

# **Multi-modal Data Processing and Implementation for Vineyard Analysis**

Georgios Alexakis



University of Burgundy

A Thesis Submitted for the Degree of  
MSc in Computer Vision (VIBOT)

· 2020 ·

## **Abstract**

Nowadays, a major challenge of the agriculture sector is to feed the increasing global population with the least possible environmental impact, while retaining the natural resources for future generations. Agriculture's negative impacts on the environment are very serious, including pollution and degradation of soil, water, and air pollution. There is a strong public demand to reduce the negative impact of agriculture on the environment by using different approaches [44]. Furthermore, the usage of phytosanitary products such as fertilizers, fungicides, insecticides, and herbicides must drastically be reduced. So, Precision Agriculture (PA) as a management strategy that takes account of temporal and spatial variability to improve sustainability of agricultural production, is developing and is gaining more and more popularity during the years [9, 46, 75]. The purpose of this project is to analyze and inspect the viticulture site. Some analysis examples are temporal follow up of the vineyard, detection of plant diseases, or others. To make the analyses, it is necessary to detect several different types of information about the plants. From a multi-sensor system, such as a Lidar sensor, depth sensor, RGB camera, multispectral camera, and other modalities can be acquired valuable data. These data must be registered and then processed. Especially the main focus of the project is the development of a method for the registration of multi-modal images in order to obtain a three-dimensional reconstruction of the vine enriched with photometric or radiometric data. Furthermore, an artificial intelligence module is developed to jointly process images from the different modalities for a detailed analysis of the plant's condition. Finally, the combination of these parts is resulting in the final robotic system for vineyard analysis.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Inspiration . . . . .	1
1.2	State Of The Art . . . . .	2
1.3	Main Objectives . . . . .	3
1.4	Timetable . . . . .	4
<b>2</b>	<b>Background</b>	<b>5</b>
2.1	Principles . . . . .	5
2.1.1	Multispectral Imaging . . . . .	5
2.1.2	Multi-modal Data Fusion & Processing . . . . .	7
2.2	Hardware . . . . .	7
2.2.1	Overview . . . . .	7
2.2.2	Multispectral Cameras: Current Market . . . . .	8
2.2.3	Multispectral Cameras Comparison . . . . .	10
2.2.4	Microsoft Kinect V2 Sensor: RGB-D Camera . . . . .	12
2.3	Software & Technologies . . . . .	13
2.3.1	Overview . . . . .	13
2.3.2	ROS (Robot Operation System) . . . . .	14
2.3.3	OpenCV Library (Open Source Computer Vision Library) . . . . .	14

<b>3 Pre-processing</b>	<b>15</b>
3.1 Overview . . . . .	15
3.2 Multispectral Camera - Parameters . . . . .	17
3.2.1 Overview . . . . .	17
3.2.2 Implementation . . . . .	18
3.3 Multispectral Camera - Acquisition & Band Separation . . . . .	19
3.3.1 Overview . . . . .	19
3.3.2 Implementation . . . . .	22
3.4 Multispectral Camera - Flat-field Correction . . . . .	23
3.4.1 Overview . . . . .	23
3.4.2 Mathematical Equation . . . . .	24
3.4.3 Implementation . . . . .	24
3.5 Multispectral Camera - White Balance Normalization . . . . .	26
3.5.1 Overview . . . . .	26
3.5.2 Mathematical Equation . . . . .	26
3.5.3 Implementation . . . . .	26
3.6 Multispectral Camera - Crosstalk Correction . . . . .	28
3.6.1 Overview . . . . .	28
3.6.2 Mathematical Equations . . . . .	29
3.6.3 Implementation . . . . .	31
3.7 Cameras Geometric Calibration . . . . .	32
3.7.1 Overview . . . . .	32
3.7.2 Methodology . . . . .	33
3.7.3 Implementation . . . . .	34
<b>4 Processing</b>	<b>35</b>
4.1 Normalized Difference Vegetation Index (NDVI) . . . . .	35
4.1.1 Overview . . . . .	35

4.1.2	Mathematical Equations . . . . .	36
4.1.3	Implementation . . . . .	37
4.2	Multi-modal Image Registration . . . . .	39
4.2.1	Overview . . . . .	39
4.2.2	Multispectral & Kinect Cameras Registration . . . . .	39
4.2.3	Mathematical Equations . . . . .	40
4.2.4	Position of Sensors . . . . .	42
4.2.5	Image Registration with Feature Matching . . . . .	43
4.2.6	Image Registration with Corner Matching . . . . .	45
4.3	3D Reconstruction . . . . .	46
4.3.1	Overview . . . . .	46
4.3.2	Methodology . . . . .	47
4.3.3	Implementation . . . . .	47
4.4	Leaf Disease Detection using SVM Classifier . . . . .	48
4.4.1	Overview . . . . .	48
4.4.2	SVMs (Support Vector Machines) . . . . .	49
4.4.3	Methodology . . . . .	49
4.4.4	Implementation . . . . .	49
4.5	Conclusion . . . . .	52
<b>5</b>	<b>Results</b>	<b>53</b>
<b>6</b>	<b>Conclusions</b>	<b>61</b>
	<b>Bibliography</b>	<b>68</b>

# List of Figures

1.1	Project Timetable	4
2.1	Electromagnetic Spectrum Illustration [73]	6
2.2	CMS-V GigE Silios Multispectral Camera [5]	9
2.3	Microsoft Kinect V2 Sensor [74]	12
3.1	Project Pipeline	16
3.2	ROS rqt_reconfigure Tool for Camera Parameters	18
3.3	Web-based Camera Controller [1]	19
3.4	Typical RGGB Bayer Mosaic for Color Image	20
3.5	Colored Band Arrangement by CMS-V Multispectral Camera	21
3.6	Raw Acquired Image by Multispectral Camera in ROS - OpenCV Interface	22
3.7	9 Band Images after Band Separation in ROS - OpenCV Interface	23
3.8	Flat-field Image Captured by Multispectral Camera	25
3.9	Dark-field Image Captured by Multispectral Camera	25
3.10	Acquired Image of Band 1 Before Flat-field Correction	25
3.11	Acquired Image of Band 1 After Flat-field Correction	25
3.12	White Reference - Rules Illustration of Pixels' Selection	27
3.13	White Reference - Pixels Selection by User Interaction with ROS and OpenCV	27
3.14	Acquired Image of Band 8 Before White Reference Normalization	28
3.15	Acquired Image of Band 8 After White Reference Normalization	28

3.16 Crosstalk Effect Illustration (bottom rectangles represent the sensor pixels, the top curved rectangles represent the sensor filter, the arrows represent the incoming light) . . . . .	28
3.17 Bands Spectrum Before Crosstalk Correction [64] . . . . .	29
3.18 Bands Spectrum After Crosstalk Correction [64] . . . . .	29
3.19 Acquired Image of Band 8 Before Crosstalk Correction . . . . .	31
3.20 Acquired Image of Band 8 After Crosstalk Correction . . . . .	31
3.21 Distortion Effect Illustration [49] . . . . .	32
3.22 Illustration of pinhole camera model with the geometric relation between a 3D point and its 2D projection [63] . . . . .	33
3.23 Flow-chart of pipeline for the projection of a 3D point to a 2D image plane with radial lens distortion . . . . .	33
 4.1 NDVI Illustration [43] . . . . .	36
4.2 NDVI Color-map Illustration . . . . .	37
4.3 NDVI Normalized Image (with values from 0 to 255) . . . . .	38
4.4 NDVI Colored Image by Using Custom Color-map . . . . .	38
4.5 Illustration of a planar surface and the image plane [48] . . . . .	41
4.6 Multispectral and Kinect Cameras Fixed Position . . . . .	42
4.7 Registration with feature matching in ROS and OpenCV interface - Multispectral image (left), kinect image (right) . . . . .	44
4.8 Kinect RGB image after homography application - Real-time result (left), best result (right) . . . . .	44
4.9 Registration with chessboard corner matching in ROS and OpenCV interface - Multispectral image (left), kinect image (right) . . . . .	46
4.10 Kinect RGB image after homography application - Real-time result (left), best result (right) . . . . .	46

4.11	3D reconstruction of an outdoor scene with band 3 of multispectral camera by using rtabmap_ros package . . . . .	48
4.12	Leaf image in RGB color-space . . . . .	51
4.13	Leaf image in HSV color-space . . . . .	51
4.14	Mask for diseased sections of the leaf . . . . .	51
4.15	Diseased sections in RGB color-space of the leaf . . . . .	51
4.16	Gray-scaled diseased sections of the leaf . . . . .	51
5.1	Raw Acquired Image By Multispectral Camera . . . . .	53
5.2	Raw Acquired Image By Kinect RGB Camera . . . . .	53
5.3	Sub-Bands After Separation . . . . .	54
5.4	Multispectral Camera - Band 8 . . . . .	54
5.5	Multispectral Camera - Band 1 . . . . .	54
5.6	Multispectral Camera - Band 8 Before Flat-field Correction . . . . .	55
5.7	Multispectral Camera - Band 8 After Flat-field Correction . . . . .	55
5.8	Multispectral Camera - Band 8 Before White Reference Normalization . . . . .	55
5.9	Multispectral Camera - Band 8 After White Reference Normalization . . . . .	55
5.10	Multispectral Camera - Band 8 Before Crosstalk Correction . . . . .	56
5.11	Multispectral Camera - Band 8 After Crosstalk Correction . . . . .	56
5.12	NDVI 8bit Image (After normalization to 0 - 255) . . . . .	56
5.13	NDVI Colored Image (Using custom color-map) . . . . .	56
5.14	NDVI Colored Image (After crosstalk correction) . . . . .	56
5.15	NDVI Color-map . . . . .	56
5.16	Background subtraction by using Otsu's method . . . . .	57
5.17	Background subtraction after erosion and dilation . . . . .	57
5.18	Image Registration with Feature Matching . . . . .	57
5.19	Image Registration with Chessboard Corner Matching . . . . .	58
5.20	3D Image (Early Experiments) - Rviz . . . . .	58

5.21	3D Image (Final Result) - Rviz . . . . .	58
5.22	Point Cloud Generation - Rviz . . . . .	58
5.23	3D reconstruction of outdoor scene with multispectral camera after multi-modal registration . . . . .	59
5.24	3D reconstruction of indoor scene with multispectral camera after multi-modal registration . . . . .	59
5.25	Remote Camera Controller - Connection settings . . . . .	60
5.26	Remote Camera Controller - Image view . . . . .	60
5.27	Remote Camera Controller - Settings view . . . . .	60

# List of Tables

2.1	CMS-V1 Multispectral Camera Specifications . . . . .	10
2.2	CMS-V1 Sensor Spectrum Specifications . . . . .	10
2.3	Multispectral Camera Comparison . . . . .	11
2.4	Microsoft Kinect V2 RGB-D Camera Specifications . . . . .	12
3.1	Crosstalk Correction Coefficients for the CMS-V Multispectral Camera (Model: CMS19110114) . . . . .	30
4.1	NDVI Values Interpretation . . . . .	38

# Chapter 1

## Introduction

### 1.1 Inspiration

Agriculture is very important for humanity, supplying humans with several goods that are necessary for daily nutritional needs. The development of agriculture enabled the human population to increase many times larger than could be sustained by its food production. As a result, the daily increasing number of the world population is demanding the increase of the production [44].

So, precision agriculture (PA) is used, not only to increase production but also to decrease the losses during the production from plant diseases, insects, bad weather conditions, etc. Precision agriculture uses modern technologies for observation, measurement, and generally the management of the plants. Satellite imagery, multispectral imagery, GPS equipped systems, and others are used for this purpose but also the combination of them. Vines need a lot of attention from the farmers to succeed in the best production and the quality of the grapes. Vineyard analysis is very important for vineyard management for multiple purposes including growth inspection, disease detection, vines' stress detection, soil conditions inspection, harvest yield estimation, resulting in quality improvement [9, 46, 75].

Inspired by other similar applications, this project is aiming to use an RGB-D camera and a multispectral camera to extract, use and visualize with 3D modeling useful information of the vineyard. The pipeline that is followed, starts from image acquisition until the plant condition analysis with artificial intelligence. Absolutely crucial, is the registration between the different modalities. Different methodologies have been examined and two approaches have been developed which are used especially in the robotic system. When multi-modal image registration is done, segmentation of the leaves, the crops, the branches can succeed, but also

the removal of unnecessary information of the image including the sky or the soil. As a result, useful information can be extracted for agronomists with the 3D models of the vineyard or using vegetation, moisture, and soil indices such as the Normalized Difference Vegetation Index (NDVI) and the Enhanced Vegetation Index (EVI). Agronomists will have the opportunity to inspect the health of the plants and take action earlier.

Finally, an artificial intelligence (AI) module is used to export precision information about plants condition. This project is divided into multiple parts that are presented below, in Chapter 3 and Chapter 4 as the project pipeline. Especially, for every section of the pipeline is presented the methodology, the steps that are followed for the implementation and the results. Generally, every part is presented as a unique part of this project with its research, methods, implementation and finally, all parts are merged.

## 1.2 State Of The Art

Wine production is very important for a lot of countries such as France, Spain or Italy. The basic economy of these countries depends on wine productions and the research on vineyard management and analysis is extensive. According to the demanding care and maintenance of the farmers, it is very important to build tools that are able to assist them in a more profitable and sustainable production. A lot of research has been done to build an autonomous robotic system for analysis, management, and generally to assist the farmers. Some of the projects are already in the market as commercial products, but others are still in the labs for research.

Several commercial or experimental robotic systems are used for vineyard analysis and management, but most of them have significant differences with the current project. The WineRobot is a similar project with the current project that uses AI to monitor grape yield, vegetative growth, water status and grape composition. This robot is equipped with several sensors such as chlorophyll-based fluorescence sensor, RGB sensor, IR sensor, and GPS sensor. The main difference with our project is that our project depends only on machine vision sensors such as RGB cameras, multispectral camera RGB-D cameras, polarized cameras, etc. Machine vision is absolutely crucial for the entire project and is used for most of the parts [45].

The Vinbot is another robotic platform that uses camera sensors, GPS for navigation, but it does not use AI. This project seems a simplified version of the current project instead of the navigation via GPS. Moreover, it uses Summit XL from Robotnik as the robotic platform of all the computer, equipment and sensors like it is used in the current project [69]. Finally, Wall-Ye company is manufacturing robots, which are used for several plants analysis and management applications. This approach is way different from the current project and provides several features such as GPS navigation, vegetation detection via camera, crop harvesting and mapping,

pruning, etc [77]. It is obvious, that this approach also uses multiple sensors and techniques, but machine vision is not the basic technique. Another similar approach of precision agriculture with robotics is the Vitirover. The Vitirover is GPS navigated robot for pruning the grass in vines, arboriculture, photovoltaic farms, railway, electricity transmission network, etc [56].

### **1.3 Main Objectives**

In this section are presented the basic objectives of the project. Generally, the main goal of the whole project is to develop and build a robotic system that will be moving in a vineyard, analyzing the condition of the plants, and finally, take actions depending on the acquired data. The current project is only one part of it and includes the multi-modal registration and the analysis of the vineyard. The final decisions also depend on the opinion of the agronomists so it is very crucial to perform further analysis about the right manipulation of several plant diseases.

The division of the whole project is required to accomplish an efficient development. The architecture of the project consists of perception, decision, and action layers as it is obvious from the project description and most of the robotic applications. For each layer, some objectives have been developed and finally connected with the other layer for a successful implementation. For the perception layer, data from sensors must be acquired (multispectral camera, Kinect sensors), processed, analyzed, and finally send to the decision layer. Especially, the main objectives are the development of a system that acquires data from the sensors, the pre-processing and processing of the data with multiple techniques, the registration of the data from both modalities, and finally the data transmission to the next layer.

The decision layer will receive the acquired data for further analysis. The artificial intelligence system will take action to extract essential information. Support Vector Machines (SVMs), Deep Learning (DL), and other machine learning techniques have been used and evaluated to extract the right decisions, but only the SVM approach is presenting. Finally, the action layer will be triggered by the decision layer, by spraying phytosanitary products or not or notify the farmer for any potential diseases, plants condition, etc. Especially, the main focus of this project is perception and decision layers. In conclusion, a very crucial objective that must be accomplished is the mobility of the system that will be used under outdoor conditions. Internet connection, computational power, power consumption, and connectivity are absolutely crucial and must be taken into account for the final implementation.

## 1.4 Timetable

During this project, it is very important to take into account the time limits and the duration of the project. Multiple sections of the project pipeline need more research to extract precision information and some sections need to be tested in the real environment that is the vineyard, for evaluation. Time management is absolutely crucial for every project and for the current project some sections absorb more time than others. In Figure 1.1 is presented the followed timetable for this project. As it can be observed, image registration, crosstalk correction, and flat-field correction consumed most of the time of research and the development of the project. Every part consists of research and development for each respectively, except for the initial research and the final report writing.

During this project period, the required software is developed and tested for the robotic system. Furthermore, the software must meet the laboratory standards, as close as possible, enabling the easy integration with the preexisting robotic infrastructure. The required robotic ecosystem for this project is ROS melodic integrated into Ubuntu 18.04 Operation System computer and the required programming language is Python. These technologies are presented in the Section 2.2.4 Software & Technologies of Chapter 2.

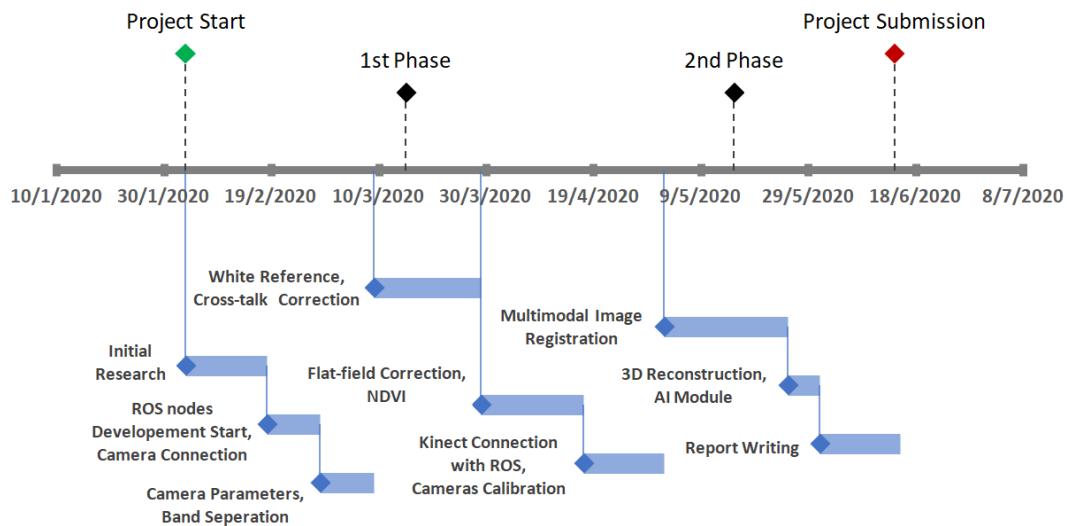


Figure 1.1: Project Timetable

# **Chapter 2**

## **Background**

### **2.1 Principles**

#### **2.1.1 Multispectral Imaging**

The electromagnetic spectrum is the range of frequencies of all types of electromagnetic radiation. Radiation is the energy that spreads out as it travels, the visible light that is visible from the human eyes, radio waves that are transmitted by radio stations, but also the microwaves that warm the food and x-rays that the doctors use for medical purposes. The electromagnetic spectrum consists of electromagnetic waves with different frequencies. Each frequency range is divided into separate different bands with different names. The visible light that human eyes can see is a part of the electromagnetic spectrum that consists of different bands [11, 42]. In Figure 2.1 is presented an illustration of the electromagnetic spectrum with different wavelengths. As it can be observed, the visible light is only a fraction of the electromagnetic spectrum.

The red, green and, blue color bands are different bands inside the visible electromagnetic wavelength that the human eye can see. A commercial color camera can capture 3 bands of visible light, but a multispectral camera can capture multiple different bands from different wavelengths and not only from the visible electromagnetic wavelength [11, 42]. The optical sensor of a color camera can capture the spectra of red, green and, blue in the range of 400 nm to 700 nm approximately. Especially, a multispectral camera can capture image data within specific wavelength ranges across the electromagnetic spectrum. For instance, the selected multispectral camera for this project can capture the spectra of 8 different bands in the range of 550 nm to 830 nm approximately, as presented in Table 2.2 below. The separation of the different wavelengths can be succeeded by using bandpass filters to image selected particular wavelength.

The human eye can see and capture a small amount of the spectral bands. Hence, multispectral imaging can enable the extraction of additional information that is not visible by the human eye. There are two main categories of multispectral imaging that are divided into multispectral imaging that uses a less than 100 bands and hyper-spectral imaging that uses often hundreds or more of spectral bands [3, 15]. Early applications of multispectral imaging are developed for the military and space sectors. In the current literature, multispectral imaging is used most in space-based applications for mapping of the earth related to vegetation, geographical measurements, etc. Especially, the most significant applications of multispectral cameras based on military and space-based imaging, such as target tracking, missile detection, weather forecasting, earth mapping, etc [14, 35].

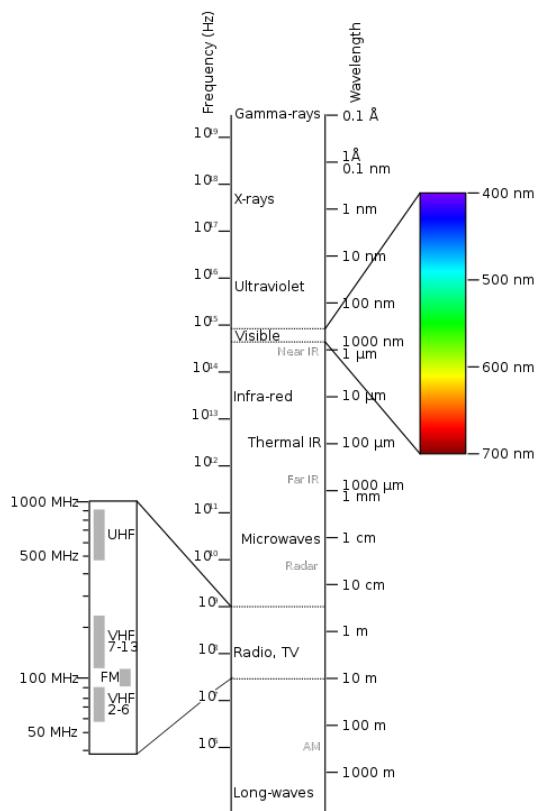


Figure 2.1: Electromagnetic Spectrum Illustration [73]

### 2.1.2 Multi-modal Data Fusion & Processing

Generally, in multiple-sensors perception, it is necessary to be used the same modality or a multi-modal approach to combine and process the data from these sensors. Especially for multi-modal approaches in robotics, several types of data acquired from different modalities such as images, videos, audios, or other types of data, that can be acquired from sensors attached to the robotic system. A robot can carry multiple types of sensors or modalities, which are able to acquire data, resulting in the need for data alignment and calibration. The precise synchronization and registration of them are absolutely crucial for valuable exported information. For instance, an RGB camera combined with a depth camera or a Lidar sensor can generate rich and useful information. Moreover, an RGB camera combined with a polarized camera or a combination of multi-modalities, such as MRI/PET in medical applications, etc. In this project, a multispectral camera and the sensors of a Microsoft Kinect V2 are combined and registered to extract useful information from the acquired images of the vineyard [33,55].

## 2.2 Hardware

### 2.2.1 Overview

In robotics, the hardware is as important as is the software. So, it is a big challenge and very vital the right selection of the hardware before the implementation. The main components of the project which are used are a CMS-V multispectral camera by Silios Technologies and a Kinect V2 sensor that is an RGB-D camera by Microsoft. Initially, several multispectral cameras of the current market will be presented and will be compared with each other and the final selection is presented. Kinect V2 has been selected as a low cost, easily integrated sensor to the robotic system that is equipped with an RGB camera and a depth camera. The Kinect V2 is the ideal sensor for this project due the outdoor usage capability, its low cost, its big availability in the market.

Very crucial is to mention that these sensors will be attached to a 4x4 outdoor robotic base that is a Summit XL by Robotnik [57]. The robotic base will provide the autonomous or remote navigation in the vineyard, which is not part of the current part of the project, while the attached sensors will acquire the multi-modal data. The final project must be able to move, acquire data, process the acquired data, and final trigger actions based on the result of processing. In the section below, are demonstrating a comparison of the different multispectral cameras in the current market, the specifications, and the reasons for the final decision. Finally, Kinect V2 sensor specifications are presented.

### 2.2.2 Multispectral Cameras: Current Market

The multispectral camera is a key component of this project. So, the right selection of multispectral camera equipment is very important for the current development of the project, but also for the future work that can be done. The current market of multispectral cameras provides cameras with different capabilities, advantages, disadvantages, and cameras for different applications such as cameras for use on Unmanned Aerial Vehicles (UAVs) or handheld cameras.

A camera for a UAV application can be used for a stationary application if it meets most of the required criteria. The cost is an absolutely crucial factor of the final decision but must take into account that the camera must meet the requirements of the project. Below are presented several multispectral cameras and their specifications and capabilities. More extensively is presented the chosen multispectral camera for this project.

- **MAIA WV multispectral camera:** The MAIA WV multispectral camera is equipped with an array of 9 sensors (1 RGB and 8 monochrome with relative band-pass filters) to capture multispectral imagery in the Visible Spectrum (VIS) and the Near-infrared Spectrum (NIR). It is designed to be attached on board the UAV systems or on board aircrafts and can be used for several applications including precision agriculture, monitoring of industrial plants, environmental monitoring, and geology applications, etc [76].
- **Hawk or HAWC (Handheld Agricultural Wireless Camera) Smart NDVI (Normalized Difference Vegetation Index) Phenocamera:** The Hawk camera is a multispectral camera, designed especially for vegetation applications by Teracam Company. This camera is designed for both handheld use and unattended gathering of images, usually for time-lapse photography. Moreover, it is equipped with a sensor that can capture, save, and process Near-infrared Spectrum (NIR), Red Spectrum (R), Green Spectrum (G) images with resolution of 2 megapixels. Also, it is equipped with a removable battery, a wireless connection, browser-based remote operation, and is PI 3 based, using Linux system [71].
- **Auk 3/3E or AWC 3/3E (Agricultural Wireless Camera) Smart NDVI (Normalized Difference Vegetation Index) Cameras:** The Auk series of multispectral cameras are entry low priced multispectral cameras especially for vegetation by Teracam Company. Moreover, the cameras are equipped with a sensor that can capture, save, and process Near-infrared Spectrum (NIR), Red Spectrum (R), Green Spectrum (G) images with resolution of 2 megapixels and sunlight sensor for onboard reflectance calculations. The cameras provide a USB2, an Ethernet and wireless connection and a Linux controller technology for remote operation [70].

- **Parrot Sequoia+ Multispectral Camera:** The Parrot Sequoia+ multispectral camera is a lightweight camera that is specially designed to be attached to UAVs. It is equipped with two types of sensor which are four multispectral sensors featuring Green (G), Red (R), Red Edge (RE) and Near-Infrared (NIR) spectrum bands (1.2 MP each), and an RGB (16MP), plus a sunshine (light) sensor [54].
- **CMS-V GigE Silios Multispectral Camera:** The CMS-V GigE camera is a lightweight multispectral camera and is developed to be used especially for vegetation applications, equipped with Gigabyte Ethernet connection. The camera imager has a modified Bayer matrix on a commercial 1.3 megapixels CMOS Sensor, made of a group of 3x3 pixels that is called macro-pixel, filtering 3x3, resulting in 8 different spectral bands and 1 panchromatic channel [38]. A raw image that is captured by the camera is built of 9 sub-images that consist of 8 color bands (8 band-pass filtered) and 1 panchromatic channel (1 panchromatic filtered with spectral response 500-900 nm), as presented in Table 2.2. The raw image resolution is 1280x1024 pixels and the resolution of the 9 sub-images is 426x339 pixels [64]. In the corners of the raw image, there are dead pixels that are used for alignment. Below in Table 2.1, are presented the specifications of the multispectral camera. In Table 2.2, are presenting the sensor specification of the multispectral camera, but also other important information about the bands, wavelength range, etc [5].



Figure 2.2: CMS-V GigE Silios Multispectral Camera [5]

<b>Multispectral Camera Specifications</b>	
Identification	CMS-V1-C-EVR1M-GigE
Connection interface	Ethernet Connection
Serial number	CMS19110114
Array type	CMOS (Si)
Raw image resolution*	1280 width x 1024 height
Band image resolution*	426 width x 339 height
Pixel pitch	5.3 m
Trigger	Software - Rising - Falling
Shutter	Global - Rolling
Pixel Clock (MHz)*	7 - 71
Frame Rate (FPS)*	0 - 50

Table 2.1: CMS-V1 Multispectral Camera Specifications

\*These specifications are important for the Camera Parameters in the Section 3.1.

<b>Sensor Specifications</b>			
Macro pixel size	3x3 bands		
Wavelength range	550 to 830 nm typical		
Type of pixel	Type of pixel 8 colors (narrow bands) + 1 B&W		
Band 1	560nm	FWHM*: 35nm	QE max: 25%
Band 2	595nm	FWHM*: 33nm	QE max: 23%
Band 3	634nm	FWHM*: 30nm	QE max: 20%
Band 4	673nm	FWHM*: 30nm	QE max: 19%
Band 5	713nm	FWHM*: 29nm	QE max: 14%
Band 6	752nm	FWHM*: 26nm	QE max: 14%
Band 7	791nm	FWHM*: 28nm	QE max: 11%
Band 8	828nm	FWHM*: 31nm	QE max: 9%
Band 9	Neutral density: QE mean = 5% over [500-900]nm		

Table 2.2: CMS-V1 Sensor Spectrum Specifications

\*(FWHM) or Full Width at Half Maximum

### 2.2.3 Multispectral Cameras Comparison

The comparison criteria have been chosen based on the requirements of the project. Dimension, the spectral bands, the connection are absolutely crucial for the project. Also, another factor of the final selection was the interfacing of the camera with the ROS ecosystem. The CMS-V multispectral camera, as the camera that is selected for this project, meets most of the criteria that we had set. Remarkably, the CMS-V camera by Silios company is based on an IDS OEM NIR camera module. As a result, the interfacing between ROS and an IDS camera module can easily be done by using the dedicated ROS drivers. Below, in Table 2.3 is presented the

comparison of the most significant specifications of the multispectral cameras for this project.

Criteria	MAIA WV	Hawk or HAWC	Auk 3/3E or AWC 3/3E	Parrot Sequoia+	Silios CMS-V
<b>Usage*</b>	UAV and Aircraft installation	Handheld or ground installation	Ground or Aircraft installation	Aircraft installation	Ground installation
<b>Sensor</b>	9 sensors (1 RGB and 8 monochrome with relative band-pass filters)	8 MP sensor	8 MP sensor	RGB sensor + 4 single-band sensors	Hybridization of a custom Bayer-like matrix on a commercial NIR CMOS Sensor
<b>Resolution</b>	1.2 MP (1280x960 pixels) each sensor	8 MP Operated at 1/2 resolution 1640x1232 pixels	8 MP operated at 1/2 (1640x1232 pixels)	16 MP (4608x3456 pixels) + 1.2 MP (1280x960 pixels) each band	1.3 MP (1280x1024 pixels)
<b>Sensor Sensitivity</b>	390 nm to 950 nm	420 nm to 950 nm	420 nm to 950 nm	550 nm to 790 nm	550 nm to 830 nm
<b>Multispectral Bands</b>	8 bands + RGB (Purple, Blue, Green, Orange, Red, Red Edge, NIR 1, NIR 2, RGB)	3 bands (Blue, Green, NIR)	3 bands (Blue, Green, NIR)	4 bands + RGB (Green, Red, Red edge, NIR, RGB)	8 bands + 1 B&W
<b>Connection Interface</b>	Wi-Fi, Gigabyte Ethernet	IEEE802.11 wireless, USB 2.0	IEEE802.11 wireless, Gigabit Ethernet, USB 2.0	N/A	Gigabyte Ethernet
<b>Weight</b>	160 g	290 g	180 g	80 g	110 g

Table 2.3: Multispectral Camera Comparison

\*The optimal usage of every camera depends on the manufacturer specifications.

### 2.2.4 Microsoft Kinect V2 Sensor: RGB-D Camera

Microsoft Kinect V2 is a motion-sensing device that has been manufactured by Microsoft initially for the gaming console Microsoft Xbox, but later also it is widely used for multiple other applications. The device consists of an RGB camera, an infrared (IR) projector, and an infrared (IR) detector which are incorporating to map depth through the time of flight calculations, enabling the real-time gesture recognition, body skeleton detection, etc [17]. This enables Kinect to be used as a hands-free natural user interface device to interact with a computer system [31].

Originally Kinect was developed as an alternative game controller of Microsoft Xbox as mentioned previously, but also it was used in academics for different purpose researches and other commercial uses. Microsoft Kinect is used for many different applications, such as in robotics, health care, etc. The Kinect V2 sensor has been selected for the project due to its low cost, high-quality images (HD), the ROS compatibility, the outdoor usage capability in difference with Kinect V1. Below, in Table 2.4 are presented the specifications of the Microsoft Kinect V2 sensor.



Figure 2.3: Microsoft Kinect V2 Sensor [74]

Kinect V2 Device Specifications	
RGB camera	1920x1080 pixels
Frame Rate (FPS)	0 - 30 Frames
Depth camera	512x424 pixels
Max depth distance	4.5 m
Min depth distance	50 cm
Horizontal field of view	70 degrees
Vertical field of view	60 degrees

Table 2.4: Microsoft Kinect V2 RGB-D Camera Specifications

## 2.3 Software & Technologies

### 2.3.1 Overview

The main part of the project depends on image processing, but also it depends on robotics. The software and the technologies which are used for this project depend on both parts and the successful fusion of them resulting in the best outcome. The base software ecosystem used is ROS melodic. A computer with Ubuntu 18.04 LTS operating system is necessary to house ROS melodic for a successful implementation.

The main available programming languages for programming in ROS are C++ and Python. For the necessary tasks that include image processing, OpenCV is used, but a lot of image processing functionalities are developed from zero base in both C++ and Python code. In the next sections (Section 2.3.2 and Section 2.3.3) are presented the most significant technologies that are used.

- Ubuntu 18.04 LTS: Operating System that is a Linux distribution based on Debian.
- ROS Melodic Morenia: Robotics framework for the development of robotic software.
- Programming Languages:
  - Matlab, Octave: Computing environments and programming languages for prototyping and testing.
  - Python 2.7: ROS packages development programming language.
  - C++ 11: ROS packages development programming language.
  - JavaScript: Client scripting language for creating web applications.
- Main Library:
  - OpenCV 3.2: Library for computer vision applications.
- ROS Packages:
  - ueye\_cam: ROS package that wraps the driver API for UEye cameras by IDS Imaging Development Systems GMBH.
  - rtabmap\_ros: ROS package for 3D Reconstruction.
  - iai\_kinect2: Microsoft Kinect V2 tools for interfacing with ROS.
  - rosbridge\_suite: ROS package that provides a JSON API to ROS functionality for non-ROS programs.
  - roslibjs: JavaScript library for interacting with ROS from the browser.
  - rviz: Visualization tool for ROS.

### 2.3.2 ROS (Robot Operation System)

ROS or Robot Operation System is a robotics framework and especially robotics framework that is used for the development of different purposes robotic software. Actually, it is not an operating system, but a middleware between operating system and the robotics interface. The main advantage of developing with ROS is code reusability, which means that it uses multiple components, which are able to work on other robots with minor changes. It provides multiple tools and libraries for robot software development, aiming in the simplification of creating complex and powerful robotic applications [51].

Very significant of ROS functionality is that it uses graph architecture, in which different structure models exist such as nodes, topics, and services. A node can be any process that can send and receive data from other nodes, sensors, and control different actuators. The data travels through the topics which are the busses of data transmission. ROS was developed by Stanford Artificial Intelligence Laboratory in 2007, it is distributed under BSD license and it is open source. Furthermore, from 2013, OSRF (Open Source Robotics Foundation) became the primary software maintainer [47]. ROS has multiple versions, but in this project, Melodic version is used [24].

### 2.3.3 OpenCV Library (Open Source Computer Vision Library)

OpenCV or Open Source Computer Vision Library is a library that contains programming functions that used for computer vision applications. Originally, OpenCV was developed by Intel to improve CPU based applications and it is free for use with open source BSD license. OpenCV library provides a lot of functionalities for computer vision, open source code and optimized code.

There are plenty applications of OpenCV especially in computer vision that include, facial recognition system, gesture recognition, human and computer interaction, mobile robotics, object recognition, segmentation, stereo vision, augmented reality, etc. Furthermore, it includes machine learning functionality to support computer vision such as decision tree learning, k-nearest neighbor algorithms, artificial neural networks (ANNs), support vector machines (SVMs), deep neural networks (DNNs) and many others [32].

# Chapter 3

## Pre-processing

### 3.1 Overview

In this chapter is presented the pipeline that is followed for this project. Declaring a project pipeline prior to the start of the project was very crucial and helpful for further research and development. The project pipeline consists of multiple different parts. Initial parts consist of image acquisition, pre-processing, processing, multi-modal registration, and finally multi-modal processing. The last part consists of the different artificial intelligence approaches that have been used.

It is very important to be mentioned that the data from the different modalities can be used separately or can be combined. Different pre-processing techniques are used after image acquisition resulting in precise exported data. Furthermore, the different techniques are analyzed and finally implemented in the ROS ecosystem. In Figure 3.1 is presented the schematic of the project pipeline. The pre-processing part of the pipeline is presented in the current Chapter 3 and the processing part is presented in the Chapter 4.

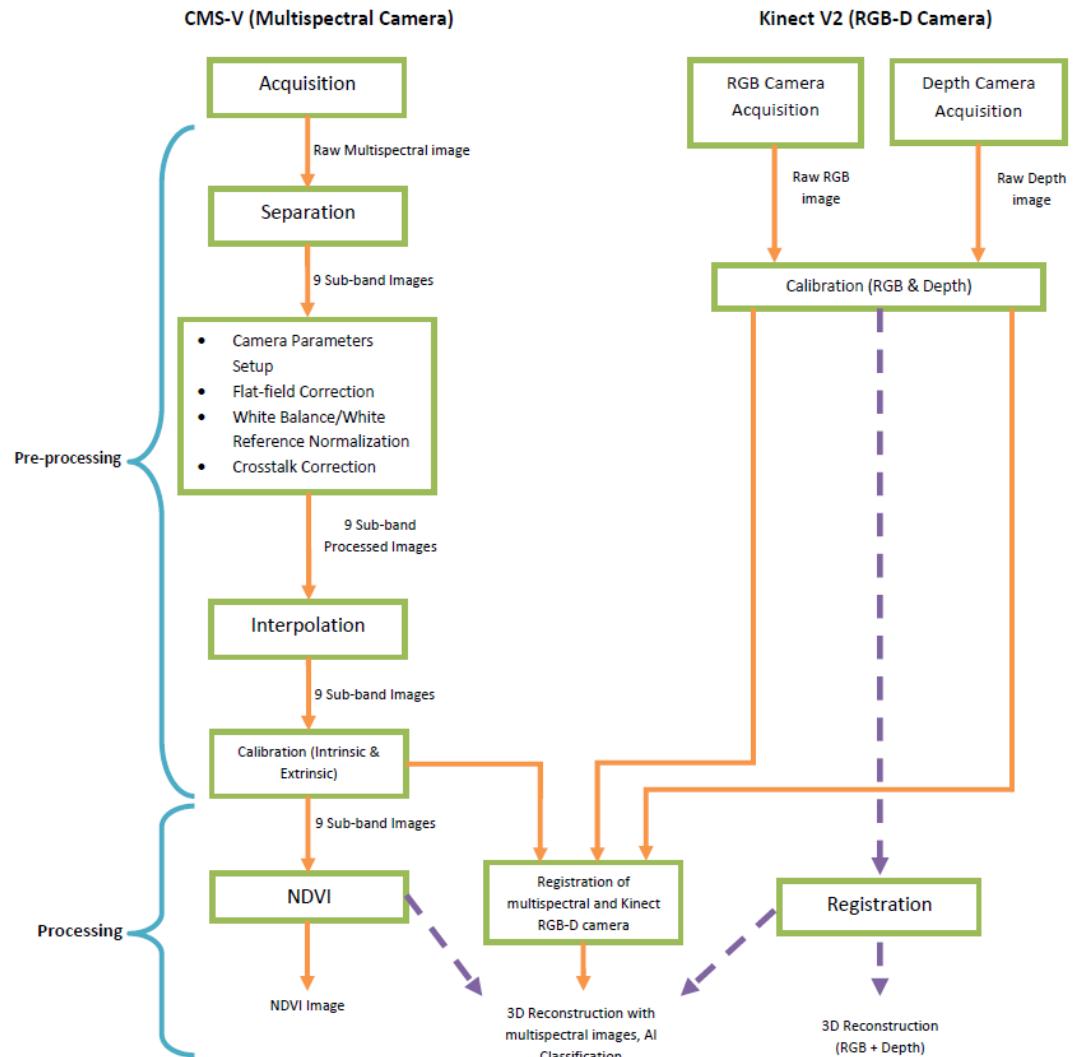


Figure 3.1: Project Pipeline

## 3.2 Multispectral Camera - Parameters

### 3.2.1 Overview

Multispectral camera parameters are a vital part of the whole project. Different luminance or different exposure time of the camera can give us entirely different results. So the right setting of camera parameters must be carefully examined and to be set before every acquisition. Absolutely crucial is also the weather conditions during operation. Cloudy conditions may need different camera parameters than sunny conditions, due to the light that can reach the camera sensor. Below are analyzed some significant parameters that they are set to the camera for the right acquisition. These parameters are used in the final implementation.

- **Pixel Clock (MHz):** The pixel clock is a high frequency pulse that determines the rate at which the pixels are acquired. Increasing the speed of a digital camera's pixel clock or acquiring more than one pixel at a time can greatly increase the camera's acquisition speed [4, 18].
- **Frame Rate (FPS):** The frame rate, expressed as fps or frames per second, is the number of frames (or images) the camera sensor can capture per second. 30 FPS is defined as the current standard from the industry that provides a smooth and clear transition to the moving objects. For a web application, fps depends also on the network's bandwidth [4, 18].
- **Aperture (F-numbers or F-stops):** Lens aperture is responsible for the amount of the light that enters the camera sensor. Especially, the aperture refers to the size of the opening in the lens' diaphragm through which the light enters the camera. The size of this opening can be adjusted and the aperture size is measured in f-stops. Smaller aperture means less light enters the camera resulting in darker images and the opposite happens for a larger aperture [4, 18].
- **Shutter Speed (Seconds):** Shutter speed is the time duration that the shutter of the camera remains open for light collection [4, 18].
- **Exposure (LUX Seconds):** Exposure is a very common parameter in photography. Especially, exposure is the time duration that the camera sensor pixels are exposed to light for light integration. Exposure depends on the lens aperture, shutter speed, and scene luminance. Generally, using low exposure on a camera may give a poor image. How bright or dark your pictures appear, depends on exposure parameter [4, 18].

### 3.2.2 Implementation

For this project, the camera parameters must be set manually. In a future approach, an automatic system that is able to set the camera parameters by the observation of the environmental conditions could be developed. Before starting up everything, initial parameters can be set. Moreover, the parameters can be changed on the fly during the running of the camera. ROS provides a plugin called `rqt_reconfigure` [61] that enables the dynamical modification of the camera parameters and it is very useful during development.

One more approach for camera parameters has been developed to provide more flexibility during outdoor experiments. For this approach `rosbridge_suite` [59] and `roslibjs` [60] have been used to develop an external camera controller that is developed with JavaScript. The camera controller is a web application that is developed to display a selected image topic and to change the camera parameters dynamically [1]. Finally, it is able to be synchronized with the existing camera plugin. In Figure 3.2 and Figure 3.3 are presented the User Interface (UI) of the ROS plugin and the web application controller for the camera parameters respectively.

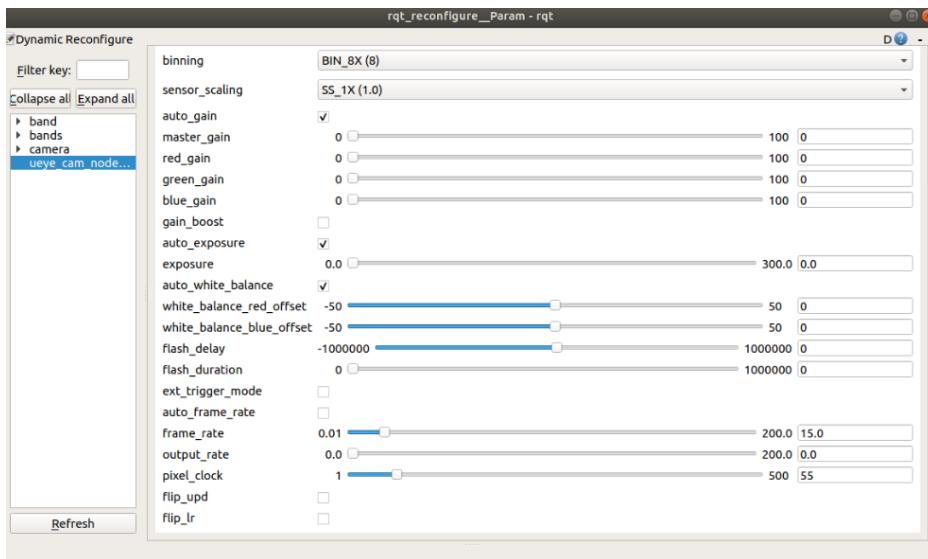


Figure 3.2: ROS `rqt_reconfigure` Tool for Camera Parameters

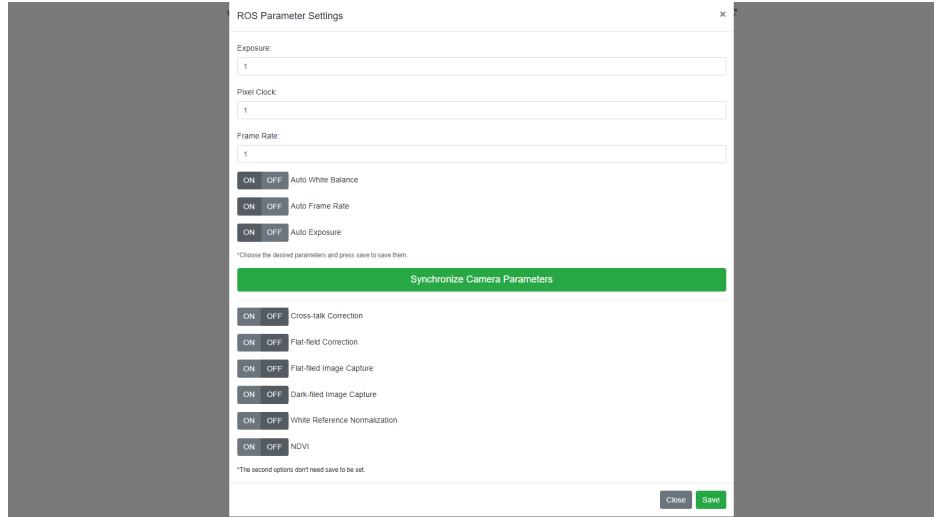


Figure 3.3: Web-based Camera Controller [1]

### 3.3 Multispectral Camera - Acquisition & Band Separation

#### 3.3.1 Overview

In this section will be analyzed the image acquisition the band separation methodology and procedures for the multispectral camera. For a standard RGB camera, it is obvious that after image acquisition, the image is ready for processing, because the embedded micro-controller does the necessary procedure. Especially, the embedded micro-controller to the camera is performing the reconstruction of a full color image from the 2x2 set of pixels (Bayer Mosaic), known as demosaicing [38]. In Figure 3.4 is presented the typical Bayer mosaic for color image capture that consists of a 2x2 set of pixels, with two green filtered pixels, and the other two have red and blue filters.

The acquisition with a multispectral camera is more challenging because needs some work to extract the useful information from the raw image by using the software. In future multispectral camera implementations, probably this functionality will be included with an embedded micro-controller. Before going deeper to image pre-processing or processing parts, it will be described how the camera sensor works and how the documentation of the camera's manufacturer describes its functionality.

The CMS sensor is made of a camera sensor and a CMS matrix filter that assemble as a single unit. The matrix filter is based on macro-pixels integrating 8 color filters (bandpass

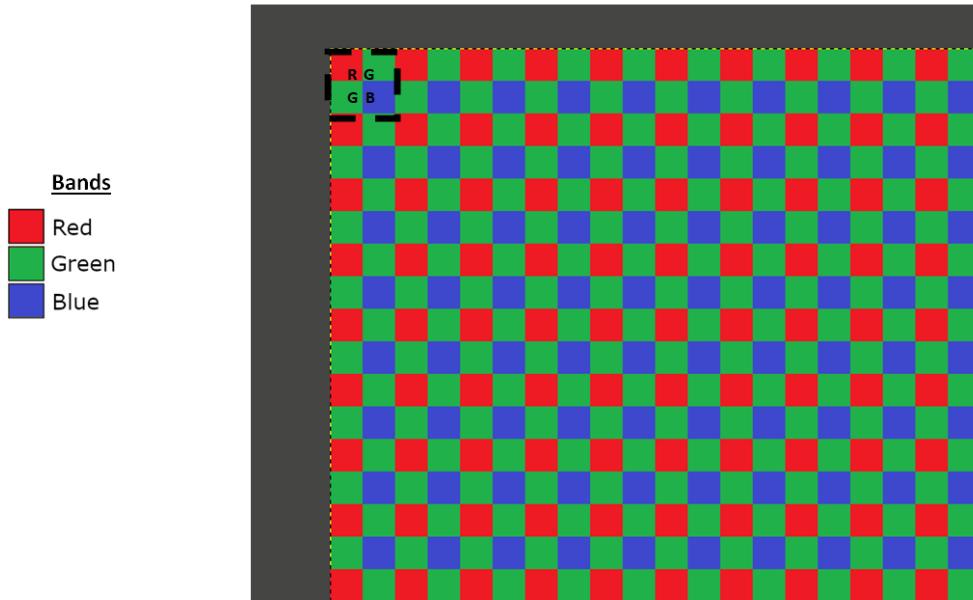


Figure 3.4: Typical RGGB Bayer Mosaic for Color Image

filters) and 1 panchromatic filter, arranged in a 3x3 matrix. As it can be observed by Figure 3.5, a macro-pixel is centered to the band 9 (panchromatic filtered). When a picture is acquired by the sensor, the pixel arrangement is seen from the sensor position and the macro-pixels are centered on the panchromatic filter. Below, in Figure 3.4, is presented the band arrangement in a 1280x1024 pixels image from the CMS-V camera sensor that is provided by the manufacturer. Finally, this arrangement is followed in the whole image.

Obviously, some of the raw image pixels are discarded due to the separation due to the division by 3 of each row and column, resulting in sub-images with resolution 426x339 pixels. As it can be observed in the list below, 2 pixels are discarded from the rows and 7 from the columns of the raw image. Furthermore, as it can be observed in Figure 3.5, the bands are mixed inside an acquired raw image and the acquired data is from sensor perspective. So, it is important for the band separation to take into account every pixel position during band separation and pixels discarding. Moreover, in Figure 3.5 are presented the colored band arrangement by the sensor and the macro-pixel that is centered in the panchromatic filter.

**Properties:**

- Offset X: 2 (from 3rd row)
- Offset Y: 0 (from 0th column)
- Raw Image Resolution: 1280x1024 pixels
- Band Image Resolution: 426x339 pixels
- Merged Sub-mages Size: Columns( $426 \times 3 = 1278$  pixels), Rows( $339 \times 3 = 1017$  pixels)
- Columns Discarded Pixels:  $1280 - 1278 = 2$  pixels
- Rows Discarded Pixels:  $1024 - 1017 = 7$  pixels

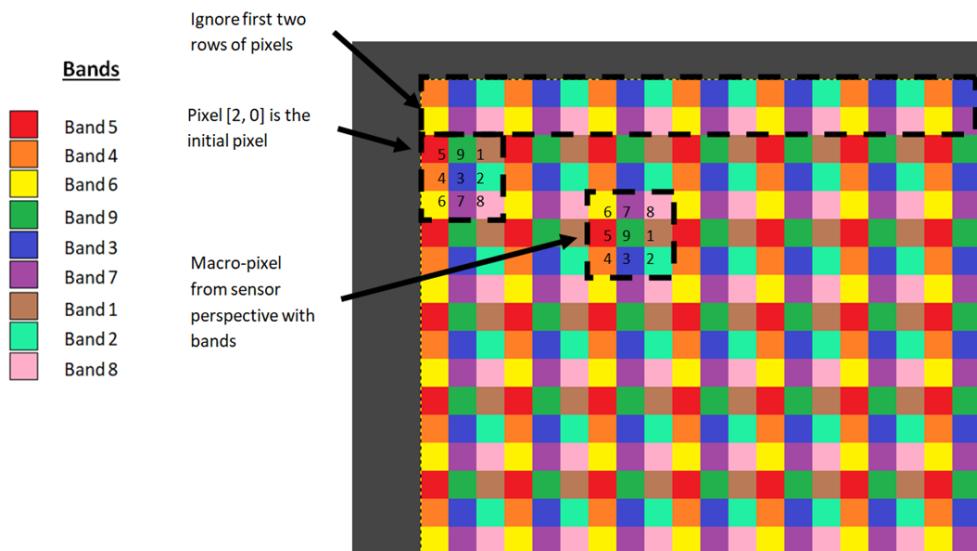


Figure 3.5: Colored Band Arrangement by CMS-V Multispectral Camera

### 3.3.2 Implementation

The CMS-V multispectral camera is compatible with IDS cameras drivers and software [19]. Moreover, ueye\\_cam ROS package is compatible with IDS cameras and is ideal for the camera interfacing. The installation of the IDS drivers and ROS package are necessary for a successful camera connection [62]. When the camera is connected the separation process can be done. The separation process that is followed generates an array of 9 arrays containing the 9 sub-images. Each array stores the pixels of each band moving to each row and each column of the raw image matrix by 3. The final result is 9 images (8-bit) that represent the 9 different bands with resolution 426x339 each. For every part of the project, 8-bit images are used which means values from 0 to 255.

A new ROS node has been developed for this process and contains multiple functionalities that are necessary for the implementation. In Figure 3.6 and Figure 3.7 are presented the raw and the resulting sub-images after separation with the Macbeth Color Checker respectively. As it can be observed in Figure 3.6, the corners of the raw image are white squares due to alignment purposes. Furthermore, some significant options of the implementation are displayed in the bottom left corner of the image, as a feedback to the user.



Figure 3.6: Raw Acquired Image by Multispectral Camera in ROS - OpenCV Interface

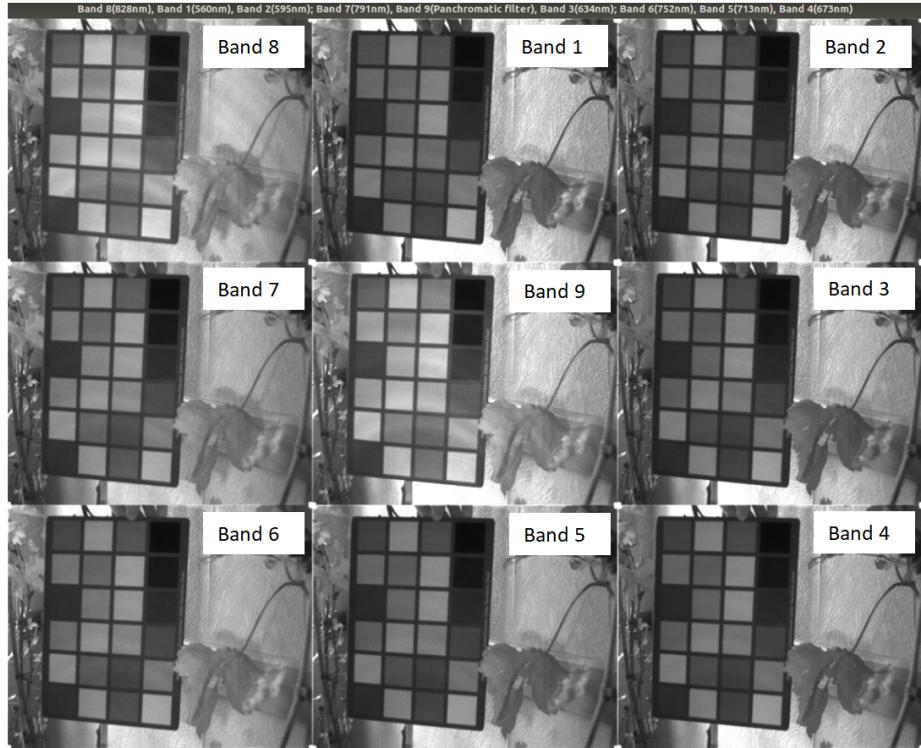


Figure 3.7: 9 Band Images after Band Separation in ROS - OpenCV Interface

## 3.4 Multispectral Camera - Flat-field Correction

### 3.4.1 Overview

Not only camera sensors and lenses have imperfections, but also in lenses appear distortions in the glass. As a result, the captured images have imperfections that add unwanted artifacts that evidence themselves as non-uniformity. Flat-field correction is used to improve image uniformity by removing or minimizing these artifacts regardless of exposure. Technically, flat-field correction corrects the spectra in the acquired picture. Especially, the goal of this correction method is to correct the pixels of the acquired raw image in such a way that when a uniform background is captured by the system (camera & lens), the resulting output image is uniform.

It is a very common technique that is applied to several types of cameras and the correction process is applied after the raw image acquisition. Many manufacturers perform flat-field correction with an embedded micro-controller to the camera sensor, resulting in high performance. The CMS-V is not equipped with an embedded micro-controller to apply this correction, so the

flat-field correction is performed by the software that has been developed. The mathematical equation and the process that is followed are analyzed below [16, 22, 23, 66].

### 3.4.2 Mathematical Equation

$$P_{i,j} = \frac{Raw_{i,j} - D_{i,j}}{F_{i,j} - D_{i,j}} * \frac{1}{(m * n)} * \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} (F_{x,y} - D_{x,y}) \quad (3.1)$$

- P: Image after the flat-field correction process.
- Raw: Raw captured image by the multispectral camera.
- D: Dark-field image that must be acquired once (Intensity range from 10% to 90%).
- F: Flat-field image that should be acquired once (Image pixels values must not be 0).
- m: Number of rows.
- n: Number of columns.

### 3.4.3 Implementation

The flat-field correction has been developed by using the equation above, with C++ and Python in ROS, especially for this project. Flat-field correction is applied after image acquisition, but it can also be applied to each sub-image. In Figure 3.8 and in Figure 3.9, are presented the flat-field and dark-field image respectively, captured by the multispectral camera. In Figure 3.10 and in Figure 3.11 is presented the image of the separated band 1, as presented in Table 2.2, before and after flat-field correction respectively.

Steps:

1. Flat-field image capturing: The image has to be a light image without any deformation and without 0 values. The intensity range of the image must be between 10% to 90%.
2. Dark-field image capturing: This image has to be a dark image. A good practice is to cover the camera lens with the protective cover, but the acquired values must not be 0.
3. Flat-field correction application: The correction can be applied in the raw image or in its band separately. In this implementation the correction is applied in the raw image.

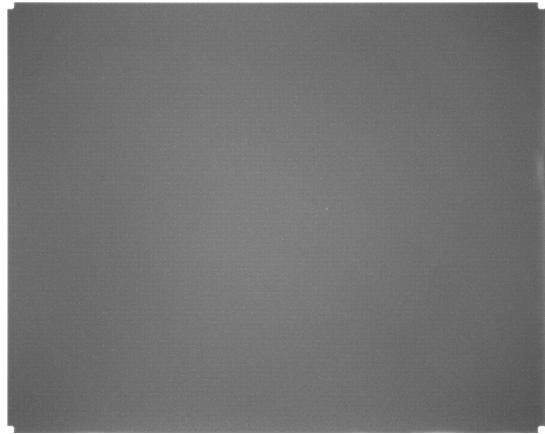


Figure 3.8: Flat-field Image Captured by Multispectral Camera

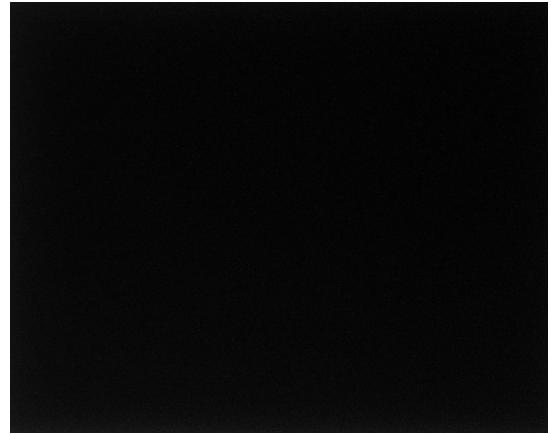


Figure 3.9: Dark-field Image Captured by Multispectral Camera



Figure 3.10: Acquired Image of Band 1 Before Flat-field Correction



Figure 3.11: Acquired Image of Band 1 After Flat-field Correction

## 3.5 Multispectral Camera - White Balance Normalization

### 3.5.1 Overview

Generally, in image processing, color balance is the adjustment of intensities of the colors in an RGB image. The final goal is the right representation of the colors. Also, color balance is referred as gray balance, neutral balance, or white balance. Especially, for the multispectral camera, the output of white reference is 9 coefficients for the normalization. The mathematical equation that is used is similar to the equation that is used for RGB camera, but with more channels. Below is presented the equation that is used for the calculation of every coefficient.

### 3.5.2 Mathematical Equation

$$\begin{bmatrix} B1 \\ B2 \\ B3 \\ B4 \\ B5 \\ B6 \\ B7 \\ B8 \\ B9 \end{bmatrix} = \begin{bmatrix} 255/AvB1' & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 255/AvB2' & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 255/AvB3' & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 255/AvB4' & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 255/AvB5' & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 255/AvB6' & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 255/AvB7' & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 255/AvB8' & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 255/AvB9' \end{bmatrix} \begin{bmatrix} B1' \\ B2' \\ B3' \\ B4' \\ B5' \\ B6' \\ B7' \\ B8' \\ B9' \end{bmatrix} \quad (3.2)$$

- $B1 - B9$ : The white balance corrected pixels for every macro-pixel.
- $B1' - B9'$ : Original pixels in a macro-pixel.
- $AvB1' - AvB9'$ : The average of the selected pixels per band, which have been selected as the white surface in the image before color balancing. If one macro-pixel is selected, then these values will contain the value of every pixel per band.
- $255/AvB1' - 255/AvB9'$ : White reference coefficient which can be stored.

### 3.5.3 Implementation

For pixels' selection of white reference pixels selection process, it is necessary the user interaction with the ROS ecosystem and OpenCV interface. The user is able to select the white surface in the raw image with the mouse pointer. If the user selects only one pixel the algorithm automatically selects 9 pixels or the whole macro-pixel, because it is necessary for the computation of the 9 bands coefficients separately.

Moreover, some rules have been applied to the selection process to avoid misuse from the user. The algorithm tries to identify the corners and shift the selected area inside the image boundaries. As a result, the algorithm will have 9 or more pixels to process for every computation. In Figure 3.12 is presented an illustration of how the algorithm acts when a misuse

happens in which the user selects pixels that are very close to the boundaries. In Figure 3.13 is presented the process of pixels' selection by the user interaction in ROS and OpenCV user interface (UI). Finally, in Figure 3.14 and in Figure 3.15 is presented the image of the separated band 8, as presented in Table 2.2, before and after the white reference normalization respectively.

Steps:

1. The user enables the option of white reference pixels' selection and selects the pixels.
2. The algorithm picks the selected pixels for further processing.
3. The algorithm identifies the band of every pixel.
4. The algorithm adds the pixels of each band separately.
5. The algorithm averages the pixels of each band separately.
6. Finally, the algorithm computes the coefficient of every band with the equation above.

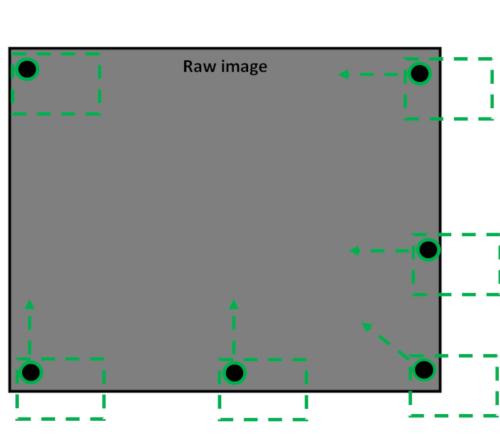


Figure 3.12: White Reference - Rules Illustration of Pixels' Selection

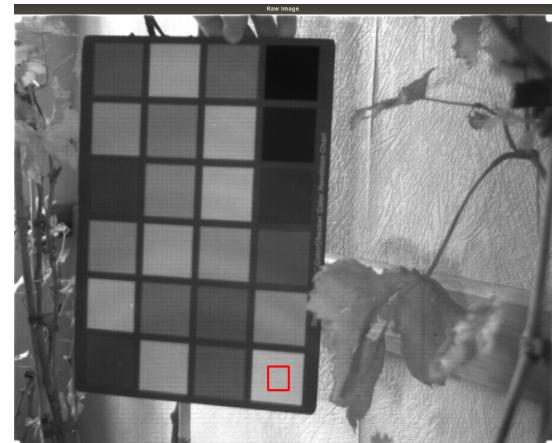


Figure 3.13: White Reference - Pixels Selection by User Interaction with ROS and OpenCV

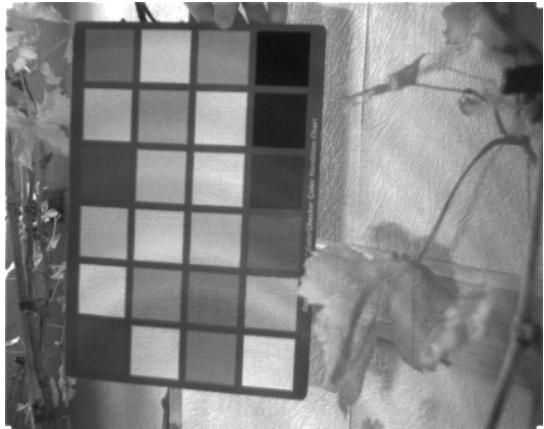


Figure 3.14: Acquired Image of Band 8 Before White Reference Normalization

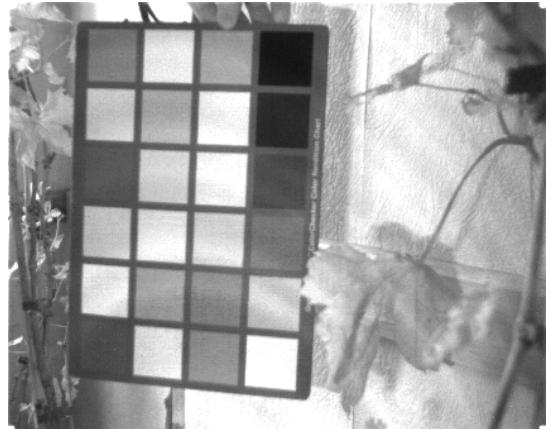


Figure 3.15: Acquired Image of Band 8 After White Reference Normalization

## 3.6 Multispectral Camera - Crosstalk Correction

### 3.6.1 Overview

Nowadays, the industry is developing smaller and smaller camera sensors, reducing the pixel size of them. The experiments reveal that these sensors suffer from different effects that decrease the quality of the acquired images. One of the problems that affect the camera sensors is crosstalk between the pixels. Especially, in a camera sensor, crosstalk is an effect where a pixel signal is affected by the neighbor pixels signal. Generally, it is the interference between the pixels. So the final response of a pixel does not depend on only the sensing light at this pixel, but also on its neighbor pixels and even much the closest neighbors. In Figure 3.16 is presented an illustration of the crosstalk effect in which the light as arrow for a pixel affects its neighbor signal response [2, 13, 64].

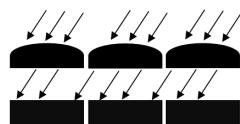


Figure 3.16: Crosstalk Effect Illustration (bottom rectangles represent the sensor pixels, the top curved rectangles represent the sensor filter, the arrows represent the incoming light)

Furthermore, it has been observed that the horizontally adjacent pixels interfere with each other much more than vertically adjacent pixels possibly due to the pixel layout. The CMS-V

multispectral camera is equipped with a multispectral filter on the top of a standard commercial monochrome sensor. Hence, with this technique, the result has a higher level of cross-talk, especially as the pixel size becomes smaller, because of the gap between the surface of the filter and the surface of the sensor. The camera manufacturer has developed after research on this phenomenon and provides some techniques to reduce it. In this project, the presented techniques have been implemented to reduce cross-talk between the pixels [21, 34]. In Figure 3.17 are presented the wavelengths of every band as they are acquired by the sensor. In Figure 3.18 are presented the crosstalk free wavelengths of the bands.

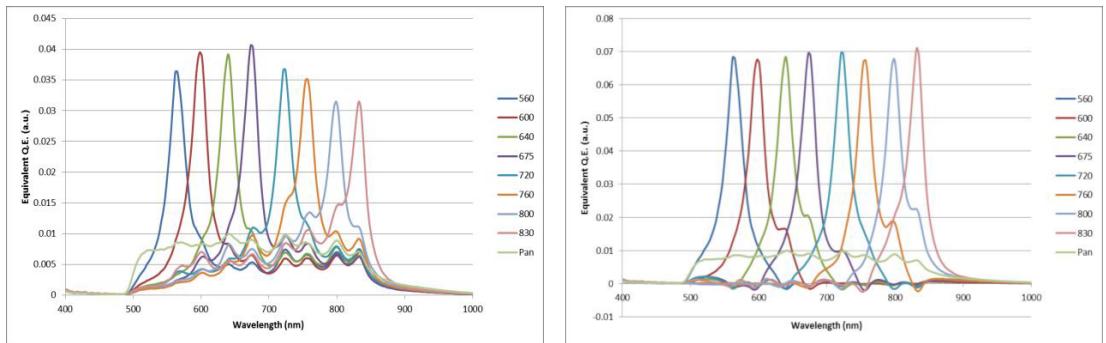


Figure 3.17: Bands Spectrum Before Crosstalk Correction [64]

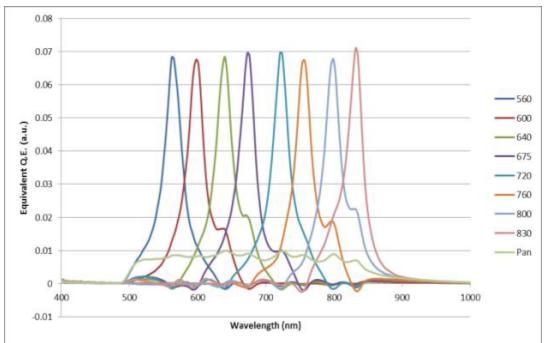


Figure 3.18: Bands Spectrum After Crosstalk Correction [64]

### 3.6.2 Mathematical Equations

Quantum efficiency equation:

$$QE_{reci}(\lambda) = \sum_{j=0}^8 CC_{i,j} * QE_j(\lambda) \quad (3.3)$$

The response of a pixel i is given by the equation below:

$$P_i = \int_{\lambda} Fr(\lambda) * QE_{reci}(\lambda) d\lambda \quad (3.4)$$

The response of a virtual pixel i is given by the equation below:

$$Prec_i = \int_{\lambda} Fr(\lambda) * QE_{reci}(\lambda) d\lambda \quad (3.5)$$

$$Prec_i = \sum_{j=0}^8 \{ CC_{i,j} * \int_{\lambda} Fr(\lambda) * QE_j(\lambda) d\lambda \} \quad (3.6)$$

The final equation is presented below. This equation is used for the final implementation to compute the value of every crosstalk free pixel.

$$Prec_i = \sum_{j=0}^8 CC_{i,j} * P_j \quad (3.7)$$

- $Fr(\lambda)$ : Optical Flux.
- $QE_i(\lambda)$ : Quantum Efficiency of a pixel filtered with the filter.
- $QErec_i(\lambda)$ : Quantum Efficiency of a so-called "virtual pixel" filtered by the reconstructed filter.
- $P_i$ : Response of a pixel filtered with the filter.
- $Prec_i$ : Response of a "virtual pixel" filtered with the reconstructed filter i.
- $\lambda$ : Electromagnetic wavelength.
- $i$ : Row.
- $j$ : Column.
- $CC_{i,j}$ : Crosstalk correction coefficients.

The crosstalk coefficients have been estimated by the manufacturer for each camera that they provide. Below, in Table 3.1 are presented the given coefficients which are used for cross-talk correction or the reduction of this effect.

	Prec1	Prec2	Prec3	Prec4	Prec5	Prec6	Prec7	Prec8	Prec9
P1	1.4872	-0.0938	-0.0084	-0.0087	-0.083	0.0113	0.0089	-0.1488	0.000
P2	0.0826	1.5340	-0.1431	-0.0554	-0.0398	-0.0112	-0.0218	-0.2073	0.000
P3	0.0191	0.0748	1.7315	-0.1353	-0.0564	-0.0378	-0.1625	-0.0779	0.000
P4	-0.0167	-0.0639	-0.0142	1.6058	-0.2237	-0.174	-0.1251	-0.1061	0.000
P5	-0.0651	-0.0447	-0.0591	0.0154	1.9477	-0.271	-0.163	-0.1278	0.000
P6	-0.0685	-0.0653	-0.0841	-0.1298	-0.112	2.0047	-0.343	-0.3548	0.000
P7	-0.0936	-0.0958	-0.1563	-0.1023	-0.1023	-0.151	2.1592	-0.4923	0.000
P8	-0.1783	-0.1753	-0.1536	-0.1557	-0.2121	-0.2969	-0.1708	2.4527	0.000
P9	-0.1667	-0.0700	-0.1127	-0.034	-0.0697	-0.0703	-0.1817	0.0398	1.000

Table 3.1: Crosstalk Correction Coefficients for the CMS-V Multispectral Camera (Model: CMS19110114)

### 3.6.3 Implementation

This method applies a linear combination of the neighbor pixel to correct cross-talk. Below are presented the steps that are followed to apply crosstalk correction. It is very crucial to notice that the first 3 steps have been provided by the manufacturer and in this project, only step 4 is necessary to be applied and to be developed.

The development for crosstalk correction implementation took place after flat-field correction and after band separation processes. For the implementation are used C++ and Python and for every pixel, the crosstalk free pixel is calculating with the given equation above. Finally, in Figure 3.19 and in Figure 3.20 is presented the image of the separated band 8, as presented in Table 2.2, before and after the crosstalk correction respectively.

Steps:

1. Measurement of the spectral response of all the pixels of the multispectral sensor.
2. Calculation of the average spectral response of each spectral sub-image.
3. Estimation of the best crosstalk reduction coefficients to minimize the cross talk in the reconstructed sub-images.
4. Usage of these coefficients directly onto the macro pixels data to reconstruct the corrected sub-images.



Figure 3.19: Acquired Image of Band 8 Before Crosstalk Correction



Figure 3.20: Acquired Image of Band 8 After Crosstalk Correction

## 3.7 Cameras Geometric Calibration

### 3.7.1 Overview

During image capturing and due to the radial distortion, the straight lines will appear in the image curved, increasing the effect when increasing the distance between the camera and the acquired object. Furthermore, defective lenses can produce not perfectly aligned images with the plane, such as some areas in an image look nearer than expected. Geometric calibration of a camera is the estimation of some parameters of the lens and sensor of the camera, which are very important to correct for lens distortion, measure the size of an object in world units, or determine the location of the camera in the scene due to cheap lenses, 3D reconstruction of images, etc [37, 67].

These parameters are divided into three different categories which include intrinsics, extrinsics, and distortion coefficients. The intrinsic parameters include the focal length, the optical center, also known as the principal point, and the skew coefficient. The extrinsic parameters consist of a rotation, R matrix, and a translation, t vector. The origin of the camera's coordinates system is at its optical center and its x-axis and y-axis define the image plane. Distortion is divided into two parts. Radial distortion occurs when light rays bend more near the edges of a lens than they do at its optical center. Tangential distortion occurs when the optical centers of the lens elements are not strictly collinear and generally when the lens and the image plane are not parallel. These affects appear due to imperfections in lens design and camera assembly [37, 49].

To estimate the camera parameters, you need to have the 3D world points and their corresponding in 2D image points. These correspondences can be acquired by using multiple images of a calibration pattern, such as a checkerboard pattern. The calibration process is a useful task in computer vision. It is used in many applications that integrates a camera and is very important to acquire precise data from several types of sensors [39, 67].

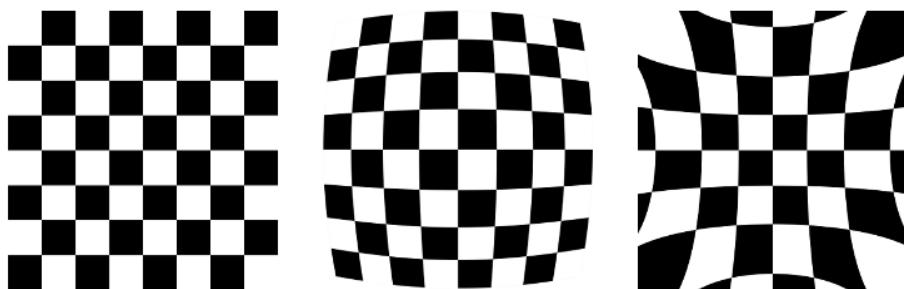


Figure 3.21: Distortion Effect Illustration [49]

### 3.7.2 Methodology

The camera calibration methodology is based on retrieving the intrinsic and extrinsic parameters of a set of 3D object points and its 2D projections points. For a camera model with lens distortion, the relation between the 3D object points and its 2D protective points are a non-linear relation. Especially, the calibration process use the points in an iterative method in order to reduce the difference between the 2D projection and the modelled one. In the equation 3.8 is presented the relation between the 3D points, the 2D points, the rotation matrix ( $R$ ), the translation vector ( $t$ ) the intrinsic camera matrix. In Figure 3.22 is presented an illustration of pinhole camera model with the geometric relation between a 3D point and its 2D projection, as it demonstrated above. In Figure 3.23 is presented the camera calibration process with radial lens distortion [63].

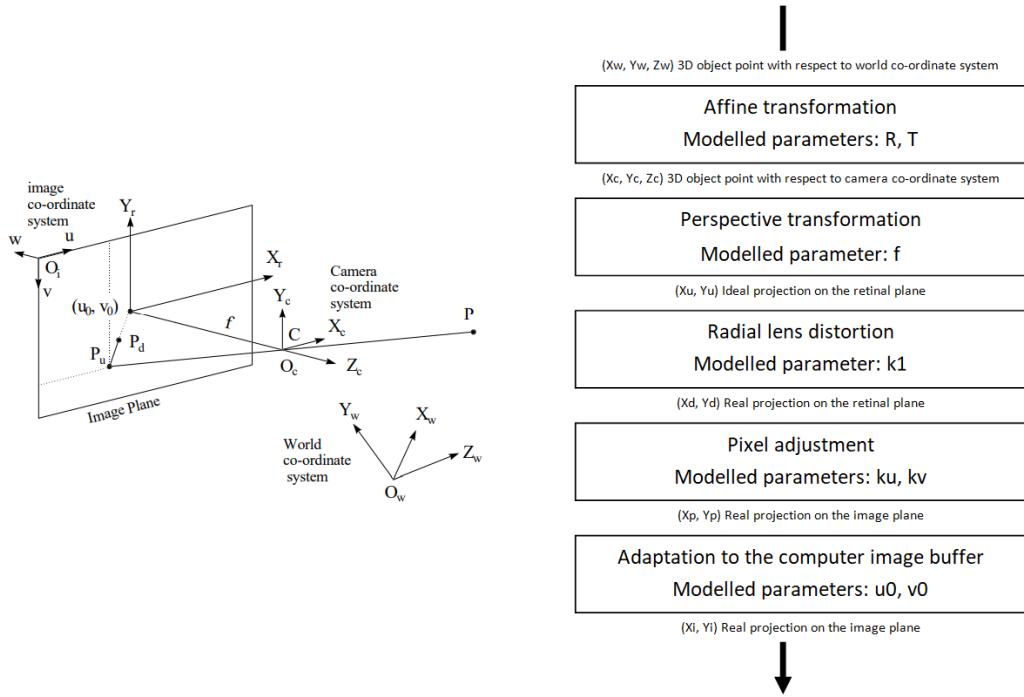


Figure 3.22: Illustration of pinhole camera model with the geometric relation between a 3D point and its 2D projection [63]

Figure 3.23: Flow-chart of pipeline for the projection of a 3D point to a 2D image plane with radial lens distortion

$$\begin{bmatrix} sX_i \\ sY_i \\ s \end{bmatrix} = \begin{bmatrix} a_u & 0 & u_0 & 0 \\ 0 & a_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_{Xw} \\ P_{Yw} \\ P_{Zw} \\ 1 \end{bmatrix} \quad (3.8)$$

$$A = \begin{bmatrix} a_u & 0 & u_0 & 0 \\ 0 & a_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (3.9)$$

- $X_i, Y_i$ : Coordinates of the projection point in pixels.
- $P_{Xw}, P_{Yw}, P_{Zw}$ : Coordinates of a 3D point in the world coordinate space.
- $u_0, v_0$ : Principal point that is usually at the image center.
- $a_u, a_v$ : Focal lengths in pixel units.
- $R$ : Rotation matrix.
- $t$ : Translation vector.
- $s$ : Scale factor parameter.
- $A$ : Camera matrix or matrix of intrinsic parameters.

### 3.7.3 Implementation

There are many available calibration packages for ROS. For the calibration of a multispectral camera, camera\_calibration package is used [58]. Furthermore, due to the special features of the multispectral camera, a calibration node is developed for research and experiments. Several sizes of a chessboard for calibration have been tested with the developed calibration node and generally, it used only for experiments. For the final calibration of the multispectral camera, camera\_calibration package by ROS is used.

A lot of calibration functionalities are provided by Kinect package iai\_kinect2 and the results are pretty good [72]. It has been decided that this package is good enough to calibrate Kinect sensors. Especially, for most of the calibration implementations, OpenCV is the backbone while provides a lot of functionalities for the development of the calibration node in this project.

# **Chapter 4**

## **Processing**

### **4.1 Normalized Difference Vegetation Index (NDVI)**

#### **4.1.1 Overview**

NDVI or Normalized Difference Vegetation Index was introduced in 1973 and is used in agriculture as an index or agronomic indice to identify the vegetation condition. NDVI can describe the vegetation density, allowing the researchers to evaluate vegetation, growth, and productivity. Especially NDVI is the contrast between the red channel (Red) and the near-infrared channel (NIR). Other popular indices are Modified Chlorophyll Absorption in Reflectance Index (MCARI), Modified Simple Ratio Index (MSR), Soil-Adjusted Vegetation Index (SAVI), Transformed Vegetation Index (TVI), etc [40,53,79].

A healthy plant will absorb blue and red light and reflect green light, which is why they appear green to our eyes. With the green visible light, plants also reflect Near-Infrared (NIR) light. This type of light, which is invisible to the human eye, also isn't actively used for the photosynthesis process, and the healthier the plant, the more NIR light is reflected. When a plant becomes dehydrated or stressed, the spongy layer of the plant collapses, and its leaves reflect less NIR light, yet they still reflect the same amount of light in the visible range. The most significant application of NDVI value is in the detection of the stressed crop because of the crop stress can be sooner detected in near-infrared (NIR) than in the visual spectrum. So the farmers can respond more quickly by taking corrective actions [43].

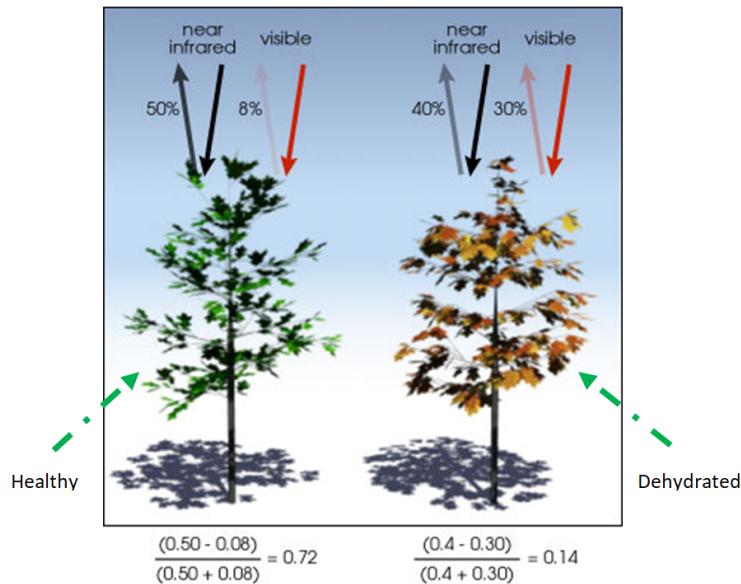


Figure 4.1: NDVI Illustration [43]

#### 4.1.2 Mathematical Equations

The reflection in red and near-infrared zones of the electromagnetic spectrum depends on the chlorophyll. So, the index is calculated as the reflection difference in the near-infrared and red spectrum divided by its total, resulting in values from -1 to 1, and the higher the index value, the greater the crop health. NDVI values between -1 and 0 correspond to non-plant surfaces that have a reflectance in the Red (band 3) that is greater than the reflectance in the NIR (band 6), based on multispectral bands as presented in Table 2.2. These could be surfaces such as equipment, water, or soil. Soil's value is close to 0. Plant values range from 0.1 to nearly 1, and as stated earlier, the higher the NDVI value, the greater their density and health.

Equations:

$$NDVI = \frac{NIR - RED}{NIR + RED} \quad (4.1)$$

Based on multispectral bands as presented in Table 2.2:

$$NDVI = \frac{Band6 - Band3}{Band6 + Band3} \quad (4.2)$$

Normalized index for values from 0 to 1, based on multispectral bands as presented in Table

2.2:

$$NDVI = 0.5 + 0.5 * \frac{Band6 - Band3}{Band6 + Band3} \quad (4.3)$$

#### 4.1.3 Implementation

Especially for the vegetation, the red spectrum (RED) reflection is lower than the near-infrared spectrum (NIR) due to the light absorption by chlorophyll, so the NDVI values for vegetation are not lower than 0.1. For this project, a custom color-map has been used to visualize the level and the condition of the vegetation. In Figure 4.2 are presented the different color shades that are used for the image coloring. In Figure 4.3 and Figure 4.4 are presented the NDVI image and NDVI colored image respectively. As can be observed from the colored image, the vegetation can be easily segmented from other materials. In table 4.1 are presented the NDVI values interpretation. Values below 0 are divided into 4 classes.

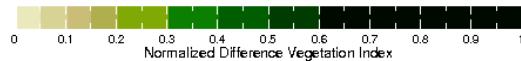


Figure 4.2: NDVI Color-map Illustration

NDVI* Interpretation	
0.00 – 0.05	Bare soil
0.05 – 0.10	Bare soil almost absent vegetation cover
0.10 – 0.15	Bare soil with very low vegetation cover
0.15 – 0.20	Very low vegetation cover (initial vegetation state)
0.20 – 0.30	Low vegetation cover
0.30 – 0.40	Medium vegetation cover
0.40 – 0.50	High vegetation cover
0.50 – 0.60	Very high vegetation cover
0.60 – 1.00	High density vegetation cover

Table 4.1: NDVI Values Interpretation

\*For values below '0', it is used 4 classes of gray and represents other materials.

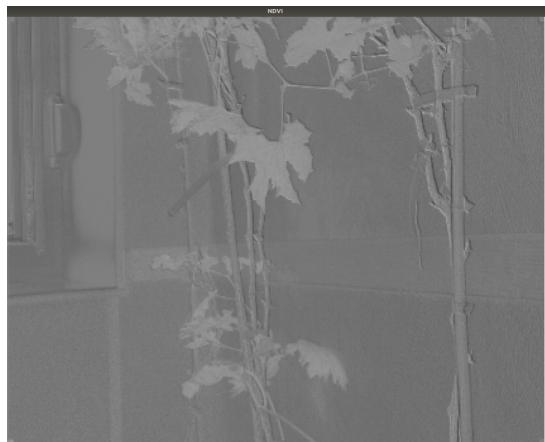


Figure 4.3: NDVI Normalized Image (with values from 0 to 255)

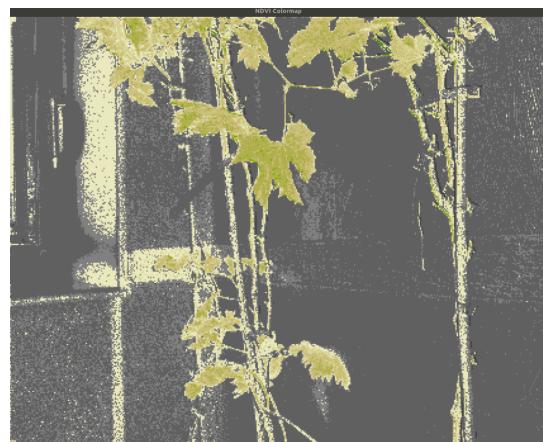


Figure 4.4: NDVI Colored Image by Using Custom Color-map

## 4.2 Multi-modal Image Registration

### 4.2.1 Overview

Image registration or image alignment is a process in which different types of data transformed into one coordinate space. The image data can be acquired from different modalities, such as RGB cameras, depth cameras, polarized cameras with different viewpoints, depth, resolution, etc. Especially in this project, the different modalities which have to be registered, are a multispectral camera and an RGB-D camera. Image registration is a very significant technique in computer vision, while new sensors become available every day in the market, but also it is a very challenging process. The different modalities can capture different types of data, resulting in more difficulties to extract information from them and finally match with each other.

In the current literature, image registration is divided into two main categories. The first is intensity-based image registration methods, in which used the similarity measurements in images via correlation metrics of intensity patterns between images to succeed the registration. The second is feature-based registration methods, which use features like edges, corners, points, lines, and contours of the images to succeed in the registration process. The procedure to succeed feature-based image registration is feature detection, feature points description, and feature points matching. There are many significant applications of image registration in computer vision for medical imaging, military application, space-based applications, etc [7, 8, 12, 20, 36, 41, 52, 65, 78]. In this project, feature-based image registration techniques are used in two main registration approaches. In the next section, the whole procedure is analyzed extensively.

### 4.2.2 Multispectral & Kinect Cameras Registration

During this project multiple experiments took place and they have been developed two different methods for image registration in ROS. Image registration proved to be very challenging for a robotic application due to the environmental conditions, the different modalities, etc. Furthermore, experiments with stereo vision techniques took place during the research and development of the image registration part. In the initial experiments were used the multispectral camera and depth camera of Kinect for the registration, but is replaced by the RGB in the final implementation. The main challenges that were faced during the experiments were the cameras' position, the different field of view of both cameras, but also the different specifications of the cameras, such as resolution, different lenses, etc.

Feature-based image registration is the basic method that is used in his project. Especially, feature detection and corner detection techniques are used. The whole idea is to detect the same features in both modalities and then register them using homography. The homography

matrix as a 3x3 matrix with 8 degrees of freedom as it is estimated up to a scale, is ideal for this procedure. Many experiments took place during the research and development of this part with all sensors of Kinect V2. The best results have been acquired using the image of the RGB camera and especially the Red Band of the Kinect V2 and the Band 3 or Red Band (as presented in Table 2.2) of the multispectral camera, due to the spectrum similarities [48].

A second approach that has been developed during this project uses chessboard corners. The chessboard corners should be detected from both modalities and then continue with the registration based on these corners. Furthermore, for the evaluation of the similarity of the registered images, the mean difference of multispectral image (Red Band) and the transformed image of Kinect (Red Band) after homography application, are used to extract a metric for acquiring the best result. The evaluation process is used in both approaches. Finally, the algorithm tries to find as many similarities in both modalities to continue to extract the homography matrix with the minimum mean difference. Below are presented the used equations which are integrated into the OpenCV functions, as the basic tool for the multi-modal image registration.

### 4.2.3 Mathematical Equations

The homography matrix is a 3x3 matrix with 8 degrees of freedom (DoF). The homography relates the transformation between two planes. When the homography is applied to every pixel, the new image is a warped version of the original image. In Figure 4.5 is presented the transformation between the planar surface and the image plane [48].

**Homography equations:**

$$H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = s \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \quad (4.4)$$

where

$$h_{33} = 1 \quad (4.5)$$

or

$$h_{11} + h_{12}^2 + h_{13}^2 + h_{21}^2 + h_{22}^2 + h_{23}^2 + h_{31}^2 + h_{32}^2 + h_{33}^2 = 1 \quad (4.6)$$

**Homography matrix:**

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \quad (4.7)$$

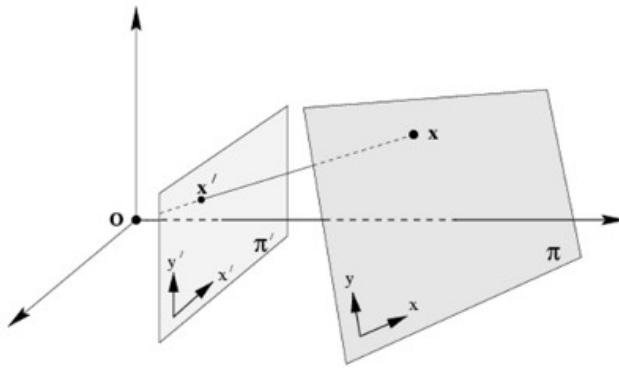


Figure 4.5: Illustration of a planar surface and the image plane [48]

**Equation for evaluation:**

$$MD = \frac{1}{m * n} * \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |M_{i,j} - K_{i,j}| \quad (4.8)$$

- MD: Mean difference of Multispectral and transformed Kinect images.
- M: Multispectral image (Band 3 - Red Band).
- K: Kinect RGB image (Red Band).
- m: Number of rows.
- n: Number of columns.

#### 4.2.4 Position of Sensors

The sensors' position could affect the final result, because the cameras that will be registered, need to see the same plane and must be as close as possible. So the final position of the multispectral camera is above the Kinect RGB camera. In Figure 4.6 is presented the final position of the cameras.



Figure 4.6: Multispectral and Kinect Cameras Fixed Position

#### 4.2.5 Image Registration with Feature Matching

During the multi-modal image registration between Multispectral Camera and RGB camera of Kinect sensor, many challenges had to be overtaken. These modalities have different image resolution, different fields of view, but also they capture images from different spectral bands, etc. During this approach, band 3 (Red Band as presented in Table 2.2) of the multispectral camera and the Red Band of the Kinect V2 RGB image sensor have been selected for registration. The raw image of the multispectral camera contains 9 pixels for every real-world corresponding pixel so it does not have precise results for the registration process. Below, are presented the followed steps of image registration for this approach and in Figure 4.7 is presented the implementation in ROS and OpenCV. Furthermore, in Figure 4.7 are presented the features of each image and the best matches between them. In Figure 4.8 is presented the RGB image from Kinect after the homography application. The left image view displays the real-time result and the right image view displays the best result that has captured.

Steps:

1. The first step is to acquire the band 3 (Red Band) from the raw image of the multispectral camera and then to interpolate for resizing by 3 to extract an image with dimensions 1273x1018.
2. The next step is feature extraction from both captured images of the different modalities. For the feature extraction, the ORB feature detector is used, as an alternative detector of SIFT and SURF detectors. No evaluation took place with the other feature extraction method, due to the time limits. Furthermore, the ORB feature detector can be easily used in ROS with OpenCV, as a significant part for the development of the final implementation. Moreover, for this approach, the library OpenCV provides a lot of the used functionalities. From both images, the ORB feature detector tries to find 1000 unique features. Then, the algorithm tries to find the matches between both modalities. A minimum threshold has been set of 40 feature matches for a successful registration [50].
3. When the algorithm has captured more than 40 features, feature matching is performed between the modalities.
4. The next step is the computation of the homography. Again, OpenCV is used to export the homography matrix. The OpenCV functions use the equations above to compute the homography matrix.
5. The homography matrix applied to the RGB image of Kinect, but also to the depth to export the registered images.

6. For evaluation purposes, the mean difference of Red Band of Kinect RGB image with band 3 (Red Band) of the multispectral image is applied. The lower the value means the better result. So the algorithm tries to capture a homography matrices with the maximum features, but with the minimum mean difference between the images.
7. The final step is the saving of the homography matrix by the user. Furthermore, the user can visually inspect the final result for any imperfections. The matrix must be captured one time and then it can be used for any further processing. The ROS tool rviz can be used to visualize the 3D images of the multispectral image with depth of Kinect sensor.

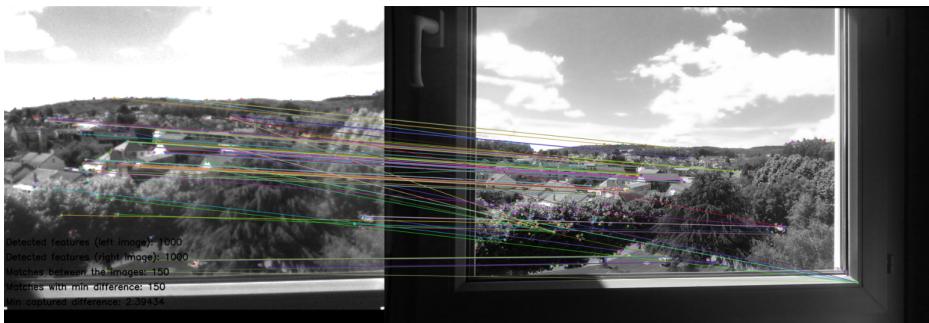


Figure 4.7: Registration with feature matching in ROS and OpenCV interface - Multispectral image (left), kinect image (right)



Figure 4.8: Kinect RGB image after homography application - Real-time result (left), best result (right)

#### 4.2.6 Image Registration with Corner Matching

The second image registration approach follows the same principles as the first approach. Some steps are similar as the feature matching approach but are included below. The main difference in this approach is that, instead of features, are used the corners of the chessboard same as for camera calibration. Below, are presented the followed steps of image registration for this approach. In Figure 4.9 are presented the captured corners of each image of each modality. In Figure 4.10 is presented the RGB image from Kinect after the homography application. The left image view displays the real-time result and the right image view displays the best result that has captured.

Steps:

1. Again, the first step is to acquire the band 3 (Red Band) from the raw image of the multispectral camera and then to interpolate for resizing by 3 to extract an image with dimension 1273x1018.
2. The next step is the corner detection of a printed chessboard, the same as the chessboard for camera calibration. The corners must be detected from both cameras. The implementation of this step uses OpenCV functionalities.
3. When the algorithm has captured the chessboard corners of each image, corner matching is performed between the modalities.
4. The next step is the computation of the homography. Again, OpenCV is used to export the homography matrix. The OpenCV functions use the equations above to compute the homography matrix.
5. The homography matrix applied to the RGB image of Kinect, but also to the depth to export the registered images.
6. For evaluation purposes, the mean difference of Red Band of Kinect RGB image with band 3 (Red Band) of the multispectral image is applied. The lower the value means the better result. So the algorithm tries to capture a homography matrices with the minimum mean difference between the images.
7. The final step is exactly the same step as the feature-based approach.



Figure 4.9: Registration with chessboard corner matching in ROS and OpenCV interface - Multispectral image (left), kinect image (right)

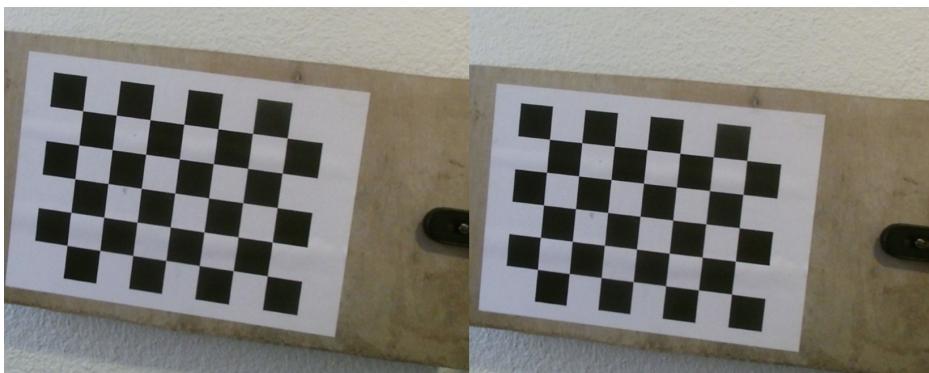


Figure 4.10: Kinect RGB image after homography application - Real-time result (left), best result (right)

## 4.3 3D Reconstruction

### 4.3.1 Overview

The 3D reconstruction is a process of creation of 3D scenes by capturing the shape using 3D points and the appearance using 2D images. The 3D reconstruction can be a very useful tool for visualization of the acquired and processed data. Data visualization with 3D modeling can be used in robotics, medical applications, space application, etc. In this project, 3D reconstruction can be very useful for agronomists. According to the acquired data, a 3D model of the vineyard can be developed.

The agronomists will have the opportunity to perform a visual inspection of the field without being present in the vineyard. The multimodal system as absolutely critical for this implementation will enrich the 3D images with more information from the multispectral camera for the inspected vegetation. The 3D reconstructed model of the vineyard can be delivered or can even

be enriched with the mapping of NDVI indexed images on the corresponding point-cloud. This model can be a very useful tool for vineyard management.

### 4.3.2 Methodology

3D reconstruction can be a highly challenging and resource-demanding process. In ROS, computer vision and 3D sensors are an integral part of the ecosystem. Rtabmap\_ros is a ROS package that is used for the 3D reconstruction of a scene. Especially rtabmap\_ros is a wrapper of package RTAB-Map (Real-Time Appearance-Based Mapping) for the ROS ecosystem, an RGB-D SLAM approach based on a global loop closure detector with real-time constraints.

In ROS, the rtabmap\_ros package is used to generate the 3D point cloud of multispectral with Kinect cameras registered images but also it can be used to create a 2D grid map for navigation. Generally, this package uses SIFT and SURF features for image stitching. Unfortunately, SIFT and SURF are not free for commercial use but can be used for research only. In the image registration section, SIFT and SURF feature methods have been avoided not only due to this reason but also due to the more challenging integration with the existing implementation [26–30].

The rtabmap\_ros package is based on a global Bayesian loop closure detector as an RGB-D Graph SLAM approach and it uses a bag of word approach. The whole approach is trying to determine if a new image is from a previously captured location or from a new location. When one of the two options is true, a new constrain is added to the map's graph and then the algorithm tries to minimize the error in the map [29, 30]. Some useful features are used for memory management to minimize the locations and graph optimization. This package can be used with a handheld Kinect, a stereo camera, an RGB-D camera, or on a robot equipped with a Lidar for mapping [26–28].

### 4.3.3 Implementation

This is one of the last parts of the project pipeline. The rtabmap\_ros package is used in the ROS ecosystem for the 3D reconstruction. So it is implemented after image registration and the whole processing and pre-processing parts. As input, it needs an image and a depth topic, but also the camera information. The registration node in both approaches, exports 3 image topics, in which one is the multispectral image (a single band), the transformed depth and RGB of the Kinect sensor. In Figure 4.11 is presented a 3D reconstructed scene.

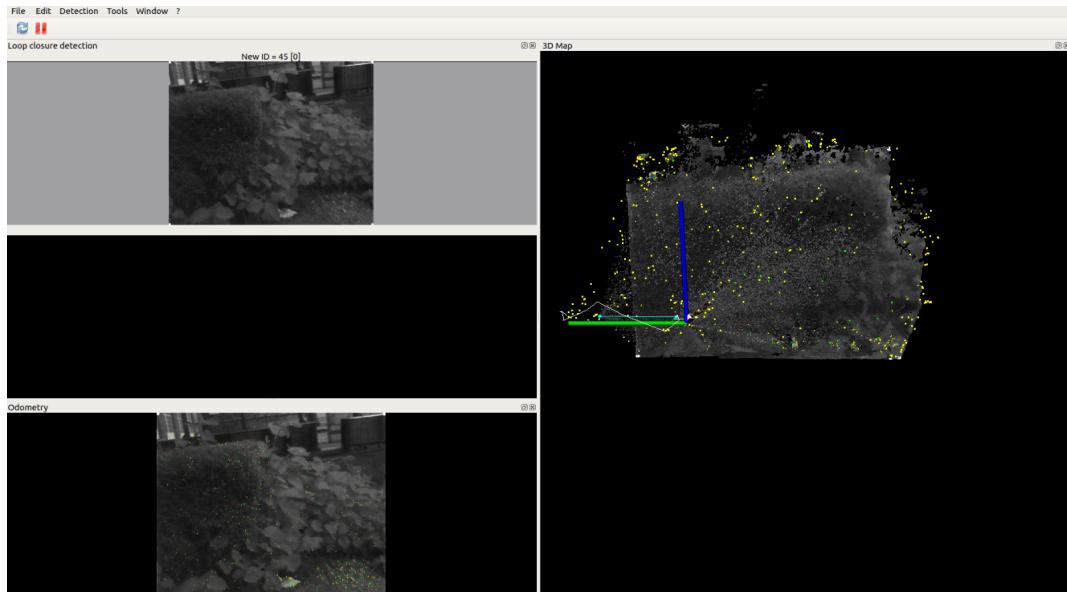


Figure 4.11: 3D reconstruction of an outdoor scene with band 3 of multispectral camera by using rtabmap\_ros package

## 4.4 Leaf Disease Detection using SVM Classifier

### 4.4.1 Overview

The decision part, as the last part, is absolutely vital for the whole project. The artificial intelligence has been chosen for this part as it can provide many capabilities and finally to export useful decisions. Many different methods have been examined and evaluated. The best methods which fit to this project have been chosen. Especially, this part receives the processed data from the previous layers, processes with the artificial intelligence methods, and finally exports useful information that can be used to the action layer.

Below are demonstrated the methodology that have been examined, some disadvantages and advantages, and the reason for the decision of the developed method. Of course, multiple methodologies can be used during this section, but the complexity of the development is increasing and the computational power also causes issues to this approach. Support vector machines (SVMs), as a method that is used for many years for different classification problems, and Convolutional Neural Networks (CNNs) that are newly used methods, providing high performance and amazing capabilities, are examined in this section, but only a SVM approach is demonstrated. Finally, this part needs more research as it can be a whole separate project.

#### 4.4.2 SVMs (Support Vector Machines)

SVMs or Support Vector Machines are supervised learning models that are used for classification or regression analysis, in machine learning. SVMs are introduced by Vladimir Vapnik and Hava Siegelmann and are used widely to categorize the data. An SVM model is a representation of the input data as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible.

The model must be trained with data that are marked as belonging to one of the existing categories. Then, the model tries to find the hyperplane with the maximum margin distance between the classes. SVM has many applications in many real-world problems such as text and image classification, handwriting recognition, data mining, bioinformatics, medicine, and even stock market [6, 10, 25, 68].

#### 4.4.3 Methodology

SVMs are very useful for classification or regression analysis. The decision of SVM for this project is due to the advantages and the disadvantages that they have in comparison with other methods. Mobility is absolutely vital for this project as a primary outdoor application. So the artificial intelligence module must be optimized to be run also offline.

The AI model must be integrated into the robotics system, but also the accuracy of the model must be taking into account. SVMs have chosen due to the accuracy in high dimensional spaces and memory efficiency. Some disadvantages that can appear are over-fitting for a big number of features and they are not computationally very efficient for big datasets. As it mentioned earlier, CNNs models for classification are examined and may be a good method for this project, are not demonstrated in this project.

#### 4.4.4 Implementation

In the current literature, SVMs are used for classification and especially for the disease detection of plants. For this task, the training data-set consists of two main categories which are healthy and diseased leaves. Initially, the conversion from RGB to HSV took place for the segmentation of the diseased parts of the leaf. The diseased parts have unique colors in HSV color space. Eventually, the segmentation of RGB image can be performed by using the mask as it is presented in Figure 4.14. Moreover, this implementation took place outside of the robotic system by using the Python programming language.

At the beginning, the feature extraction is performed for the important features that can divide the leaf into one of the two categories. The features that are used are Contrast, Correlation, Energy, Homogeneity, Mean, Standard Deviation, Entropy, RMS, etc. The next step is

the training of the model with the dataset that consists of 2000 images of leaves and the trained model extraction for further usage. A validation dataset, of 20 images of leaves, has been used to measure the accuracy with results about 95% to 97% accuracy of the model [10, 25]. Below are presented the selected features for extraction. In Figure 4.12 and in Figure 4.13 are presented the images in RGB and HSV color-space respectively. In Figure 4.14, in Figure 4.15 and in Figure 4.16 are presented the mask of diseased sections of the leaf, the diseased RGB sections of the leaf and the gray-scaled diseased sections of the leaf respectively.

#### Features:

- Contrast: The difference in luminance or colour that makes an object distinct.
- Correlation: The joint probability occurrence of the specified pixel pairs.
- Energy: The sum of squared elements in the Gray-Level Co-Occurrence Matrix (GLCM). When grey level intensities are very close to each other, the value of energy is small. It is also known as uniformity.
- Homogeneity: The closeness of the distribution of elements in the GLCM to the GLCM diagonal.
- Mean: The average of image array.
- Standard Deviation (STD): The computation of standard deviation of image array.
- Entropy: It is opposite of energy.
- Root Mean Square (RMS): The root mean square level of image array.
- Variance: The variance of the image.
- Smoothness: The smoothness of the image.
- Kurtosis: Kurtosis is a measure of whether the data are peaked or flat relative to a normal distribution.
- Skewness: It is a measure of symmetry, or more precisely, the lack of symmetry.
- Inverse Difference Moment (IDM): It is the opposite of contrast and it is also called local homogeneity.

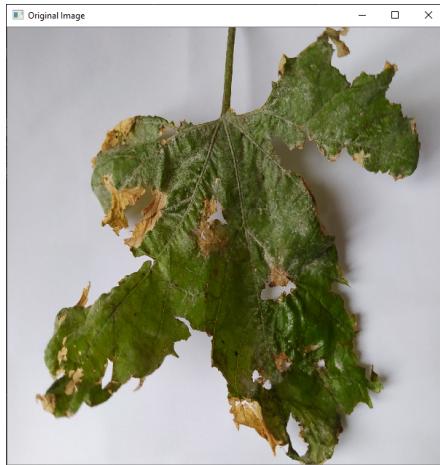


Figure 4.12: Leaf image in RGB color-space

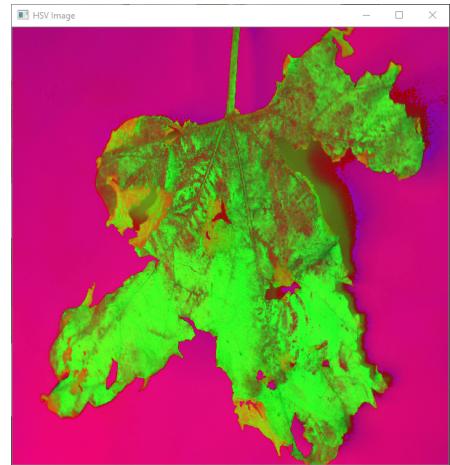


Figure 4.13: Leaf image in HSV color-space

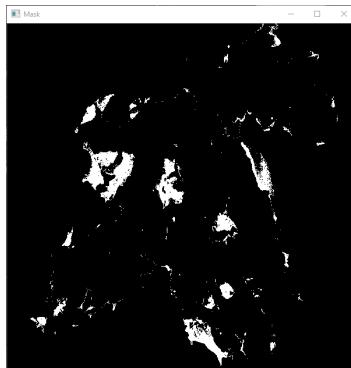


Figure 4.14: Mask for diseased sections of the leaf

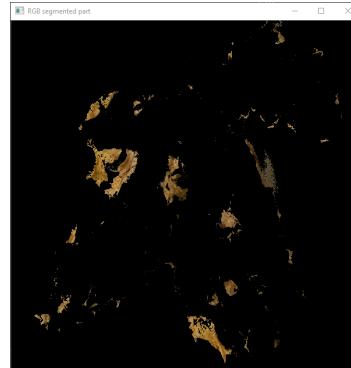


Figure 4.15: Diseased sections in RGB color-space of the leaf



Figure 4.16: Gray-scaled diseased sections of the leaf

## 4.5 Conclusion

In the Chapter 3 (Pre-processing) and in the Chapter 4 (Processing) was presented the whole pipeline that is followed, divided into these two chapters. Every part consumed enough time for research and development. As a result, some sections are not implemented in the Robotics System such as the leaf disease detector. Other parts need more work and more experiments, not only indoors but also outdoors, for better results.

Especially, the whole Pre-processing Chapter is fully implemented. From the Processing Chapter, a lot of Vegetation indices have been examined, developed, and tested to the robotic system. Furthermore, the 3D reconstruction implementation, is not so accurate in outdoor conditions, due to the multispectral special features. Most of the experiments took place in indoor conditions resulting in not so accurate results for implementation in the vineyard. Hence, some tools for simulation are developed and other packages from ROS ecosystem are used to simulate outdoor conditions. Finally, leaf disease detector and generally the last section has only implemented and tested under ideal conditions and not in the robotic system.

# Chapter 5

## Results

In this chapter, are presented the results of the project, but also some functionalities of the ROS implementation of this project. Almost all outside experiments took place near the robotics lab, by using a plant in front of a brick wall as a background as presented below. The other experiments took place indoors. Furthermore, the results of the leaf disease detector with the SVM classifier are not presented in this chapter because the only acquired results have been presented during in the Chapter 4 of Processing.

Below are demonstrated the figures for every part of the project as demonstrated in the pipeline. In Figure 5.1 and in Figure 5.2 are presented the raw acquired images by both modalities. In Figure 5.3 are presented the sub-band images after the separation. In Figure 5.4 and in Figure 5.5 are presented the separated band 8 and band 1 respectively, as it is presented in Table 2.2.



Figure 5.1: Raw Acquired Image By Multispectral Camera



Figure 5.2: Raw Acquired Image By Kinect RGB Camera

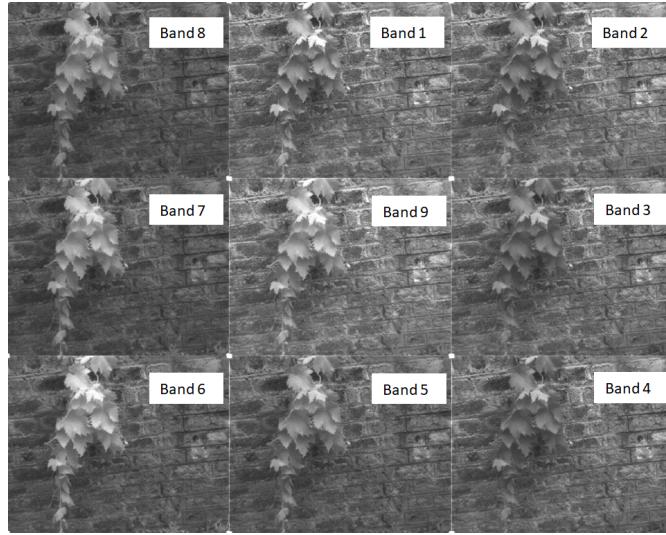


Figure 5.3: Sub-Bands After Separation



Figure 5.4: Multispectral Camera - Band 8



Figure 5.5: Multispectral Camera - Band 1

In Figure 5.6 and in Figure 5.7 is presented the image of the separated band 8 before and after flat-field correction. As can be observed, the quality of the corrected image has improved. Most of the unwanted artifacts have been removed and even the pixels that used for alignment in the corners have removed.



Figure 5.6: Multispectral Camera - Band 8  
Before Flat-field Correction



Figure 5.7: Multispectral Camera - Band 8  
After Flat-field Correction

In Figure 5.8 and in Figure 5.9 is presented the image of the separated band 8 before and after white reference normalization. For this procedure is used the Macbeth Color Checker to select the white reference area. The white color is in the green box in both images. As can be observed by the images, the white color in the normalized image is almost fully corrected.

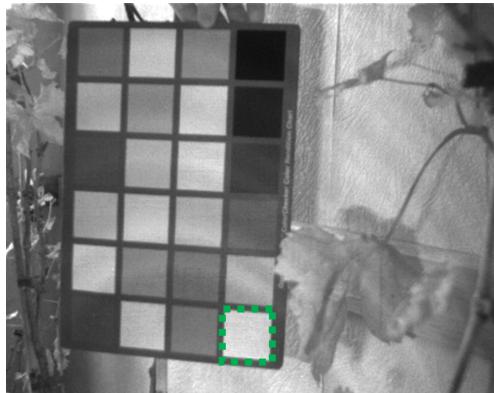


Figure 5.8: Multispectral Camera - Band 8  
Before White Reference Normalization

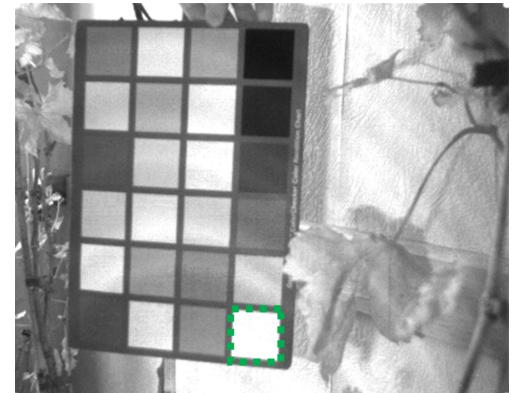


Figure 5.9: Multispectral Camera - Band 8  
After White Reference Normalization

In Figure 5.10 and in Figure 5.11 is presented the image of the separated band 8 before and after crosstalk correction. As can be observed in Figure 5.11, the contrast has improved significantly and the plant can easily be segmented by the brick wall.



Figure 5.10: Multispectral Camera - Band 8 Before Crosstalk Correction



Figure 5.11: Multispectral Camera - Band 8 After Crosstalk Correction

In Figure 5.12 and in Figure 5.13 are presented the images after the computation of NDVI, its normalization, and finally the application of the custom color-map. As can be observed in Figure 5.12, the vegetation is distinguishable. Especially, for the colored image in Figure 5.13, the values of the vegetation have interpreted, and the values for the other materials have ignored. Coloring can be proved very useful for the segmentation of the vegetation. In Figure 5.14 is presented the colored image after crosstalk correction. As can be observed the contrast is increased and the vegetation is more distinct.

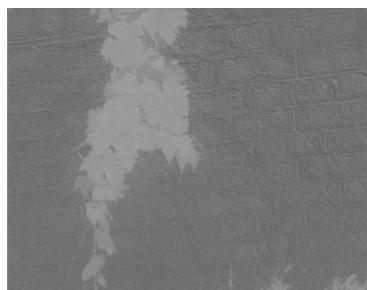


Figure 5.12: NDVI 8bit Image  
Figure 5.13: NDVI Colored Im-  
(After normalization to 0 - 255) age (Using custom color-map)

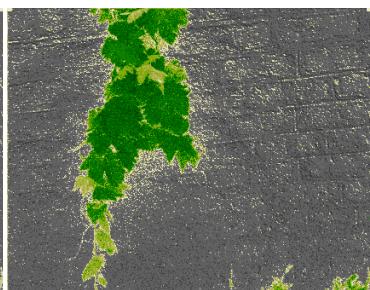


Figure 5.14: NDVI Colored Im-  
(After crosstalk correction)



Figure 5.15: NDVI Color-map

In Figure 5.16 is presented the segmented vegetation by using Otsu's method for automatic image thresholding. In Figure 5.17 is presented the segmented vegetation after erosion and dilation for quality improvement.

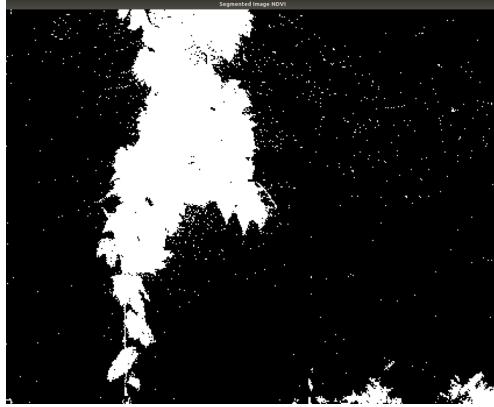


Figure 5.16: Background subtraction by using Otsu's method



Figure 5.17: Background subtraction after erosion and dilation

In Figure 5.18 and in Figure 5.19 are presented the images for both modalities that are going to be registered with feature matching and with chessboard corner matching respectively. This Figure demonstrates the ROS and OpenCV interface with the detected features in both images, the matches, and other information in the left of the Figure. It has been followed a similar procedure with feature-based registration, but here are used corners of the chessboard instead of features as it is presented above. When the corners are detected from the camera, it is displayed that the chessboard is detected but also the corners are displayed in the image view.

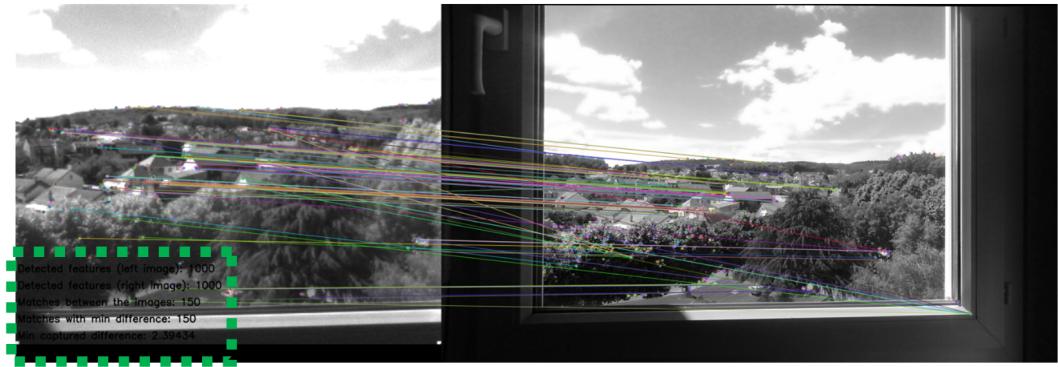


Figure 5.18: Image Registration with Feature Matching

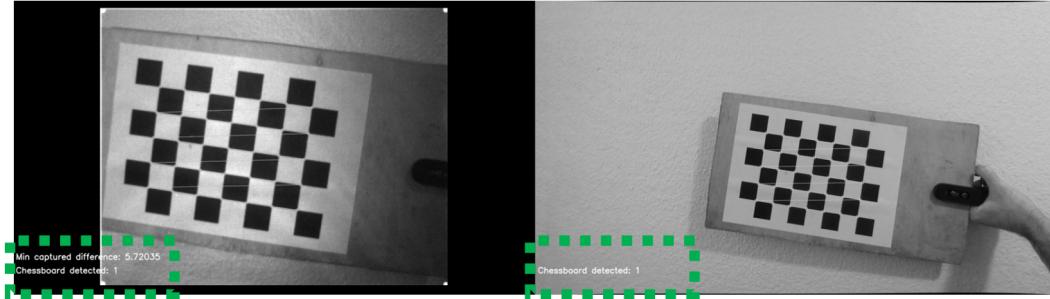


Figure 5.19: Image Registration with Chessboard Corner Matching

In Figure 5.20 and in Figure 5.21 are presented the 3D images after multi-modal image registration. In Figure 5.20 is presented an early test of 3D image visualization. In Figure 5.22 is presented the Point Cloud of them. All experiments took place indoors and the rviz package is used for the visualization.



Figure 5.20: 3D Image (Early Figure 5.21: 3D Image (Final Figure 5.22: Point Cloud Generation - Rviz  
Experiments) - Rviz                      Result) - Rviz                      - Rviz

In Figure 5.23 and in Figure 5.24 are presented the 3D reconstructed scenes in outdoor and indoor conditions respectively. It is used the rtabmap\_ros package for the procedure. The 3D reconstruction procedure is very sensitive to outdoor environmental conditions. Finally, in Figure 5.25, in Figure 5.26 and in Figure 5.27 is presented the web application that is developed for the remote controlling of the camera during the outdoor experiments.

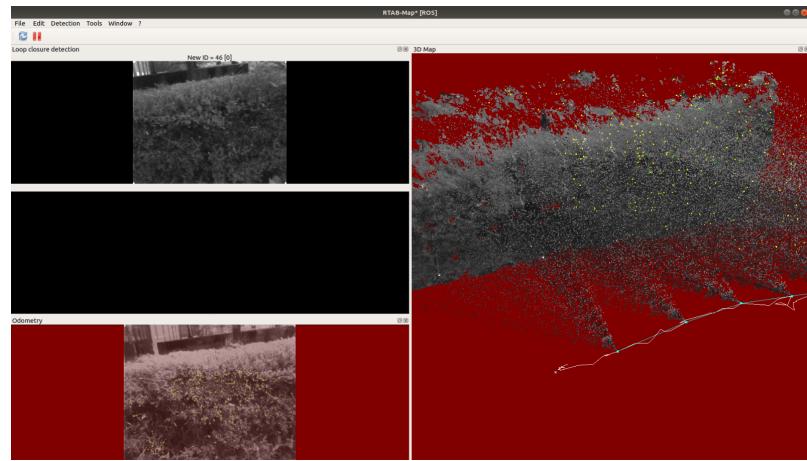


Figure 5.23: 3D reconstruction of outdoor scene with multispectral camera after multi-modal registration

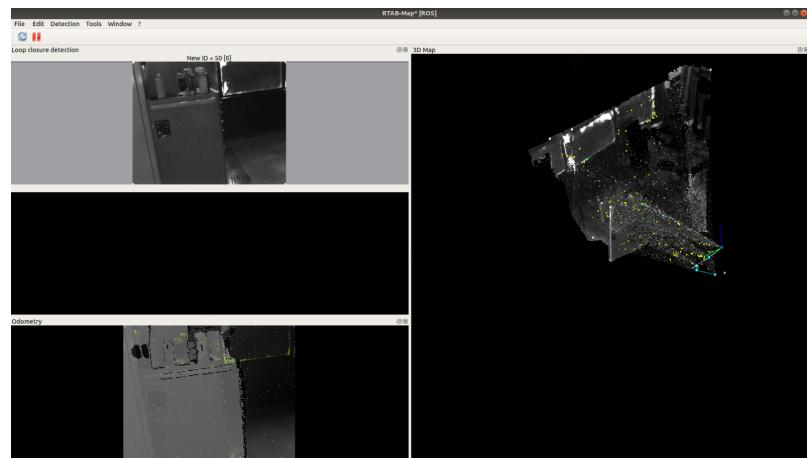


Figure 5.24: 3D reconstruction of indoor scene with multispectral camera after multi-modal registration

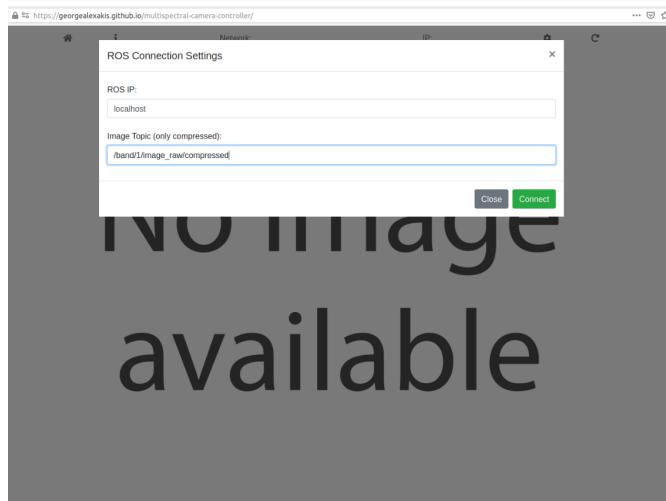


Figure 5.25: Remote Camera Controller - Connection settings

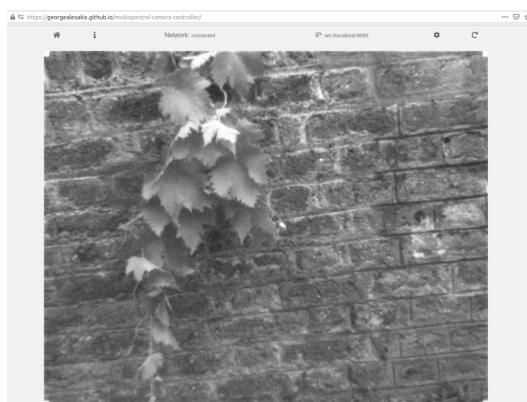


Figure 5.26: Remote Camera Controller - Image view

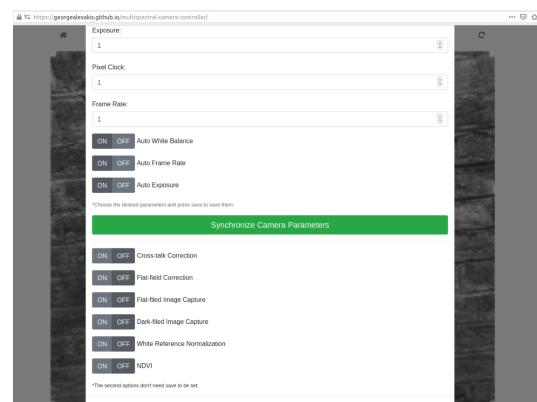


Figure 5.27: Remote Camera Controller - Settings view

# Chapter 6

## Conclusions

This project demonstrates the pipeline of the full implementation of vineyard analysis. Especially, the pipeline of the project has divided into Chapter 3 and Chapter 4. In this project are presented all steps, by beginning from real-time acquisition, pre-processing, processing, multi-modal image registration, and lastly the analysis with artificial intelligence. Every step is presented as a unique project in which is presented the methodology and its implementation. For a successful implementation, all parts of the pipeline must be integrated into the ROS robotic ecosystem and to work in real-time.

All steps consumed a lot of time and work for research, development, and finally the evaluation of the results. Many challenges were faced during the development, most of them have been addressed, but many new arose during the project. Most of the experiments took place indoors, a few experiments took place outdoors, and no experiments took place in the vineyard, due to the lock-down. Furthermore, due to the limited access to the lab equipment, an RGB camera, and some developed simulation nodes are used for most of the research and the development of the project. Although, during the limited outdoor experiments, it observed that the difference in luminance and even the wind can modify the final results.

ROS Melodic and OpenCV are the basic tools that have been used for this project. The most functionalities of them are optimized, resulting in real-time and accurate data processing. The pre-processing part of the project is almost completed, but the evaluation of each sub-part is necessary. Generally, an extensive evaluation of every implemented part must be done in future work. In the processing part, two approaches of multi-modal image registration were presented, which seem accurate, but need more time to evaluate and compare with other preexisting methods. Moreover, the AI module needs to be integrated into the robotic system. In future work, it can be extended and improved. Finally, the 3D reconstruction with NDVI images or with other vegetation indices must be examined as a very useful tool for the agronomists.

# Bibliography

- [1] Georgios Alexakis. Multispectral camera controller. <https://georgealexakis.github.io/multispectral-camera-controller>.
- [2] Stefano Andriani and Harald Brendel. Crosstalk correction technique for single sensor camera provided with bayer color filter array. In *2013 IEEE International Conference on Image Processing*, pages 2252–2255. IEEE, 2013.
- [3] Chein-I Chang. *Hyperspectral imaging: techniques for spectral detection and classification*, volume 1. Springer Science & Business Media, 2003.
- [4] Jon Chouinard. The fundamentals of camera and image sensor technology. [https://www.visiononline.org/userassets/aiauploads/file/cvp-the-fundamentals-of-camera-and-image-sensor-technology\\_jon-chouinard.pdf](https://www.visiononline.org/userassets/aiauploads/file/cvp-the-fundamentals-of-camera-and-image-sensor-technology_jon-chouinard.pdf).
- [5] SiliOS Company. Multispectral cameras cms series. <https://www.silos.com/cms-series>.
- [6] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [7] Nathan Crombez, Ralph Seulin, Olivier Morel, David Fofi, and Cédric Demonceaux. Multi-modal 2d image to 3d model registration via a mutual alignment of sparse and dense visual features. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6316–6322. IEEE, 2018.
- [8] Mário André Pinto Ferraz de Aguiar. 3d reconstruction from multiple rgb-d images with different perspectives. 2015.
- [9] Universitat de Lleida. Precision agriculture definitions. [http://www.grap.udl.cat/en/presentation/pa\\_definitions.html](http://www.grap.udl.cat/en/presentation/pa_definitions.html).

- [10] T. Gayathri Devi, A. Srinivasan, and S. Sudha. Computer vision based detection and classification of tomato leaf diseases. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, pages 2278–3075, 2019.
- [11] Science Mission Directorate. Introduction to the electromagnetic spectrum. [http://science.nasa.gov/ems/01\\_intro](http://science.nasa.gov/ems/01_intro).
- [12] Clément Douarre, Carlos Crispim-Junior, Anthony Gelibert, Laure Tougne, and David Rousseau. A strategy for multimodal canopy images registration. 2019.
- [13] Alexander Getman, Timofei Uvarov, Y Han, Bumsuk Kim, J Ahn, and Y Lee. Crosstalk, color tint and shading correction for small pixel size image sensor. In *International Image Sensor Workshop*, pages 166–169, 2007.
- [14] AC Goldberg, B Stann, and N Gupta. Multispectral, hyperspectral, and three-dimensional imaging research at the us army research laboratory. Technical report, Army Research Lab Adelphi MD, 2003.
- [15] Hans Grahn and Paul Geladi. *Techniques and applications of hyperspectral image analysis*. John Wiley & Sons, 2007.
- [16] Nathan Hagen. Flatfield correction errors due to spectral mismatching. *Optical Engineering*, 53(12):123107, 2014.
- [17] Radu Horaud, Miles Hansard, Georgios Evangelidis, and Clément Ménier. An overview of depth cameras and range scanners based on time-of-flight technologies. *Machine vision and applications*, 27(7):1005–1020, 2016.
- [18] IDS. Bandwidth management with adjustable pixel clock. [https://fr.ids-imaging.com/tl\\_files/downloads/techtip/TechTip\\_PixelClock\\_EN.pdf](https://fr.ids-imaging.com/tl_files/downloads/techtip/TechTip_PixelClock_EN.pdf).
- [19] IDS. Ids camera drivers. <https://en.ids-imaging.com/download-ueye-lin64.html>.
- [20] Md Baharul Islam and Mir Md Jahangir Kabir. A new feature-based image registration algorithm. *Computer Technology and Application*, 4(2), 2013.
- [21] Akira Iwasaki and Hideyuki Tonooka. Validation of a crosstalk correction algorithm for aster/swir. *IEEE transactions on Geoscience and Remote Sensing*, 43(12):2747–2751, 2005.
- [22] Peet Kask, Kaupo Palo, Chris Hinnah, and Thora Pommerencke. Flat field correction for high-throughput imaging of fluorescent samples. *Journal of microscopy*, 263(3):328–340, 2016.

- [23] Alexander Kokka, Tomi Pulli, Eija Honkavaara, Lauri Markelin, Petri Kärhä, and Erkki Ikonen. Flat-field calibration method for hyperspectral frame cameras. *Metrologia*, 56(5):055001, 2019.
- [24] Anis Koubaa. *Robot Operating System (ROS): The Complete Reference*, volume 3. Springer, 2018.
- [25] Pooja Kulinavar and Vidya I. Hadimani. Classification of leaf disease based on multi-class svm classifier. *International Journal of Advance Research, Ideas and Innovations in Technology*, 3:321–327, 2017.
- [26] Mathieu Labb   and Fran  ois Michaud. Memory management for real-time appearance-based loop closure detection. In *2011 IEEE/RSJ international conference on intelligent robots and systems*, pages 1271–1276. IEEE, 2011.
- [27] Mathieu Labb   and Fran  ois Michaud. Appearance-based loop closure detection for online large-scale and long-term operation. *IEEE Transactions on Robotics*, 29(3):734–745, 2013.
- [28] Mathieu Labbe and Fran  ois Michaud. Online global loop closure detection for large-scale multi-session graph-based slam. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2661–2666. IEEE, 2014.
- [29] Mathieu Labb   and Fran  ois Michaud. Long-term online multi-session graph-based slam with memory management. *Autonomous Robots*, 42(6):1133–1150, 2018.
- [30] Mathieu Labb   and Fran  ois Michaud. Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *Journal of Field Robotics*, 36(2):416–446, 2019.
- [31] E Lachat, H Macher, MA Mittet, T Landes, and P Grussenmeyer. First experiences with kinect v2 sensor for close range 3d modelling. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(5):93, 2015.
- [32] Robert Lagani  re. *OpenCV 3 computer vision application programming cookbook*. Packt Publishing Ltd, 2017.
- [33] Dana Lahat, T  lay Adali, and Christian Jutten. Multimodal data fusion: an overview of methods, challenges, and prospects. *Proceedings of the IEEE*, 103(9):1449–1477, 2015.
- [34] Wanqing Li, Philip Ogunbona, Yu Shi, and Igor Kharitonenko. Cmos sensor cross-talk compensation for digital cameras. *IEEE Transactions on Consumer Electronics*, 48(2):292–297, 2002.

- [35] MacDonald Lindsay, Giacometti Alejandro, Campagnolo Alberto, Robson Stuart, Weyrich Tim, Terras Melissa, and Gibson Adam. Multispectral imaging of degraded parchment. In Shoji Tominaga, Raimondo Schettini, and Alain Tréneau, editors, *Computational Color Imaging*, pages 143–157, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN: 978-3-642-36700-7.
- [36] Huajian Liu, Sang-Heon Lee, and Javaan Singh Chahl. Registration of multispectral 3d points for plant inspection. *Precision Agriculture*, 19(3):513–536, 2018.
- [37] Dieu Sang Ly, Cédric Demonceaux, Pascal Vasseur, and Claude Pégard. Extrinsic calibration of heterogeneous cameras by line images. *Machine vision and applications*, 25(6):1601–1614, 2014.
- [38] Henrique S Malvar, Li-wei He, and Ross Cutler. High-quality linear interpolation for demosaicing of bayer-patterned color images. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages iii–485. IEEE, 2004.
- [39] MathWorks. What is camera calibration? <https://www.mathworks.com/help/vision/ug/camera-calibration.html>.
- [40] Takeshi Motohka, Kenlo Nishida Nasahara, Hiroyuki Oguma, and Satoshi Tsuchida. Applicability of green-red vegetation index for remote sensing of vegetation phenology. *Remote Sensing*, 2(10):2369–2387, 2010.
- [41] Sayan Nag. Image registration techniques: a survey. *arXiv preprint arXiv:1712.07540*, 2017.
- [42] NASA. The electromagnetic spectrum. <https://imagine.gsfc.nasa.gov/science/toolbox/emspectrum1.html>.
- [43] NASA. Normalized difference vegetation index (ndvi). [https://earthobservatory.nasa.gov/features/MeasuringVegetation/measuring\\_vegetation\\_2.php](https://earthobservatory.nasa.gov/features/MeasuringVegetation/measuring_vegetation_2.php).
- [44] OECD. Agriculture and the environment. <https://www.oecd.org/agriculture/topics/agriculture-and-the-environment>.
- [45] University of La Rioja. Winerobot. <http://www.vinerobot.eu>.
- [46] International Society of Precision Agriculture. Precision ag definition - language modal. <https://www.ispag.org/about/definition>.
- [47] Inc. (OSRF) Open Source Robotics Foundation. Open source robotics foundation, inc. (osrf). <https://www.openrobotics.org>.

- [48] OpenCV. Basic concepts of the homography explained with code. [https://docs.opencv.org/master/d9/dab/tutorial\\_homography.html](https://docs.opencv.org/master/d9/dab/tutorial_homography.html).
- [49] OpenCV. Camera calibration and 3d reconstruction. [https://docs.opencv.org/2.4/modules/calib3d/doc/camera\\_calibration\\_and\\_3d\\_reconstruction.html](https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html).
- [50] OpenCV. Feature matching. [https://docs.opencv.org/master/dc/dc3/tutorial\\_py\\_matcher.html](https://docs.opencv.org/master/dc/dc3/tutorial_py_matcher.html).
- [51] Robot Operating System (ROS) Organization. Ros. <https://www.ros.org>.
- [52] Cristina Palmero, Albert Clapés, Chris Bahnsen, Andreas Møgelmose, Thomas B Moeslund, and Sergio Escalera. Multi-modal rgb-depth-thermal human body segmentation. *International Journal of Computer Vision*, 118(2):217–239, 2016.
- [53] Sudhanshu Sekhar Panda, Daniel P Ames, and Suranjan Panigrahi. Application of vegetation indices for agricultural crop yield prediction using neural network techniques. *Remote Sensing*, 2(3):673–696, 2010.
- [54] Parrot. Parrot sequoia+. <https://www.parrot.com/business-solutions-us/parrot-professional/parrot-sequoia>.
- [55] Cristiano Premebida, Rares Ambrus, and Zoltan-Csaba Marton. Intelligent robotic perception systems. In Efren Gorrostieta Hurtado, editor, *Applications of Mobile Robots*. IntechOpen, 2018. DOI: 10.5772/intechopen.79742.
- [56] Vitirover Micro Winery Robotics. Saint-Émilion: Vitirover. <http://www.vitirover.com>.
- [57] Robotnik. summit-xl. <https://www.robotnik.eu/mobile-robots/summit-xl>.
- [58] ROS. camera\_calibration. [http://wiki.ros.org/camera\\_calibration](http://wiki.ros.org/camera_calibration).
- [59] ROS. rosbridge\_suite. [http://wiki.ros.org/rosbridge\\_suite](http://wiki.ros.org/rosbridge_suite).
- [60] ROS. roslibjs. <http://wiki.ros.org/roslibjs>.
- [61] ROS. rqt\_reconfigure. [http://wiki.ros.org/rqt\\_reconfigure](http://wiki.ros.org/rqt_reconfigure).
- [62] ROS. ueye\_cam. [http://wiki.ros.org/ueye\\_cam](http://wiki.ros.org/ueye_cam).
- [63] Joaquim Salvi et al. *An approach to coded structured light to obtain three dimensional information*. Universitat de Girona, 1998.

- [64] Vincent Sauget, Marc Hubert, Aurélien Faiola, and Stéphane Tisserand. Application note for cms camera and cms sensor users: post-processing method for crosstalk reduction in multispectral data and images. *white paper, SiliOS Technologies*, 2018.
- [65] Nick Schneider, Florian Piewak, Christoph Stiller, and Uwe Franke. Regnet: Multimodal sensor registration using deep neural networks. In *2017 IEEE intelligent vehicles symposium (IV)*, pages 1803–1810. IEEE, 2017.
- [66] James Anthony Seibert, John M Boone, and Karen K Lindfors. Flat-field correction technique for digital detectors. In *Medical Imaging 1998: Physics of Medical Imaging*, volume 3336, pages 348–354. International Society for Optics and Photonics, 1998.
- [67] Aaron Staranowicz, Garrett R Brown, Fabio Morbidi, and Gian Luca Mariottini. Easy-to-use and accurate calibration of rgbd cameras from spheres. In *Pacific-Rim Symposium on Image and Video Technology*, pages 265–278. Springer, 2013.
- [68] Cecilia Sullca, Carlos Molina, Carlos Rodríguez, and Thais Fernández. Diseases detection in blueberry leaves using computer vision and machine learning techniques. *International Journal of Machine Learning and Computing*, 9:656–661, 10 2019. doi: 10.18178/ijmlc.2019.9.5.854.
- [69] Vinbot Team. Vinbot. <http://vinbot.eu>.
- [70] Tetracam. Auk 3e (awc3e). <http://www.tetracam.com/Products-Auk3E.htm>.
- [71] Tetracam. Hawk (hawc). <http://www.tetracam.com/Products-Hawk.htm>.
- [72] Thiemo Wiedemeyer. Iai kinect2. [https://github.com/code-iai/iai\\_kinect2](https://github.com/code-iai/iai_kinect2).
- [73] Wikipedia. Electromagnetic spectrum. [https://en.wikipedia.org/wiki/Electromagnetic\\_spectrum](https://en.wikipedia.org/wiki/Electromagnetic_spectrum).
- [74] Wikipedia. Kinect. <https://en.wikipedia.org/wiki/Kinect>.
- [75] WillAgri. The digital revolution, precision agriculture and conservation farming. <https://www.willagri.com/2020/01/20/the-digital-revolution-precision-agriculture-and-conservation-farming>.
- [76] MAIA WV. Maia wv – the multispectral camera. <https://www.spectralcam.com/maia-tech>.
- [77] Wall ye Softwares and Robots. Wall-ye. <http://www.wall-ye.com>.

- [78] Barbara Zitova and Jan Flusser. Image registration methods: a survey. *Image and vision computing*, 21(11):977–1000, 2003.
- [79] Veronika Zuzulova and Jaroslav Vido. Normalized difference vegetation index as a tool for the evaluation of agricultural drought. *Ecocycles*, 4(1):83–87, 2018.