Master Thesis in ICT for Internet and Multimedia

Continuous 3D Reconstruction of Plants with Multispectral Information

Supervisor
Pietro Zanuttigh
Università di Padova

Co-Supervisor
Yalcin Incesu
Sony Europe B.V.

Co-Supervisor
Francesco Michielin
Sony Europe B.V.

Master Candidate
Maria Teresa Benato

Padova, 7 October 2019
Academic Year 2018/2019

Università degli Studi di Padova

Department of Information Engineering

Master Thesis in ICT for Internet and Multimedia

Continuous 3D Reconstruction of Plants with Multispectral Information

Supervisor
Pietro Zanuttigh
Università di Padova

Co-Supervisor
Yalcin Incesu
Sony Europe B.V.

Co-Supervisor
Francesco Michielin
Sony Europe B.V.

Master Candidate
Maria Teresa Benato

Padova, 7 October 2019

Academic Year 2018/2019

Abstract

Phenotyping is the process of identifying desirables traits of plants, i.e. drought resistance or yield productivity, and their health status by analysing them. These traits do not depend only on the genome of the plant but also on the environment in which it grows. For this reason a great amount of data has to be gathered to have a complete analysis of a plant species. Imaging techniques can be applied on this field to help relieving the bottleneck caused by manual gathering technique.

Climate changes represent a challenge to satisfy the demand of food of the increasing world population. Phenotyping can help to relieve this problem. Once the mapping and the characterization of the plant is completed, through the analysis of the plant in the first stages of its growth, we can see if e.g. it meets the optimal traits for food production and continue its growth only in the positive case. Moreover a plant variety that has good drought resistance property can be selected to relieve the problem that desertification will cause in the next years.

In this work I developed a robust pipeline to build a 3D model of plants with multispectral information as a basis for phenotyping. Fusing multispectral information coming from different images, we created a 3D multispectral point cloud. Multispectral information, especially the Near-Infrared band, can help to better understand the status of the plant. Then we registered the behaviour of plants over time and under water stress to see how their spectral reflection changes. Finally we used vegetation indices to see the evolution of the plant during the acquisition period.

Contents

1 Introduction

Phenotyping is the procedure to understand the good characteristic and the future behaviour of a plant from its structure and composition. Given that these desirable traits are not only the expression of the genome of the plant but depends also on the environment in which the plant grows, a lot of information needs to be gathered to perform this task.

In the last years imaging techniques were applied to this field to replace destructive, manual phenotyping procedures that were labour-intensive, time consuming and that require the presence of an expert. Simple imaging technique, e.g. processing of RGB images, are limited because they cannot gather information about the structure or the health of the plant, unless the plant is already largely compromised.

For this reason in this thesis we created a 3D multispectral reconstruction of a plant to be used as a basis for phenotyping. Thanks to the fusion procedure applied on multispectral information coming from different images of the same plant, we have a more robust measure of the multispectral values for each 3D point in the point cloud. Acquiring information of the same plant over time, we studied the changes in its spectral behaviour under water stress. This can be done thanks to the non-destructiveness of the used imaging technique.

Using the 3D reconstruction information, we have a better understanding of the plant structure while the multispectral data can give us information about the environmental stress that affects the health of the plant.

This thesis is structured as follows: in chapter 2 we can find the description of the 3D re-

construction procedure using structure from motion and COLMAP. The description of the setup used to gather data can be found in chapter 3. Chapter 4 describes the basis of phenotyping and the advantages to apply multispectral information to this problem. Chapter 5 contains the description of the approach that was used to perform the reconstruction and the processing of the data. The result of the work can be found in chapter 6 while chapter 7 contains the conclusions and possible extensions of this thesis.

2

Structure from Motion and 3D Reconstruction

3D reconstruction is a complex task due to the high number of information required to perform it. The main problem is to estimate the depth of the points that was lost with their projection in the 2D world.

This chapter will give an overview on the main approaches to build a 3D model starting from at least two images [1][2].

Firstly camera calibration will be describe to estimate the mapping parameters from the 3D to the 2D world. Then triangulation, matching and stereo calibration will be considered. Finally Structure-from-Motion (SfM) and dense point cloud reconstruction, as performed by COLMAP, will be described

2.1 Calibration and pose estimation of a single camera

A picture is the projection of the 3D world into the 2D domain. To define this mapping, the parameters that model the camera, that performs the projection, have to be retrieved. This process is called calibration.

To know the number of parameters that have to be estimated, the camera model has to be introduced.

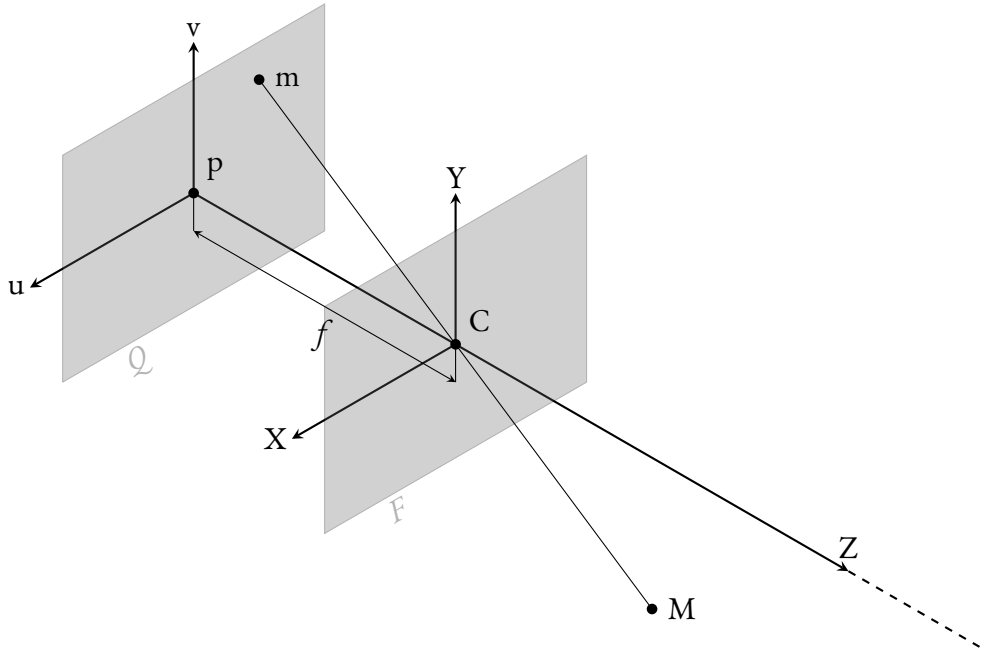


Figure 2.1: Projection of the 3D point **M** into **m**, 2D point.

2.1.1 Pinhole model

Pinhole model is the basic model of a camera and holds under some assumptions.

In this model the world reference system is represented by (X, Y, Z) and it is centred in C, called *Centre Of Projection* (COP). The Z axis is called principal axis. In this model it coincides with the optical ray, so the world reference system and the camera one coincide. The plane *F*, that is the perpendicular plane to the principal ray and it is centred in the COP, is called focal plane. The parallel plane to *F*, called *Q* in figure 2.1, is called image plane. Its distance from *F* is *f*, where *f* is called *focal length*. It is centred on the principal point p and it contains the camera reference system (u, v).

Therefore we can write a 3D point, in Cartesian coordinates, as $\widetilde{\mathbf{M}} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$ and its 2D projection through C, belonging to *Q*, as $\tilde{\mathbf{m}} = \begin{bmatrix} u \\ v \end{bmatrix}$.

From similar triangles, figure 2.2, we can see that

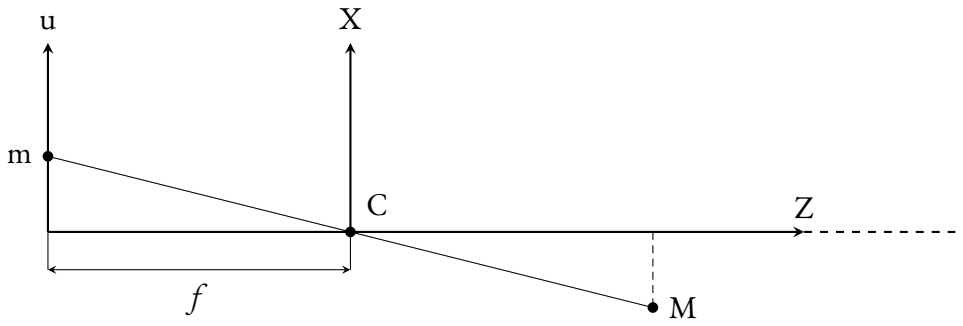


Figure 2.2: 2D view of the projection of the 3D point **M** into **m**.

$$\frac{f}{Z} = -\frac{u}{X} = -\frac{v}{Y} \qquad \begin{cases} u = -\frac{f}{Z}X \\ v = -\frac{f}{Z}Y \end{cases} \tag{2.1}$$

This mapping is non linear if we consider Cartesian coordinates. To have a linear relation, homogeneous coordinates need to be used instead.

The 2D and 3D points in homogeneous coordinates are represented as:

$$\mathbf{m} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \qquad \mathbf{M} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.2}$$

The mapping can then be written as:

$$Z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = Z \begin{bmatrix} -f\frac{X}{Z} \\ -f\frac{Y}{Z} \\ 1 \end{bmatrix} = \begin{bmatrix} -fX \\ -fY \\ Z \end{bmatrix} = \begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.3}$$

Therefore the projection of a 3D point into a 2D one is:

$$\mathbf{m} = \frac{1}{Z} P\mathbf{M} \qquad or \qquad \mathbf{m} \simeq P\mathbf{M} \tag{2.4}$$

where the last equation gives us equality up to a constant and with P, called *Camera Pro-*

jection Matrix, defined as:

$$P = \begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.5}$$

2.1.2 General Model

This model is an extension of the pinhole model: it takes in consideration differences between camera and world coordinate frames and the conversion from meters to pixels.

To have this last transformation, the matrix A is used:

$$A = \begin{bmatrix} -k_u & 0 & u_0 \\ 0 & -k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.6}$$

with (u_0, v_0) coordinates of the principal point and k_u and k_v the number of pixels per unit distance in image coordinates in u and v directions.

P can now be defined as:

$$P = \begin{bmatrix} -k_u & 0 & u_0 \\ 0 & -k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = K \begin{bmatrix} I & | & \mathbf{0} \end{bmatrix} \tag{2.7}$$

where

$$K = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

with $\alpha_u = fk_u$ and $\alpha_v = fk_v$.

K contains the internal characterization of the camera defined by four internal camera parameters: α_u, α_v, u_0 and v_0.

To completely describe the internal structure of the camera, we have also to consider a parameter called *skew*: it is the angle between the u and v axis. If it is considered, we can

rewrite K as:

$$K = \begin{bmatrix} fk_u & -\frac{fk_u}{tan\theta} & u_0 \\ 0 & \frac{fk_v}{sin\theta} & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

where the skew is defined as $\gamma = -\frac{fk_u}{tan\theta}$.

Usually θ is considered equal to $\frac{\pi}{2}$ therefore $\gamma = 0$, i.e. the skew can be neglected.

When the camera reference system is different from the world coordinate frame, we need to introduce an isometry to make the two systems coincide.

It is composed by a rotation, R, followed by a translation, t. We can write this transformation as

$$\mathbf{M}_c = V\mathbf{M} \tag{2.8}$$

where $\mathbf{M}_c$ is the point in the camera coordinate frame, $\mathbf{M}$ is the one in the world reference system and V is the view matrix:

$$V = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \tag{2.9}$$

V provides information about the orientation and position of the camera with respect to the world reference system.

The parameters contained in V are called *external* parameters of the camera.

Therefore we can describe the mapping from the 3D world to the 2D camera sensor as:

$$\mathbf{m} \simeq K \begin{bmatrix} I & | & \mathbf{0} \end{bmatrix} V\mathbf{M} = K \begin{bmatrix} R & | & \mathbf{t} \end{bmatrix} \mathbf{M} = P\mathbf{M} \tag{2.10}$$

with

$$P = K \begin{bmatrix} R & | & \mathbf{t} \end{bmatrix}$$

2.1.3 Zhang calibration technique

To compute the internal parameters of our cameras, the Zhang calibration [3] method was used. It requires several pictures of the same planar pattern, usually a checkerboard-like pattern, taken with different orientations without changing the internal parameters of the camera.

Without loss of generality, we can assume that the checkerboard is on the plane $Z = 0$ of the world coordinate system. For each picture, a projection matrix $P = K \begin{bmatrix} R_i & | & \mathbf{t}_i \end{bmatrix}$ is

computed. The intrinsic parameters, K, are the same for each image while R_i and $\mathbf{t}_i$ change for each image.

Using the identified corners of the board, for each picture the homography can be computed up to a scale. It has 8 degrees of freedom so to estimate the parameters of the projection matrix, that has 11 degrees of freedom, at least 3 picture are needed. 5 of these 11 degrees of freedom are given by the intrinsics, so they are the same for all the images, while 6 are for the extrinsics and they differ from image to image.

Re-projecting the corner projection back to the image, we can minimize the error with an iterative process. In this way the calibration procedure gives more accurate results.

2.1.4 Radial distortion

The introduction of real lenses in the camera model generates distortion effects.

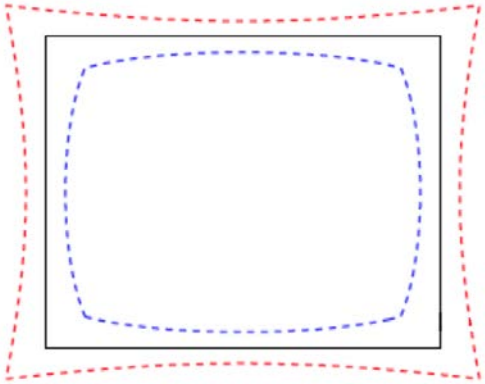


Figure 2.3: Distortion effects caused by the lens. The black rectangle is the undistorted ideal case, the red one represents the effect of pin-cushion distortion while the blue one is the barrel distortion case.

We can write the transformation from undistorted, ideal coordinates (u, v) to the distorted ones $(\hat{u}, \hat{v})$ as:

$$\begin{cases} \hat{u} = (u - u_0)(1 + k_1 r_d^2) + u_0 \\ \hat{v} = (v - v_0)(1 + k_1 r_d^2) + v_0 \end{cases} \tag{2.11}$$

where $r_d^2 = (\frac{(u-u_0)}{\alpha_u}) + (\frac{(v-v_0)}{\alpha_v})$ and (u_0, v_0) are the coordinates of the principal point. The coefficient that represents radial distortion is k_1.

Writing $x = \frac{(u-u_0)}{\alpha_u}$ and $y = \frac{(v-v_0)}{\alpha_v}$ we have:

$$\begin{cases} \hat{x} = x(1 + k_1(x^2 + y^2)) \\ \hat{y} = y(1 + k_1(x^2 + y^2)) \end{cases} \tag{2.12}$$

To compute the distortion coefficient, the camera projection matrix P is needed but to compute P, we need to know k_1.

This problem is therefore solved iteratively: firstly P is estimated. Using this result k_1 is then computed and P is refined after distortion removal. This process is repeated up to convergence.

OpenCV calibration considers 5 distortion coefficients to achieve more accuracy. 3 of them are given by radial distortion, k_1, k_2 and k_3, and 2 by tangent one, p_1 and p_2.

2.2 Stereo rig

When a 3D point is projected into a 2D one, information regarding its position in the 3D world are lost.

To recover them, we can use two different images taken from a slightly different point of view one from the other. This operation is called *triangulation* and can be performed when calibration parameters of the two cameras are known.

Of course the correspondence between the same point in the two images has to be known.

2.2.1 Triangulation

Triangulation is the procedure that allows us to estimate the position of a point in the space starting from its projections on two images (conjugate points). To do that, we have to know the camera matrix of the two different cameras that took the pictures.

Simple stereo case

Let us consider a couple of cameras with the same focal length f and with parallel image planes. C and C', the corresponding centres of projection, are on the X axis at a distance b called *baseline*. The left camera reference system corresponds to the world coordinates (figure 2.4).

The 3D point M (in the figure only X and Z axes are represented) is mapped into two 2D points: m = (u, v) and m' = (u', v'). The first belongs to the image plane of the left camera, the second to the one of the right camera.

Using the similitude among triangles, we can write:

$$\begin{cases} \frac{f}{z} = -\frac{u}{x} \\ \frac{f}{z} = \frac{u'}{b-x} \end{cases} \tag{2.13}$$

where b and f depend on the camera: the former is related to the extrinsic parameters and the latter to the intrinsic ones. From 2.13 we can write:

$$x = \frac{-bu}{u' - u} \tag{2.14}$$

Using equations 2.13 and 2.14, we can finally write the equation for the z component:

$$z = \frac{bf}{u' - u} \tag{2.15}$$

We have now the relation between the depth z and the disparity $d = u' - u$ given by equation 2.15: z is inversely proportional to the disparity.

Computing the disparity we can therefore estimate the depth z and reconstruct the coordinates of M.

General case

The situation described in the previous section is very restrictive, therefore we need a more general solution.

We can write the two camera matrices in the following way:

$$P = \begin{bmatrix} \mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \mathbf{p}_3^T \end{bmatrix} \qquad P' = \begin{bmatrix} \mathbf{p}_1'^T \\ \mathbf{p}_2'^T \\ \mathbf{p}_3'^T \end{bmatrix} \tag{2.16}$$

where each element of the vector is a row of the camera matrix.

We can then write:

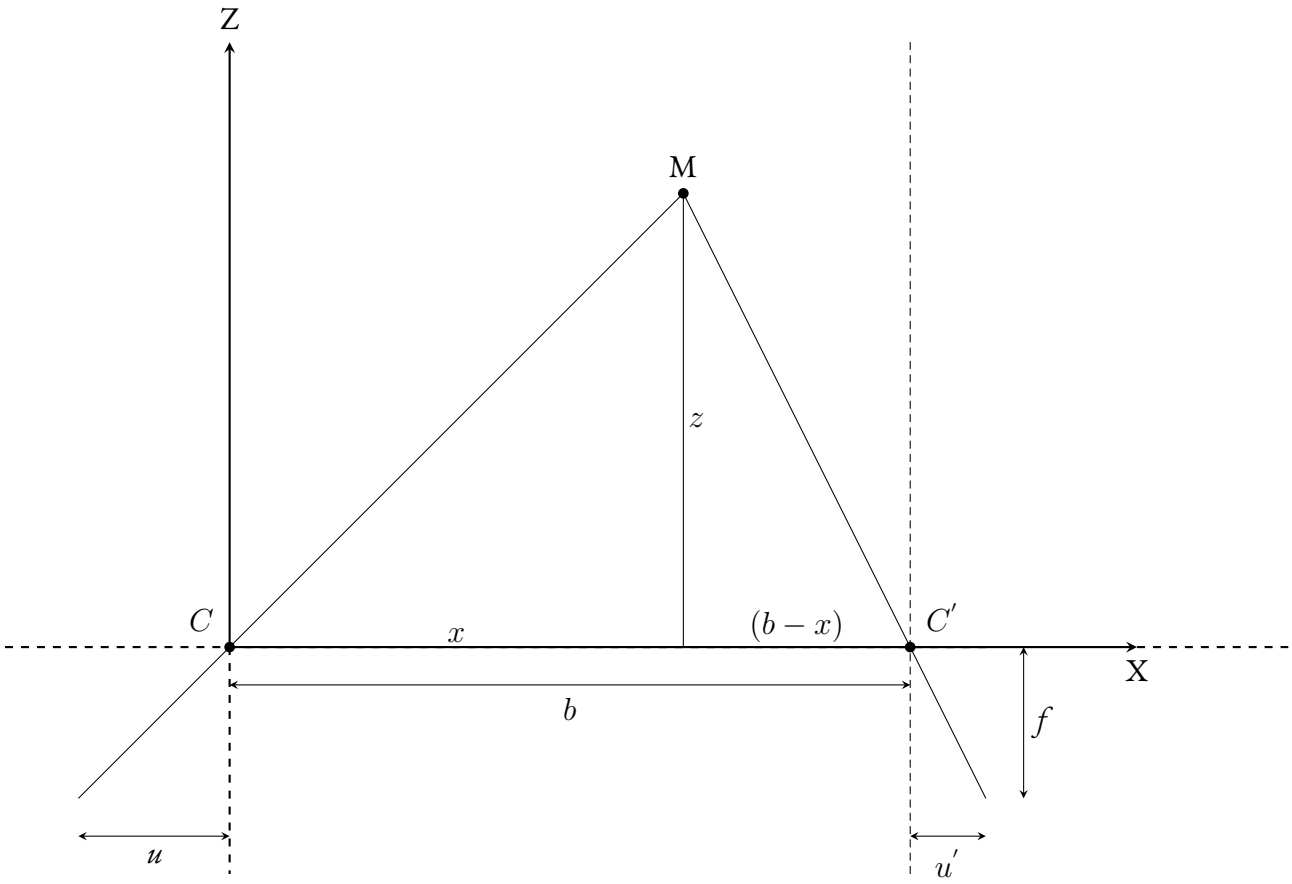


Figure 2.4: Naive triangulation case.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = P\mathbf{M} \qquad \begin{cases} u = \mathbf{p}_1^T\mathbf{M} \\ v = \mathbf{p}_2^T\mathbf{M} \\ 1 = \mathbf{p}_3^T\mathbf{M} \end{cases} \tag{2.17}$$

from which we obtain:

$$\begin{cases} u\mathbf{p}_3^T\mathbf{M} = \mathbf{p}_1^T\mathbf{M} \\ v\mathbf{p}_3^T\mathbf{M} = \mathbf{p}_2^T\mathbf{M} \end{cases} \qquad \begin{bmatrix} \mathbf{p}_1^T - u\mathbf{p}_3^T \\ \mathbf{p}_2^T - v\mathbf{p}_3^T \end{bmatrix} \mathbf{M} = \mathbf{0} \tag{2.18}$$

Repeating this process also for the second camera we have:

$$A\mathbf{M} = \begin{bmatrix} \mathbf{p}_1^T - u\mathbf{p}_3^T \\ \mathbf{p}_2^T - v\mathbf{p}_3^T \\ \mathbf{p}_1^{'T} - u^{'}\mathbf{p}_3^{'T} \\ \mathbf{p}_2^{'T} - v^{'}\mathbf{p}_3^{'T} \end{bmatrix} \mathbf{M} = \mathbf{0} \tag{2.19}$$

The solution of the equation 2.19 is the kernel of the matrix A. We can decompose A using SVD and selecting the last eigenvector. The problem with this solution is that it is algebraic and does not consider the geometric error. To minimize the geometric cost, we can re-project back the point and compute the error with respect to its original position and iteratively try to minimize it. The cost function is the following:

$$\epsilon(\mathbf{M}) = \left\| \begin{bmatrix} u \\ v \end{bmatrix} - \begin{bmatrix} \frac{\mathbf{p}_1^T\mathbf{M}}{\mathbf{p}_3^T\mathbf{M}} \\ \frac{\mathbf{p}_2^T\mathbf{M}}{\mathbf{p}_3^T\mathbf{M}} \end{bmatrix} \right\|^2 - \left\| \begin{bmatrix} u^{'} \\ v^{'} \end{bmatrix} - \begin{bmatrix} \frac{\mathbf{p}_1^{'T}\mathbf{M}}{\mathbf{p}_3^{'T}\mathbf{M}} \\ \frac{\mathbf{p}_2^{'T}\mathbf{M}}{\mathbf{p}_3^{'T}\mathbf{M}} \end{bmatrix} \right\|^2 \tag{2.20}$$

2.2.2 Rectification

The rectification operation consists in bringing back the two cameras to the configuration presented in section 2.2.1, without changing the coordinates of the two centres of projections C and C'. That setup simplifies the computation because the two conjugate points, that we are trying to match, lay on the same line: we reduce the number of possible conjugate points (our search becomes unidimensional from bi-dimensional) therefore the number of possible

false matches.

We need the rectification operation because usually we have a different system with respect to the naive one.

To perform this operation, the two cameras are rotated around their center of projection (to make the two image planes parallel) and their reference systems are translated to obtain the situation in which $v = v'$. We can write the rotation matrix:

$$R = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix} \qquad \begin{cases} \mathbf{r}_1 = \frac{\tilde{\mathbf{C}}' - \tilde{\mathbf{C}}}{||\tilde{\mathbf{C}}' - \tilde{\mathbf{C}}||_2} \\ \mathbf{r}_2 = \mathbf{k} \times \mathbf{r}_1 \\ \mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2 \end{cases} \tag{2.21}$$

where k is an arbitrary versor that is orthogonal to the new Y axis. The optical ray associated to m and m' do not change.

If, after these operations, we have still $v \neq v'$, we can obtain the equality translating the reference systems.

2.2.3 Stereo rig calibration

Stereo calibration is the process that allows to estimate the transformations between two camera orientations of a rig. These transformations will be constant during the acquisitions if the rig is not modified.

The needed information to perform stereo calibration are the intrinsic parameters of the two cameras and pairs of acquisitions of the same planar cheeseboard-like pattern. It will be therefore possible to compute R_{stereo} and $\mathbf{t}_{stereo}$ that are the rotation matrix and the translation vector that defines the cameras orientation difference. From them it is possible to estimate also the baseline as $b = ||t_{stereo}||$.

2.3 Matching strategies

Having two views of the same object taken with a small, but not too small, baseline, it is possible to find the same point that is projected in the two different images: we can associate m with m'. Using triangulation, we can then recover the position of the point in the space. Repeating this procedure for all the pixels in the picture, we can build a dense 3D model. We call this process *disparity* estimation.

To perform the matching process, two different approaches can be used:

- local methods: they consider only a small window around the pixel that we want to match;
- global methods: they pose the constraint on the whole line or on the whole image.

2.3.1 Block matching

Block matching is a local method to compute the *disparity*.

Having two images I and I', we want to find the correspondent (u + d, v) in I' for the point (u,v) in I . The window centered in m $= (u, v)$, with dimensions (2N + 1)(2N + 1), is compared with the one of the same dimensions centered in m' $= (u+d, v)$ using a coupling metric.

The procedure is repeated for different disparity values d in a range $[d_{min}, d_{max}]$ where d_{min} depends on the largest distance z we want to measure.

The final disparity $d_0(u, v)$ for m is the value that minimize or maximize the coupling metric.

There are different coupling metrics that can be used:

- based on the intensity differences (we want to minimize them): SSD, SAD;
- based on correlation (we want their maximization): NCC, ZNCC;
- based on the intensity transformations: Census transform.

The metric used by COLMAP is the *Normalized Cross Correlation* (NCC). It is a similarity metric to be maximized. NCC is defined as the normalized scalar product of the two considered windows:

$$NCC(u, v, d) = \frac{\sum_{k,l} I(u + k, v + l)I'(u + k + d, v + l)}{\sqrt{\sum_{k,l} I(u + k, v + l)^2}\sqrt{I'(u + k + d, v + l)^2}} \tag{2.22}$$

and the disparity is:

$$d_0(u, v) = \underset{d\in[d_{min},d_{max}]}{\arg\max} \; NCC(u, v, d) \tag{2.23}$$

Once the disparity is known, the depth can be found using the equation 2.15.

The accuracy of the estimation is affected by occlusions and uniform regions due to false detection of conjugate points.

2.4 COLMAP

COLMAP is a general-purpose program to perform 3D reconstruction through incremental Structure-from-Motion [4] [5].

2.4.1 SfM in COLMAP

SfM aims to build a 3D reconstruction of an object starting from more than two images of the object taken with different points of view. Images can be taken also with uncalibrated cameras: SfM is able to estimate internal and external camera parameters during the reconstruction procedure.

First of all the matches regarding the same point among the different images have to be found. SIFT [6] is used to extract the images features in a robust way, invariant to radiometric and geometric changes.

Then the most similar features are searched and matched so correspondences among the relative points are created. To have robust matches, SfM computes the transformation to project linked points from an image to another. If the majority of the mapped points are correct, the matches are verified. RANSAC [7] is used to remove outliers during this phase.

Once the robust matching is computed, the basis for the 3D reconstruction procedure is available. COLMAP uses the incremental SfM procedure: the 3D model is initialized performing the reconstruction from two images. Then a new image, the one that introduces more information having still enough redundancy, is iteratively added to the model and new points are added to the reconstruction by triangulation.

Redundancy and newly added points are fundamental for the camera parameters estimation and for the *Bundle Adjustment* [8] procedure, a joint non linear refinement of the camera parameters.

COLMAP applies a bundle adjustment run on the most connected images after each added image, this is called *local* bundle adjustment, and a *global* bundle adjustment on the whole model when it grows of a certain percentage.

When no more images are available or when the addition of a new image does not improve the model, the incremental procedure ends and a global bundle adjustment is performed on

the entire model.

Once the final 3D sparse model is built, COLMAP gives the possibility to run a final bundle adjustment with the so called *rig constraint*: if a stereo rig is used, the cameras parameters are tuned to keep the baselines between the defined cameras constant.

There is not the possibility to set the value for the baseline so the reconstruction given by COLMAP will be up to a scale.

2.4.2 Depth and normal estimation

View selection, depth and normals estimation processes performed by COLMAP are based on the work of Zehng et al.[9].

View selection is performed considering the probability of taking similar patches to the considered one from the colour point of view.

In [9] the NCC is used as matching metrics. COLMAP [5] uses a bilaterally weighted adaptation of the NCC to be more robust against blurred depth discontinuity. Weights are function of the grayscale distance between the considered pixel and the central one and their spatial distance. This allows also to take in consideration the photometric consistency. Moreover geometric consistency is considered to avoid outliers due to noise and occlusions. This is done by integrating multi-view geometric consistency constraint to the analysis. A filtering noise process is then applied to discard the noise. Differently from Zehng, in COLMAP per pixel depth and normals are estimated.

To perform the fusion operation of depth and normal maps, a graph of consistent pixels is built considering their normal and depth similarity and re-projection error. The fusion is then performed by taking the median of the consistent points to remove outliers. Finally the dense point cloud is created using the fused information.

3

Capturing setup description

During my internship, I worked with an experimental setup kindly provided by Sony Europe B.V. in the Stuttgart Technology Center.

Different attempts were made to optimize the background and illumination before reaching the final setup. Saturation, change of natural illumination, fixed matches found on the background were major issues during the reconstruction procedure. Moreover the sensor position in two of the cameras had to be tuned (see difference in figure 3.1a and 3.1b), otherwise useful information was lost by two of the three cameras.

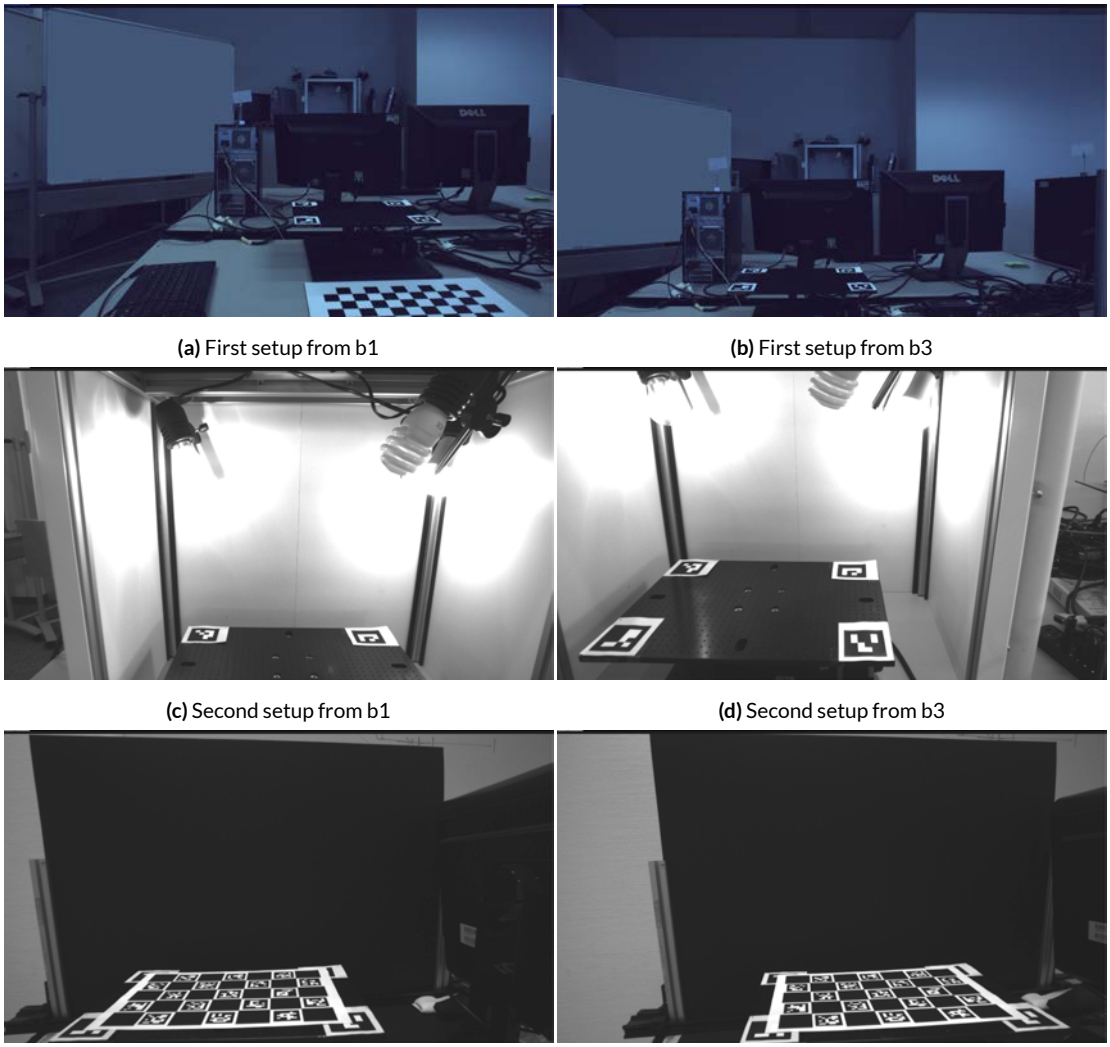
The final setup was composed by a stereo rig of three cameras, a turntable, two halogen lamps, a programmable socket, all controlled by a computer, and by a ChArUco board and some pieces of uniform black cardboard to be used as background.

3.1 Stereo rig

The rig were formed by three cameras: one multispectral camera and two RGB ones.

3.1.1 Multispectral camera

Multispectral sensors are devices that can acquire visible and non-visible portion of the spectrum. The gathering of spectral information can be done by capturing a band at a time in subsequence exposure times (scanning technique) or simultaneously (snapshot technique)



(a) First setup from b1

(b) First setup from b3

(c) Second setup from b1

(d) Second setup from b3

(e) Final setup from b1

(f) Final setup from b3

Figure 3.1: Evolution of the setup.

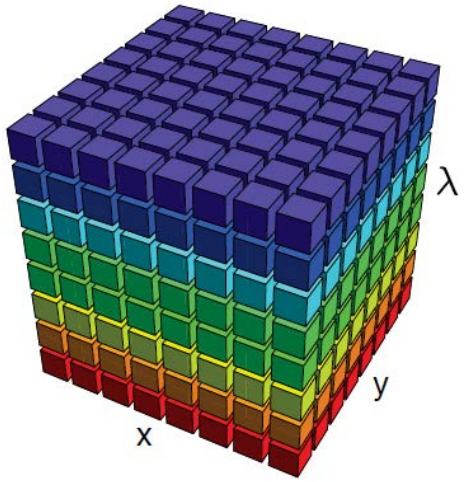


Figure 3.2: Spectral data cube created by a snapshot multispectral sensor [11].

[10]. The collected data are organized in a three-dimensional matrix called datacube (figure 3.2).

The main advantage of scanning technique is that every band is acquired with full resolution but multiple exposures lead to motion artifacts and to an higher cost of production due to a more complex architecture (e.g. turnable filters and tunable illumination).

Using snapshot techniques, we have a less artifacts and lower costs. One of the most used technique to build a snapshot sensor is the *MultiSpectral Filter Array* (MSFA, figure 3.3): it is an extension of the *Color Filter Array* (CFA) for more than three bands. Given that the final multispectral image is the result of a demosaicing procedure, a good design of the MSFA is fundamental and it is usually application specific with respect to the number of bands it has to capture. If the geometric filters arrangement of the MSFA is not well designed, there can be a huge loss in the image resolution.

Multispectral sensor of the setup

The multispectral camera we used during the acquisition procedure has a snapshot sensor that captures spectral information in the wavelength interval from $450nm$ to $800nm$ with uniform spacing of $10nm$. This range allows to gather information regarding the Near-InfraRed (NIR) component of the spectrum, band of particular interest for phenotyping purposes.

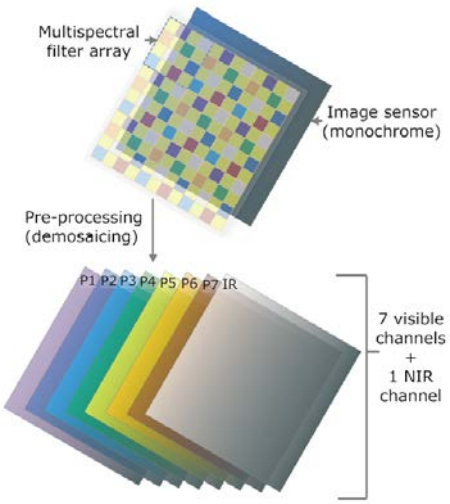


Figure 3.3: Visualization of the Multispectral Filter Array [10].

3.2 Illumination of the scene

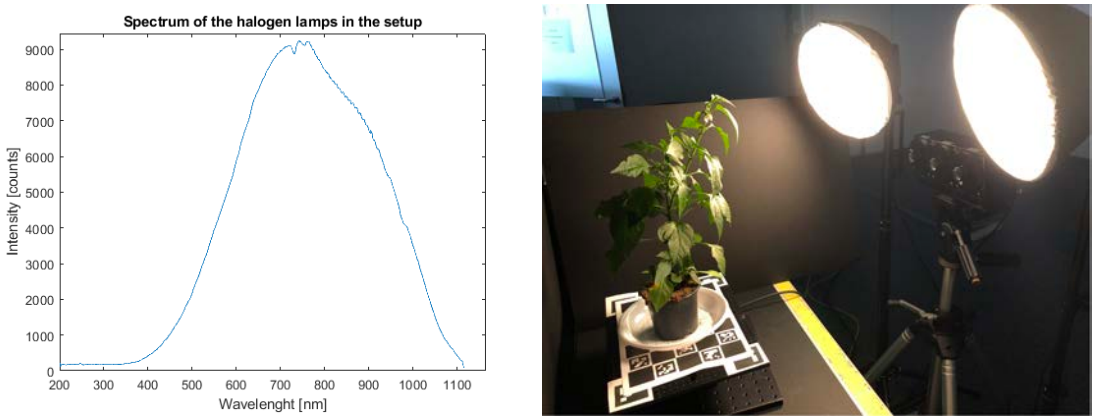
The setup was placed in a room without windows to have more control on the light conditions. Two $28W$ halogen light bulbs were chosen as light sources given their broad spectrum (figure 3.4). To avoid saturation and have an uniform illumination, the two lamps were placed inside two light diffusers situated on two stands on the left and on the right of the camera rig.

3.3 Elements to aid data gathering for SfM

The 3D reconstruction of the plants was performed using SfM technique combined with a turntable: the object to be reconstructed was placed on the turntable and acquisitions were performed during the rotation of the table.

The background was composed by some pieces of black cardboard to avoid too many outliers in the matching procedure. The problem caused by those fixed matches was a wrong pose estimation and a wrong parameters correction performed by *COLMAP* during the bundle adjustment procedure.

The turntable had an high precision rotation mechanism. However the information provided by the precise rotation angle were not used to estimate the external parameters of the cameras. The speed and acceleration of the table were set to avoid movements in the leaves.



(a) Spectrum of the light sources measured with a spectrometer

(b) Position of the light sources with respect to the stereo rig

Figure 3.4: Spectrum of the light sources measured with a spectrometer and their positions with respect to the stereo rig.

The poses of the cameras during the rotation of the table were obtained thanks to the ChArUco board placed on top of the turntable.

3.3.1 ArUco markers and ChArUco Board

ArUco markers are fiducial square markers that can be used to estimate the camera pose by finding correspondences between the known marker and its respective camera projection [12].

The advantage of square markers is that the pose can be estimated from their four corners, if the camera calibration parameters are known.

The inner part of the marker, that contains a binary code, is used for marker identification, false positives rejection and errors correction. Once the actual markers are detected, the pose can be estimated by iteratively minimizing the reprojection errors of the corners.

To have an approach that is robust against occlusions, more markers are placed in a structured grid called marker board, figure 3.5. In this way we achieve not only robustness against occlusions, unless the board is mostly covered, but also less influence of the noise, given that more corners can be used to compute the pose.

However ArUco markers have a drawback: the estimation of the corner position is not so accurate.

To overcome this problem a ChArUco board, figure 3.6, can be used: it is a combination

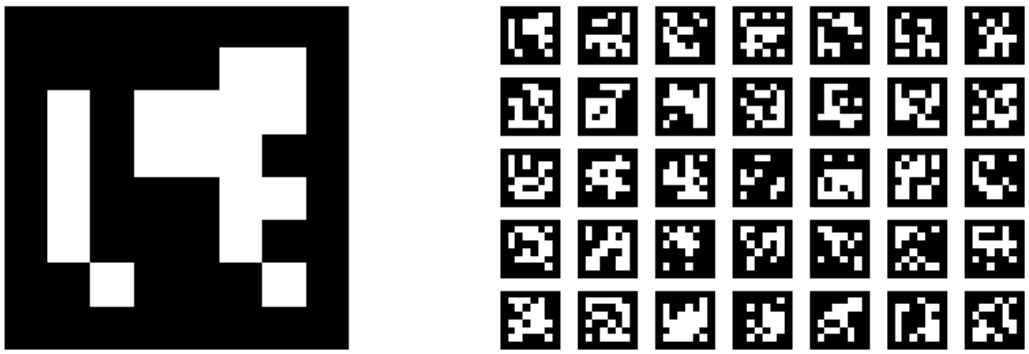


Figure 3.5: ArUco marker and ArUco board.

of a chessboard pattern and several ArUco markers placed in the white squares of the chessboard. In this way high precision corner detection can be achieved by using the corners of the chessboard together with a sub-pixel refinement procedure and robustness against occlusions is given by the ArUco markers.

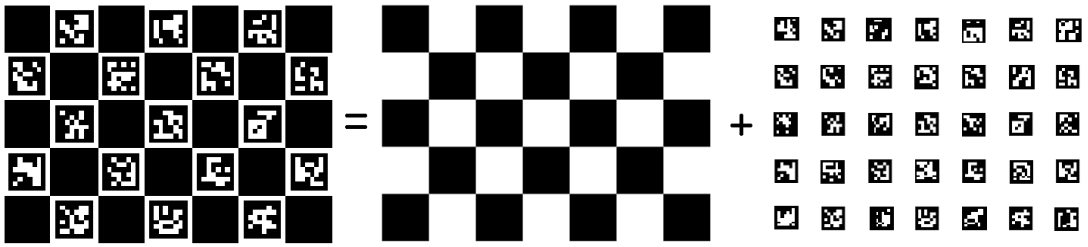


Figure 3.6: A ChArUco board: it is a combination of a chessboard pattern and of ArUco markers.

3.4 Automation of the setup

One of the main task performed to improve the setup performance was completely automatize the acquisition procedure.

The first step was synchronizing data acquisition with the rotation of the turntable. Three pictures (one for each camera) were taken each five degrees of rotation of the table. This procedure was repeated 72 times to have a complete 360° rotation. In the end 72 images per each camera were available.

The second step was having a system that automatically performed one acquisition every six hours (4 acquisitions per day) for one week. This allowed to capture also the temporal

evolution of the plants.

To optimize energy consumption and assure a correct cycle of light to the plant, a programmable socket was added to the setup. In this way the lights were switched on during the acquisition procedure and remained on twelve hours per day.

3.5 Plants

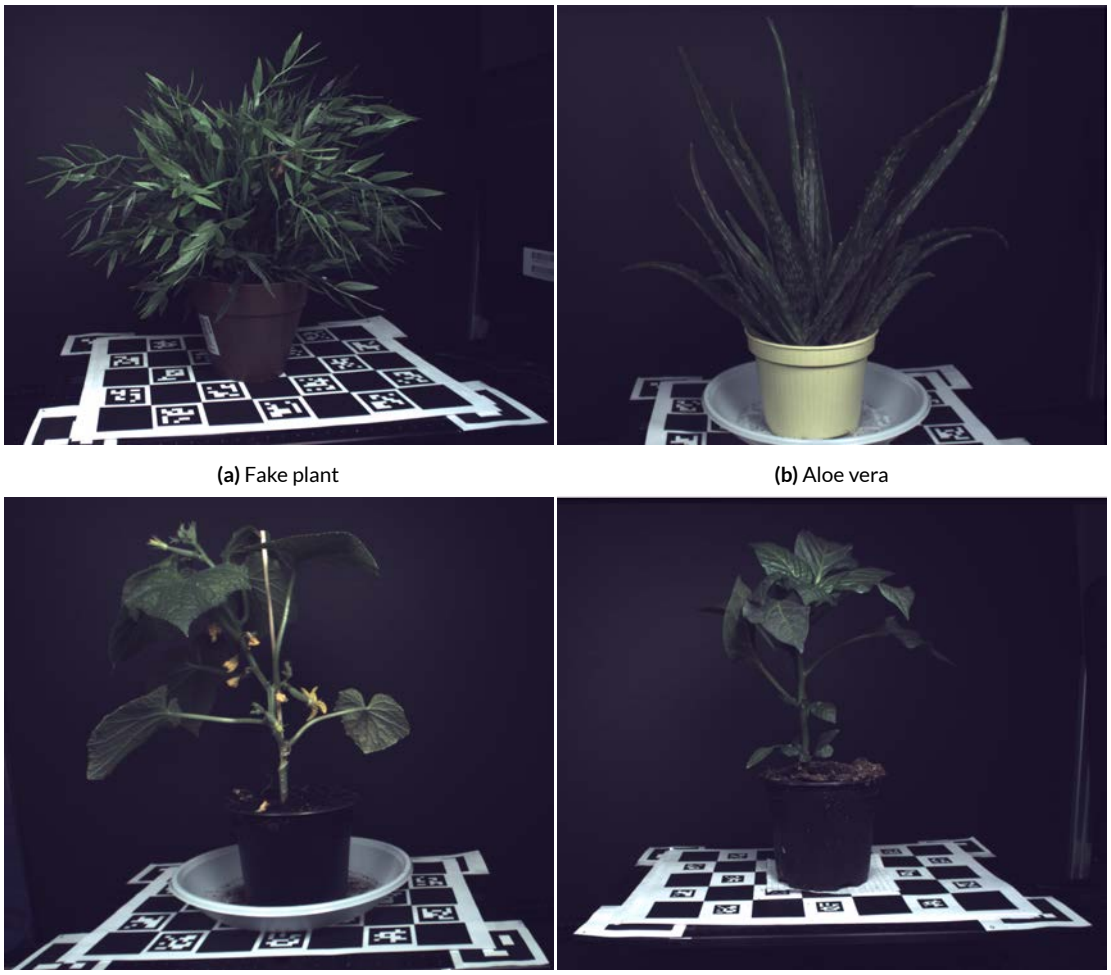
The plants used for the acquisitions, figure 3.7, were a fake plastic plant, an aloe vera, a *cucumis sativus* (cucumber plant) and a *capsicum annuum* (pepper plant). The first two were used to perform 3D reconstructions trials and mapping tests, the other two to study the behaviour of plants under water stress.

3.6 Limitations of the setup

The setup works in a controlled environment: changes in illumination and weather condition, e.g. wind, can affect the result of the 3D model reconstruction. Moreover the multispectral sensor will saturate if illumination conditions are too strong.

The turntable and the fix camera rig position set a constraint on the maximum height of the plants to be acquired.

Lastly an initialization for the camera poses, computed using the ChArUco board, is needed to have an accurate result to further process the 3D data.



(a) Fake plant

(b) Aloe vera

(c) *Cucumis sativus*

(d) *Capsicum annuum*

Figure 3.7: Plants used to perform the 3D reconstructions

4

Phenotyping with multispectral and 3D information

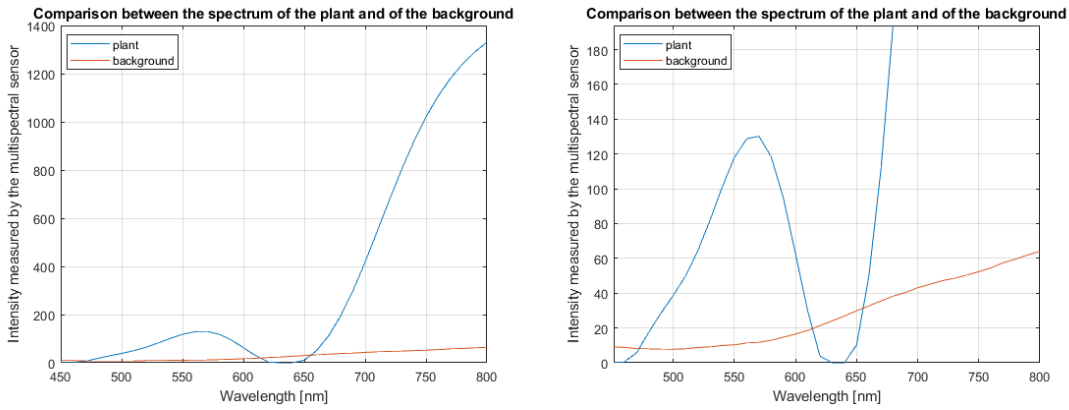
The increase of world population together with the changes in the climate represent a new challenge to meet the growing demand of food. Therefore new approaches to help identifying desirable agricultural traits in crops, e.g. stress tolerance, yield stability, yield potential, are needed.

4.1 Phenome and genome

To understand why identifying the good characteristic of crops is a complex task, we have to distinguish among phenome, phenotype and genome. Genome is the total set of genes that are present in an individual. A phenotype is a trait or a characteristic that is observable in an organism. The phenome refers to the phenotype as a whole [13], i.e. to all the phenotypical traits that belong to an organism. Plant phenomics is therefore the study of plant growth, performance and composition. [14].

Crop selection, i.e. choose the best genotype that has the desired characteristics, makes use of plant phenotyping by analysing the phenotypic expression of crops.

It is possible to have a complete characterization of the gene, e.g. using DNA sequencing, while is extremely difficult to have a complete description of phenome. This because phe-



(a) Spectrum of a pixel belonging to the *Cucumis sativus* (blue line) and the one of the background (orange line)

(b) Zoomed in version of the comparison

Figure 4.1: Comparison between plant spectrum and the background spectrum.

nome can be described as $phenome = genome \times environment$ [14]. So plants which have the same genotypical constitution but growing in different environment, can have different phenotypical traits [15].

To have a complete description of a phenotypical trait of interest, there is the need of replicated information gathering made in different environment and in different season. This work is destructive for crops, extremely time-consuming, labour intensive and requires a lot of precision. Moreover experts are needed to study the resulting data.

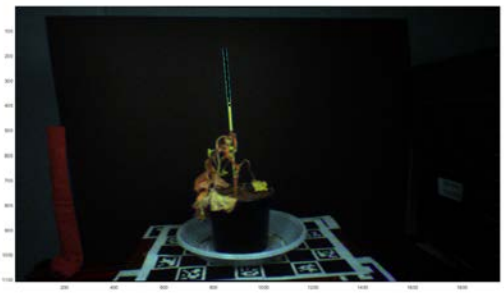
To overcome this phenotyping bottleneck, imaging technique started to be applied to this problem to reach non-invasive and non-destructive high-throughput phenotyping.

4.2 Multispectral sensor applied to plant phenotyping

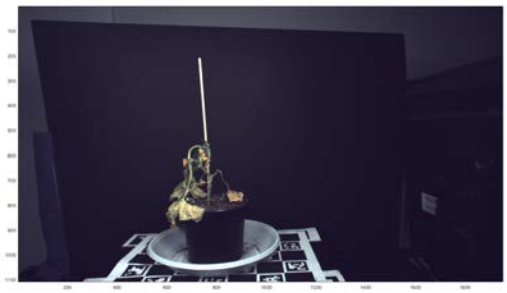
Multispectral imaging can enhance the information we can gather from plants. Vegetation has in fact a specific spectral behaviour that can help distinguish it from other elements present in the environment [16]: plants have a low reflection in the blue and red bands, due to strong absorbance performed by chlorophyll, while they have high reflection in the green and Near-InfraRed (NIR) bands, figure 4.1. Moreover the spectral property of plants can change among different species and depending on the health of the plant. The blue band is generally considered to study carotenoids content of the leaves [17], aspect on which we did not focus in our study.

(a) Color-InfraRed image of the alive cucumber plant

(b) RGB image of the alive cucumber plant



(c) Color-InfraRed image of the dead cucumber plant



(d) RGB image of the dead cucumber plant

Figure 4.2: Visualization of a plant using the CIR false color composite representation created from the multispectral information compared with the correspondent RGB image.

4.2.1 False colour composite images

Giving the high number of bands of a multispectral image, new possibility of representation, other than RGB, are available combining different bands. False colour composite images are image that does not maintain the original colour of objects because instead of the red, blue and green bands, other ones are used in the RGB representation.

To visualize vegetation, the most common false colour composite representation, called *Color-InfraRed* (CIR), has the following structure: the NIR band is used instead of the red one, the red band takes the position of the green one and the green band can be found instead of the blue one (NIRRB instead of RGB).

In this schema the parts belonging to plants appear with different tone of red depending on the characteristics of the plants themselves (figure 4.2a). This is due to the fact that we visualize the image as if it was a normal RGB image.

4.2.2 Vegetation indices

A non destructive way to gather information about the health, growth and stress of plants is computing Vegetation Indices (VIs). There are multiple indices computed using different combinations of spectral bands and their application can vary depending on the species of the plants. These indices were used mainly to monitor the vegetation from the satellites, for this reason a lot of variations take in account clouds and atmosphere sensibility.

Ratio Vegetation Index

The *Ratio Vegetation Index*, also called simple ratio vegetation index or ratio vegetation index, is one of the basic vegetation indices that employs the spectral behaviour of plants [18]:

$$RVI = \frac{NIR}{RED} \tag{4.1}$$

Regions corresponding to vegetation areas will have a RVI value larger than one and the value will depend on the type of vegetation and its health. Soil will have values close to one because its NIR reflection si similar to the red one while water and snow will have values smaller than one given that their red band reflectance is larger than the NIR band one.

This index in not bounded, so if there is a big difference of intensity between NIR and red component, it can assume very high values. For this reason normalized indices were introduced.

Normalized Difference Vegetation Index

One of the most widely used VI is the *Normalized Difference Vegetation Index* (NDVI) [19]. This index exploits the fact that healthy plants absorb visible light and reflect NIR band while unhealthy plants reflect less NIR light and more visible spectrum:

$$NDVI = \frac{NIR_{800} - RED_{680}}{NIR_{800} + RED_{680}} \tag{4.2}$$

NDVI can range from -1 to 1: it assumes a value close to 1 when a high density of green leaves is detected, 0 if no plant is detected and negative values if there is the presence of snow, water or clouds. Its behaviour is similar to the RVI one but it has the advantage to be normalise so it is easier to interpret its output.

This index is sensible to green vegetation.

Red Edge NDVI

Red Edge NDVI is similar to NDVI but it is more sensible to smaller changes in vegetation health:

$$Red\_Edge\_NDVI = \frac{RED_{750} - RED_{705}}{RED_{750} + RED_{705}} \quad [17] \tag{4.3}$$

Enhanced Vegetation Index

The *Enhanced Vegetation Index* corrects the distortion, present in the NDVI, that particles of air cause due to reflection.

$$EVI = G \times \frac{NIR - RED}{NIR + C_1 RED - C_2 BLUE + C_3} \tag{4.4}$$

where $C_1 = 6, C_2 = 7.5, C_3 = 1$ and $G = 2.5$ [20].

Kim in [21] used thirteen different VIs to study the evolution of five different sets of apple trees under water stress. Each set has a different water treatment: 100%, 90%, 75%, 60% and 45% of needed water over about two months. The used setup was composed by one RGB camera, two normalized difference vegetation index sensors and an hyperspectral camera with range $[385, 1000]$ nm with 5nm interval.

The author noticed that the two sets with least water (60% and 45%) showed an increase in the red band reflection and a decrease in the NIR with respect to the other sets. Kim identified NDVI and Red Edge NDVI as the indices with the highest correlation with water stress of the plants.

The conclusion of his work is that a difference in reflectance is found among water stressed and non-stressed plants even before some visible symptoms are present. This is of fundamental importance because it allows to act in advance and avoid damages on the plants and on the yield.

4.3 Phenotyping with depth information and 3D plant models

Usually plants have a very complex canopy so 2D images can only give limited information on their structure. For this reason 3D models of plants are used instead: they can give infor-

mation about the angle of the leaves, plant topology and can be useful if robots are used to prune them [22].

A stereo rig approach with two RGB cameras was used by Biskup et al. [23] to perform a partial 3D reconstruction of plants with triangulation. To do that the calibration parameters and the stereo calibration of the cameras were needed. The authors segmented the plant from the background by using the colour of the leaves in the *HSV* colour space performing a thresholding operation. They also segment the single leaves using graph-based segmentation algorithm combined with plane fitting and keeping only the segmented region with a certain area. A plane fitting approach was also used to compute the orientation of the leaves to study how it changes during the day and under water stress.

Santos et al. in [24] perform 3D reconstruction of plants from images acquired by a single camera that was moved producing a short baseline. SIFT [6] was used to find local invariant features to be employed in the SfM technique. Also calibration parameters were estimated during SfM. A region growing approach and multiple view stereo procedure were applied to reach a dense model. Ball pivoting algorithm was finally used to obtain a triangular mesh for the surfaces. To perform SfM Santos used *Bundler* [25] while in our work the reconstruction was performed using *COLMAP*: the latter has better performance than the former [4] and produces also the dense model as output.

An extension of this work can be found in [22]. Here the proper scale of the model is recovered using eight points of a known planar pattern placed on the pot of the plant. The plant was then segmented and clusters corresponding to leaves were produced with spectral clustering. In [26] Santos compute also plant height and leaves length. Moreover an computer-aided image acquisition was used to capture the optimal number of images moving the camera around the plant by hand.

In [27] McCormick et al. tried to perform different measurement on the 3D models of different *sorghum* plants acquired more than once in a 17 days interval. The 3D point clouds were created from 12 depth maps acquired by a time-of-flight camera and 12 RGB images taken as the sorghum rotated on a turning table. The 12 partial point clouds were than registered and registration errors were corrected manually. A mesh was generated from the point clouds. In this work the idea of the turning table is used: it allows to have a stable baseline among the images and a more structured and repeatable acquisition procedure.

The plants were then clustered: shoot and inflorescences were classified manually while leaves were segmented by using supervoxel adjacency.

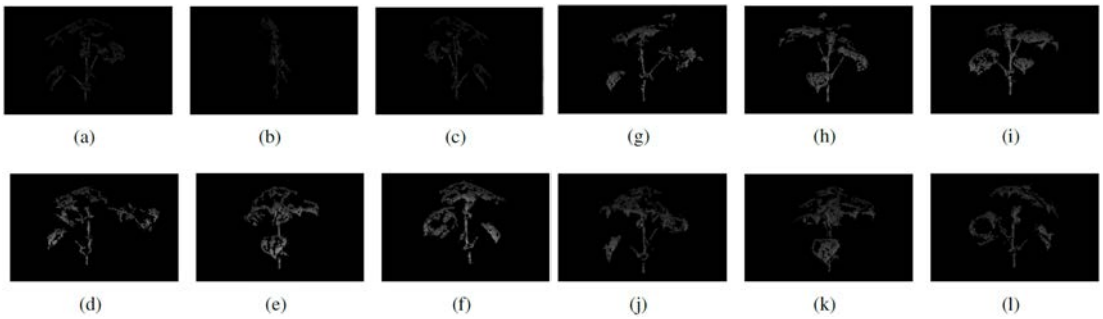


Figure 4.3: Views of point clouds of the same plant created using 600, 800, 850, 900 nm wavelength information [28].

From the 3D models, different measures were computed e.g. shoot height, shoot surface area, shoot centre of mass, leaves length, leaves surface area and leaf angle.

Liang et al. [28] created a 3D model directly from hyperspectral information obtained with a acousto-optical tunable filter. Consistent illumination conditions and a turning platform were used to facilitate the reconstruction procedure. An acquisition was performed by the hyperspectral camera for each three degrees of rotation of the turning table. The plant was segmented from the background and from the pot using Support Vector Machine (SVM): the classifier was trained on manually labelled images and then was used to classify parts related to the plant. SIFT keypoints were extracted on the output of Canny edge detection performed on the image to avoid the problem of insufficient features due to low resolution images.

One 3D model was built per each band (61 in total). A problem of Liang's approach is that one of the 3D model alone cannot reconstruct the entire plant (figure 4.3).

Differently from the authors of this work, we created a complete point cloud starting from RGB information and mapping the selected multispectral band or index on it. In this way more points are reconstructed.

Also Behmann in [29] underlines the importance that a hyperspectral 3D model can have in plant phenotyping and the problems that need to be solved to create one.

5
Reconstruction Approach

The work that was performed during the internship can be divided in four main parts:

- data acquisition,
- image processing,
- 3D reconstruction,
- 3D data processing.

Data acquisition, 3D reconstruction and data processing were the tasks that required more time, resources and effort.

Data acquisition implied the tuning of the setup and its automation. The generation of dense depth and normal maps during the 3D reconstruction was time consuming and required a GPU.

Finally the 3D data processing was a complex operation due to the high dimensionality of data belonging to the fused 3D reconstructions.

5.1 Acquisition pipeline

The optimization of the setup was of fundamental importance to have good data, so a good starting point, to perform all the further processing.

The procedure to perform an acquisition, described in section 3.4, generated three images (one per camera) for the 72 different positions of the object to have a complete $360°$ rotation. In the end 72 images per each camera were available. When we started to consider the temporal behaviour of the plants, this process was repeated 4 times per day, once every 6 hours.

5.2 Image processing

The acquisitions provided raw data for the multispectral and RGB images. From them it was possible to create the RGB and luminance images for the two bayer cameras and the luminance image for the multispectral one.

The luminance versions of the images were used both for calibration and pose estimation.

5.2.1 Camera calibration and pose estimation

The intrinsic parameters of the three cameras of the stereo rig were computed independently with the Zhang calibration method [3] using a $11x7$ chessboard pattern. In this way the camera matrix and the five distortion coefficients (k_1, k_2, p_1, p_2, k_3) were available.

The extrinsic parameters of the camera were estimated, using the intrinsic parameters, with the ChArUco board in figure 3.6. It was placed on top of the turntable and 72 acquisitions per camera, covering all the $360°$ range, were performed to have information about every camera position. The estimation can be done both without the object (more accurate result, figure 5.1) or with the object placed on the board, if it does not occlude the majority of the ArUco markers (figure 5.2).

Estimate the pose directly from the object acquisition was only possible with the fake plant and not with real ones: to water the plant during the acquisitions a saucer was needed and it covers most of the marker.

The rotation vectors were converted in quaternions because this is the structure of the rotation information required by COLMAP.

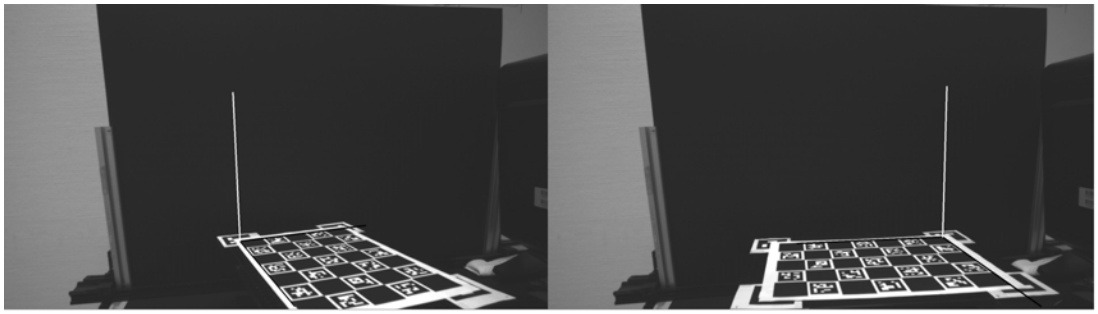


Figure 5.1: Pose estimation performed using ChArUco board without occlusions.

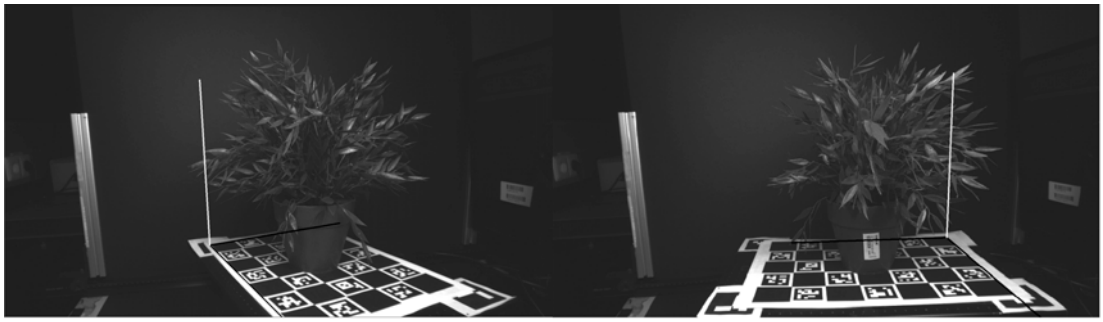


Figure 5.2: Pose estimation performed using ChArUco board with occlusions.

5.2.2 Undistortion and padding

Using the camera matrices and distortion coefficients obtained by the calibration, the images were undistorted. For the RGB and luminance images this procedure was performed using the OpenCV undistortion function directly on the images.

The multispectral images, in the first two dimensions, had a different size with respect to the images of the RGB cameras and the luminance ones of the multispectral camera (1920×1080 compared to 1952×1110). To have correct undistortion and alignment with the luminance images, a fixed padding was applied along the four borders of the multispectral images.

After that each channel was undistorted independently given the high dimensionality of the data.

5.3 3D model creation and processing

The 3D reconstruction performed by *COLMAP* was one of the most time consuming part of the procedure. To perform the bundle adjustment, the global bundle adjustment with the

stereo rig constraint and to build the sparse 3D model of a plant using 216 images (72 per camera), *COLMAP* took less than 2 minutes. To build the dense 3D model, i.e. the fused depth maps and normal maps with both photometric and geometric constraint were created, the time required by *COLMAP* was about 253 minutes. The final fusion procedure took about 4 minutes.

To perform the dense reconstruction procedure, we used a Nvidia GeForce GTX 1060 6GB. *MeshLab* [30] was used to visualize the point clouds.

5.3.1 COLMAP input and output

To have a correct reconstruction of the plant, four inputs were required:

- the 72 undistorted images for each camera,
- intrinsic parameters for each camera,
- pose estimation for each camera for all the 72 poses,
- a file containing the matching procedure.

This was due manly to the structured acquisition procedure performed with the turntable.

Input files

First of all the undistorted images obtained by processing the acquisitions, 216 in total (72 for each camera), were given as input.

Then the intrinsic and extrinsic parameters of the cameras were given to *COLMAP* in two different files. Given that COLMAP changes the size of the images while undistorting them (it crops images to avoid black padding due to undistortion) and as a consequence also of the depth maps related to the images, the distortion coefficients were fixed to 0. In this way the images considered by COLMAP were the same given in input, already undistorted, for all the three cameras and no manual padding had to be applied to find the correct alignment with the depth maps. The program was allowed to change only the provided pose estimations during the bundle adjustment procedure and no tuning was performed on the internal parameters of the cameras.

A file containing the matching pattern among images was also created taking in consideration the acquisition procedure. Let ${i_t}^{b1}$ be the t-th image taken with the first RGB camera,

${i_t}^{ms}$ be the same image but taken with the multispectral camera and ${i_t}^{b3}$ taken with the second RGB camera. For each camera, each image was matched with five images before and after it. The matches were searched in the interval $[i_{t-5}, ..., i_{t-1}, i_t, i_{t+1}, ..., i_{t+5}]$. Cross matches regarding the same image but taken with a different camera were also considered: ${i_t}^{b1}$ was matched both with ${i_t}^{ms}$ and ${i_t}^{b3}$ with $t \in [0, ..., 71]$.

This file was required to optimize the time and efficiency of the matching procedure: a lot of non visible images by the camera point of view were not considered so the number of wrong matches decreased.

Output files

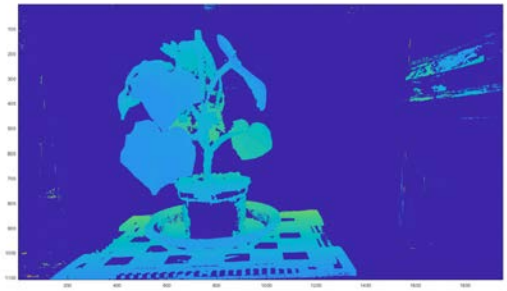
The outputs of *COLMAP*, that were used for the multispectral reconstruction, were the dense point cloud given in output by the dense fusion and fused depth and normal maps (figure 5.3). Moreover the refined pose estimations resulting from the bundle adjustment with the stereo rig constraint were considered: they are composed from quaternions to describe the rotation component and translation vectors.

5.4 Naive approach to create a multispectral point cloud

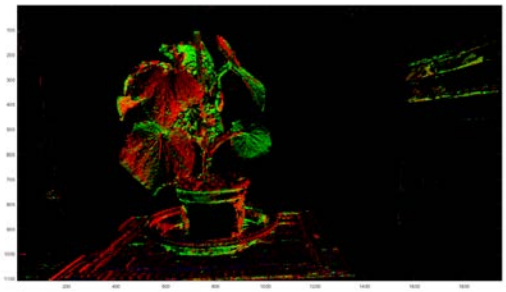
To generate a multispectral point cloud, the simplest approach was taking 4 depth maps correspondent to the multispectral images acquired at $0°$, $90°$, $180°$ and $270°$. From these information, 4 partial multispectral point clouds were created. Using the pose estimation given in output by *COLMAP*, we registered these point clouds and create a unique 3D model.

This approach is very simple but it does not take in account duplicate points. They can have different multispectral values because they are acquired by different angle with respect to the camera. Moreover if pose estimation is not too accurate, the different point clouds are not perfectly aligned so further registration procedures are needed. Another problem is that not all the points are reconstructed in this way.

To have a more precise 3D model, we used the spatial information coming from image and depth maps created considering one of the two RGB cameras. Using the stereo calibration parameters, we mapped the multispectral information from the multispectral images to the RGB ones and then on the point clouds. This approach leads to a problem: *COLMAP* perform the 3D reconstruction up to a scale factor. To perform this mapping, the scale was needed.

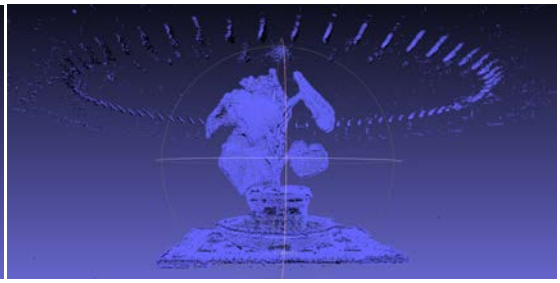


(a) Cucumber fused depth map

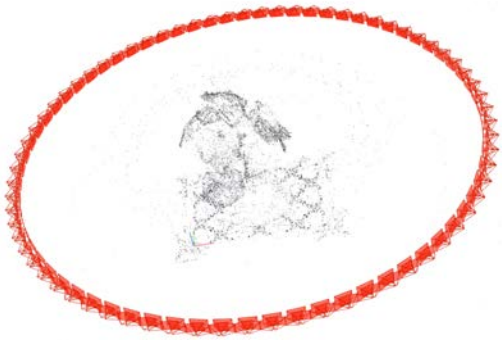


(b) Cucumber fused normal map

(c) *Cucumber dense point cloud*



(d) *Cucumber dense point cloud with normals*



(e) *Position of cameras plotted with COLMAP*

Figure 5.3: Fused depth map, normal map, dense point cloud (with and without normal visualization) and cameras pose estimation given as output by COLMAP.

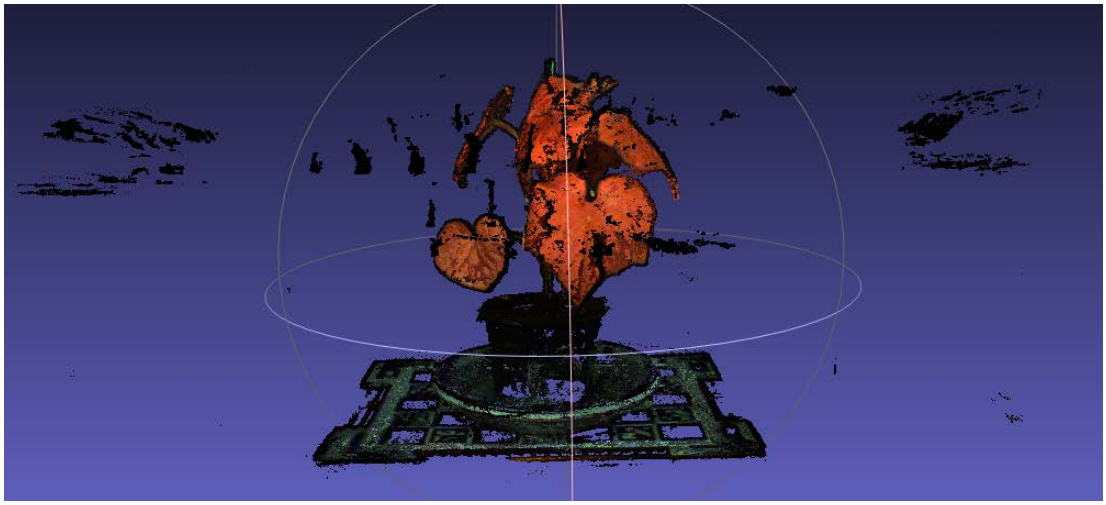


Figure 5.4: Superposition of four different point cloud with CIR mapping created from four depth maps.

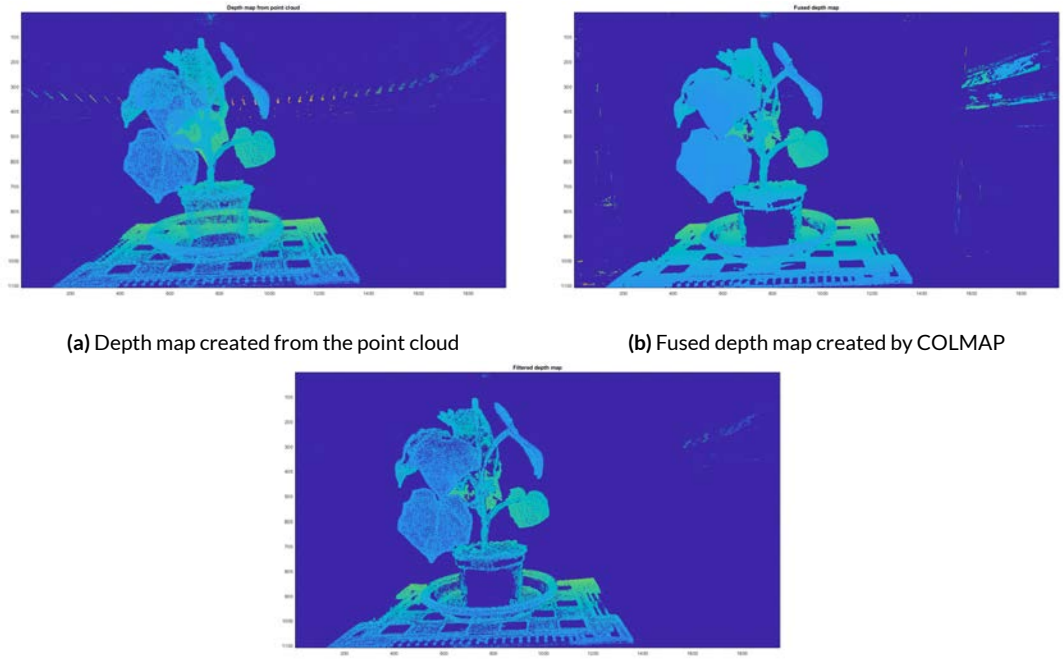
To recover it, we used the information coming from the stereo calibration and the rig constraint on the bundle adjustment of *COLMAP*. The actual baseline in mm was computed from the translation vector (t) given by the stereo calibration taking the RGB camera as reference:

$$baseline = ||\mathrm{t}|| \tag{5.1}$$

We perform a similar operation on the data coming from the pose estimation of *COLMAP*: we took the difference between the translation vectors for every pose of the multispectral camera and the ones of the selected RGB camera taken as reference. We compute the difference in rotation between the two using their respective quaternions. This difference was converted in a rotation matrix and we used equation 5.1 to compute the baselines used by *COLMAP* between the two cameras in different poses. To have an unique baseline, we took the average of all the baselines obtained with the aforementioned procedure (one per each pose). The scale factor was then recovered:

$$scale\_factor = \frac{calibration\_baseline}{COLMAP\_baseline} \tag{5.2}$$

The advantage of this approach is that the created point cloud is already scaled back to the original size of the plant. Also in this case we have duplicate points and the pose estimations need to be accurate. The resulting point cloud looks therefore noisy, figure 5.4.



(a) Depth map created from the point cloud

(b) Fused depth map created by COLMAP

(c) Filtered depth resulting from the comparison between (a) and (b)

Figure 5.5: Depth map creation and filtering

5.5 Information mapping

The best approach to have a robust measure of the multispectral values on the point cloud was mapping multispectral information coming from all the 72 multispectral images on the RGB point cloud resulting from the 3D *COLMAP* reconstruction. Given that a 3D point is reconstructed from more than one image, the multispectral values coming from different images had to be fused before this mapping.

5.5.1 Tracking of points in the images

The first step to map any information on the 3D model was understanding the correspondences among 3D points of the point cloud and 2D points of the images.

The tracking of the points was performed on one of the two RGB cameras ($b1$): the 3D points were projected on the RGB images and the valid ones were selected. Those were then projected into the multispectral images. In this way some precision was lost due to occlusions between the two cameras but we had more accuracy in the identification of the valid points:

the multispectral camera has a lower resolution with respect to the RGB ones, therefore also the depth maps connected to the multispectral images have less resolution.

Mapping from one single view

To create a map between the *COLMAP* point cloud in the world reference system and an image, the 3D points were brought in the camera reference system. The depth components were sorted and then projected on the image reference system creating a depth map (figure 5.5a). This depth map does not account for any visibility constraint: a lot of non visible points are displayed.

To filter out the wrong points, this depth map was compared to the one given as output by *COLMAP*:

$$valid\_depth = \frac{|cloud\_depth - fused\_depth|}{cloud\_depth} < t \tag{5.3}$$

where $cloud\_depth$ is the depth created projecting the depth components of the point cloud, $fused\_depth$ is the fused depth of *COLMAP* and $t = 0.0099999997764825821$ is the threshold used by *COLMAP* during the depth maps fusion procedure.

In this way the indices of the visible 3D points were found. These points are then projected to the multispectral camera reference system and to the multispectral image to assign them their correspondent multispectral values. We can see the result of the mapping of one view containing the CIR representation of the plant in figure 5.6.

This procedure was repeated for all the 72 images of the multispectral camera and a table containing the 3D points indices and their respective multispectral values was created.

Different types of multispectral information can be gatherd for each 3D point e.g. the NIR value, the NDVI value, the CIR representation or the NDVI mapped in the RGB domain.

5.6 Fusion

For each 3D point, different values coming from different images are available. To have an unique and more robust representation, a fusion procedure among these values is needed.

5.6.1 Fusion metrics

I considered four different metrics to fuse the information coming from the mapping procedure: mean, median, robust statistic and weighted average.

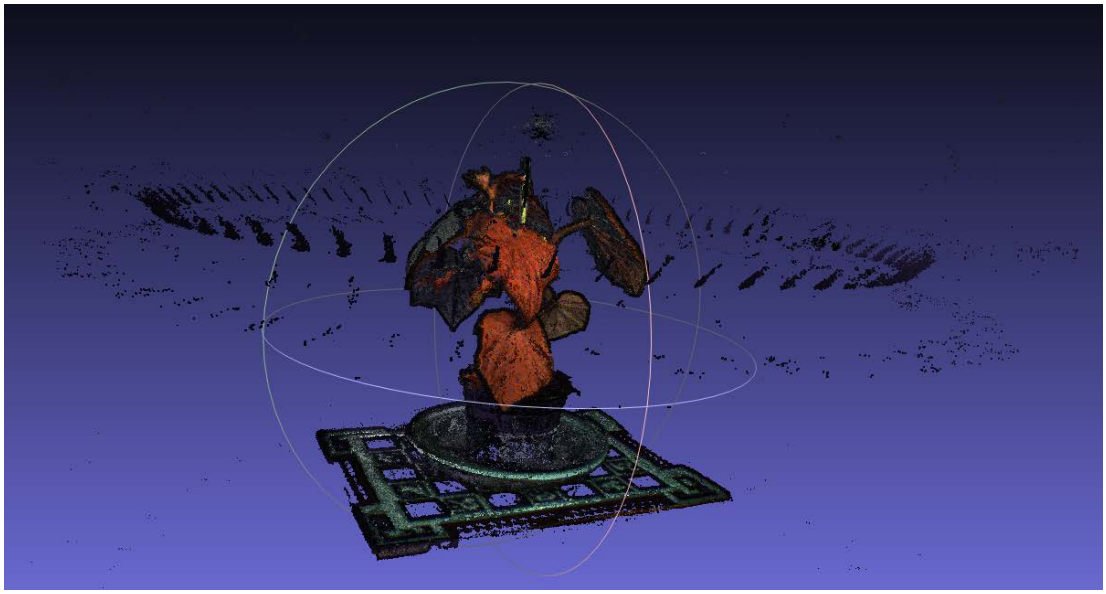


Figure 5.6: Mapping of CIR multispectral information coming from a single image to the point cloud.

Mean is the most simple of the four but its main disadvantage is that it is not robust against outliers while median is less affected by presence of outliers. In fact the breakdown point, so the largest number of outliers that can be tolerated before the model breaks, for the average is 0 while for the median is $\frac{1}{2}$ [31].

5.6.2 Weighted average

To compute a weighted average among the different values corresponding to the same 3D point, we decided to consider the normals information of the point compared with the direction of the camera.

We computed the camera normal taking the third row of the rotation matrix of the camera and compare it to the normal direction of each point in the image that was considered:

$$normal\_diff = normal\_pt \cdot -normal\_cam^T \tag{5.4}$$

where we did not consider the normalization factors because the norm of a normal vector is 1.

The angle between the two normals can be computed in radiants by:

$$\theta = \arccos(normal\_diff) \tag{5.5}$$

Then the conversion into degrees was performed.

We repeated this computation for all the 72 different poses and images. At the end of this procedure we had a difference value for each 2D point corresponding to a 3D point. These angles were normalized to be used as weight during the averaging process.

The weights are larger if the angle between the camera normal and the point normal is small. As this angle becomes bigger, the weights are smaller and smaller.

This can help to decrease the way in which outliers affect the final value associated with the 3D point: usually the most reliable values of the point are given when the point is facing the camera they will have the larger impact on the total value.

However in the weighted average we consider outliers, also if in a small part.

5.6.3 Robust statistic

Outliers can highly affect the accuracy of the final fused result. To overcome this problem, robust statistic can be used. Robust statistic aims detecting the outliers by searching for the model fitted by the majority of data [32].

In the algorithm we considered, algorithm 5.1, we start from an initialization value, the median, and iteratively update its value shifting it towards the majority of the data. The process is repeated until the $\triangle s$ converges to a specified value or if the number of iteration reaches a maximum fixed by the user. The convergence is however fast. In this way, assuming that the outliers are not the most of the data, we will have a final value that is only lightly affected by them. This because the weighting function will give a larger weight to the values close to the median or, after the first iteration, to its update values. So given that median is robust against outliers, we can assume that they will have smaller weights. Therefore they will affect less the updating procedure of the value.

Algorithm 5.1 Robust statistic

Input: s vector of values, max number of iteration

Output: robust estimation

1: $\underline{s} \leftarrow median(s)$
2: $threshold \leftarrow e^{-5}$
3: $\triangle s \leftarrow Inf$
4: $\sigma \leftarrow 8$ ▷ number of bits needed to represent the number
5: while $\triangle s >$ threshold or i > max number of iterations, do
6: $\quad err = \underline{s} - s$
7: $\quad w = \frac{1}{1+(\frac{err}{\sigma})^2}$ ▷ weights computation
8: $\quad \underline{s}\_up = \frac{\sum s \cdot w}{\sum w}$ ▷ value update
9: $\quad \triangle s = |\underline{s} - \underline{s}\_up|$
10: $\quad \underline{s} = \underline{s}\_up$
11: $\quad i++$
12: end while
13: return $\underline{s}$ ▷ robust estimation

5.7 Segmentation process

Given that in the 3D reconstruction some noise is present due to matches on the background and some wrong reconstruction of the background around the leaves, a segmentation technique was used to divide the plant information from the noise.

As explained in section 4.2, the plants have a peculiar spectral behaviour. Therefore we decided to use this property to perform segmentation.

5.7.1 Thresholding segmentation

Once the data fusion procedure was performed, we have one single values corresponding to the NIR representation per each metric. We were able to discard points belonging to the pot or to the background just performing a simple thresholding procedure on the NIR values.

5.7.2 Neighbourhood segmentation

To preserve more points related to plant information, we also threshold the NIR values considering the average and median of the neighbourhood of the point. This procedure required

Figure 5.7: 4 closest neighbours (in red) of the considered point (in white).

the identification of the neighbours in the 3D domain.

An exhaustive neighbours search among so many points is computational expensive so we decided to simplify the problem: the points were sorted using the height component. Then each point of the point cloud was considered and its four nearest neighbours were searched inside a window of 1000 points centred around that point 5.7.

Then the average and the median of the NIR values belonging to the point and to its neighbours were computed, in this way a more robust measure was obtained and used to perform the segmentation procedure.

6 Results

In this chapter the results obtained during my internship will be presented. Firstly we will introduce the 3D reconstruction and multispectral mapping results. Then the temporal information analysis will be presented

6.1 3D reconstruction

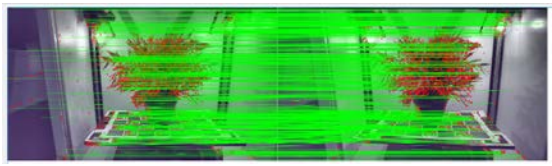
After the partial automation of the second setup (figure 3.1c and 3.1d), we started testing the 3D reconstruction procedure using COLMAP. To perform it, a plastic plant was used.

First of all the poses estimation was required otherwise false matches in the background would affect the capability of SfM procedure of directly estimating them. With a wrong or not accurate poses estimation, a precise reconstruction and mapping would not be possible, as can be seen in figure 6.1.

Using the poses obtained with the ChArUco board, COLMAP already had the initialization of the poses. It tuned them using the bundle adjustment and the rig bundle adjustment considering only the valid points, coherent with the poses.

With the introduction of the last setup, to solve the saturation and noise problem, and the injection of the poses, a precise 3D model of the fake plant could be created, figure 6.2.

To test our 3D reconstruction pipeline, we tried also to build a 3D model of an aloe plant: it is very complex to have a good reconstruction of this plant due to its structure and the

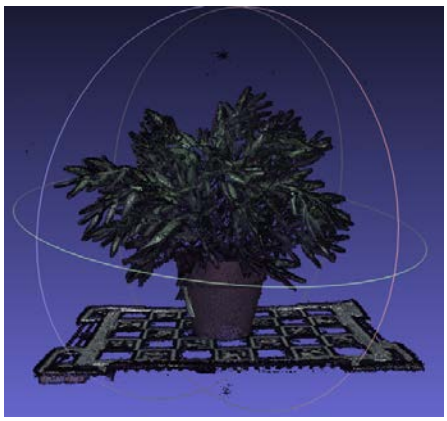


(a) Matches between a pair of images used to perform the reconstruction and the bundle adjustment procedure.

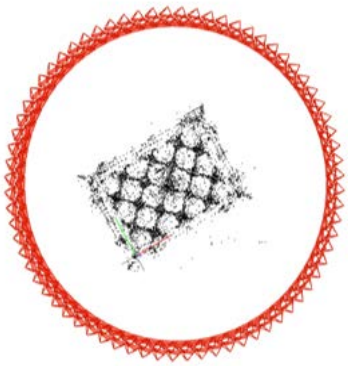
(b) Wrong poses estimation and reconstruction by COLMAP

Figure 6.1: Wrong poses estimation due to fixed matches belonging to the background.

(a) One of the images used for the reconstruction.



(b) Plastic plant 3D reconstruction



(c) Correct poses estimation

Figure 6.2: Correct 3D reconstruction thanks to the injection of the poses estimation in COLMAP.

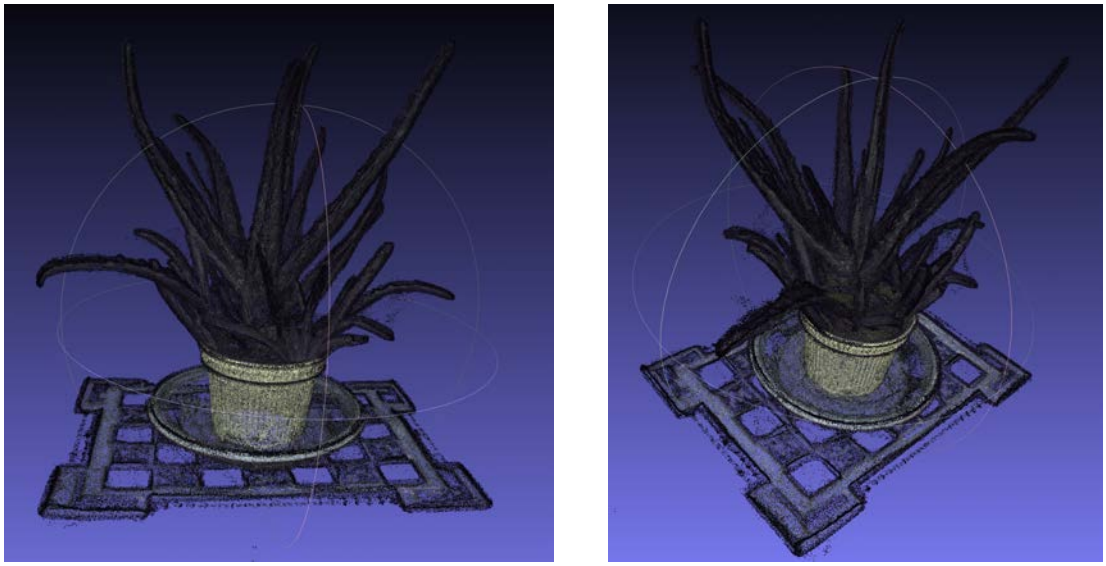


Figure 6.3: 3D model of the aloe plant that was used to test the reconstruction pipeline.

high number of occlusions given by the leaves. The result was however a quite precise reconstruction: thanks to the acquisition procedure, also the parts of the plant, that were occluded from one view, could be rebuilt using information given by another view 6.3.

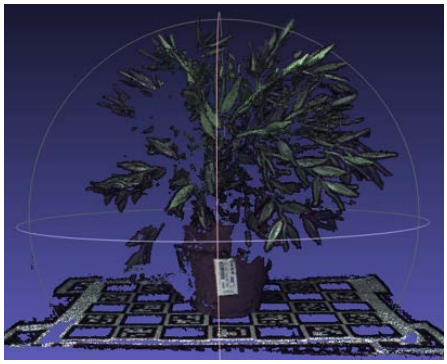
At this point we had a robust pipeline to perform 3D reconstruction performing acquisitions with our. The outputs of this process were one fused depth map and normal map for each input image, a dense point cloud and the refined poses estimation with the rig constraint, as the ones shown in figure 5.3.

6.2 Mapping procedure

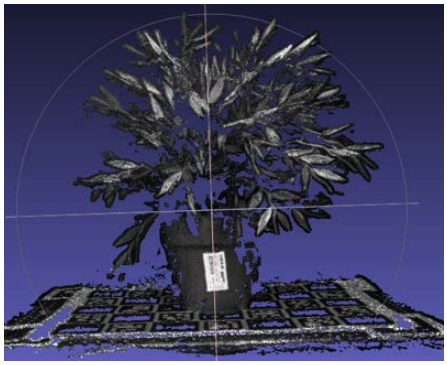
Thanks to the rig constraint, the scale of the point cloud can be recovered from the average of the 72 baselines provided by COLMAP.

Baseline values in mm						
Calibration baseline	82.346					
COLMAP baselines	2.2029	2.2017	2.2005	2.1992	2.1968	2.1975

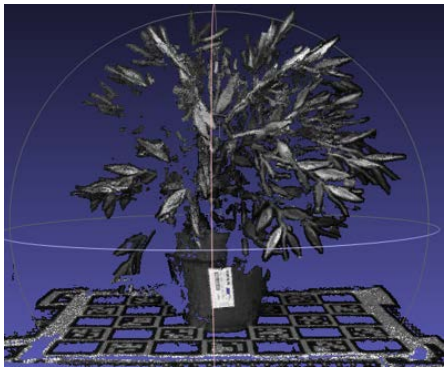
Table 6.1: Values of the baseline between the first RGB camera and the multispectral one coming from stereo calibration and some of the ones provided by COLMAP for the fake plant reconstruction.



(a) Partial point cloud created from image and depth map related to the first RGB camera



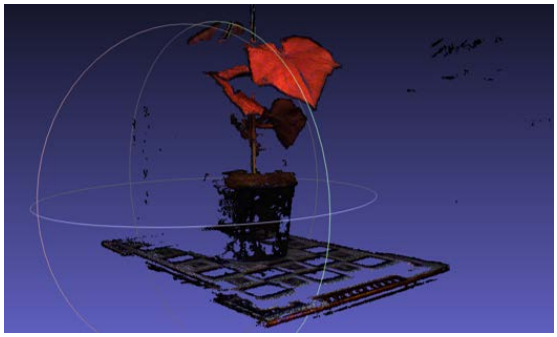
(b) Partial point cloud created from gray scale image and depth map related to the multispectral camera



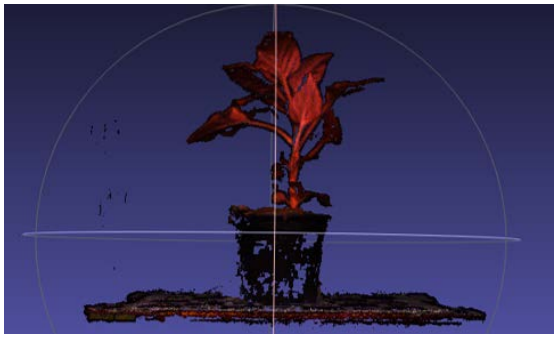
(c) Partial point cloud created from depth map related to the first RGB camera and image taken with the multispectral one

Figure 6.4: Information mapping from gray scale image of the multispectral camera to the partial point cloud created from depth information linked to the first RGB camera.

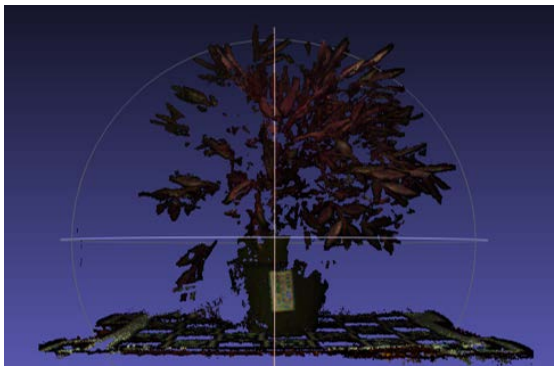
This allowed the mapping of information from the multispectral camera to a partial point cloud built using the fused depth map connected to a RGB camera. With a wrong scale factor, this information mapping was not effective because the values coming form the multispectral camera were mapped into the wrong points. Once this mapping was correct, mapping other values like wavelength informations, NDVI values or CIR colour map was a straightforward process. The difference of multispectral information connected with real and fake plant can be seen from the CIR representation obtained with the partial mapping.



(a) CIR representation of the cucumber plant



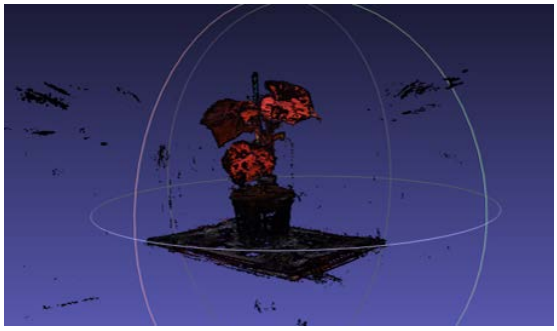
(b) CIR representation of the pepper plant



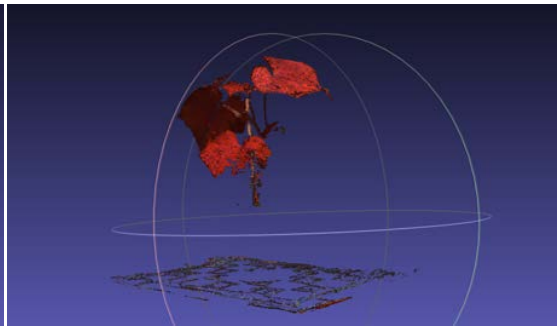
(c) CIR representation of the plastic plant

Figure 6.5: Comparison of the CIR representations of two real plants with the one of the fake plastic plant.

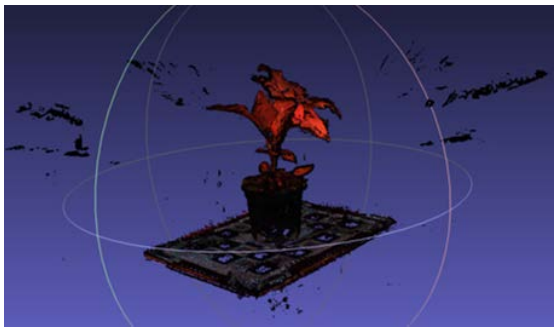
Superimposing four of these partial mappings, we obtained a complete point cloud of the plants. Performing a thresholding operation on the NIR values, we discarded all the points with $NIR < 800$. In figure 6.6, we can see that the information regarding the real plants are still preserved while the one of the fake plant are completely rejected. Therefore considering multispectral values, NIR band in particular, can help us to distinguish real plants from fake ones.



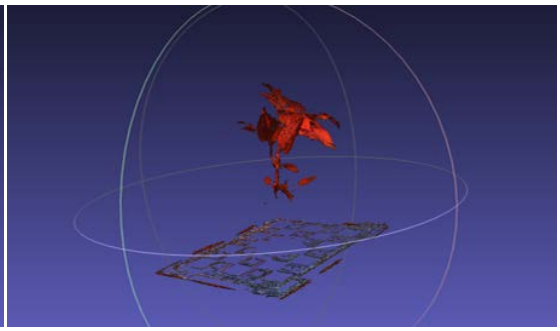
(a) Naive 3D reconstruction of the cucumber plant



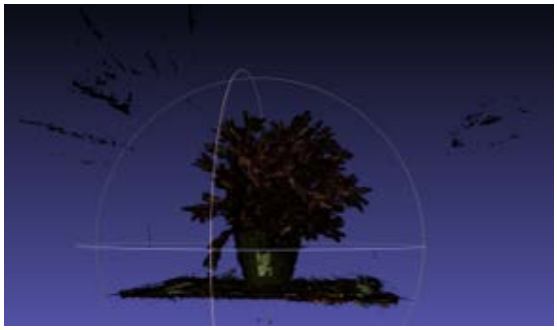
(b) Thresholding operation on the naive reconstruction of the cucumber plant



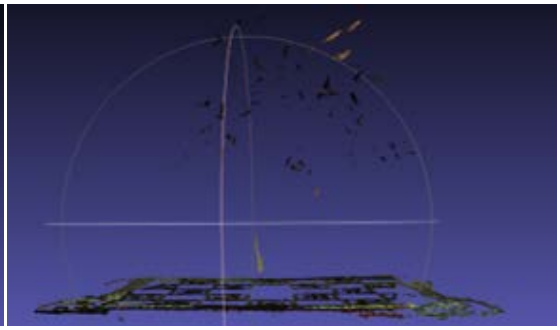
(c) Naive 3D reconstruction of the pepper plant



(d) Thresholding operation on the naive reconstruction of the pepper plant



(e) Naive 3D reconstruction of the plastic plant



(f) Thresholding operation on the naive reconstruction of the fake plant

Figure 6.6: Naive 3D multispectral reconstructions with CIR information and their corresponding thresholded version.

6.3 Acquisition over time

Our interest was to see how plants change during time, especially under water stress, both from their structure and from a spectral point of view. This acquisition over time was performed on two different real plants: the cucumber and the pepper one. Acquisitions on the cucumber plant were performed from the 28.06.2019 to the 22.07.2019 while the ones on the pepper plant were performed from the 22.07.2019 to the 19.08.2019, figure 6.7 and 6.8. The acquisition of one plant ended when it was dead.

6.4 Tracking of points and fusion procedure

The mapping of the multispectral values on the dense point cloud created with COLMAP was performed using data that were acquired during the temporal acquisitions of the real plants.

Performing the mapping on the dense point cloud, the number of points that were considered was higher than the one of the naive mapping, table 6.2. Some of them belong to the non useful part of the reconstruction (background, pot, turntable) but we have also to take in consideration that in the naive approach duplicate points are present.

Number of points in the point cloud		
Plant	Naive approach	Complete mapping
Cucumber	1545838	1819548
Pepper	1214389	1742578

Table 6.2: Number of points that were reconstructed in the 3D model using the naive approach and the one with tracking of points from the dense point cloud.

6.4.1 Fusion metrics and multispectral information mapping

Evaluating the quality of the four fusion metrics was not easy because we did not have a ground truth as reference. Therefore we based our analysis on the percentage of reconstructed points with an high NIR value and the visual result of the mapping (e.g. considering the CIR representation).

(a) Reconstruction from acquisition performed on 28.06.2019

(b) Reconstruction from acquisition performed on 09.07.2019

(c) Reconstruction from acquisition performed on 13.07.2019

(d) Reconstruction from acquisition performed on 16.07.2019

(e) Reconstruction from acquisition performed on 19.07.2019

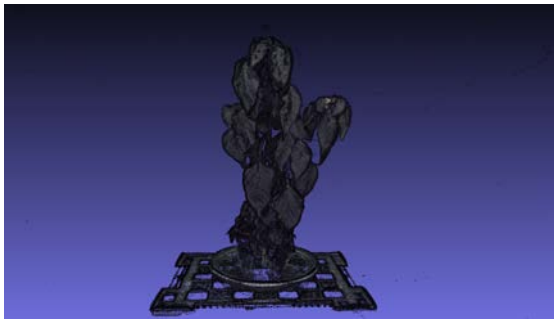
(f) Reconstruction from acquisition performed on 22.07.2019

Figure 6.7: Reconstruction of the cucumber plant over time.

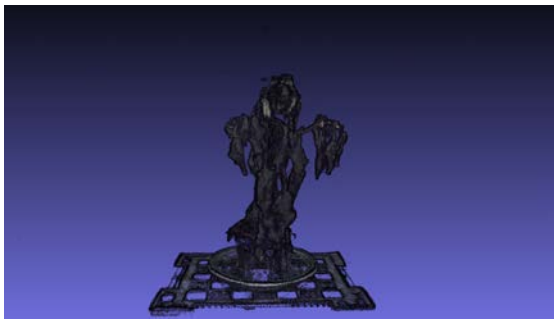
(a) Reconstruction from acquisition performed on 25.07.2019

(b) Reconstruction from acquisition performed on 01.08.2019

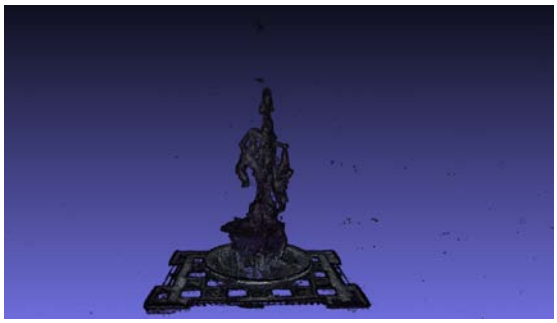
(c) Reconstruction from acquisition performed on 05.08.2019



(d) Reconstruction from acquisition performed on 09.08.2019



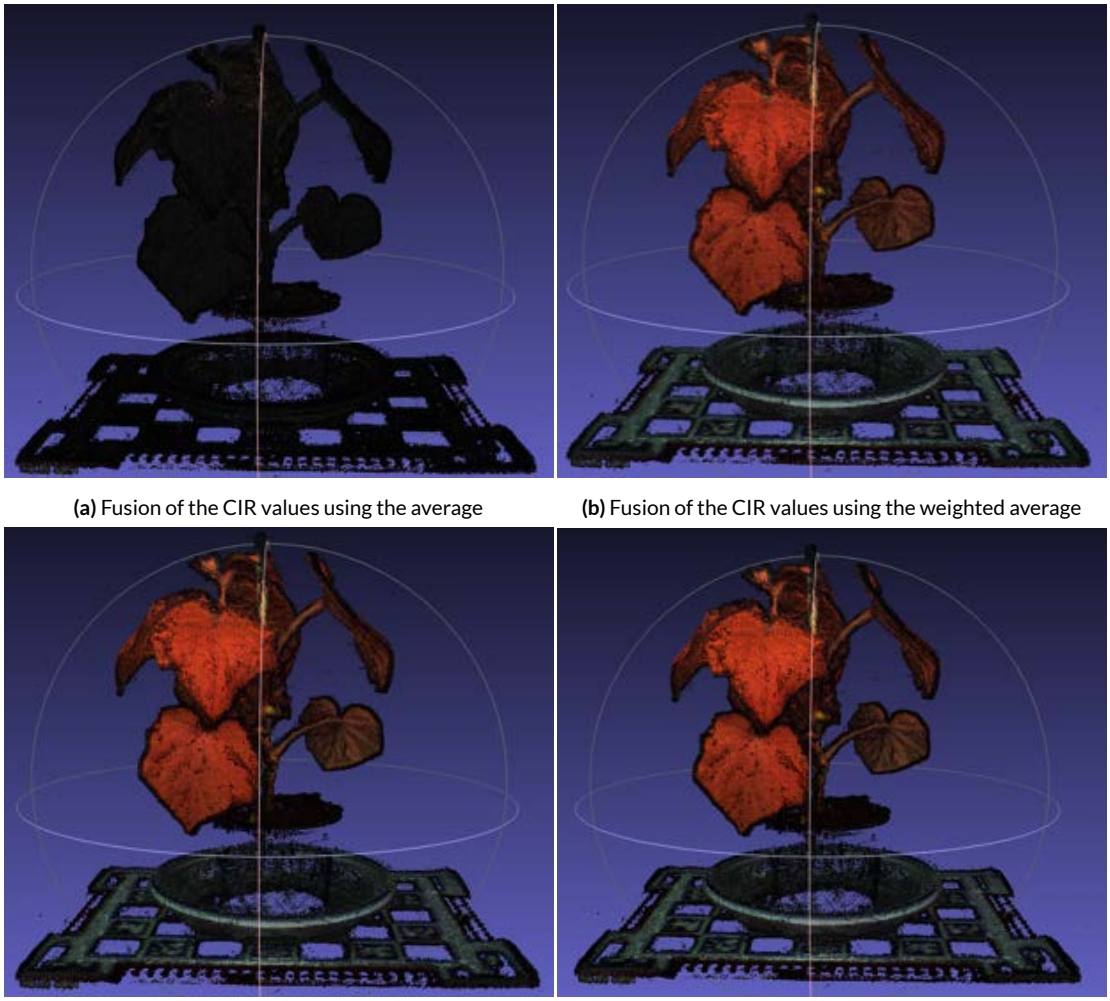
(e) Reconstruction from acquisition performed on 11.08.2019



(f) Reconstruction from acquisition performed on 19.08.2019

Figure 6.8: Reconstruction of the pepper plant over time.

As can be seen in figure 6.9, only the result obtained with the simple average is much worse that the others. This because the average is highly affected by outliers while the other metrics are more robust against them.



(a) Fusion of the CIR values using the average

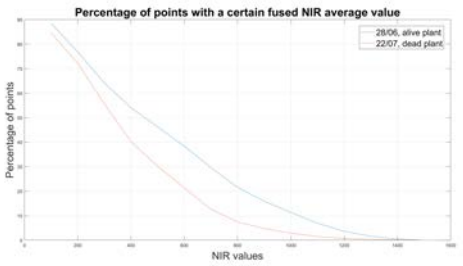
(b) Fusion of the CIR values using the weighted average

(c) Fusion of the CIR values using the median

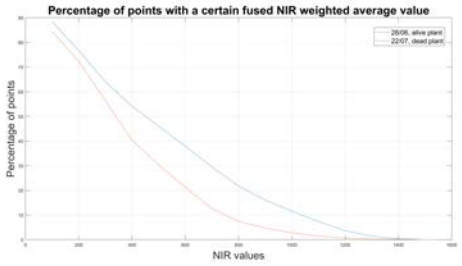
(d) Fusion of the CIR values using the robust statistic

Figure 6.9: Visual comparison of the values of the CIR representation fused using the four metrics described in section 5.6.

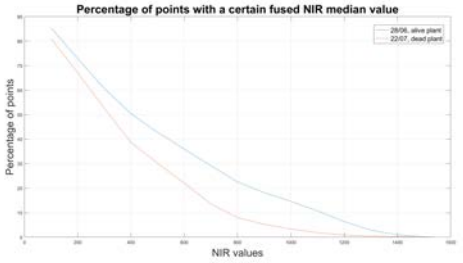
To see the differences among the other three metrics, the percentage of points having a certain NIR value was considered. The plots show how the percentage of points changes while a NIR value taken as threshold increases. In the plots the percentage of both the alive and dead cucumber plant are shown.



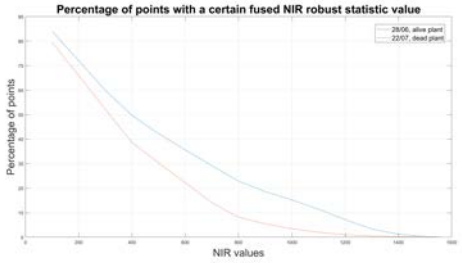
(a) Fusion of the CIR values using the average



(b) Fusion of the CIR values using the weighted average



(c) Fusion of the CIR values using the median



(d) Fusion of the CIR values using the robust statistic

Figure 6.10: Comparison of the NIR information fused using the four metrics described in section 5.6.

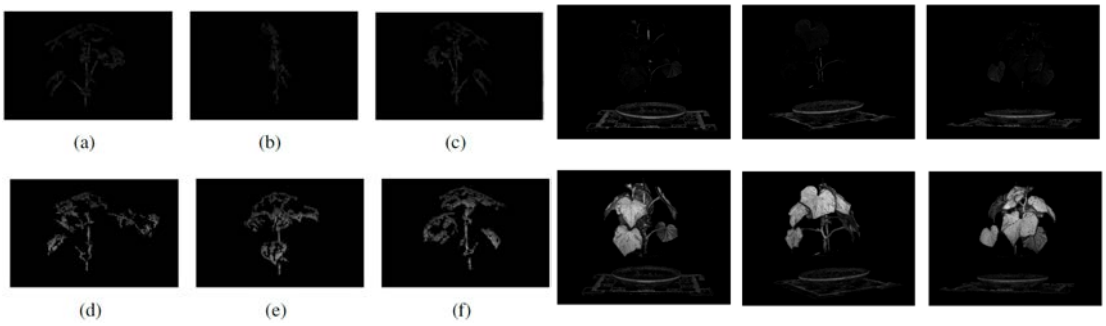
As we can see from this comparison, median and robust statistic are less affected by outliers so the percentage of points with a NIR value greater than 700 is larger than the one in the two average metrics. In fact weighted average considers outliers anyway, even if with a small weight.

After the fusion procedure we also mapped information related to vegetation indices (with and without colour map) and wavelength information.

To visualize the other mappings, we decided to use the robust statistic fused values given that it should be the less affected by outliers.

6.4.2 Wavelength mapping

Differently from the Liang's approach [28], we already had a complete point cloud on which we mapped multispectral information related to the wavelength, so for each band a complete reconstruction was possible. However we used a multispectral camera so the comparison of the result will be displayed only in the available common bands (600 and 800 nm).



(a) Liang's approach

(b) Reconstructions obtained using our approach by mapping wavelength information

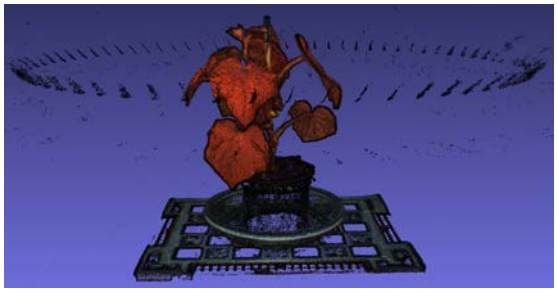
Figure 6.11: Comparison between the reconstruction of Liang and ours considering 600 and 800 nm bands.

The difference in the 800 nm reconstruction is quite visible: our approach allows to have a dense point cloud without merging different partial point clouds.

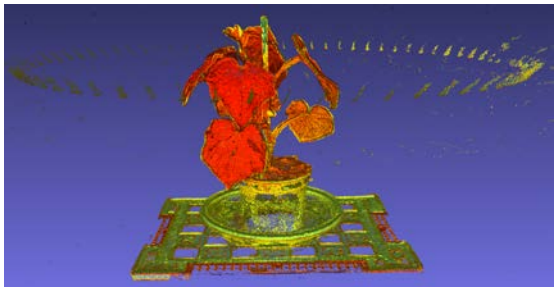
6.4.3 NDVI mapping and CIR mapping

To map the NDVI, a colour map is usually used: the NDVI values close to 1 are mapped to a red colour while low values, close to -1, to blue. This false colour map allows to highlight the presence of healthy vegetation. In figure 6.12 we can see the NDVI and CIR mapping on the cucumber and pepper 3D models.

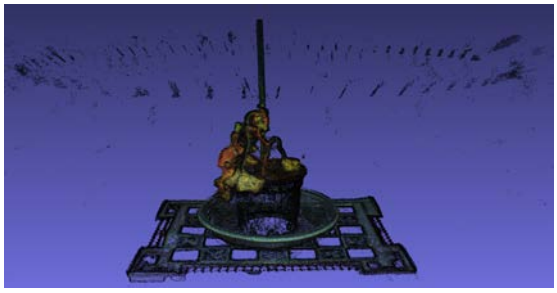
From the NDVI mapping, we can see clearly that the part of the point cloud associated with the plant has very high NDVI values while the pot, background and turntable have lower values (yellow is associated with values close to 0).



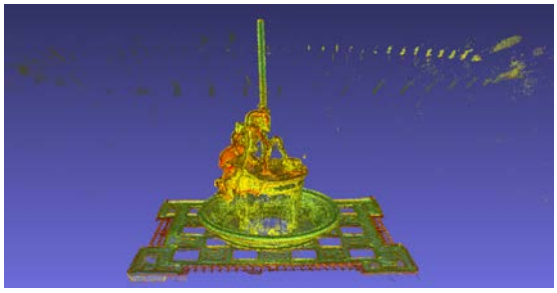
(a) CIR mapping on cucumber plant with acquisition of 28.06.2019



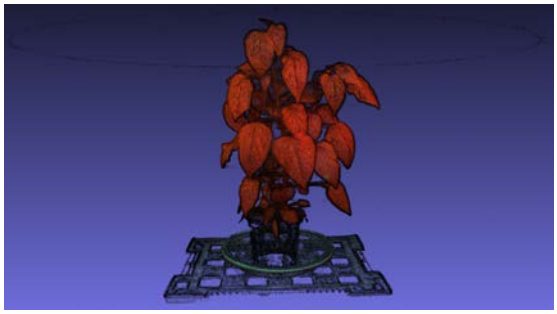
(b) NDVI mapping on cucumber plant with acquisition of 28.06.2019



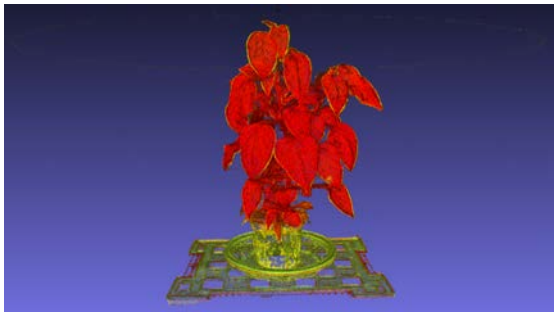
(c) CIR mapping on cucumber plant with acquisition of 22.07.2019



(d) NDVI mapping on cucumber plant with acquisition of 22.07.2019



(e) CIR mapping on pepper plant with acquisition of 25.07.2019



(f) NDVI mapping on pepper plant with acquisition of 25.07.2019

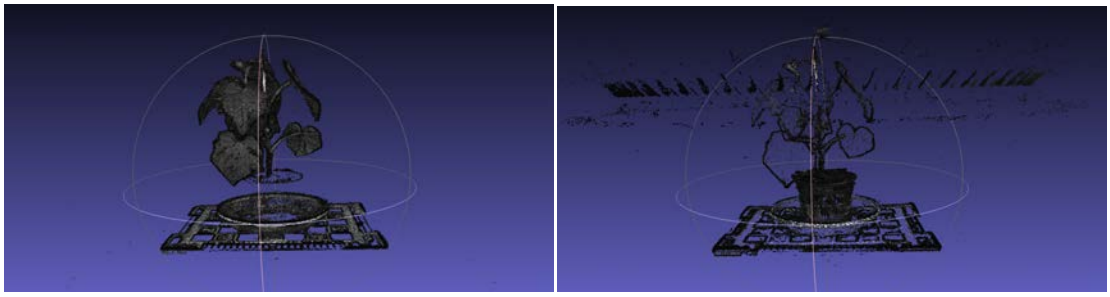
Figure 6.12: CIR and NDVI mapping in false colours on cucumber and pepper plants.

6.5 Segmentation result

To segment the plant from the background, we considered the NIR values for the naive mapping and the fused NIR values with robust statistic for the complete mapping. We decided to use robust statistic to preserve the highest number of points belonging to the plant.

6.5.1 Simple thresholding

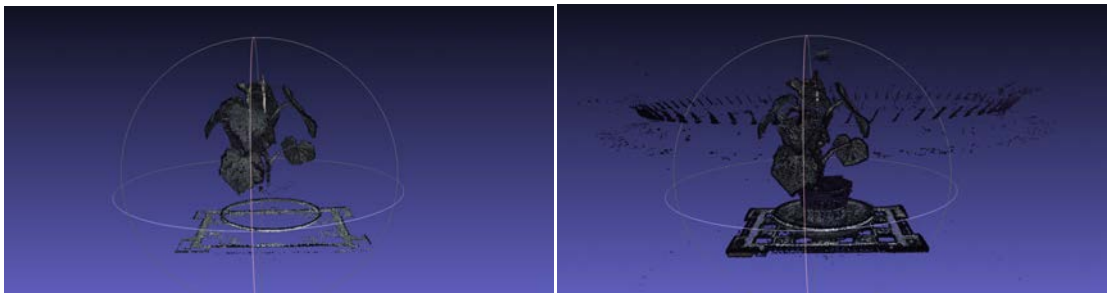
The thresholding operation on the NIR value is quite effective to separate the plant form the other element in the setup. As can be seen from figure 6.13, just using a low NIR threshold, $NIR > 150$, we are able to discard a lot of points belonging to the background while preserving the ones belonging to the plant. However most of the points belonging to the white part of the ChArUco board are not filtered out by the thresholding operation: this is due to the fact that white regions not only reflect the visible light but also the NIR band[33]. Therefore an higher threshold is needed to discard those points.



(a) Segmented point cloud **(b)** Discarded points

Figure 6.13: Segmentation of the plant performed with a threshold of 150 on the NIR robust fused value.

When using an higher threshold, $NIR > 600$, we start losing also points belonging to the plant, figure 6.14.



(a) Segmented point cloud **(b)** Discarded points

Figure 6.14: Segmentation of the plant performed with a threshold of 600 on the NIR robust fused value.

To have a fairer comparison between the naive reconstruction and the complete mapping, figure 6.15, the number of points of the point clouds segmented with NIR values will be considered. In this way we will consider mainly the points belonging to the plant, thus the most interesting for us.

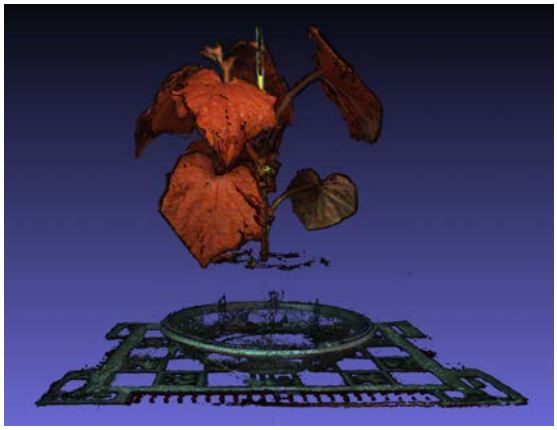
Number of points in the point cloud		
Plant	Naive approach	Complete mapping
Complete alive cucumber	1545838	1819548
Alive cucumber NIR > 150	1173432	1495161
Alive cucumber NIR > 500	697087	768570
Complete dead cucumber	838508	1155991
Dead cucumber NIR > 150	562674	908293
Dead cucumber NIR > 500	251065	349879
Complete alive pepper	1214389	1742578
Alive pepper NIR > 150	1040377	1458798
Alive pepper NIR > 500	598379	834429
Complete dead pepper	603719	910896
Dead pepper NIR > 150	445548	675493
Dead pepper NIR > 500	186327	260057

Table 6.3: Number of points that were reconstructed in the 3D model using the naive approach and the complete one over the time evolution of the cucumber and pepper plant considering also thresholds on the NIR band.

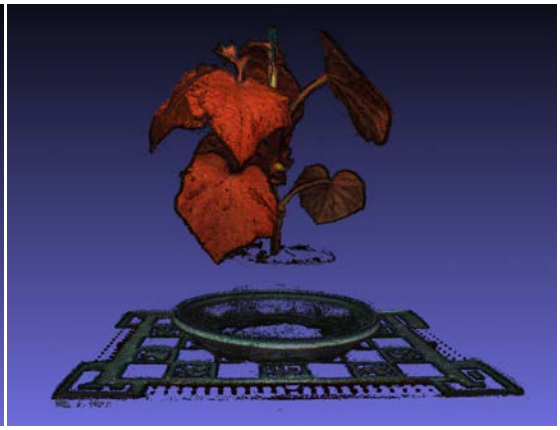
As we can see from table 6.3, the complete mapping approach preserves more points even if an high NIR threshold is used. Therefore the complete mapping gives us more complete information regarding the plant. We also have to consider the fact that in the naive approach duplicate points belonging to the plants are not discarded with the NIR thresholding operation.

6.5.2 Neighbourhood thresholding

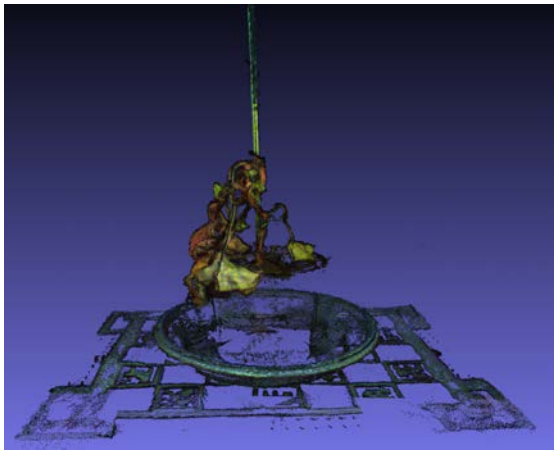
To try to preserve the most of the plant information, we used a thresholding operation that considered also the neighbourhood of the point. Here we considered the average of the neighbourhood NIR values (5 in total, the ones belonging to the 4 closest neighbours and one of the considered point) previously fused with the robust statistic. We considered the average because it discard less plant points with respect to the median, figure 6.16 with threshold $NIR > 600$ applied on the neighbours NIR value.



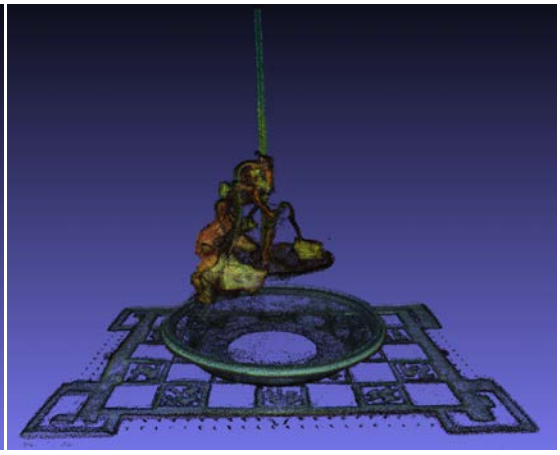
(a) Segmented naive point cloud with NIR > 150. Acquisition 28.06.2019



(b) Segmented complete point cloud with NIR > 150. Acquisition 28.06.2019



(c) Segmented naive point cloud with NIR > 150. Acquisition 22.07.2019



(d) Segmented complete point cloud with NIR > 150. Acquisition 22.07.2019

Figure 6.15: Comparison of the thresholding segmentation approach on the naive and complete mapping point cloud of the alive and dead cucumber plant.

(a) Simple thresholding, 1170854 points are discarded

(b) Discarded points using the average of the neighbourhood. 1123456 points are discarded

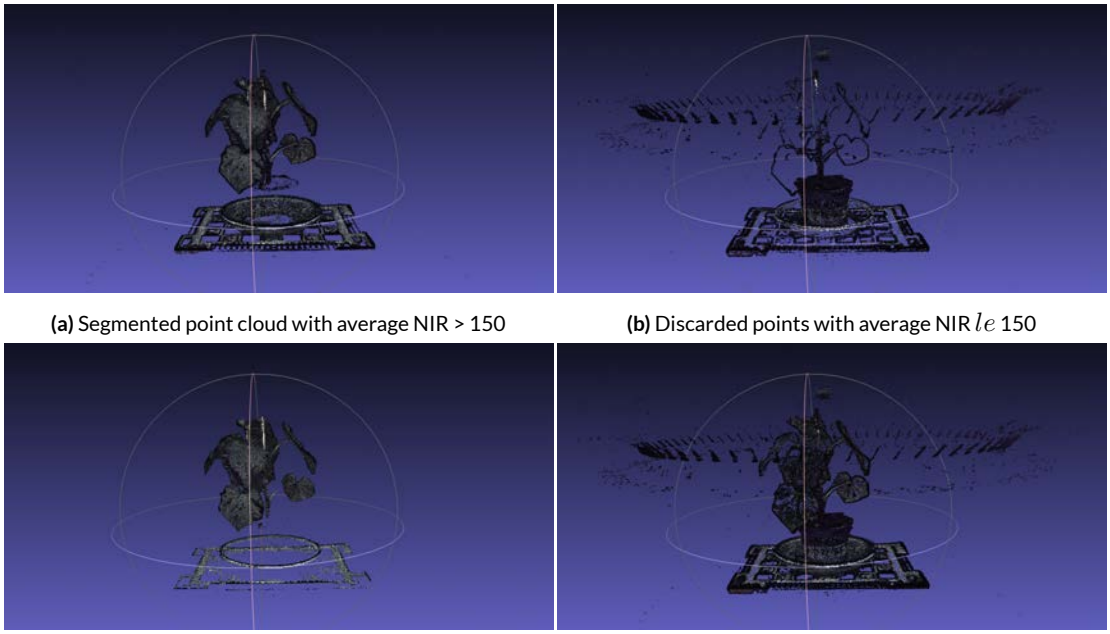
(c) Discarded points using the median of the neighbourhood. 1140921 points are discarded

Figure 6.16: Discarded points using NIR threshold of 600 on the simple robust NIR, on the average of the robust NIR of the neighbours and on the median of the robust NIR of the neighbours.

For low threshold of the NIR band, $NIR > 150$, we can say that the algorithm performs slightly worse with respect to the simple thresholding because it discards less noise. However when we consider high threshold values, $NIR > 600$, the neighbourhood approach preserves more plant points than the simple approach, see figure 6.17 and table 6.4.

Number of points in the point cloud	
Thresholding	Number of points
Simple threshold NIR = 150	1404696
Threshold NIR = 150 on neighbourhood	1433115
Simple threshold NIR = 600	648694
Threshold NIR = 600 on neighbourhood	696006

Table 6.4: Comparison of the segmented number of points obtained by simple thresholding and neighbourhood thresholding approach.



(a) Segmented point cloud with average NIR > 150

(b) Discarded points with average NIR le 150

(c) Segmented point cloud with average NIR > 600

(d) Discarded points with average NIR le 600

Figure 6.17: Segmentation of the plant using the average of the robust NIR values of the closest 4 neighbours of the considered point.

6.6 Water stress multispectral results

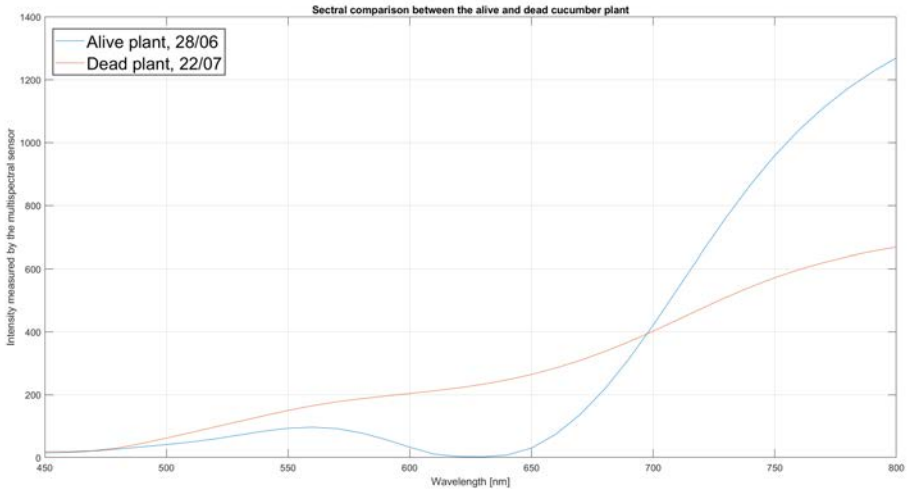
To consider the spectral evolution of the plant under water stress, we considered a 5×5 window centred around a point belonging to a leaf of the plant tracked manually in the different acquisitions. We then averaged the information coming from the considered pixels for each channel and we computed the Vegetation indices (VIs) on them.

From the spectral plots of the two plants, figures 6.18 and 6.19, we can see a difference in their evolution. The cucumber plant shows the expected behaviour: the NIR band reflection decreases while the red one increases. This is more evident in the period that goes from the 16.07.19 to the death of the plant because the leaves started to become reddish. This behaviour is reflected also from the plotted VIs: their values start to decrease when the plant starts to die. We can see that the value of the NDVI, figure 6.21, one of the easiest VIs to be interpreted, changes from 0.7071 to 0.3310, so from a value connected to an healthy plant to the value of a dead one.

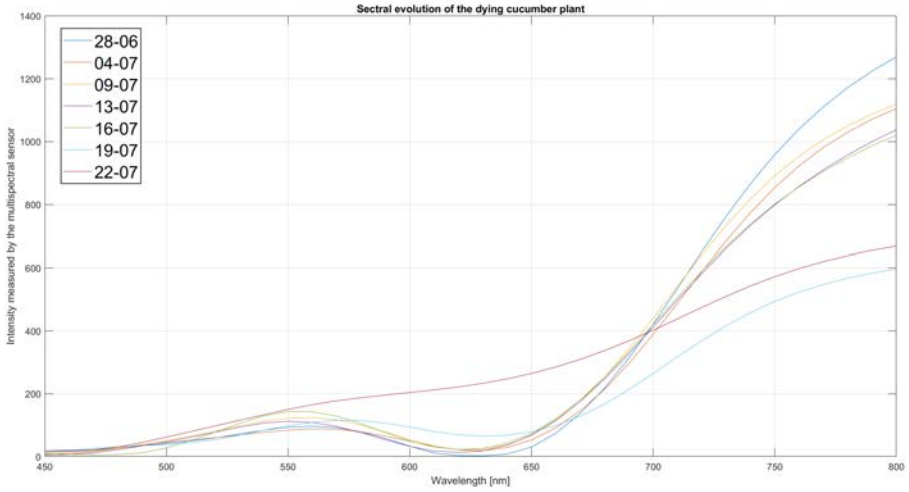
Using this information we can understand that the plant is under stress and act to avoid its death.

The pepper plant, however, shows a different evolution from the spectral point of view: the NIR reflection did not decrease so much as in the cucumber case and the reflection of the red band did not increase. This behaviour can depend on the plant species: the pepper plant did not become reddish while the plant was dying. They turned to a darker green. As a consequence of the spectral evolution, the VIs values did not change during the evolution of the plant status. The NDVI in this case stays inside the range $[0.6801, 0.7725]$ that can be considered as values related to a quite healthy plant.

In this last case, studying the evolution of a plant from the 3D point of view, e.g. the direction of the leaves and their area, can give complementary information to understand the health status of the plant. This can be done by e.g. clustering the leaves, so we know also their number, and compute their direction, figure 6.23. We can then study the changes of the canopy structure over time. The clustering procedure was applied on a filtered point cloud using fused $NIR > 600$ as a threshold. We used k-means++ to have a better initialization and we set the number of centroid to 9.

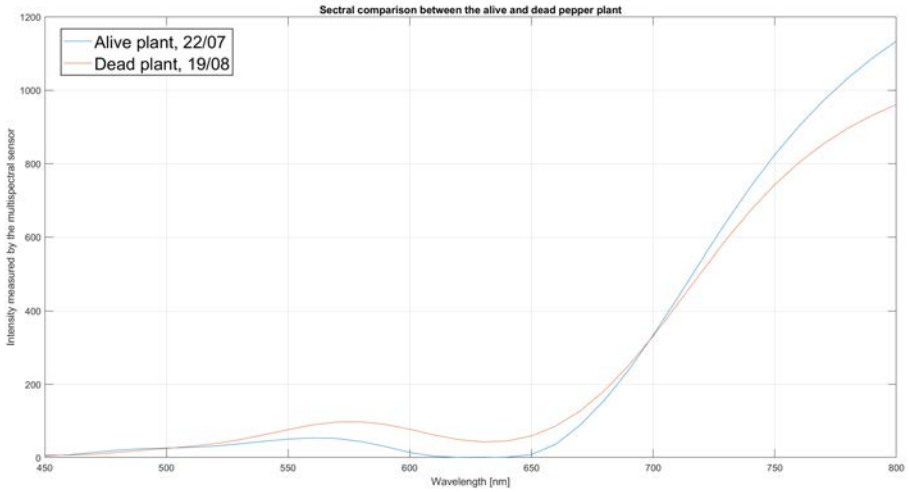


(a) Comparison of the spectrum between alive and dead cucumber plant

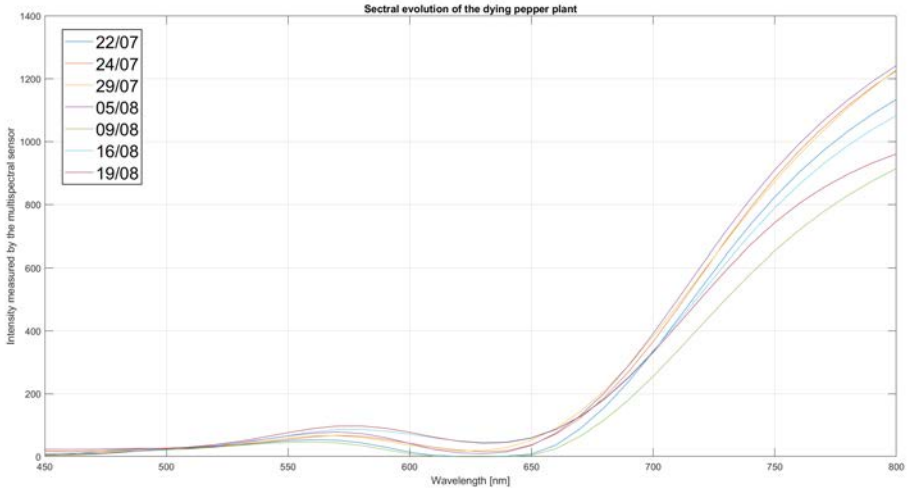


(b) Evolution of the spectrum of the cucumber plant

Figure 6.18: Evolution of the spectrum of the cucumber plant during the acquisitions over time.

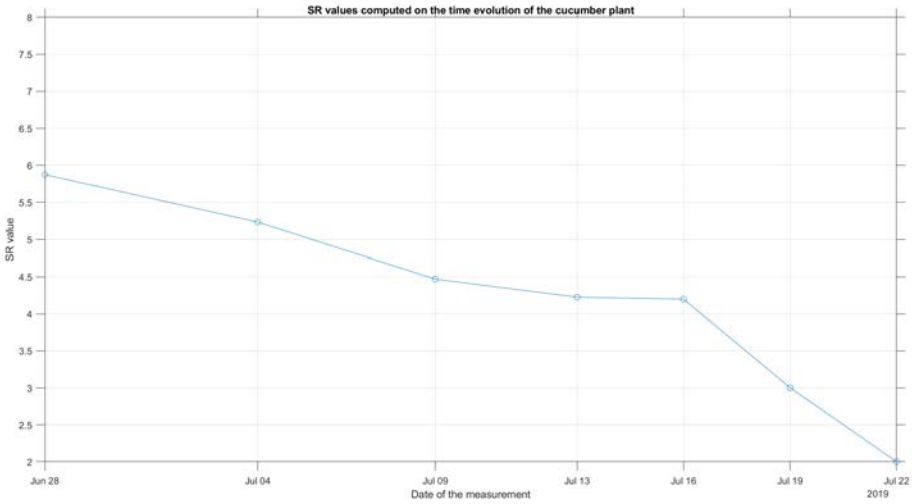


(a) Comparison of the spectrum between alive and dead pepper plant

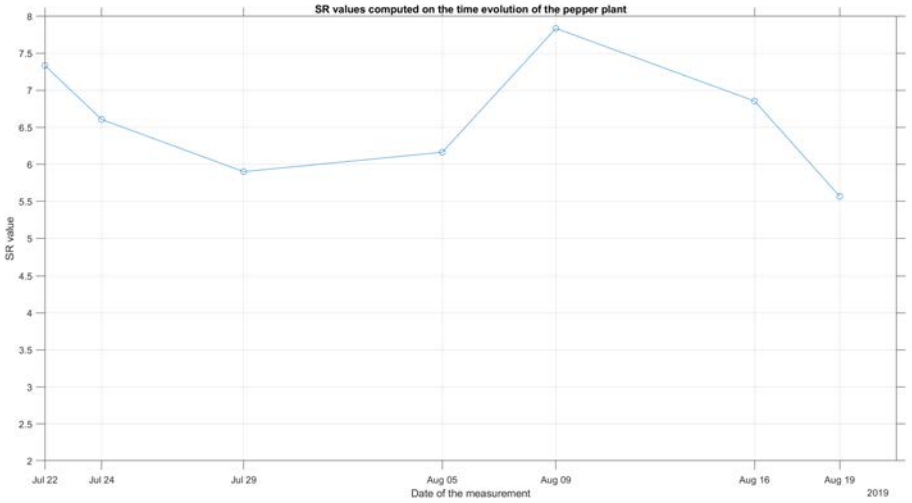


(b) Evolution of the spectrum of the pepper plant

Figure 6.19: Evolution of the spectrum of the pepper plant during the acquisitions over time.

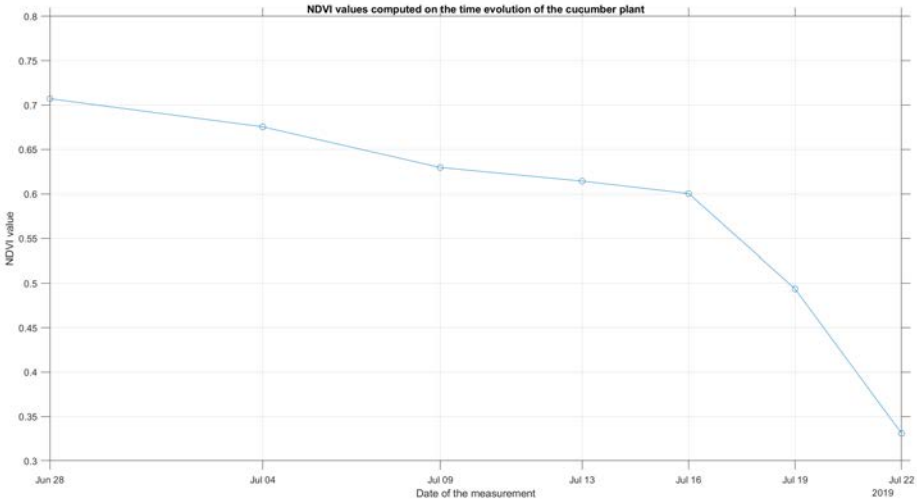


(a) Simple ratio index values of the evolution of the cucumber plant

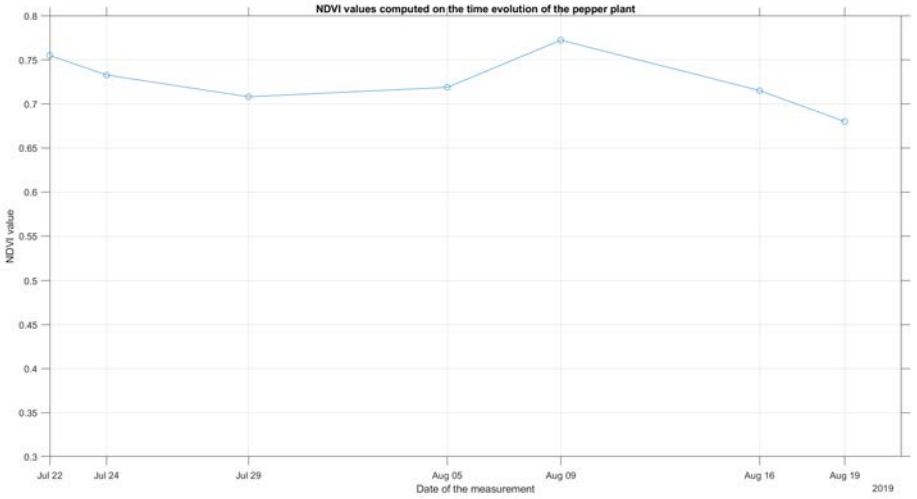


(b) Simple ratio index values of the evolution of the pepper plant

Figure 6.20: Comparison between the Simple Ratio evolution of the cucumber and the pepper plant.

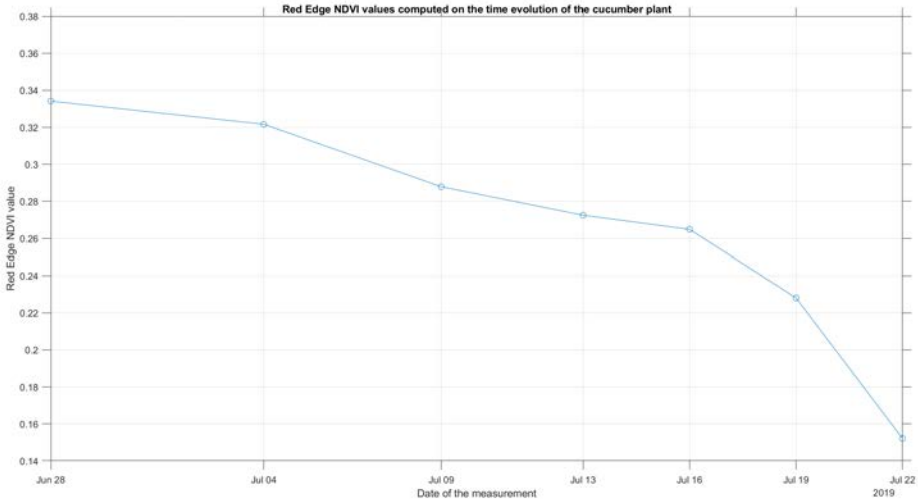


(a) NDVI index values of the evolution of the cucumber plant

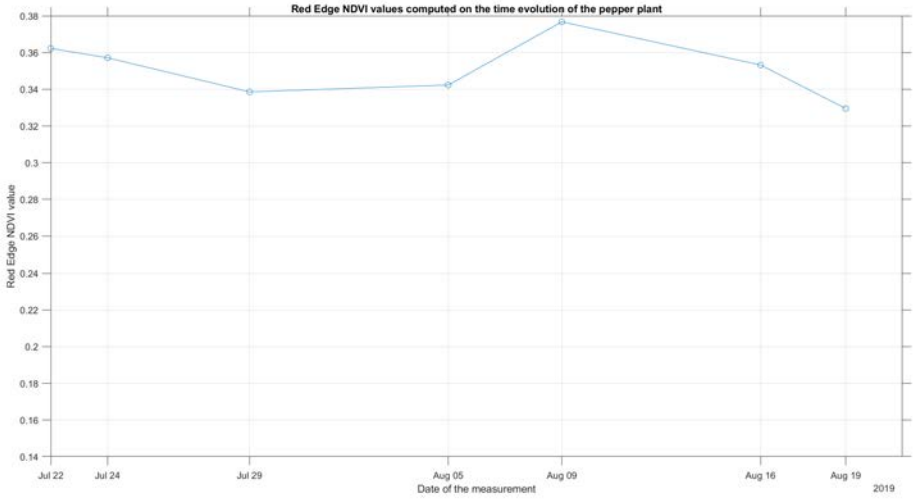


(b) NDVI index values of the evolution of the pepper plant

Figure 6.21: Comparison between the NDVI evolution of the cucumber and the pepper plant.

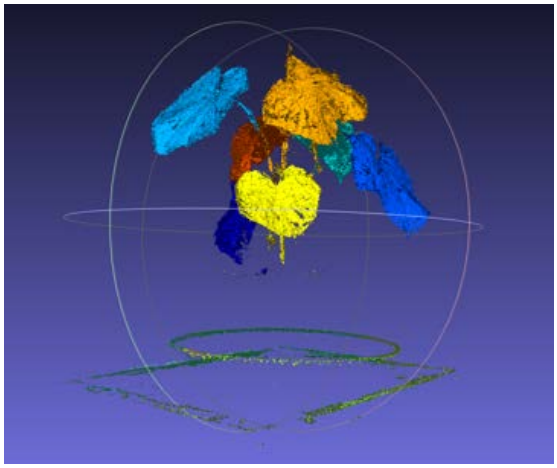


(a) Red Edge NDVI index values of the evolution of the cucumber plant

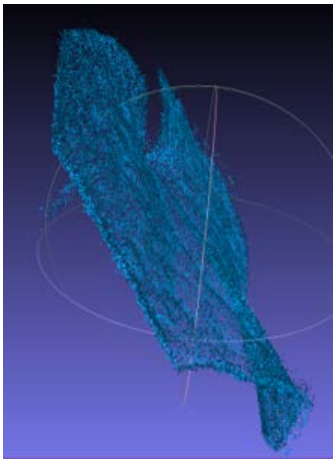


(b) Red Edge NDVI index values of the evolution of the pepper plant

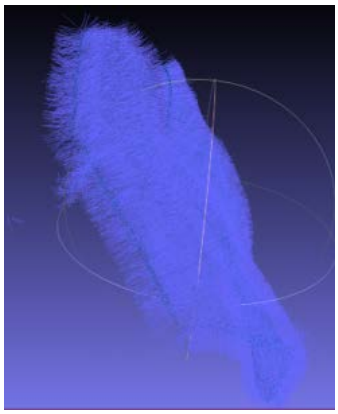
Figure 6.22: Comparison between the Red Edge NDVI evolution of the cucumber and the pepper plant.



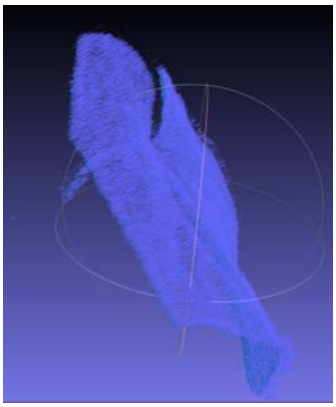
(a) Clustering of the filtered cucumber plant



(b) Cluster corresponding to a leaf



(c) *Normals of the point belonging to the leaf*



(d) *Direction of the leaf*

Figure 6.23: Leaf direction computed by averaging the normals direction of the points belonging to the leaf. The clustering procedure was performed using k-means++ with 9 centres.

7

Conclusions and future works

During my internship I developed a robust pipeline to build a 3D multispectral model to be used as basis for phenotyping. Starting from a camera rig and a turntable, an automatized setup was built to gather information regarding plants in different time instants. In this way the behaviour of the plant under water stress could be captured. The fusion of the multispectral information allowed us to have a more robust value for each of the point of the 3D model and exploiting them we could segment the plant information from noise and background data. In the end, from the multispectral information, we could study the changes in the NIR band reflection due to lack of water. Vegetation indices were also computed to have information about the status of the plant. One of the main advantage of our approach is that it gathers information in a non destructive way: it is therefore possible to acquire information regarding the same plant multiple time. In this way a complete monitoring of a plant during all its growth stages can be obtained.

The setup conformation conditioned the entire reconstruction procedure, especially the way in which we gave input data to COLMAP. It also introduce limitations to the generalization of the approach: some of them are linked with the 3D reconstruction process, other are connected to the plant. Our setup was placed in a controlled environment regarding both the light conditions and the weather ones. Wind and changes in illumination can strongly affect the precision during the matching procedure for the 3D reconstruction. Moreover a strong illumination can saturate the multispectral sensor so multispectral information are lost. The structure of the setup gives also a limit to the size of the plant if an accurate 3D

model is the aim of the reconstruction. Lastly depth fusion and dense 3D reconstruction required a lot of time and the use of at least one GPU.

Regarding future works, 3D information could be exploited more as complementary information to the multispectral one. Therefore a clustering procedure that is able to cluster correctly and automatically, without having priors, the complex structure of different plants has to be developed. Moreover, to reduce the reconstruction time and computational complexity, the number of images, needed to have an accurate reconstruction, can be optimized.

Listing of figures

Listing of tables

Listing of acronyms

BA	Bundle Adjustment
CFA	Color Filter Array
CIR	Color InfraRed
MSFA	MultiSpectral Filter Array
NDVI	Normalized Difference Vegetation Index
NIR	Near-InfraRed
SfM	Structure from Motion
VI	Vegetation Index

References

[1] A. Fusiello, *Visione computazionale: tecniche di ricostruzione tridimensionale*. FrancoAngeli, 2018, pp. 152–194.

[2] R. Hartley and A. Zisserman, *Multiple View Geometry in computer vision*. Cambridge University Press, 2003, pp. 152–194.

[3] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[4] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise view selection for unstructured multi-view stereo," in *European Conference on Computer Vision (ECCV)*, 2016.

[6] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, 11 2004.

[7] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, June 1981.

[8] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment – a modern synthesis," in *Vision algorithms: theory and practice, LNCS*. Springer Verlag, 2000, pp. 298–375.

[9] E. Zheng, E. Dunn, V. Jojic, and J.-M. Frahm, "Patchmatch based joint view selection and depthmap estimation," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '14. IEEE Computer Society, 2014, pp. 1510–1517.

[10] P.-J. Lapray, X. Wang, J.-B. Thomas, and P. Gouton, "Multispectral filter arrays: Recent advances and practical implementation," *Sensors (Basel, Switzerland)*, vol. 14, pp. 21 626–59, 11 2014.

[11] N. A. Hagen and M. W. Kudenov, "Review of snapshot spectral imaging technologies," *Optical Engineering*, vol. 52, no. 9, pp. 1 – 23 – 23, 2013.

[12] S. Garrido-Jurado and R. M.-S. et al., "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280 – 2292, 2014.

[13] J. Kumar, A. Pratap, and S. Kumar, *Plant Phenomics: An Overview*. Springer India, 2015, pp. 1–10.

[14] T. Furbank, Robert and M. Tester, "Phenomics – technologies to relieve the phenotyping bottleneck," *Trends in Plant Science*, vol. 16, no. 12, pp. 635 – 644, 2011.

[15] W. Johannsen, "The genotype conception of heredity," *The American Naturalist*, vol. 45, no. 531, pp. 129–159, 1911.

[16] C. Liew, S., "Principles of remote sensing," http://www.crisp.nus.edu.sg/~research/tutorial/rsmain.htm, 2001, [Online; accessed 5-August-2019].

[17] A. D. Sims, Daniel, and J. Gamon, "Relationships between leaf pigment content and spectral reflectance across a wide range of species, leaf structures and developmental stages," *Remote Sensing of Environment*, vol. 81, pp. 337–354, 08 2002.

[18] B. Abdou, D. Morin, and F. Bonn et al., "A review of vegetation indices," *Remote Sensing Reviews*, vol. 13, pp. 95–120, 01 1996.

[19] J. Weier and D. Herring, "Measuring vegetation (NDVI & EVI)," *Earth Observatory*, 2000.

[20] R. Colombo, D. Bellingeri, and D. Fasolini et al., "Retrieval of leaf area index in different vegetation types using high resolution satellite data," *Remote Sensing of Environment*, vol. 86, pp. 120–131, 06 2003.

[21] Y. Kim, D. Glenn, and J. Park et al., "Hyperspectral image analysis for water stress detection of apple trees," *Computers and Electronics in Agriculture - COMPUT ELECTRON AGRIC*, vol. 77, pp. 155–160, 07 2011.

[22] T. Santos, L. Koenigkan, J. Barbedo, and G. Rodrigues, "3d plant modeling: Localization, mapping and segmentation for plant phenotyping using a single hand-held camera," 09 2014.

[23] B. Biskup, H. Scharr, and U. Schurr et al., "A stereo imaging system for measuring structural parameters of plant canopies," *Plant, Cell & Environment*, vol. 30, no. 10, pp. 1299–1308, 2007.

[24] T. Santos and A. Oliveira, "Image-based 3d digitizing for plant architecture analysis and phenotyping," *Workshop on Industry Applications (WGARI) in SIBGRAPI 2012 (XXV Conference on Graphics, Patterns and Images)*, pp. 21–28, 01 2012.

[25] N. Snavely, "Bundler: Structure from motion (sfm) for unordered image collections," http://www.cs.cornell.edu/~snavely/bundler/, [Online; accessed 13-August-2019].

[26] T. Santos and G. Rodrigues, "Flexible three-dimensional modeling of plants using low- resolution cameras and visual odometry," *Machine Vision and Applications*, vol. 27, 11 2015.

[27] R. F. McCormick and S. K. Truong et al., "3d sorghum reconstructions from depth images identify qtl regulating shoot architecture," *Plant Physiology*, vol. 172, no. 2, pp. 823–834, 2016.

[28] J. Liang, A. Zia, and J. Zhou et al., "3d plant modelling via hyperspectral imaging," in *2013 IEEE International Conference on Computer Vision Workshops*, 12 2013, pp. 172–177.

[29] J. Behmann and A. K. Mahlein et al., "Generation and application of hyperspectral 3d plant models: methods and challenges," *Machine Vision and Applications*, vol. 27, pp. 611–624, 2015.

[30] P. Cignoni, M. Callieri, and M. Corsini et al., "MeshLab: an Open-Source Mesh Processing Tool," in *Eurographics Italian Chapter Conference*, 2008.

[31] F. Hampel, "Robust statistics: A brief introduction and overview," *Research report, Seminar für Statistik, Eidgenössische Technische Hochschule (ETH)*, vol. 94, 2001.

[32] P. Rousseeuw and M. Hubert, "Robust statistics for outlier detection," *Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery*, vol. 1, pp. 73–79, 01 2011.

[33] K. Dornelles and M. Roriz, "A method to identify the solar absorptance of opaque surfaces with a low-cost spectrometer," *PLEA 2006 - 23rd International Conference on Passive and Low Energy Architecture, Conference Proceedings*, 2006.

Acknowledgments

Firstly, I would like to thank Prof. Pietro Zanuttigh and Yalcin Incesu for their essential support and guidance throughout the duration of the internship and of the development of this thesis.

I would also like to express my sincere gratitude to Francesco Michielin and Piergiorgio Sartor for their valuable advices and for the time they dedicated to me.

Finally I am very thankful to my family and my friends. Their support was fundamental during the whole duration of my studies.