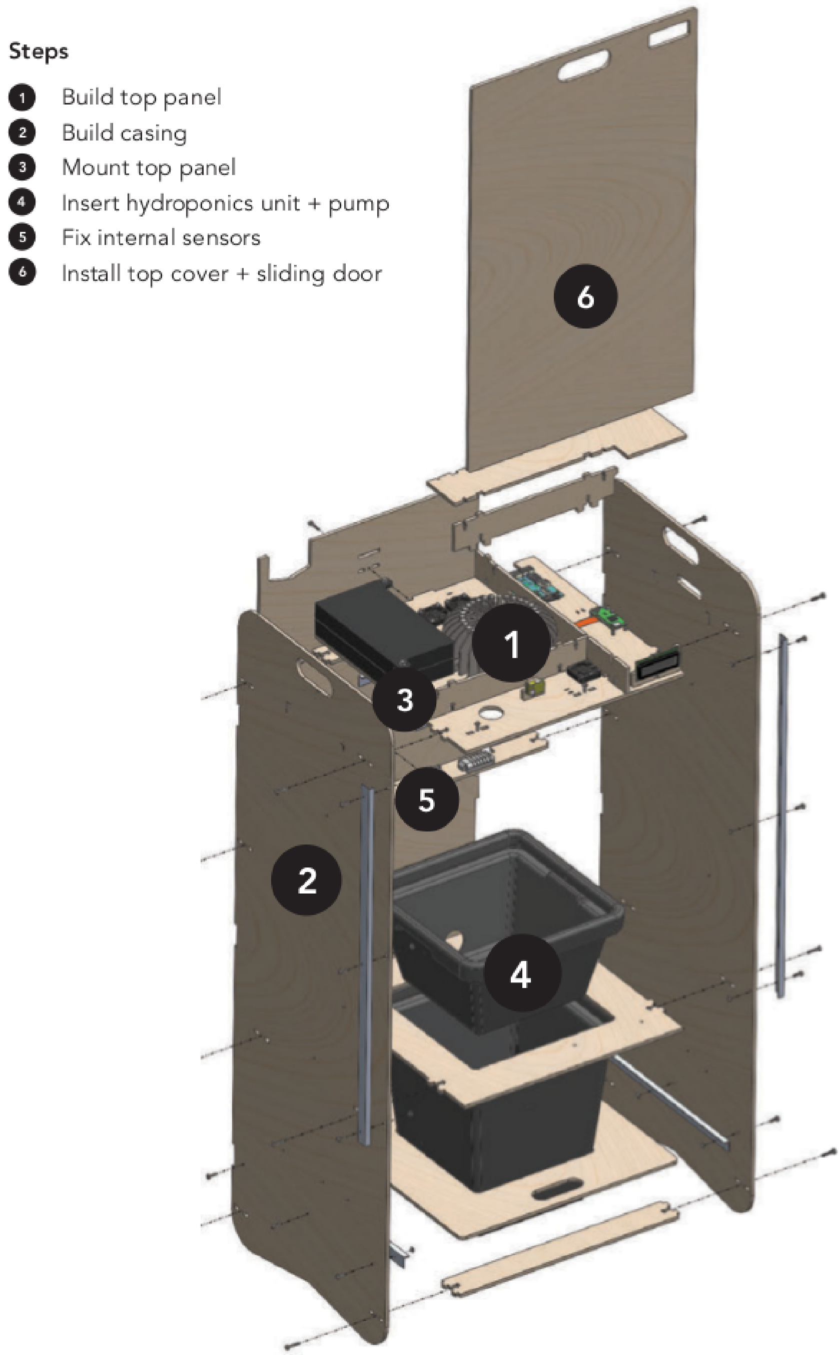


Figure 2: Overview of how the Astroplant kit is built up from the inside. This image comes from the manual for building the case.

The last — and possibly most important — part that is already available is the growth lighting. This part is panel with several LEDs that provides the lighting for the plant to grow. To be specific, the LED panel consists of:

- 4 blue LEDs of type GD CSSPM1.14 (rated at 451 nm, bandwidth around 20 nm)
- 8 red LEDs of type GH CSSPM1.24 (rated at 660 nm, bandwidth around 20 nm)
- 4 far-red LEDs of type GF CSSPM1.24 (rated at 730 nm, bandwidth around 30 nm)

Assuming that the kit will be closed and running on its own for a certain period of time, the camera will be somewhat dependent on the lighting provided by the LED panel in the case, if results can be obtained with this lighting, this further reduces costs required to incorporate the camera system in the kit. It is worth mentioning that the led panel is strictly the only part of the kit that will be constant over all kits supplied to customers. While most will probably choose to also order the wooden casing, hydroponics system and other sensors, some users might choose to build their own casing or use their own set of sensors. For this project however, we will largely assume that the setups of the systems will be similar.

2.3 Implementation strategies

2.3.1 Camera

From this point onward several strategies can be chosen. Since the work is performed in a controlled environment, it would make sense to first look at camera's that are as sensitive as possible. The main focus of the camera system is multispectral imaging, meaning that a camera that has as wide a spectral response as possible would be desirable. In the ideal case a camera could be chosen without a Bayer filter (the RGB pixel overlay), since the paint in this filter blocks plenty of light in useful wavelengths. These monochrome camera's however start at around 150 euro, going up as high as 400 euro for decent models. On top of that, all these systems are USB connected, which means they cannot benefit from the GPU image processing pipeline of the Raspberry Pi. Since the system of sensors already in place is built around the Raspberry Pi Zero W, it therefore makes sense to then consider the Raspberry Pi Camera Module V2. Since this camera can also be delivered without an IR cut-off filter in place, this seems to be a good candidate to start with for multispectral imaging. On top of this, the module is relatively cheap, and doesn't require some form of hardware hacking to remove the IR cut filter that is placed in most modern digital camera's. Furthermore, an extensive research paper was written on the use of this camera in scientific environments, including parts about linearity with ISO values and exposure times. [11] This camera supports 10 bit raw recording of images, further increasing its value for recording in a repeatable way every time. Especially because the measurements will be seperated in time by several minutes or hours this increases the chance of reliable results. It should be noted that research was performed on the regular V2.1 camera module, which uses the exact same Sony IMX219 8 megapixel sensor, but with an IR cut filter in place to filter out the wavelengths above $\sim$700 nm. A spectral response curve for this camera module can be found online. [2] The spectral response shows (figure 3) that much leaves to be desired if we would want to build a really sensitive camera that registers all wavelengths of light with a similar sensitivity. It has been shown before [12] that it is possible to remove the Bayer filter layer (the layer of red, green and blue die that makes your image have color) off of the camera sensor of a Pi Camera, but specifics list that this is a risky and time consuming process that requires a microscope and a very steady hand, something likely not to be done by most users of the Astroplant kit. The focus on cheap, of-the-shelf products in this kit is one of its powers, so this seems a bit excessive to consider.

2.3.2 Lighting/Filters

There are two main approaches to create multi spectral images of plants with the sensor described above. One could either shine a white light (wide spectrum, similar to daylight, that is) at the plant and cover the lens with different filters, or one could adjust the lighting in such a way that only the relevant wavelengths are present in the source light to begin with. Both have their advantages and disadvantages, but the second option seems to fit the problem a lot better than the first. The main advantage of the filter option is that it works in regular daylight, meaning most prior research assumes that plants are exposed to daylight and most definitions are based on measurements performed in daylight. However, this is surmountable for the lighting option, as corrections can be made if one knows what kind of parameters the lights and the sensor have. The main disadvantage of the filter method is that filters in general are a lot more expensive and harder to come by than LEDs in different colors and sizes. Also, the kit itself does not have a wide spectrum light installed already (artificial light at specific wavelengths of blue, red and far-red), which would mean that a new light source has to be added, or the box has to be opened and exposed to sunlight to perform measurements. This defeats the purpose of the kit and hurts repeatability big time, and is therefore undesirable. Not to mention the need for moving parts within the kit to put different filters in front of the camera, which in itself is not that bad, but the fewer moving parts the better.

The filterless approach in general is preferred above the method with filters because it is mechanically simpler and can generally be controlled to a higher level. The main downside is that the lighting conditions need to be heavily controlled in order for this method to work. Luckily the Astroplant kit is build around the idea of a controlled environment and therefore is isolated from the outside world on many levels, including

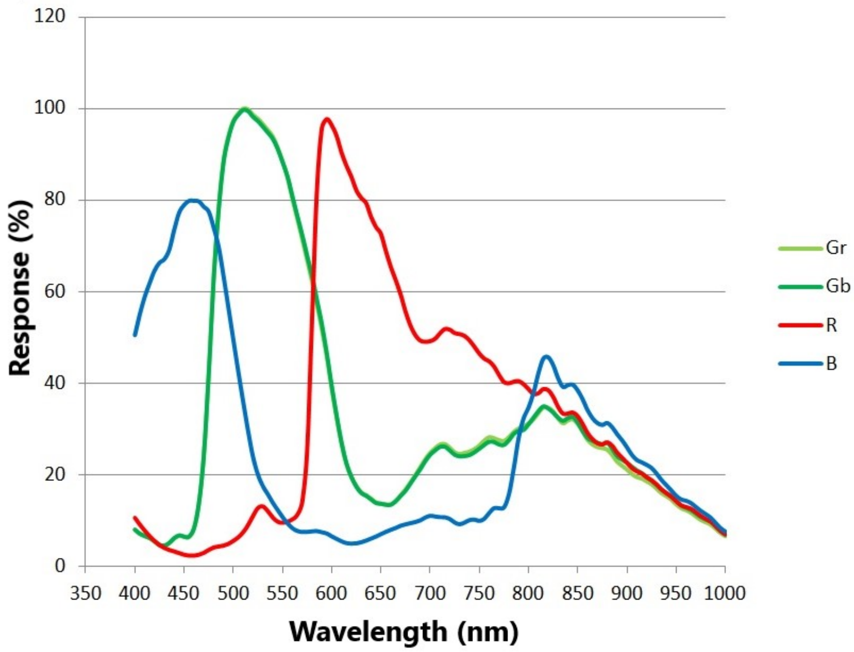


Figure 3: Spectral response of the Sony IMX219 sensor in the Pi Camera NoIR V2.1. Response is calculated relative to the maximum response of the green channel. Note that all channels respond in some way to NIR wavelengths, which means making regular pictures with this sensor will produce weird results in daylight. [2]

lighting, which automatically solves the biggest problem for this method. Since the camera sensor cannot record light that is not emitted from the source, simple RGB images or pure NIR images can be made without the need of a filter, with only the appropriate LEDs. Even white LEDs in general are fake news, since the cheaper models just produce a mixture of red, blue and some green light, as opposed to a nice (almost) even spectrum. This is only good in our case, since it means that we don't have to worry about the IR spectrum when taking a regular RGB picture and vice versa. The downside is that measurements of the spectra of the LEDs are necessary to obtain quantifiable results. Also, results of prior research might need a correction factor to apply to this research, since most of that research was done in broad daylight. On the other hand, results will probably be more consistent, since all the variables regarding the image can be controlled.

2.3.3 Fulfilling requirements

The first must have of the system is that it return information about plant health and growth over time. One important quantity here is the NDVI, which normally is measured using the red and infrared spectra reflected/absorbed by the plant. It has been shown that a useful measurement for this quantity can be obtained using both a regular and a NoIR Pi camera in the past. [3] Other research uses two similar camera's (one with the IR cut-off filter removed) to do something similar for flying over crops in the field. [8] Countless other research in this field can be found, all concluding that NDVI is a quantity that represents current plant health relatively well and is not too hard to measure. [13] The general idea here will be acquiring a few LEDs in the red and the infrared spectrum. Cheap ones seem to be available at around 630 nm and 850 nm, both well within what the camera can actually pick up. Field tests will have to show whether simple small LEDs are enough, or if larger 1-10 W LEDs are needed to provide sufficient light for the camera to pick up. Obviously the 1-10 W LEDs will be more expensive (around a dollar a piece at Aliexpress) and will require more advanced control circuitry, but the total cost of a camera module that delivers on the must haves should not cost more than 10-15 euros when build in somewhat more bulk amounts. If small

LEDs combined with long exposure times prove to be sufficient, costs could go down as low as a few euros. Since the grow lights will be off during imaging (to provide repeatable results), the 24 V rail can also be used to power these somewhat higher power LEDs. Both LEDs should be measured with a spectrometer to determine their power output and spectral properties, since data sheets for cheaper LEDs tend to be unavailable or unreliable, or — unsurprisingly — both. NDVI is based on the proportions of red light and NIR light present in sunlight, and directly compares the reflection of both wavelengths to each other. This means that for accurate measurements, the sensitivity of the camera needs to known, the amount of light the LEDs emit and the proportion of red and NIR light in regular sunlight. This will require some measuring, but should certainly be possible.

In order to make reliable measurements regarding plant growth with PlantCV (as for example in this experiment [1]) we can use both information from the visible spectrum and the NIR spectrum. Since PlantCV [5] is based OpenCV [4] functions for sectioning off certain parts of the image are built in and can be stacked to obtain the right cutouts of the plant. Since the environment is controlled, and the camera settings can be too it should be possible to tweak the parameters in such a way that a certain approach will work for multiple plants. It is to be determined whether this will work in practice, but since a wide variety of light sources should be available and tweakable, there are a lot of parameters that will make imaging easier. For example, using only a green light source or a NIR light source should produce a nice contrast between plants and the background that would be harder to achieve with natural lighting.

Moving on to the integration with the monitoring system, this will require coordination with the developer of that system. Other than that it should be possible, not much can be said about it at this point in time.

A more advanced option to look at involves some form of fluorescence imaging. Previous research has shown that usefull results can be obtained by lighting up the plant with a saturating pulse of red light and measuring the response when the plant was in a dark adapted state before. [10] Fluorescence measurements however generally require very sensitive equipment, since the signal emitted by live plants is very weak. The environment is ideal for these kinds of measurements (dark, controlled lighting, right wavelength of saturating pulse already available), but the camera will most likely be the lacking sensitivity to pick up the resulting signal. While some adjustments can be made (long exposure times, raw recording), this limit will probably make this idea impossible to do cheaply. Another research looked into the option of capturing fluorescence measurements with a UV excitation source that would produce certain wavelengths of fluorescence from chlorophyll in the plants. [7] This could in theory be done, since the camera is not sensitive to UV light, meaning with constant illumination a long exposure time could be set for the camera to read. The equipment used in the research is much more advanced though, so again sensor sensitivity or noise might be a problem.

The time lapse of the plant in the could-have section should be easy enough to implement. Set the system to take pictures at set times on the day (when the growth lighting is off, maybe at the end of a light cycle) and save them somewhere on the device. In order to test this, I plan on doing a small field test with a type of plant (maybe a herb) that sprouts and grows fairly fast to get results quickly. This test can also nicely be integrated with testing the leaf area accuracy (so a plant that mainly grows sideways instead of up) and keeping track of the NDVI (possibly not watering it for some time at some point to see if the stress this induces can be measured remotely by looking at the pictures).

2.3.4 Concrete Plan

To fulfill the must-haves described above, the following list of items is needed:

- Raspberry Pi Camera Module V2.1 (NoIR edition)
- White light source ($\sim$ 400 - 700 nm) to make regular RGB pictures without the influence of NIR light

(either white LED or RGB LED)

- Near-infrared light source (850 nm) to make pictures for the NDVI measure
- Red light source (630 nm) to make the other pictures for the NDVI measurements
- Green light source to test whether contrast is higher than with white light (to make PlantCV measurements easier)
- Supporting elements to make the LEDs work with the Pi (resistors, LED drivers, MOSFETs?)
- Possibly some random other light sources (UV, 940 nm infrared, blue LED) just to see what happens to the picture of the plant under those

3 Production

3.1 Building the module

After some testing of small LEDs it can be concluded that there is enough light to move on with this strategy. With very limited light and exposure times of around a fifth of a second the image is still decent. The shorter this time, the better, because plants tend to move around somewhat, especially in a kit that has active air circulation, such as the Astroplant kit. Sensitivity in the NIR range is also perfectly fine and delivers well on image quality. To measure the NDVI three red LEDs (630 nm) and three NIR LEDs (850 nm) will be used initially. A test setup was made containing these LEDs, as well as two narrow angle white LEDs, two wide angle white LEDs for the regular photos, and a single amber, green and blue LED, just for further testing. All LEDs are high quality Cree LEDs producing considerably more light than the cheap LEDs used initially. The LEDs are all driven by the 5V rail of the Pi via a ULN2003 transitor array, resulting in an extra volt of play room when matching the LEDs with a resistor. All LEDs were matched with a resistor such that the current through the LEDs is around 15 mA, with the exception of the wide angle white LEDs, which were matched at 20 mA, since they have a higher continuous current rating.

Further design of the module and the boundary conditions within the kit requires some more looking into the desired behavior of the module and what we are actually trying to measure.

3.1.1 NDVI

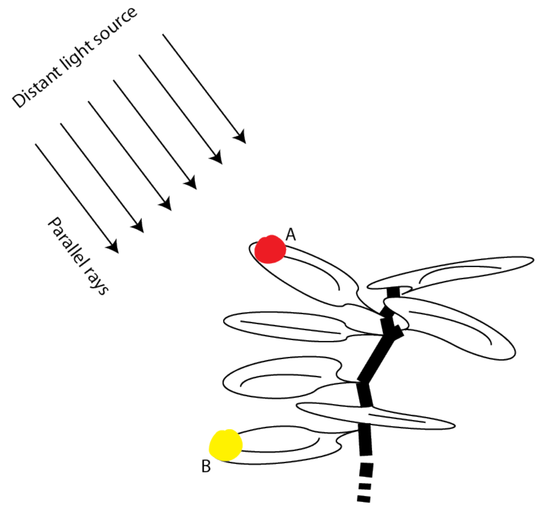
The NDVI (Normalized Difference Vegetation Index) gives a very rudimentary value for the amount of live vegetation (and the quality of the vegetation up to some point) contained in a certain portion of an image. It is defined as follows:

$$NDVI = \frac{R_{NIR} - R_{Red}}{R_{NIR} + R_{Red}} \tag{1}$$

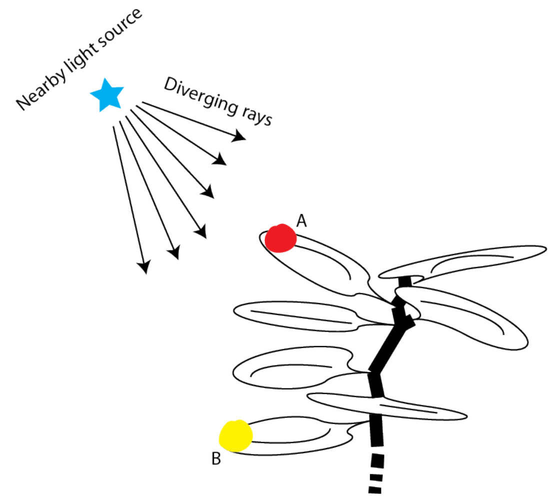
Where R_X is the reflectance (the ratio between incoming and reflected light) of a certain spectral band X. Values lie between -1 and 1 and typical values for vegetation are between 0.3 and 0.8. Other mass in the photos will tend to have a lower NDVI. Because it is built up out of reflection ration's, this immediately raises an important point for the lighting used to image the plant.

3.1.2 Lighting considerations

Since the module is aimed at used within a closed environment, there are several advantages, but also several disadvantages. An important difference between regular NDVI measurements using the sun and using artificial lighting is the distance to the light source and the scale at which the measurement is being performed. An illustration of this is provided in figure 4. Where with regular measurements in sunlight the light level does not differ too much from the top side of the plant to the lower leaves (except for shadows et cetera), within a closed environment, especially one that is relatively small such as the Astroplant kit the light level will vary significantly from the leaves closer to the light source to the leaves further down. This is a problem, because we want to calculate the reflection of certain spectral bands of light, meaning we should also be able to derive or determine the light received at a certain point reliably. This produces a problem, because we are able to calculate the amount of light received at a certain point, but we do need to know where the point we want to measure is in space. For small plants that mainly grow sideways one can probably get away with assuming that the plant is as high as the bottom plate, of which an intensity map can be produced when calibrating the camera. For all other instances however, some kind of depth map of the plant is going to be needed to accurately predict where the leaves are in space, thus correcting for the increased amount of light they capture. This is not so easy, since there are limited cheap ways in which such a depth map can be obtained. One option is using stereoscopic view (using two camera's) to create a dispersion map and then filter out an actual depth map. This increases the cost of the camera system tremendously however, since some kind of extension board is needed to support two camera's on one Raspberry Pi. An alternative is



(a) *Graphic illustration of a parallel collimated beam (sun rays) hitting a plant. Point A and B will receive approximately the same amount of light, provided that B is not in the shadow.*



(b) *Graphic illustration of artificial lighting hitting a plant. Point A and B will receive different amounts of light, purely because B is further away from the light source than A.*

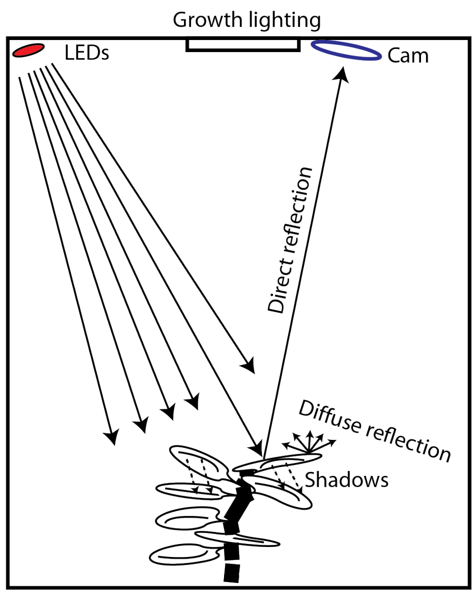
Figure 4: Illustrations of the differences between distant and nearby light sources.

moving around a single sensor and taking two pictures seperated in time, but this introduces moving parts and is hard to realize in a reliable and cheap way. A third option is to try and distill a pseudo depth map from a 2D image, which has been shown to work reasonably ok in prior research. [6]

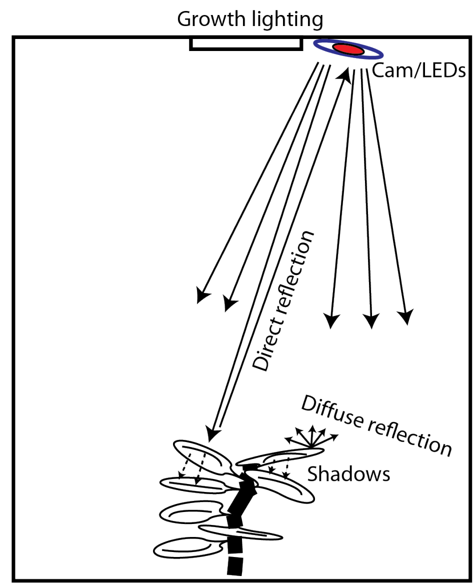
Another point of interest is the placement of the lighting with respect to the camera inside the kit. There are basically two choices, around the camera, or in a corner of the kit, illustrated in figure 5. When placing the lights in a corner of the kit, direct reflections will generally be less of a problem due to the angle the light has to make to reach the camera. This combined with the relatively diffuse nature of random leaves makes that direct reflections will generally not be messing up the picture. In this case there are, however, problems with shadows in the image. Since the light is relatively close and the plant is only lit by direct light (in the ideal case), hard shadows will be present in the image that will be hard to correct for. In the other case, where a LED array is placed around or in the neighborhood of the camera, direct reflections tend to be more of a problem. Shadows disappear however, since we are basically shooting light "out of the camera". Another advantage in this case is that calculation of the amount of light at a certain point in space is relatively easy. Assuming we are in the far-field of the LEDs the intensity of the light will distribute along the lines going out from the camera. This means that corrections are only necessary taking into account the distance from the camera to the plant. This does require a depth map, which — as mentioned before — poses a problem, but some kind of depth map would be necessary either way. Direct reflections generally present themselves as white spots in the image, which can be detected using their unique low saturation and high value in the HSV color space according to prior research. [9] When obtaining an average value for the NDVI of a plant these spots could just be left out of the calculation.

3.1.3 Other aspects

Some quick initial testing has furhtermore shown that blacking out the bottom plate is a good idea for contrast in the image. To this end a piece of black photo paper (diffuse cardboard) was cut around the hole in the bottom plate of the kit. To calibrate and account for the spots formed on the bottom plate by the LEDs a piece of diffuse white paper was also cut to size to provide a flat diffuse reflective surface for



(a) Illustration of the lights being at another position than the camera. This will result in relatively little direct reflections and good diffuse scattering, but shadowns will be a problem. Also calculating the incoming light a certain position will be harder, since we can't simply calculate backwards from the bottom plate.



(b) Illustration of the lights being at approximately the same position as the camera, still resulting in some direct reflections, but reducing shadows in the image and providing an easier way of calculating light intensity at different distances from the camera. Once the spot pattern is known, intensity can be corrected.

Figure 5: Illustration of the different options for light placement in the kit.

calibration. Furthermore we want to image the older leafs below as well, because the young leafs at the top of the plant will have a generically good NDVI value. This is somewhat hard, because the leafs tend to block view of the lower leafs. There is however not much that can be done about this except hoping that the plants will not grow directly towards the camera. Initial testing with a Mint plant indicated that this would generally not be the case, as most of the stems grew in different directions, providing clear view of the lower leafs.

3.2 Refining the module

Some preliminary testing with the test setup described in the previous chapter resulted in the images displayed in figures 7 and 6 for the angled light sources and the lights near the camera respectively. Some interesting things can be spotted in both cases. Both have their advantages and disadvantages, as described in the previous chapter. On the bright side, doing calculations purely relative provides a way of calculation without the need for a depth map. This in turn requires that flatfielding is performed in a decent way (either the radiation patterns of nir and red need to be highly similar or the field needs to be flat in advance), which poses a new problem. Initial results show that lighting near the camera is not a good idea due to the lack of contrast in the resulting image, which in itself is enough to discard this option for the continuation of this research. It cannot be guaranteed that the plant can be tilted away from the camera for every photo, that is not how plants work.

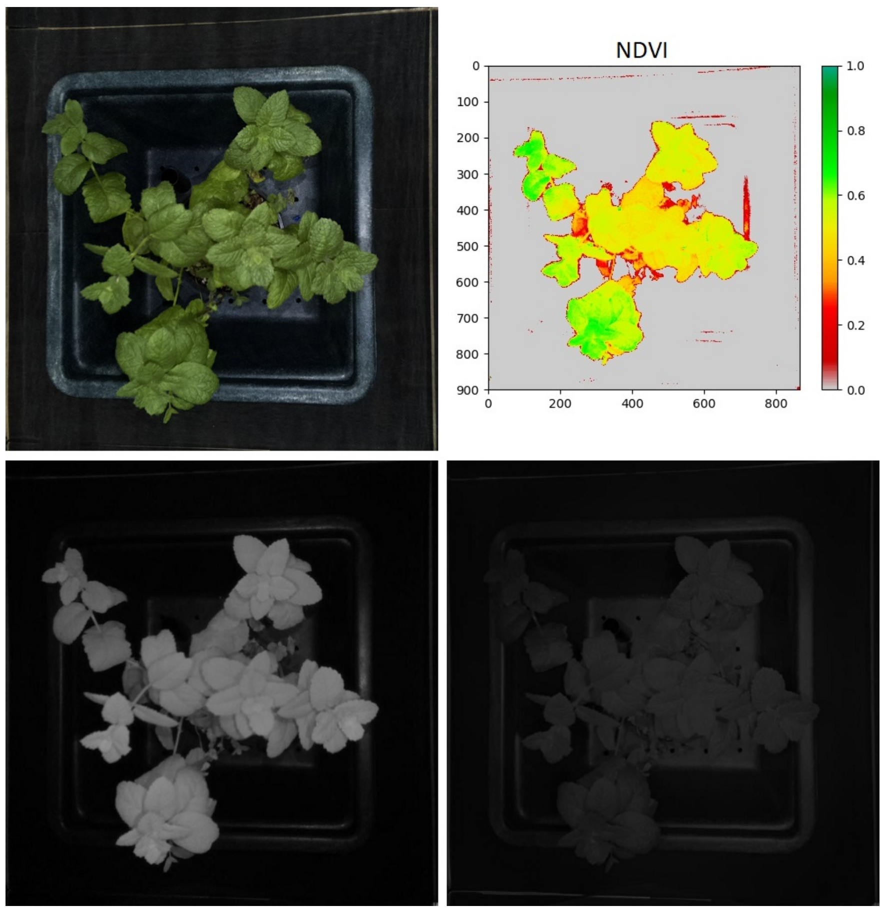


Figure 6: Representation of the first results using lighting near the camera, as depicted in figure 5b. (clockwise: white, NDVI, red, nir.) Two immediate problems can clearly be spotted directly, a lack of contrast in the NDVI image (new leafs at the top should have a higher NDVI than leafs lower down, which is not the case here) and direct reflections on the plant container. Actually these problems go hand in hand, since the lack of contrast is most likely caused by direct reflections on the plant. Portions of the plant that are angled away from the camera show better contrast in general. Flatfielding does work effectively in this case, since the percieved radiation patterns stay constant with height due to the placement of lights with respect to the camera.

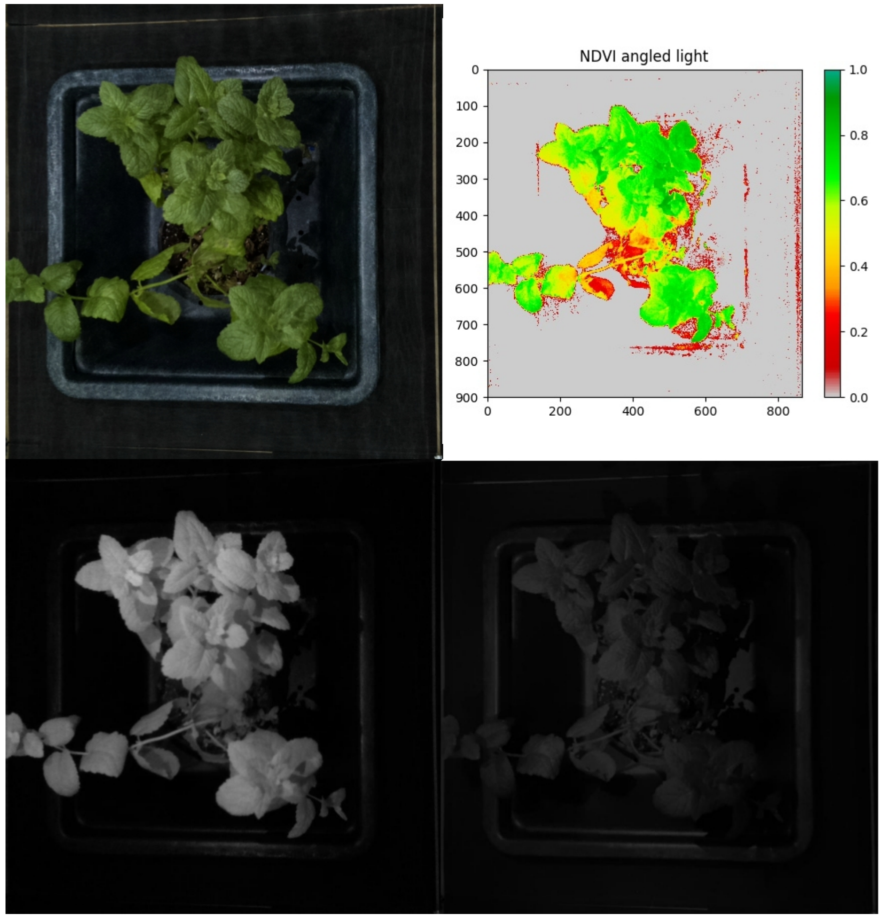


Figure 7: Representation of the first results using lighting at an angle, as depicted in figure 5a. (clockwise: white, NDVI, red, nir.) Results are fairly ok, contrast is better than in the case with the lights near the camera. Direct reflections are still a problem, as the plant container still shows in the NDVI picture at those locations. Problems are less severe though. Another problem that can be spotted is that flatfielding is not working as intended here. Since the light is coming from a different location than the camera, the height of the plant starts to cause discrepancies between the flatfield image of a white surface at the bottom and the actual distribution of the light in the image. This is a problem, because the red and nir light sources have different radiation patterns.

When the figures with the light sources under an angle are considered, contrast is better. A significant amount of green shows at the top of the plant, which should be the case for new leafs. Direct reflections are still a problem, as they show up in the final image on the plant container. Some highlights can also be spotted on the leafs of the plant, which will cause further problems when 'shinier' plants are used in some subsequent experiment. The main problem here is that shadows form a problem (the NDVI in the shadow areas seems excessively high). Flatfielding also only works to a limited extent, since the radiation pattern of the LEDs changes with height. The distance from the camera to the plant is similar to that of the lights to the plant, which causes this discrepancy. This would not be as much of a problem if the LEDs had similar light cones, but in practice this hard to realize. Also small angle differences between LEDs of different kind show in the flatfield images. This method requires a lot of calibration to work correctly, and for ideal measurements a depth map is still required. Unfortunately, a depth map is difficult to produce from a single camera. Adding another camera will increase the costs drastically, for relatively little gain.

3.2.1 Diffuse light source

Since all of the calculations are done relative to an original flatfield image, we can also look into taking the flatfielding out all together. If the radiation pattern of the lights is flat enough, this step can be skipped. Now this is hard, since diffuse light sources are not that easy to come by (most LEDs have a relatively small cone of light shooting outward) and it still will not fix the problem with direct reflections. Both can be fixed at the expense of total amount of light by introducing somewhat of a trick. If the light sources are pointed upwards towards a diffuse reflective surface (the same white paper used for flatfielding before) we can simulate a diffuse light source that has some spatial size, reducing direct reflections. If the field is flat enough for both the red and the nir channels, the flatfield image used before can be reduced to a flatfield number, to which the relative calculations can be performed. This means the process of calculating the NDVI will look as follows in formula form:

$$NDVI = \frac{\frac{I_{nir}}{I_{nir,ff}} - \frac{I_{red}}{I_{red,ff}}}{\frac{I_{nir}}{I_{nir,ff}} + \frac{I_{red}}{I_{red,ff}}} \quad , \tag{2}$$

where the $I_{xxx,ff}$ terms are simply floating point numbers captured from a reference image with a white diffuse surface on the bottom plate. The gains used by the camera are recorded as well, providing enough information to obtain relative values for the amount of light emitted at the location of a certain pixel in the image. Assuming that the camera works linearly (which will be shown in another section) the apparent intensity of light of pixel P of a certain channel is proportional to:

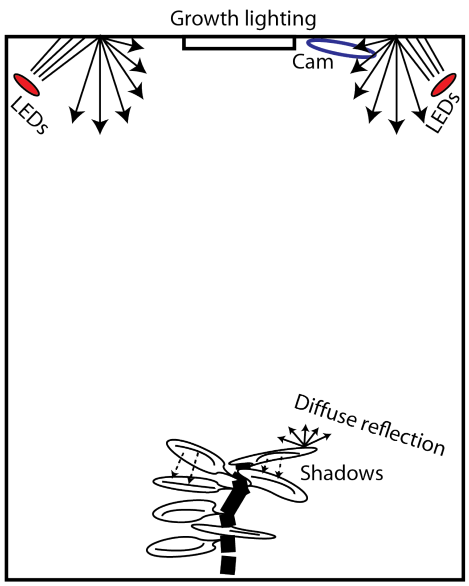
$$I_P \propto \frac{PV_P}{PV_{ff}} \frac{G_{ff}}{G_{photo}} \quad , \tag{3}$$

where PV is the pixel value of a certain pixel or part of the image, and G_x represents the gain of the camera in case x. In the case where diffuse lighting is used PV_{ff} can be taken the average of the flatfield image, which will be approximately constant. This approach will work when the pixel values remain far enough away from the clipping 0 and 255 values and if the flatfields in the infrared and red spectra are sufficiently similar in dropoff. The result is a setup as displayed in figure 8a. Note that because of the layout, reflections will start to play a larger role in the total amount of light reaching a certain part of the plant, which is shown in figure 8b.

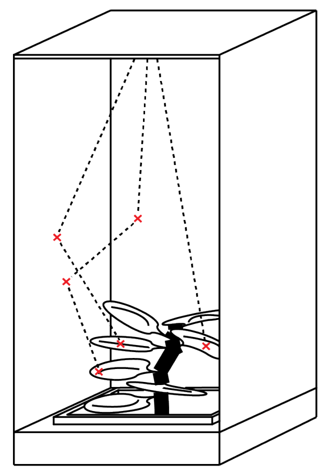
What can be observed when looking at images in the red and infrared spectra is that the reflection coefficient for the foil on the sides of the kit is different for different kinds of light. The reflection coefficient was determined by comparing the pixel values in the corners of the kit with their reflections, compensating for a black reference point in the image. This calculations resulted in a reflection coefficient for red light of 0.92 ± 0.02 and a reflection coefficient of 0.75 ± 0.02 for near infrared light. This discrepancy poses some problems for the intensity distribution of the light when plants grow taller. Since the second and third order reflections play a relatively important role in the total amount of light that reaches a target surface, near infrared light will increase in intensity faster when moving the target surface up than the red light intensity will. Luckily, we can simulate what happens to some extend to better understand this phenomenon. In order to do this we model the indirect light source as a point source that behaves according to the rules of Lambertian reflectance, i.e. we assume that the white surface at the top of the kit is a perfect matte reflective surface. This means that its intensity distribution will behave according to Lambert's cosine law. This law describes that the intensity of light that shines a certain direction depends on the angle α to the normal of the reflecting surface like this:

$$I(\alpha) = I_0 \cos(\alpha) \tag{4}$$

In our simulation, we can then assume one or more sources with a radiation pattern that behaves according to this law. This information can be used to gather data about the influence of different orders of reflection in the kit, as described above. To simulate this behavior, we can take advantage of the geometry of the Astroplant kit, which can be simplified to a square box. The reflections indicated in figure 8b can be simulated by laying out a field that consists of multiple copies of the ground plane next to each other. This grid



(a) *Illustration of the lights pointing upwards and reflecting off of a white diffuse surface, creating diffuse lighting. Light bouncing off of the reflective surfaces on the sides of the kit now plays a significant role in the total amount of light reaching a target surface.*



(b) *Illustration of the first orders of reflection when diffuse lighting is used. In previous cases the light reflecting off of the walls was relatively minimal, since the beam was pointed directly at the bottom surface, here however, the role of indirect light becomes more serious.*

Figure 8: Illustration of the indirect lighting method.

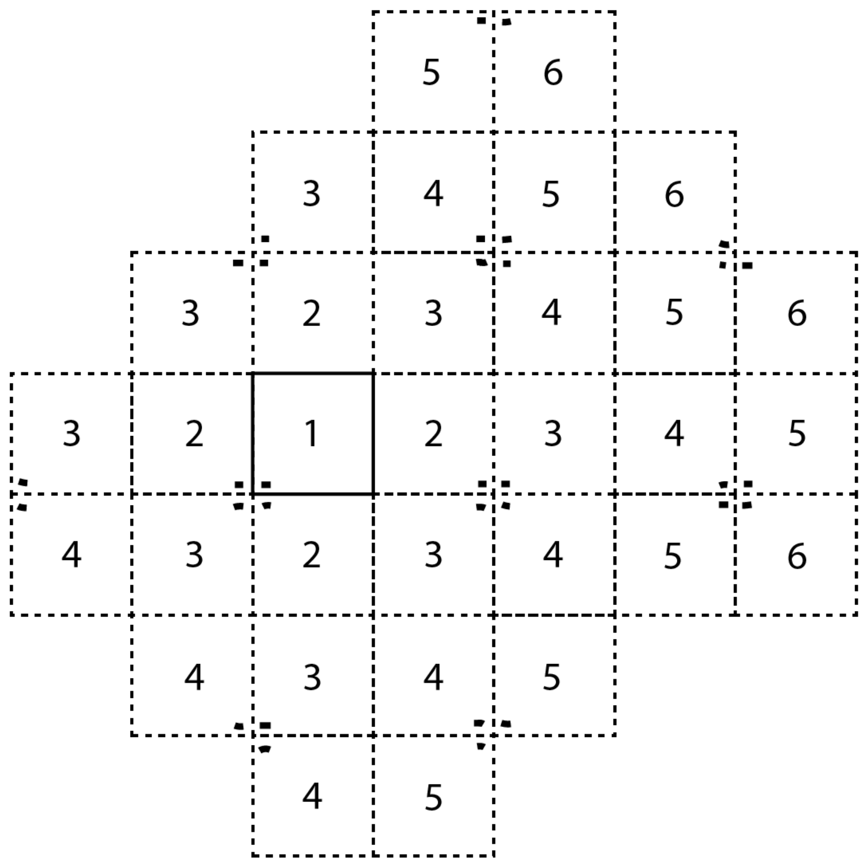


Figure 9: Overview of the simulation grid used to simulate light intensity captured from different orders of reflection in the Astroplant kit. The square with the 1 inside is the direct light received from the reflection off of the top panel. All neighboring squares are second order reflections (light rays that hit the reflective walls once), and so on. A small black square in the corner indicates in which direction a certain plane is mirrored when mapped back to the original ground plane.

is displayed in figure 9. Each neigboring square maps back to the original ground plane via the reflective surfaces. This creates a simple plane in which calculations can be done easily with the help of the formula above. Because the diffuse reflection of the light on the bottom plane also behaves according to the cosine law, we can multiply by this term again to obtain the light intensity captured by the camera.

To gain insight in the different orders of reflection, we can split them out into their separate contributions. These contributions can be seen in figure 10. Note that in this simulation four perfectly diffuse light sources were placed halfway between the middle and each of the four corners. Clear is that the second and third orders contribute significantly to the total intensity in the final picture. Also the radiation pattern changes significantly after the second order reflection. When summing all contributions we get the final image displayed in figure 11. As can be seen there, the result is a decently flat field, which strokes with the fields we see when taking actual pictures in the red spectrum. It is nearly flat, but falls off a little near the corners of the ground plane.

Taking the average intensity of the ground plane when moving up the target plane inside the kit, we can determine what happens to the ratio between near infrared and red light for different distances to the light

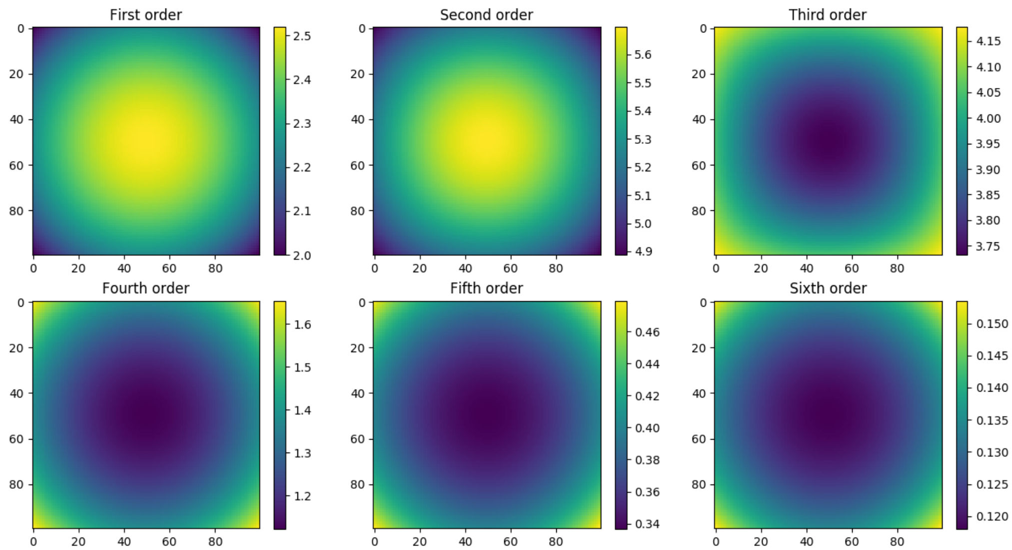


Figure 10: Result of the simulation of the intensity at the ground plane for different orders of reflection for a reflection coefficient of 0.92 and reference intensity $I_0 = 1$. Note that the second order reflections contribute significantly more to the final image than the first order reflections. Contributions only lower from the fifth order onwards.

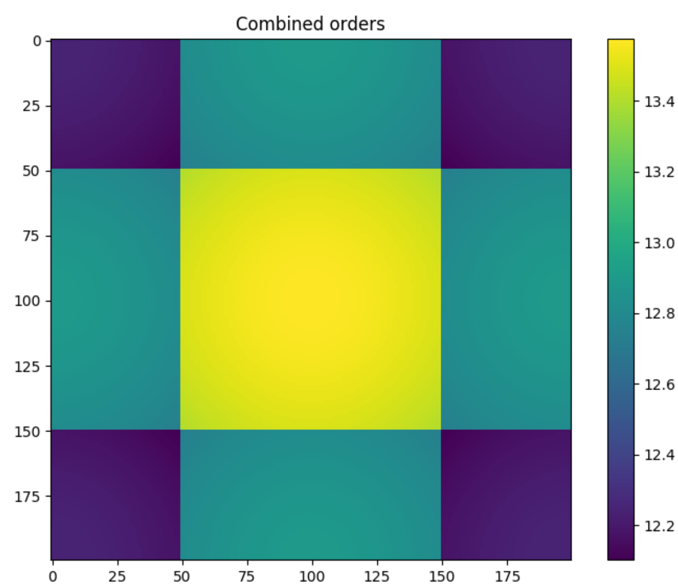


Figure 11: Total intensity mask when summing the results from figure 10. The field is nearly flat, with a falloff from 13.6 in the center to 13.3 at the sides. This is acceptable as flat enough to take the average of the ground plane as the flatfield value.

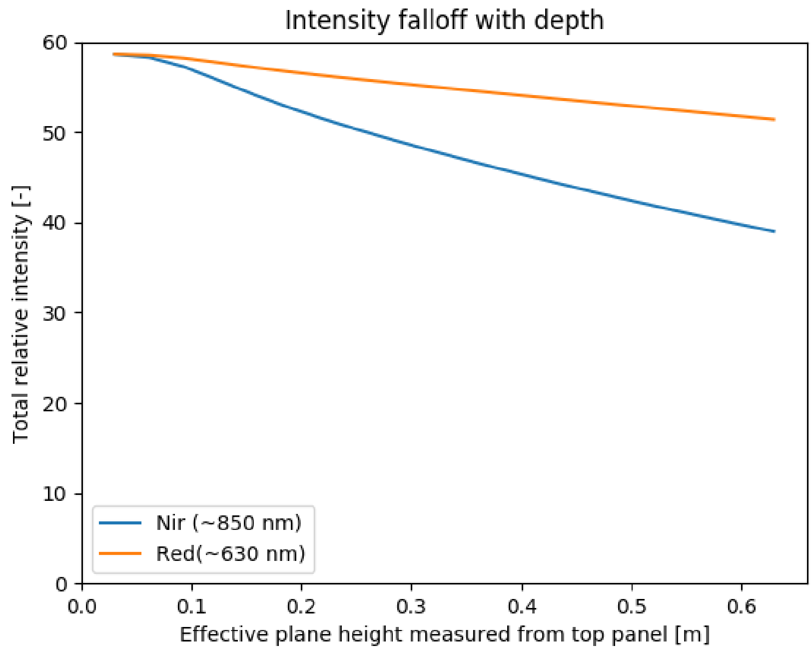


Figure 12: Intensity falloff with depth in the kit. Clear is that red does not gain much intensity when moving the target plane up to the lights, but near infrared does. This significantly impacts NDVI calculations, as the result almost differs by a factor two between the bottom of the kit and the top. This characteristic can be fixed by using foil that reflects both channels in a similar fashion.

sources. The result of this analysis can be seen in figure 12. Clear is that the NDVI will suffer from this if the kit is calibrated to the bottom plane. When moving up higher NDVI will increase as the near infrared channel will gain intensity faster than the red channel. This can be fixed by having a depth map of the plant (but as discussed before, this is hard to achieve, and impractical), by replacing the reflective surfaces on the sides of the kit with material that reflects both channels equally well (the exact reflection coefficient doesn't matter, as long as it is the same for both channels) or by estimating an error based on the average height of the plant. The second option is the best here, since it fixes the problems at the source, and does not try to patch problems later in the process.

When taking a photo with the indirect light approach, we can immediately spot the differences between this approach and the two direct approaches. A summary of the different images is displayed in figure 13. Results are definately better than the two previous tries, and the results above are reflected in a way that would be expected. The wooden bar in the right of the image lights up in the NDVI image, as would be expected from a relatively high-reflective object placed at that height in the kit. All unwanted reflections present in the previous images are gone here, which is good. Contrast is also better, but again, NDVI will be a little too high for the leafs that are higher up, which capture relatively more near infrared light than red light.

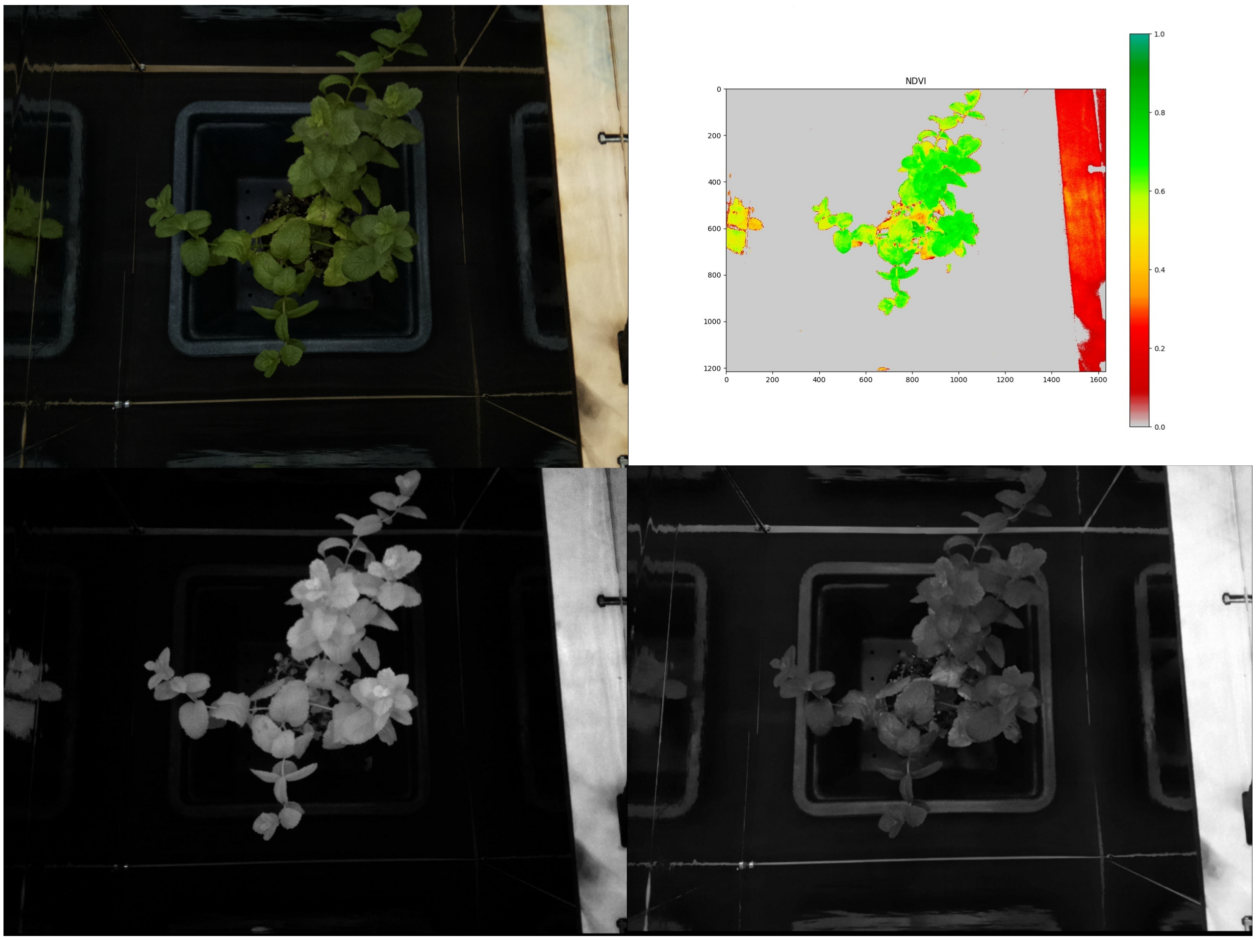


Figure 13: Representation of the first results using indirect lighting, as depicted in figure 8a. (clockwise: white, NDVI, red, nir.) Results are better than with the two direct approaches, giving better contrast between different leafs and getting rid of the reflections on the plant container. The wooden bar on the right is visible in the NDVI image due to the effect described above, where the reflective material on the sides reflects red light better than near infrared light. The bar is just above the halfway mark inside the kit, so it isn't strange that it becomes visible in the NDVI image.

3.3 Coding

The source code for this project can be found in a git repository at `https://github.com/Liveshort/astroplant-camera-module`. It contains the source code, a README and some results from testing. A few choices that were made deserve to be highlighted, as there are other approaches available, or the specific implementation required some creative problem solving. To keep things organized, the structure of the code will be highlighted first.

3.3.1 Code Structure

The code within the Astroplant camera module Python module is organized as follows:

```
+ astroplant-camera-module
|
+--+ cameras
|  |
|  +--+ pi_cam_noir_v21.py
|  +--+ pi_cam_v21.py
|
+--+ core
|  |
|  +--+ camera.py
|  +--+ ndvi.py
|
+--+ misc
|  |
|  +--+ debug_print.py
|  +--+ helper.py
|
+--+ __init__.py
+--+ setup.py
+--+ typedef.py
```

The `cameras` folder contains the different implementations for specific camera's. If users want to use their own camera, this is the place to add another file with implementations specific to their camera. During the project focus was laid on the Pi Camera modules available commercially for around 30 euros. An implementation is given for the regular camera, capable of shooting regular pictures with white light, or with the growth light. The NoIR camera implementation allows for NDVI photos as well, if the user installed appropriate light sources within the kit. Note that these different camera files contain child objects of the actual camera objects. Requirements to functions supplied to the module are given in the README.

The `core` folder contains the actual brains of the module. The `camera.py` file contains routines for making pictures in a single channel, calibration and the logic behind the command comprehension. The `ndvi.py` file contains routines to calculate the average NDVI, but also take NDVI pictures. Other quantities of interest acquired in similar fashion (two or three sequential photos in different bands) could be programmed in a similar manner, with minimal changes to the existing code.

The `misc` folder contains some helper functions and a print function that can be disabled at will for debug purposes.

The other files are related to the setup of the module and the indication that it is actually a Python module. The typedef.py file contains the allowed commands that a user can send to the module. This approach was chosen to minimize errors due to wrong spelling. Also, because these are separate objects, Python can perform a type check when functions get called that use a command.

3.3.2 Code workings

The first step in using this code is creating a camera object in the user code. Settings specific to each version of the kit are supplied with the camera objects and need to be supplied when creating a camera object. This will set up the work space for the camera (generate configuration folders and a place to save the images). If the work space is already set up the code will try to load previous configuration files so that calibration is not necessary every time the camera is called. For first time use a calibration cycle has to be run, in which the white balance is determined and the flatfield values are calibrated for NDVI measurements. The code assumes that the user has closed the kit and put a white diffuse surface on the bottom plate. This routine will take about 5 minutes to complete, and does not have to be repeated until significant changes are made to the dimensions or the insides of the kit. Notable here is that the white balance calibration uses the `picamera` Python module, and determines the right balance by gradually increasing or decreasing the red and blue gains until the image is white for the white, growth and nir channel.

Taking a picture loads up the camera and executes a `raspistill` command in a separate subprocess. This was done because the gains can be set manually using this command, whereas this is not possible with the current version of the `picamera` module. Workings are fairly simple, the code turns on the appropriate light using a `light_control` function supplied by the user, then proceeds to take a bright and a dark frame (where the light is off again) and performs dark frame subtraction to eliminate light leaking in form the outside. The image is saved to disk and a dictionary with a small report is returned to the user. This `dict` is standardized for all commands and is built up as follows:

```
{
  'photo_kind':
    [
      'raw NDVI',
      'processed NDVI'
    ],
  'encountered_error': False,
  'value_kind': ['NDVI'],
  'value_error': [0.0],
  'timestamp': '20190606-140728',
  'contains_value': True,
  'contains_photo': True,
  'value': [0.3598392680470938],
  'photo_path':
    [
      '.../astroplant-camera-module/tests/cam/img/ndvi1_20190606-140728.tif',
      '.../astroplant-camera-module/tests/cam/img/ndvi2_20190606-140728.jpg'
    ]
}
```

Because of this structure, the user can always check whether the operation executed successfully, and see whether a value or photo is present in the result. After a photo is taken, the code will check whether the current gain settings are still up to date. If the last update was more than a day ago, an `update` function will be called that will determine the correct gain settings. This will take around a minute per active channel. The reason this is necessary is that plants grow over time, and when they get taller, light gets significantly more intense. This requires the camera to be adjusted accordingly. An important note here is that brighter pixels in the photos for high analog and digital gains do not scale perfectly linear. To overcome this behavior, gains are limited between 0.67 and 1.5 times the flatfield gain for analog, and limited to maximum 1.7 for digital gain.

For the NDVI photos, the code gets a little more interesting. In essence it is not much more than taking two pictures in different color channels and laying them on top of each other, but in practice some difficulties arise. First a red and nir photo are taken and translated to a float numpy array. As the photos are recorded in 8 bit by the camera, the original matrix contains values from 0 to 255 in all color channels. Values are translated to reflectivity values by dividing by 255 and comparing their values to the flatfield values obtained during the calibration process. Values are corrected for differences in gain according to equation 3. Then the NDVI matrix can be calculated using the regular NDVI formula. Since we assume that nir and red intensities rise similarly when the plant grows, this formula is not corrected for different plant heights. We know that this is in fact not the case for the reflective foil, but since there is no depth map, appropriate corrections can not be made. It is left to user to interpret the data appropriately and realize that higher plant parts will have slightly lower NDVI than depicted in the image. Two images are delivered to the user, a raw `tiff` file (0 maps to -1, 255 to 1) that users can use to create their own NDVI maps, and a jpg containing a processed NDVI image with the Polariks color map and a scale on the side. `matplotlib` is used to create this image, which introduces a few programming difficulties. `matplotlib` is not built to be run in an endless loop, and for some reason produces a memory leak when used to simply create and save an image. This was overcome by placing the `matplotlib` routine in a separate subprocess that frees its memory when it gets destroyed after it is finished. Furthermore, `matplotlib`'s regular backend requires an active X server to be running. Since most of the Raspberry Pi's are running headless (without connected peripherals that is) another backend has to be used. This can be achieved by calling `matplotlib.use('Agg')` when the module is imported.

4 Results and Discussion

Tests were performed with LED panel as described in the previous chapters. A schematic reference of this setup is given in figure 14.

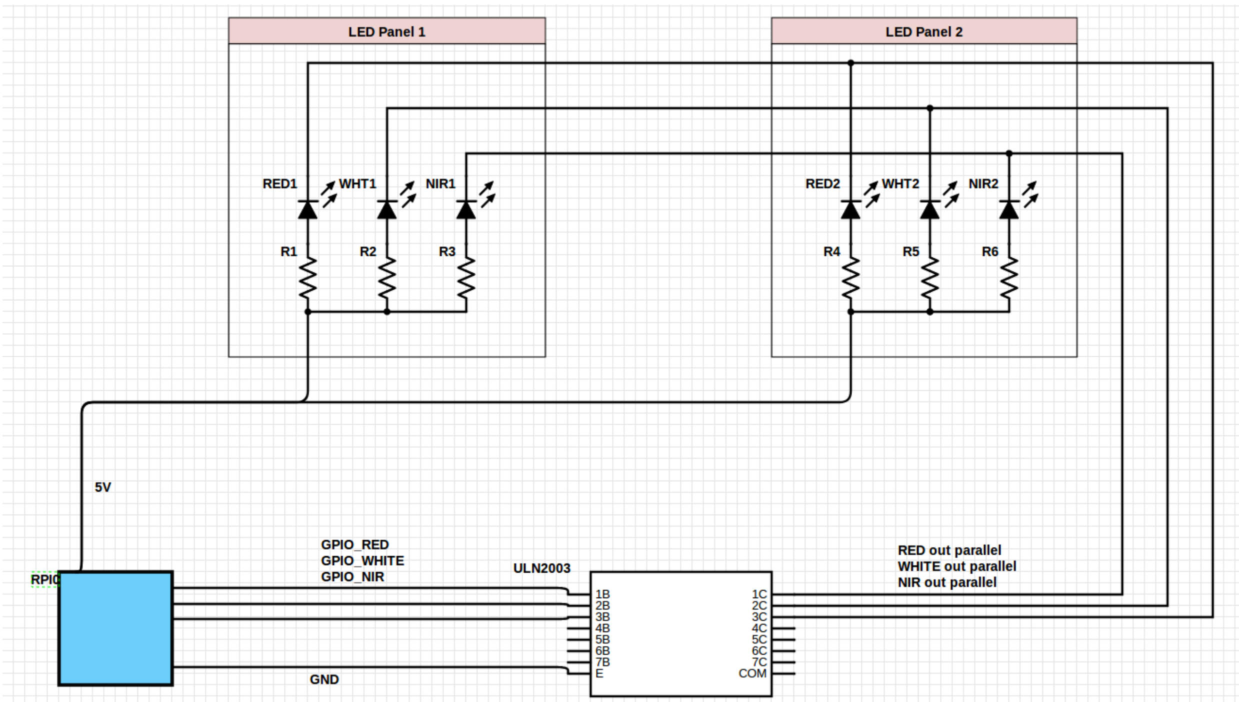


Figure 14: Schematic representation of the test setup used for the following tests. Two boards of three LEDs are present in the kit, mounted in such a way that they radiate upwards on a white diffuse surface. Light from this surface lights the scene with the plant. Resistors in this schematic are chosen to limit the current through the LEDs to 15 mA, and are (for red, nir and white respectively) 150, 200 and 100 Ohm's. A driver circuit is used in order to be able to use the full 5V rail of the Raspberry Pi.

4.1 Linearity Test

It was shown in previous research that the raw mode of the Pi Camera V2 behaves linearly. [11] To check if this behavior is also accurate for regular pictures taken with the camera, a small test was performed. Sheets of paper with different shades of gray were put at the bottom of the kit, and were lit by two white LEDs. White balance is set up such that the image is white (red, green and blue channels similar values for white parts). Then a set of different settings was run to determine linearity in exposure time, analog gain and digital gain. Default settings for both gains are 1, default setting for the exposure is 40 ms, corresponding to a frame rate of 25. Results are given in figures 15, 16 and 17.

It becomes clear from the graphs that the camera mainly struggles with brighter pixels with higher gains. Exposure scales almost linearly, until clipping starts to play a role. This is not the case for the analog and the digital gain, where values flatten out for higher gains. This is a problem that can actually occur when using the camera module. Mainly the red channel has a large deficit between gains during calibration and operation. Since plants have low red reflectance (when they're healthy at least), but the white calibration paper has high reflectance, the difference in gain is significant. The way this is solved in the code is limiting the maximum gain the camera can use. This will result in slightly darker pictures that lose some color resolution, but linearity is preserved.

4.2 Stability Test

During the last part of the internship, a two week stability test was run to check code operation during longer periods of time. To this end photos were taken in the white channel, as well as NDVI photos at regular two hour intervals. During the first three days problems arose surrounding the memory leak in the `matplotlib` module, which were subsequently fixed after which the test was resumed. Two identical plants were placed inside the kit, one of which received water, while the other did not. At the end, both plants did not receive water for a while. This test had to be run with an incomplete kit and natural light, which means that some of the photos taken during daytime are unusable due to light leakage from outside. Results still show a good time lapse of the plant health in the photos that are usable, which means the test in general was a success. Note that for this test the inside of the kit was padded with white paper, to also check whether the problem with different reflectance values for nir and red light could be solved in this way. Test was started on May 20th in the morning, with both plants healthy and watered. Some results along the way are shown in figures 18 through 23.

A few things can be taken away from this test. First off, the goal of making NDVI pictures in order to be able to say a little more about the plant health seems to be achieved. In both cases the NDVI pictures show a significant change before the regular photo with the white lighting does. This shows that plant stress can be detected earlier with this method than with regular camera checking. Secondly, results seem fairly consistent over time, there are no huge spikes in NDVI over time. Thirdly, even though white paper in principle should give better results, it does hurt the image quality. Using reflective foil makes the area surrounding the plant dark (because that is what is directly reflected from the bottom plate), which in turn makes the camera make better decisions with respect to gains. The images with reflective foil depicted in the previous chapter have considerably better contrast between the plant and the background. Because of this, I conclude it is better to have the error margin from the reflective material and the better photos, than the theoretically more accurate white paper. When it is calibrated to the middle of the kit the error should not exceed ~ 10 percent at both maxima (the bottom and top of the kit).

4.3 Intensity Falloff Test

As described in a previous chapter, intensity falls off faster for nir light than for red light, due to the difference in reflectance coefficient between red and nir light. To quantify this difference, a simulation has been described above that gives some insight in this phenomenon. To verify the correctness of the simulation, a test was performed with the test set up. In order to do this the white diffuse surface used for calibration was placed at various heights inside the kit, with intervals of 10 cm's. The average corrected values for nir and red intensity with respect to the calibration values were calculated both for the case with reflective foil on the walls of the kit and the case with diffuse white paper on the walls of the kit. Results are listed in figures 24, 25 and 26.

From these figures, it becomes clear that the simulation is fairly accurate for the actual real life situation. Nir light does becomes stronger faster than red light in the case of reflective foil, and the difference between white paper and reflective foil seems to match the simulation as well. A point of discussion for this test is the hard to determine error margin. The white paper was mounted as good as possible inside the kit, but differences of 0.5-1 cm could have occurred. This would not have had a dramatic influence on the results, but even so. The error of the relative intensity values is harder to determine, since it depends on the camera settings, lighting setup and light conditions during the test. Photo to photo differences were not that large under the same circumstances, but differences of a few percent could have arisen.

4.4 Memory/CPU Utilization

During the stability test, resources of the Raspberry Pi were monitored. The system reported 443,868 kB of memory present, where the rest of the 512 MB in the system were dedicated to the GPU. When idling, resource usage was minimal [0% CPU usage, only idle Python 3 interpreter usage (which is still 15-16%, but this will be the case in the Astroplant software too)]. When taking photos and performing calculations CPU usage rose to $\sim 95\%$, indicating efficient use of resources by `numpy`. Taking a white picture takes just under minute (actual photo is taken after about 20 seconds), while taking an NDVI picture takes up to 2 minutes. Most of this time is in the plotting with `matplotlib`, which takes up over a minute in this process. Memory usage is also highest during this operation. The `matplotlib` process takes up between 22% and 27% of the total memory available, next to the already running test process that takes up to 25% memory. Taking a regular picture in a single channel opens up a new process that uses only 10% to 15% of memory. Both numbers are not problematic, but might require some background stuff from the kit to be temporarily postponed in order to have enough memory. If this is impossible, it is also possible to cut out the `matplotlib` part and rely on the raw `tiff` file that can be converted to a nicer view on another pc (or the server, for that matter). The CPU should not be the problem, since it will just calculate as fast as it is allowed to. If another process is running in the background, it will take a little longer, but it should not stop. This is also demonstrated by the `pigpio` module, which can sometimes take up 5% to 10% of memory, while the plotting subroutine is running. Both will still finish, and there does not appear to be a problem. An entire cycle of updating the gains, taking a white photo and taking a NDVI photo takes about 4-5 minutes. This should be ok to do twice a day without the plant caring much.

5 Conclusion and Recommendations

Reflecting on the MoSCoW analysis at the beginning of this report, all must haves have been satisfied. As can be concluded from the tests described in the previous chapter, NDVI images and measurements can be made in such a way that significant differences are clear when the plant is stressed out (lack of water in this case). System is cheap, consisting of a camera that was already in place and a small LED board that can be produced somewhere in the future. Images are saved to disk at a location the user can specify. Integration with the existing system was made as easy as possible, but will be handled by the lead programmer of the kit. Since the code is written as a Python module, and the documentation is fairly clear, integration should not be too hard. The main difficulty here is that the back end does not yet support images being sent over. This is something I can't do much about at this time, but once that works, implementing the camera module code should work similar to other sensors. The could have has also been satisfied if the user chooses to keep images from the past. It is certainly possible with the framework that is now in place.

Regarding recommendations, the most important thing on the list of next steps is the development of a LED panel according to the specifications used in this project. A schematic with different options can be found in figure 27. The most ideal scenario would be that the LED boards can be connected through the I2C bus, as there are multiple free spots for a connection on the Astroplant extension board. This is certainly possible, but requires someone with knowledge of circuit design, so that the test set up can be professionalized. Once this is done, the LED board could be shipped to users that want to engage in NDVI measurements. For this, the code also needs to be implemented in the regular code for the Astroplant kit. As mentioned, this is doable, but requires some work.

As far as the kit goes, the ideal situation would be that the reflective material inside the kit could be replaced with material that has similar reflectance properties for red and nir light. However, most (if not all) of these foils are made out of aluminium, which inherently has these characteristics. Getting real silver or gold would help, but this will not be cheap, one of the key requirements of this kit. There might be some form of (spray) paint that works better, this could be looked into in the future.

Something that could be researched in the future is the addition of other quantities like NDVI. There are several small band quantities that are calculated in similar fashion. Implementation would be similar to NDVI, if the light sources are commercially available.

5.1 Learning Process

The nature of the internship required a lot of working essentially by myself, both because the company consists of nearly only part timers, but also because I was brought in as the 'expert' on the subject. There were regular meetings with Polariks, which gave me the chance to ask questions and spar a bit with the people over there, but most of the work had to be done independently. This required some creativity in solving problems and I believe I gained some insight in how to tackle larger problems by myself. Furthermore the end result relied upon programming skills and some electrical engineering as well. There is some experience in both subjects, but I certainly learned new things about Python's quirks and some basic circuit design and building. I've also gained experience in sieving through literature to find relevant research that could be used for my own research.

References

[1] Computer Vision/LED Plant Measurement System. `https://publiclab.org/notes/MaggPi/03-15-2018/computer-vision-led-plant-measurement-system`. Accessed: 2019-02-06.

[2] IMX219 spectral response curve. `https://github.com/hyperspy/hyperspy/issues/1775`, mail response from Sony. Accessed: 2019-02-07.

[3] NDVI pictures generated with RPi regular and NoIR camera. `https://publiclab.org/notes/khufkens/05-10-2015/multispectral-raspberry-pi-first-light-ndvi-images`. Accessed: 2019-02-06.

[4] Open Source Computer Vision Library. `https://opencv.org/`. Accessed: 2019-02-11.

[5] PlantCV. `https://plantcv.readthedocs.io/en/latest/`. Accessed: 2019-02-11.

[6] H. Dong, S. Yin, W. Xu, Z. Zhang, R. Shi, L. Liu, and S. Wei. An automatic depth map generation for 2d-to-3d conversion. *IEEE ISCE*, 2014.

[7] M. S. Kim, J. E. McMurtrey, C. L. Mulchi, C. S. T. Daughtry, E. W. Chappelle, and Y.-R. Chen. Steady-state multispectral fluorescence imaging system for plant leaves. *Appl. Opt.*, 40(1):157–166, Jan 2001.

[8] L. Mendonca, C. Chaves, E. De Lima, and L. E. Vicente. Low-Cost Multi-Spectral Camera Platform for In-Flight Near Real-Time Vegetation Index Computation and Delivery. 05 2017.

[9] A. Morgand and M. Tamaazousti. Generic and real-time detection of specular reflections in images. *IEEE*, 2014.

[10] E. Murchie and T. Lawson. Chlorophyll fluorescence analysis: a guide to good practice and understanding some new applications. *Journal of Experimental Botany*, 64(13):3983–3998, 08 2013.

[11] M. A. Pagnutti, R. E. Ryan, G. J. Cazenavette, M. J. Gold, R. Harlan, E. Leggett, and J. F. Pagnutti. Laying the foundation to use Raspberry Pi 3 V2 camera module imagery for scientific and engineering purposes. *Journal of Electronic Imaging*, 26:26 – 26 – 13, 2017.

[12] T. C. Wilkes, A. J. S. McGonigle, J. R. Willmott, T. D. Pering, and J. M. Cook. Low-cost 3D printed 1 nm resolution smartphone sensor-based spectrometer: instrument design and application in ultraviolet spectroscopy. *Opt. Lett.*, 42(21):4323–4326, Nov 2017.

[13] H. Xiang and L. Tian. Development of a low-cost agricultural remote sensing system based on an autonomous unmanned aerial vehicle (uav). *Biosystems Engineering*, 108(2):174 – 190, 2011.

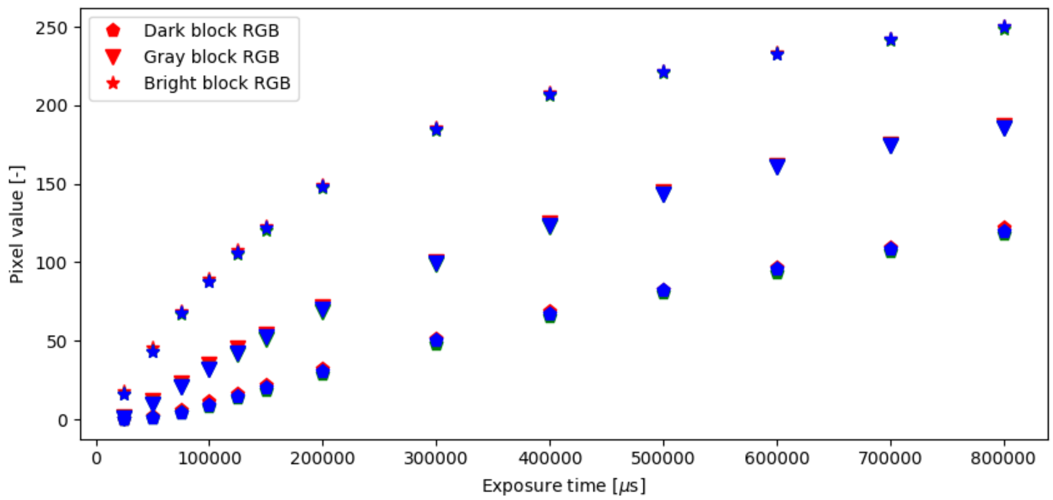


Figure 15: Results for the linearity test for the exposure of the Pi Camera V2. Behavior in the lower exposure times is linear, but for larger exposure times behavior becomes less linear. This is largely due to clipping of the pixels (the darker blocks perform better). Since exposure time is locked in the implementation, this non-linearity does not matter for the results obtained with our code.

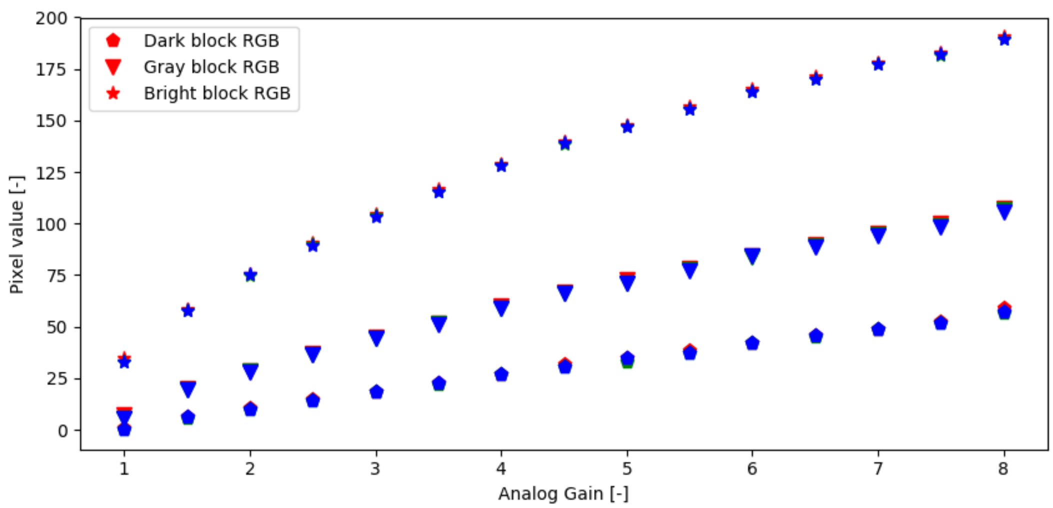


Figure 16: Results for the linearity test for the analog gain of the Pi Camera V2. Behavior in the lower gains is linear, but for larger gains behavior becomes less linear. Clipping should not be a problem yet around pixel values of 150-200, but values increase slower than they should. This is circumvented by limiting the maximum achievable analog gain to 3 in the working code.

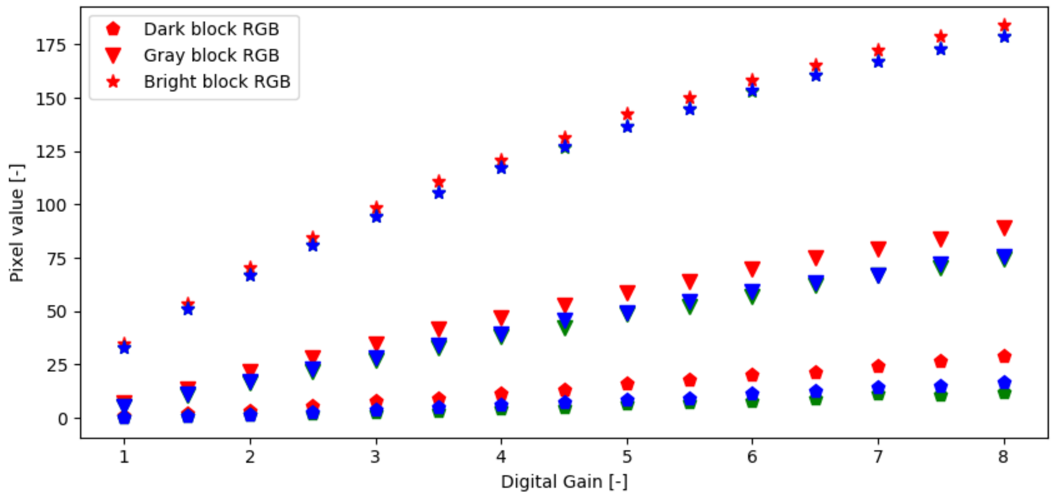


Figure 17: Results for the linearity test for the digital gain of the Pi Camera V2. Behavior in the lower gains is linear, but for larger gains behavior becomes less linear. Clipping should not be a problem yet around pixel values of 150-200, but values increase slower than they should. This is circumvented by limiting the maximum achievable digital gain to 2 in the working code.

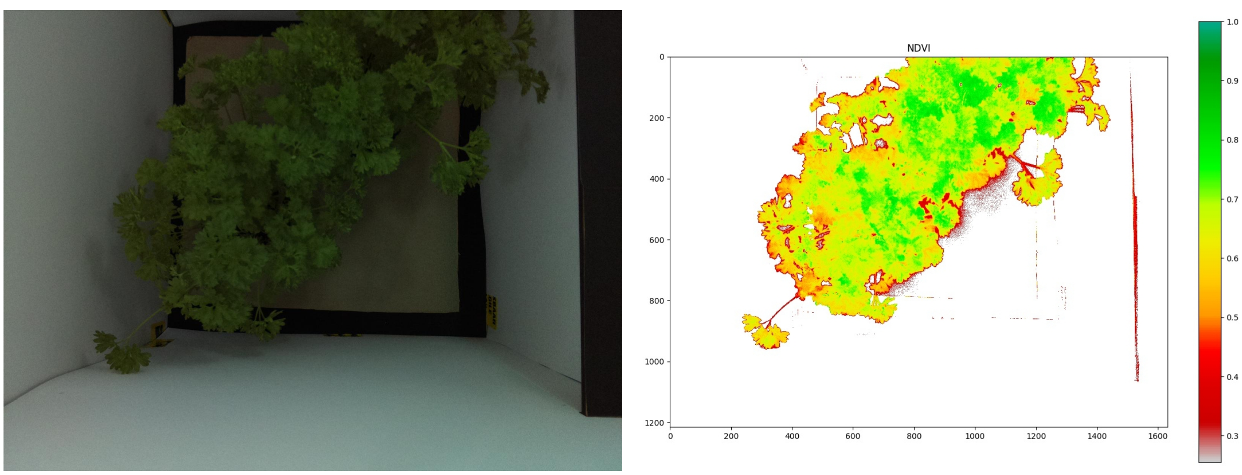


Figure 18: Situation on 20/05 at 14:57. Both plants are healthy, which can also be concluded from the NDVI image. Note that there is some noise in the image, as light is diffusely reflected from the plant on the white paper. In this regard, reflective foil tends to perform better.

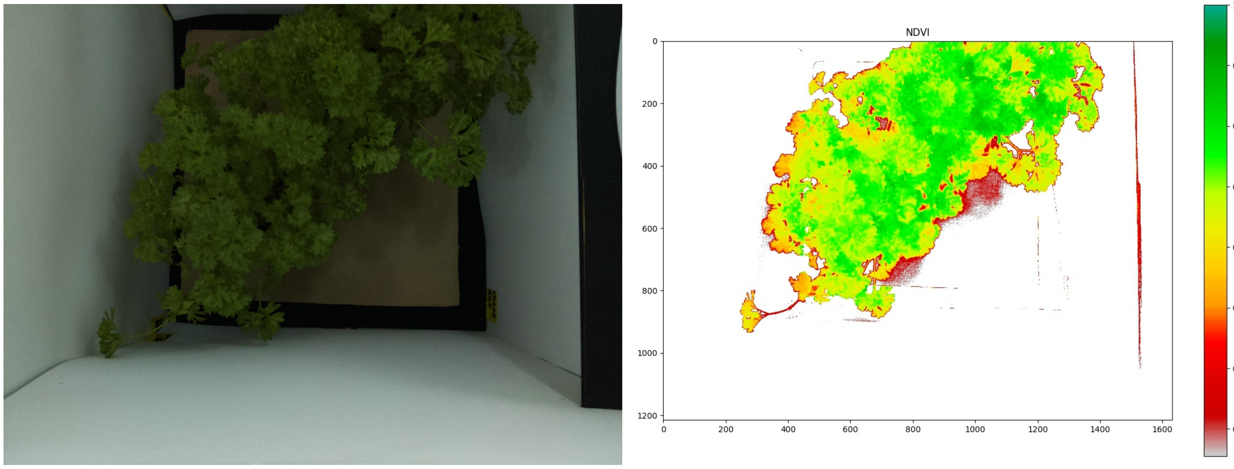


Figure 19: Situation on 20/05 at 23:08. Both plants are healthy and have taken up the water given to them earlier on the day.

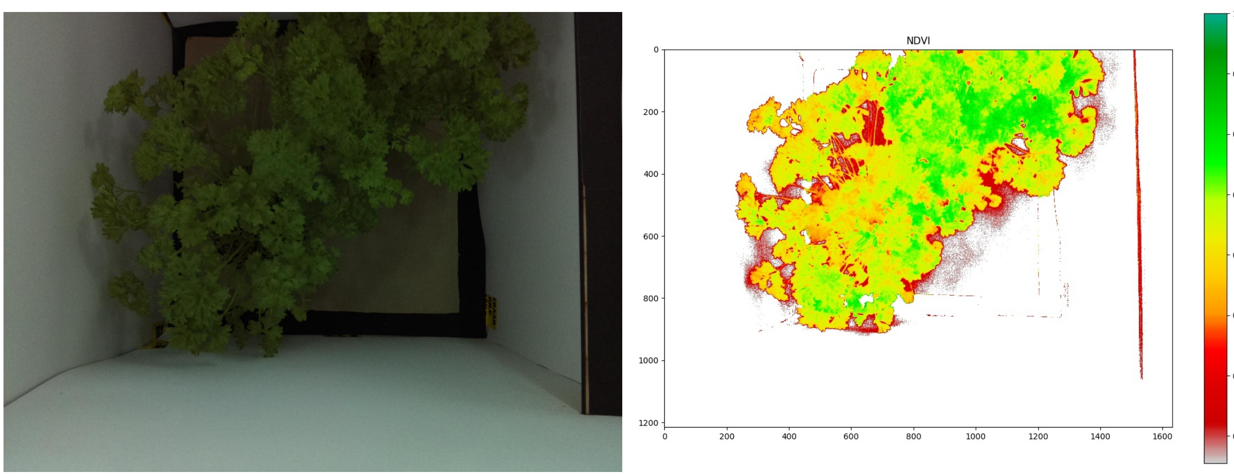


Figure 20: Situation on 23/05 at 14:20. The upper plant has received water, while the other has not. Decay starts to show in the NDVI image, but is still hard to spot in the regular image.

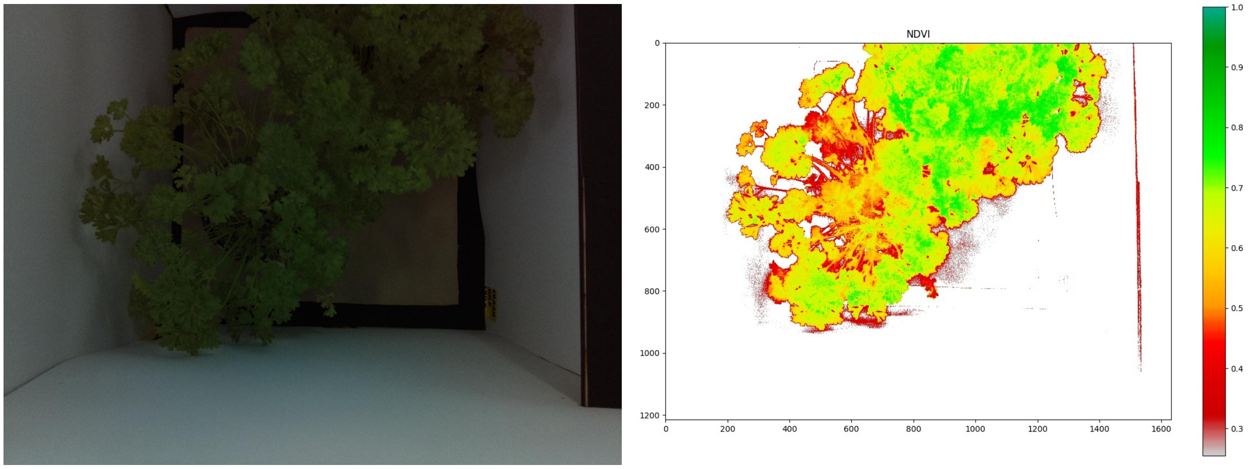


Figure 21: Situation on 24/05 at 22:06. The upper plant is doing pretty well. The lower plant is now clearly dying in the NDVI image, and it starts to show in the regular image as well.

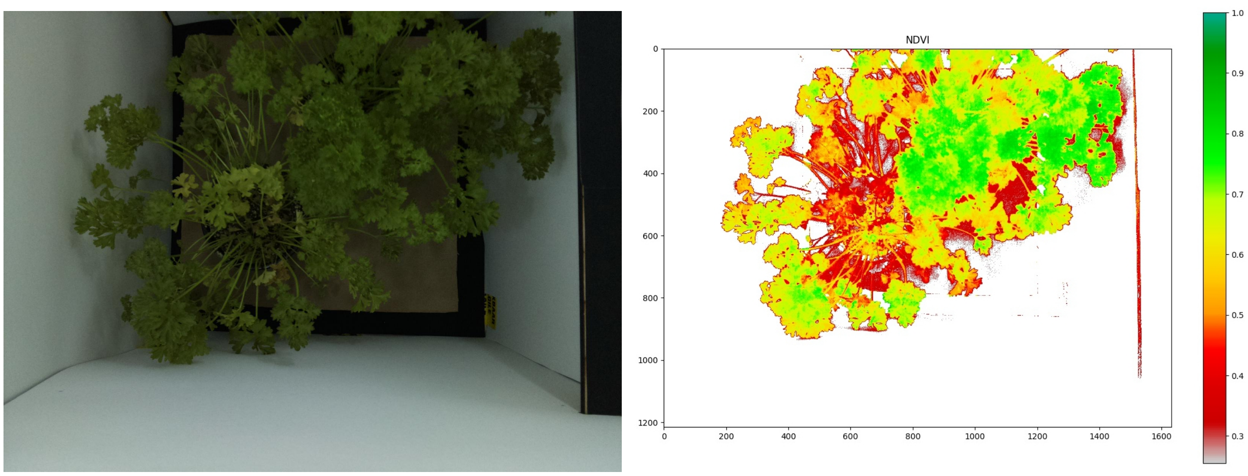


Figure 22: Situation on 26/05 at 21:52. The upper plant is still doing pretty well, but starts to suffer from presumably a little lack of daylight (as it starts to hang slightly). The lower plant is is still dying, with leaves turning notably yellow in the white image. These leaves correspond with a low NDVI, as shown in the right picture.

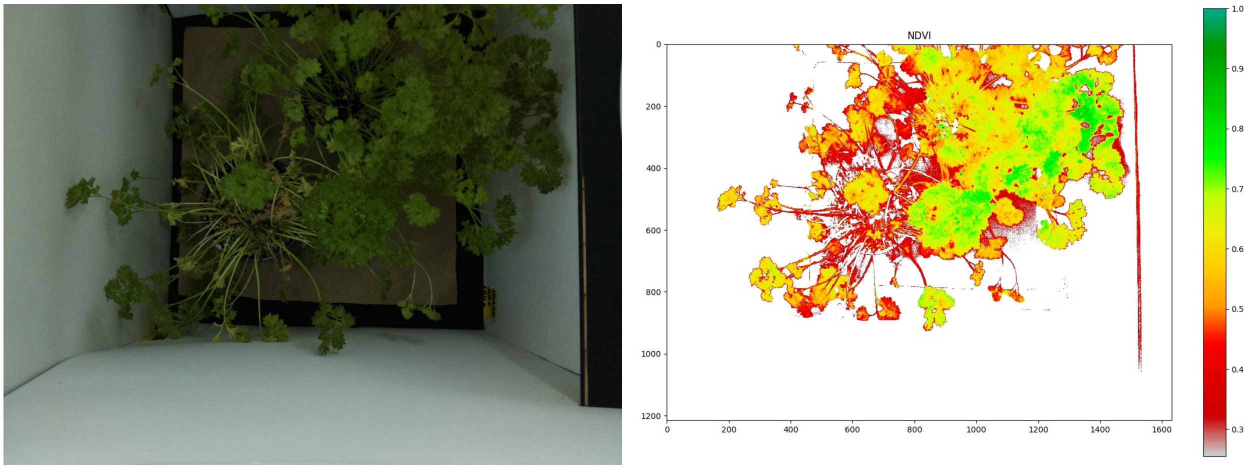


Figure 23: Situation on 30/05 at 04:37. The upper plant now also has not received water for a few days. Clearly the NDVI starts going down with respect to the previous set of photos. Again, nothing seems too bad on the regular image, but the NDVI image shows a less healthy plant. The lower plant is close to dead by now, as can be seen in both photos.

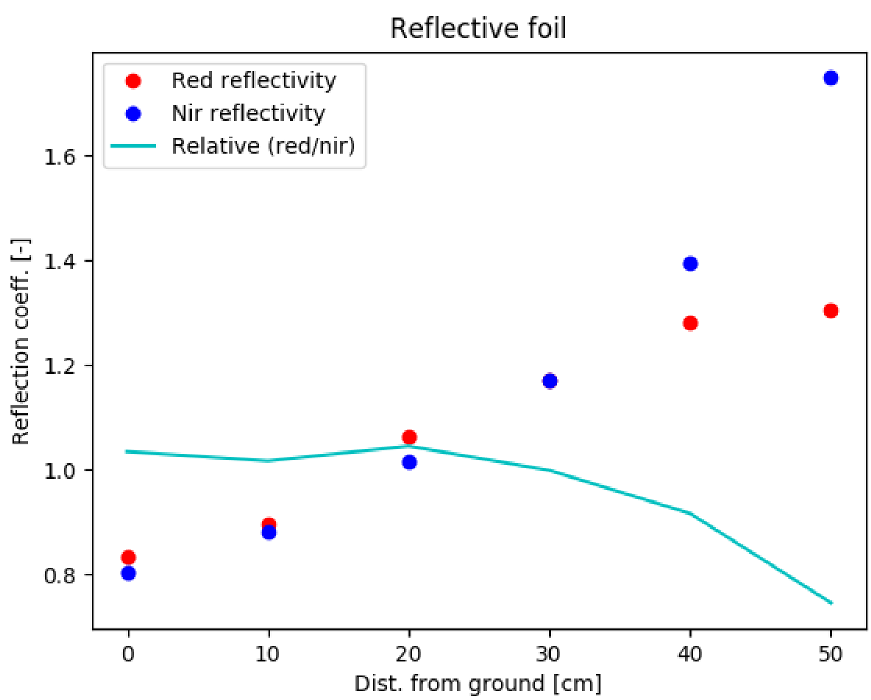


Figure 24: Intensity falloff when the reflective foil is put on the walls of the kit. The first few data points seem to keep the difference between red and nir light fairly constant, but after the 30 cm point, nir light starts to get relatively stronger and stronger, as expected based on the simulation.

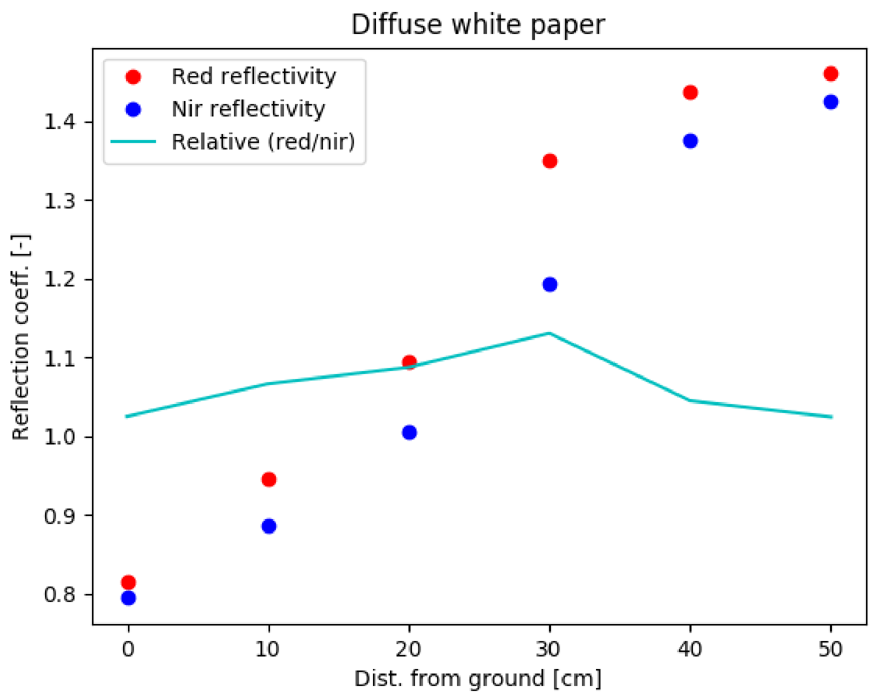


Figure 25: Intensity falloff when white paper is put on the walls of the kit. The difference between nir and red light stays fairly constant, as expected. There is a bit of a strange peak at 30 cm's, which could be due to sudden light leakage from outside.

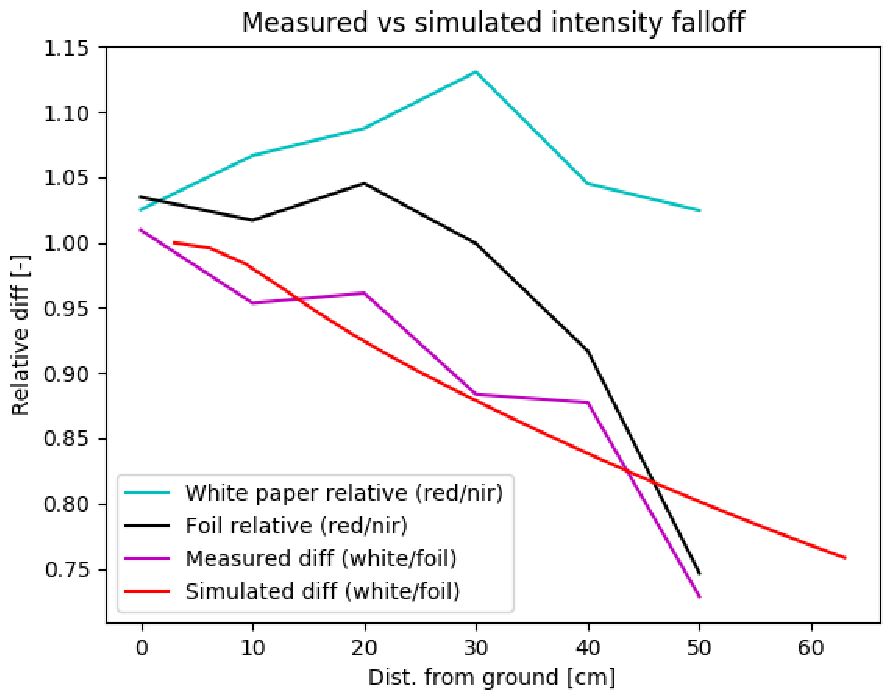


Figure 26: Comparison between the simulated falloff and the measured falloff. The two upper lines represent the the relative intensity falloffs from the previous figures, calculated as the red reflectance over the nir reflectance. The red line represents the simulated difference between white paper and reflective foil, while the purple line represents the measured difference between the red and the nir channel. The measured and simulated difference match fairly well, except for the last point at 50 cm. This is to be expected, since the conditions in the kit at the uppermost 10 cm's are not as ideal as the simulation (material does not extent up perfectly, cables and fans might interfere etc.)

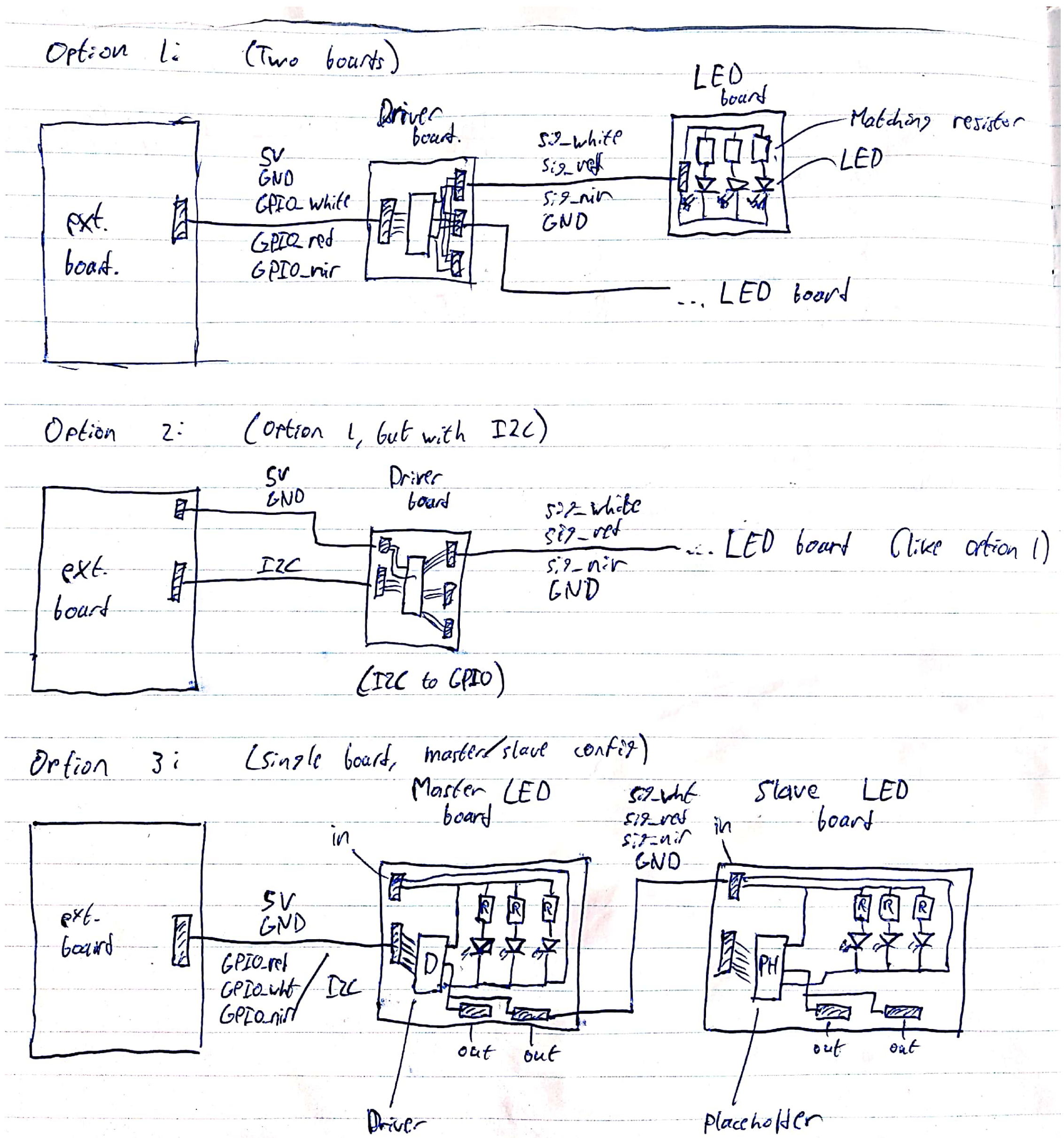


Figure 27: Proposition for the LED board production. The first two options list a driver board and separate LED boards (regular GPIO vs I2C bus), while the last option lists a master/slave configuration where one LED board acts as a master with the driver on board, while the other board(s) act(s) as (a) slave(s).