

**UNIVERSITÀ DEGLI STUDI DI MILANO**

**SCUOLA DI DOTTORATO IN INFORMATICA**

**Dipartimento di Informatica**

Via Comelico 39/41, 20135 Milano



*Elaborazione dei Segnali Multi-dimensionali e Applicazioni  
XXIV Ciclo*

**Adaptive processing architecture of multi-sensor  
signals for low-impact treatments of plant diseases.**

**INF/01**

Studente di dottorato: **Paolo Tirelli**

Tutor: Prof. N. A. Borghese

Correlatore: Prof. Roberto Oberti

Direttore: Prof. Ernesto Damiani

**A.A. 2010/2011**



## Summary

<b>UNIVERSITÀ DEGLI STUDI DI MILANO</b> .....	<b>1</b>
<b>1. INTELLIGENT SENSING FOR PRODUCTION OF HIGH-VALUE CROPS</b> .....	<b>5</b>
1.1. SCIENTIFIC AND TECHNICAL QUALITY .....	5
1.1.1. <i>Concept and objectives</i> .....	5
1.1.2. <i>Progress beyond the state-of-the-art</i> .....	6
<b>2. PROBLEM DEFINITION</b> .....	<b>8</b>
<b>3. LITERATURE REVIEW</b> .....	<b>12</b>
3.1. PRECISION AGRICULTURE .....	12
3.2. PRECISION CROP PROTECTION AND MONITORING OF PLANT DISEASES .....	12
3.3. OPTICAL SENSOR SYSTEMS .....	13
3.4. REFLECTION OF VEGETATION .....	14
3.5. MULTISPECTRAL SENSORS FOR DISEASE DETECTION .....	17
3.6. ANALYSIS OF MULTISPECTRAL DATA.....	18
<b>4. BACKGROUND THEORY</b> .....	<b>20</b>
4.1. PATTERN RECOGNITION METHODS.....	20
4.1.1. <i>Overview</i> .....	20
4.1.2. <i>Uses</i> .....	22
4.1.3. <i>Algorithms</i> .....	22
4.2. IMAGE PROCESSING METHODS .....	41
4.2.1. <i>Image Arithmetic</i> .....	42
4.2.2. <i>Point Operations</i> .....	43
4.2.3. <i>Geometric Operations</i> .....	45
4.2.4. <i>Image Analysis</i> .....	45
4.2.5. <i>Morphology</i> .....	46
4.2.6. <i>Digital Filters</i> .....	47
4.2.7. <i>Feature Detectors</i> .....	49
4.2.8. <i>Image Transforms</i> .....	50
4.2.9. <i>Image Synthesis</i> .....	54
4.3. ALGORITHMS EVALUATION METHOD.....	55
4.3.1. <i>PPV, NPV, Sensitivity and Specificity</i> .....	55
<b>5. MATERIALS AND METHODS</b> .....	<b>56</b>
5.1. DATA MEASUREMENTS .....	56
5.1.1. <i>Spectral Reflectance Measurements</i> .....	56
5.1.2. <i>Multispectral Imaging Measurements</i> .....	58
5.2. MATERIALS .....	60
5.2.1. <i>Laboratory tests</i> .....	60
5.2.2. <i>Greenhouse tests</i> .....	66
5.2.3. <i>Field tests</i> .....	69
5.3. SPECTRAL REFLECTANCE PROCESSING .....	84
5.3.1. <i>Laboratory Methods</i> .....	84
5.4. MULTI-SPECTRAL IMAGE PROCESSING.....	93
5.4.1. <i>Laboratory Methods</i> .....	93
5.4.2. <i>Greenhouse Methods</i> .....	104
5.4.3. <i>Field Methods</i> .....	110

<b>6. CONCLUSIONS</b> .....	<b>129</b>
<b>COLLABORATIONS</b> .....	<b>130</b>
<b>PUBLICATIONS</b> .....	<b>131</b>
<b>ACKNOWLEDGEMENTS</b> .....	<b>132</b>
<b>BIBLIOGRAPHY</b> .....	<b>133</b>
<b>APPENDICES</b> .....	<b>144</b>
<b>A. COMMERCIAL SOFTWARE USED</b> .....	<b>144</b>
A.1. <i>Matlab R2009b v7.9</i> .....	<i>144</i>
A.2. <i>Microsoft Visual Studio 2008 v9.0</i> .....	<i>144</i>
A.3. <i>MySQL v5.5</i> .....	<i>144</i>
A.4. <i>Paint .NET v3.5</i> .....	<i>145</i>
A.5. <i>Mercurial SCM v1.9</i> .....	<i>145</i>
<b>B. APPLICATION SOFTWARE DEVELOPED</b> .....	<b>146</b>
B.1. <i>Images Acquisition</i> .....	<i>146</i>
B.2. <i>Filters Chain Definition</i> .....	<i>147</i>
B.3. <i>ROIs Selection</i> .....	<i>148</i>
<b>C. HARDWARE SPECIFICATIONS</b> .....	<b>150</b>

# **1. Intelligent sensing for production of high-value crops**

## **1.1. Scientific and technical quality**

This thesis has been realized within the CROPS project.

CROPS will develop scientific know-how for a highly configurable, modular and clever carrier platform that includes modular parallel manipulators and intelligent tools (sensors, algorithms, sprayers, grippers) that can be easily installed onto the carrier and are capable of adapting to new tasks and conditions. Several technological demonstrators will be developed for high value crops like greenhouse vegetables, fruits in orchards, and grapes for premium wines. The CROPS robotic platform will be capable of site-specific spraying (targets spray only towards foliage and selective targets) and selective harvesting of fruit (detects the fruit, determines its ripeness, moves towards the fruit, grasps it and softly detaches it). Another objective of CROPS is to develop techniques for reliable detection and classification of obstacles and other objects to enable successful autonomous navigation and operation in plantations and forests. The agricultural and forestry applications share many research areas, primarily regarding sensing and learning capabilities. The project started in October 2010 and will run for 48 month.

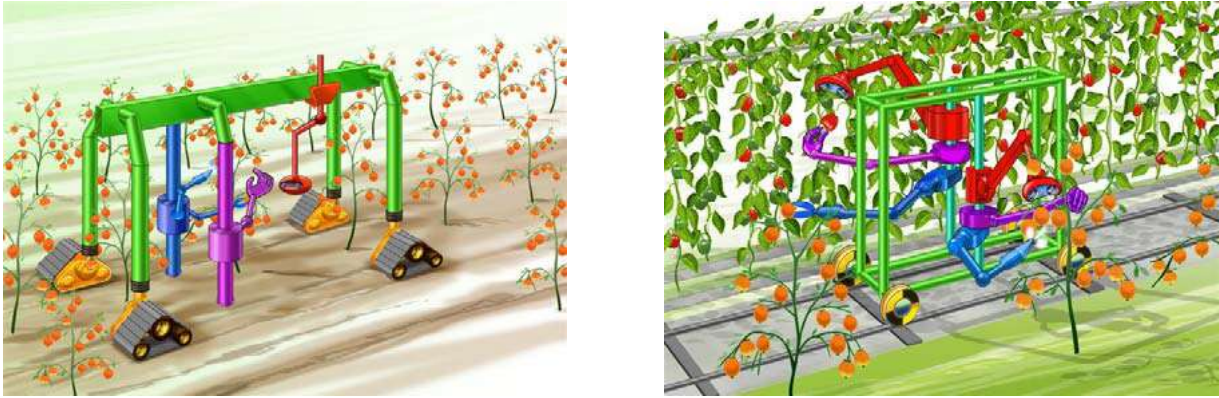
The aim of this thesis is to lay the foundations, suggesting the guidelines, of one task addressed by the CROPS project, in particular, the aim of this work is to study the application of a VIS-NIR imaging approach (intelligent sensing), based on a relatively simple algorithm, to detect symptoms of powdery mildew and downy mildew disease at early stages of infection (sustainable production of high-value crops). Also a preliminary work for botrytis detection will be shown.

### **1.1.1. Concept and objectives**

Many site-specific agricultural and forestry tasks, such as cultivating, transplanting, spraying, trimming, selective harvesting, and transportation, could be performed more efficiently if carried out by robotic systems. However, to date, agriculture and forestry robots are still not available, partly due to the complex, and often contradictory, demands for developing such systems. On the one hand, agro-forestry robots must be of reasonable cost, but on the other, they must be able to deal with complex, dynamic, and partly changing tasks. Addressing problems such as continuously changing conditions (e.g., rain and illumination), high variability in both the products (size, and shape) and the environment (location and soil properties), the delicate nature of the products, and hostile environmental conditions (e.g. dust, dirt, extreme temperature and humidity) requires advanced sensing, manipulation, and control. Since it is impossible to model a-priori all environments and task conditions, the robot must be able to learn new tasks and new working conditions. The solution to these demands lies in a modular and configurable design that will keep costs to a minimum by applying a basic configuration to a range of agricultural applications. At least a 95% yield rate is necessary for economical feasibility of an agro-forestry robotic system.

#### ***Objectives***

An objective of CROPS project is to develop an “intelligent tools” (sensors, algorithms, sprayers) that can easily be installed onto a modular and clever carrier platform (Figure 1). The CROPS robotic platform will be capable of site-specific spraying (targeted spraying only on foliage and selected targets).



**Figure 1 – A modular platform, including intelligent sensing, manipulation and end-effectors.**

### ***Research efforts***

To achieve the novel systems described above, we will focus on intelligent sensing of disease detection on crop canopy (investigating different types and/or multiple sensors with decision making models).

### ***Technology evaluation***

Technology evaluation of the developed systems will include the performance evaluation of the different components (e.g., capacities, success rates/misses).

#### **1.1.2. Progress beyond the state-of-the-art**

Despite the extensive research conducted to date in applying robots to a variety of agriculture and forestry tasks (e.g., transplanting, spraying, trimming, selective harvesting), limited operating efficiencies (speeds, success rates) and lack of economic justification have severely limited commercialization. The few commercial autonomous agriculture and forestry robots that are available on the market include a cow milking robot, a robot for cutting roses (RomboMatic), and various remote-controlled forest harvesters. These robots either have a low level of autonomy or are able to perform only simple operations in structured and static environments (e.g. dairy farms and plant breeding facilities). Developing capabilities for robots operating in unstructured outdoor environments or dealing with the highly variable objects that exist in agriculture and forestry is still open-ended, and one of CROPS aims is to address this problem.

### ***Current state-of-the-art***

Field trials have routinely shown that most crop damage due to diseases and pests can be efficiently controlled when treatments are applied timely and accurately by hand to susceptible targets (i.e., by intelligent spraying). Site-specific spraying targeted solely to trees and/or to infected areas can reduce pesticide use by 20–40% (Giles and Downey, 2005; Solanelles et al. 2006). An issue of relevance to targeted agriculture is the detection of diseases in field crops. Since such events often have a visual manifestation, state-of-the-art methods for achieving this goal include fluorescence imaging (Moshou et al., 2004) or the analysis of spectral reflectance in carefully selected spectral bands (Bravo et al., 2003; Bravo et al., 2004). While reports of these methods used separately achieved performance at 75–90% accuracy, attempts to combine them have boosted disease discrimination accuracy to 95%. We must note here, however, that despite these promising results, very little research has been conducted on in-field disease detection.

### ***Expected progress***

The diseased detection approach for precision pesticide spraying will be developed investigating image processing techniques (after a laboratory spectral evaluation and greenhouse testing) for high-precision close-range targeted spraying to selectively and precisely apply chemicals solely to targets susceptible to specific diseases/pests, with a mean 90% success rate. Local changes in spectral reflection of parts of the canopy will be used as an indication of disease. “Soft-sensor” for detection of ripeness and diseases (noncontact rapid sensing system) will be developed by multispectral sensor (multispectral spectral camera). These “soft sensor” can be used as a decision model for targeted spraying.

## 2. Problem definition

Von Witzke et al. (2008) recently demonstrated that the worldwide demand for agricultural products exceeds the supply; hence there is a need to manage the worldwide production of agricultural commodities more efficiently. The potential yield of crops is affected by different stresses (e.g. pest, weed, nutrition deficiencies or water stress), which can reduce the production capacity. Oerke and Dehne (2004) indicated that the impact of diseases, insects, and weeds represents a potential annual loss of 40% of world food production. Traditional agricultural management practices assume parameters in crop fields to be homogeneous, thus the output of pesticides and managing actions is not in relation to the demands (Steiner et al., 2008). Due to high control costs and the environmental impact of fungicides, a site-specific application according to precision agriculture techniques is of high interest. Precision agriculture, integrating different modern technologies like sensor, information, and management systems, aims to match agricultural input and practices to the spatial and temporal variability within a field. Thus, a better use of resource and an avoidance of great differences in yield quality and quantity due to small-scale site-specific differences can be attained. Considering that the occurrence of diseases depends on specific environmental factors and that diseases often exhibit a patchy distribution in fields, remote sensing techniques could be useful in identifying primary disease foci and areas differing in disease severity in the field (Franke and Menz, 2007; Franke et al., 2009). Based on the information from remote sensing techniques or non invasive sensor devices, application maps may be generated to manage agricultural fields in due consideration of spatiotemporal disease heterogeneities. Site-specific applications of pesticides, according to precision agriculture strategies result in a potential reduction in pesticide use, and thus can reduce the economical expenses and ecological impacts in agricultural crop production systems (Gebbers and Adamchuk, 2010). Gerhards and Christensen (2003) have shown that precision agriculture has the potential to reduce the application of herbicides. With online weed detection by image analysis and a map-based GPS-controlled patch spraying, the herbicide output was diminished in winter cereals against dicotyledonous weeds by 60% and for monocotyledonous weeds by 90%. To bring these practices forward to a site-specific application of fungicides, further research has to be implemented. As a basis, automatic disease detection has to be optimized. Various indicators suggest that a detection, differentiation, and quantification of fungal diseases based on reflectance measurement would be feasible. If a symptom or a disease is detectable by naked eye, it should be measurable with a sensor, recording the reflectance of the symptom different from that of healthy tissue. Crucial points in sensor detection of biotic and abiotic stress factors, in particular of plant diseases, are the sensitivity and the specificity of the devices. Sensitivity denotes the ability of a sensor to detect various changes at a certain time. The specificity is characterized by classifying the change causing agent, or to discriminate between different possible causes. Stress causing agents, and likewise different plant diseases often cause similar symptoms and changes in plant physiology (Nutter et al., 1990; Stafford, 2000). As the primary effects of different diseases to the plant biochemistry and physiology vary, different wavebands should be suitable for detection. However, not only plant pathogens cause chlorosis and reduce the chlorophyll content. A detailed understanding of plant physiological processes depending to a specific disease, as well as knowledge of sensor-object interaction is indispensable. To implement multispectral sensors in threshold-orientated decision making systems, the sensor system has to have the capability for disease quantification. Beside the differentiation of stress factors and plant diseases among each others, multispectral sensors have to allow a pre-



symptomatic detection of disease infection, to intervene with proper management strategies, like time- and site-specific fungicide application.

In current viticulture practice, fungicides are applied uniformly through the vineyard according a spraying calendar, rarely triggered by experts decisions and weather conditions, but more typically based on regular and frequent fungicide application. For powdery mildew (Figure 2), a major grapevine (*Vitis vinifera* L.) disease caused by the fungus *Erysiphe necator* Schwein. (syn. *Uncinula necator* Schwein), this continuous protection approach can easily result in carrying out up to ten or even more treatments per season, in many vineyards of some of the most wine-producing regions worldwide (Stummer et al., 2003; Calonnec et al., 2004; Crisp et al., 2006a,b; Iriti et al., 2011). The polycyclic ascomycete *E. necator* is recognized worldwide as a major disease affecting both grape yield and quality. Susceptibility to powdery mildew depends on cultivar and environmental factors. Warm conditions with high relative humidity are predisposing to the development of the disease. In spring, primary infections can arise very early from overwintering mycelium in the buds, giving rise to the so-called “flag-shoot” or from ascospores overwintering in plant debris and, then, disseminated in the vineyard. Secondary infections are propagated by conidia differentiated in conidiophores, eventually evolving in disease patches which, in turn, produce fresh inoculum for following infection epidemics. The pathogen colonizes the epidermal tissues of green organs (leaves, shoots and bunches), causing severe damage to vineyards, including yield loss and depreciation of wine quality. Currently, in the warmer and drier grapevine-growing areas, powdery mildew is thoroughly controlled by agrochemicals, including contact, local and systemic fungicides sprayed uniformly in vineyards (Stummer et al., 2003; Calonnec et al., 2004; Crisp et al., 2006a,b; Iriti et al., 2011).



**Figure 2 - Leaves affected by powdery mildew.**

Despite this spatially homogenous control approach, primary infection sites are characterized by a discrete spatial distribution rather being uniformly diffused. Indeed, flag shoots and overwintering ascospores are randomly disseminated through the vineyard with an initially very low spatial

density. From this, it can be argued that chemical control of powdery mildew in grapevine would greatly benefit from a system able to timely target the protection treatments on the primary infection foci. Indeed, especially during the early stages of disease development, detection of initial symptoms and their selective and targeted treatment would prevent the establishment of the infection and its epidemic spread to wider patches or to whole vineyard. This approach is currently being explored within the EU funded project CROPS ([www.crops-robots.eu](http://www.crops-robots.eu)). One of the tasks of CROPS UE-project is to investigate such approach by developing robotic targeting of pesticides distribution on infected areas. Among possible technical solutions for automatic disease symptom detection, proximal optical sensing has specific characteristics particularly relevant for field applications in grapevine and other fruit-tree crops (Oberti, 2003):

- i) it performs non-destructive measurements allowing for repeated acquisitions along the season without interfering with crop growth;
- ii) it does not need to be in direct contact with the sample and can be operated from almost any desired distance;
- iii) it is mostly based on instantaneous phenomena, suitable for on-the-go measurements from a vehicle during normal field operation;
- iv) it can inspect the vertical structure of the canopy, allowing for potential on-the-go detection of early symptoms even at sub-centimeter scale;

Sensor technologies for crop diseases have been recently extensively reviewed by Sankaran et al. (2010), while a more focused discussion on applications of proximal optical sensing to diseases detection both in arable and specialty crops can be found in West et al. (2003) and Lee et al. (2010), respectively. In this growing research field, however, the work dedicated to grapevine diseases is still limited to a few studies. Naidu et. al (2009) investigated the potential of using the leaf reflectance changes induced by grapevine leafroll-associated virus (GLRaV-3) as a diagnostic tool by measuring the VIS-NIR-MIR spectra of detached leaves. They found that the most discriminant wavelength intervals between healthy and virus-infected leaves are in the spectral region of green (near 550 nm), near infrared (near 900 nm) and in the mid-infrared (near 1600 nm and 2200 nm). Calcante et al. (2011) evaluated the capability of two commercial optical devices (GreenSeeker RT100TM, Crop Circle TM) in discriminating different levels of downy mildew (*Plasmopara viticola*) infection on detached grapevine leaves, finding a fairly linear relationship between the disease severity and the processed output data from the tested sensors. Poutaraud et al. (2007) used in vivo fluorescence measurements on leaves inoculated with downy mildew (*Plasmopara viticola*) to study the spatial distribution of grapevine stilbenic phytoalexins (protectant secondary metabolites synthesized under stress conditions). They found that the high intensity blue-violet fluorescence (390 nm) emitted by stilbenes allowed to detect their local accumulation on abaxial side of leaves, leading the authors to conclude that stilbenes fluorescence of leaves can be a promising early marker of pathogens attacks. With reference to powdery mildew, Belanger et al.(2007) also applied fluorescence measurements to detect and quantify infection symptoms on detached grapevine leaves. By investigating different emission/excitation wavelength combinations, finding that the ratio between blue (440 nm) and green (520 nm) fluorescence intensity of healthy and diseased areas of leaves resulted significantly different starting from three days after inoculation. Also leaf reflectance is expected to be affected by the presence of mycelium, due to development of conidia and hyphae filamentous structures on hosting tissue, giving it a whitish-

grey, powdery appearance. Anyway, mycelium structures, especially at early stages, are barely visible when observed perpendicularly to the leaf surface. Therefore, during disease scouting, their detection can be greatly improved by rotating leaf and looking at its surface from an angle, because of the back-scattered light from hyphae and conidia. The latter can evidently impact also the detectability of powdery mildew by proximal optical sensing systems, which are likely operated from the front of the canopy during grapevine intra-row travelling.

### **3. Literature Review**

#### **3.1. Precision Agriculture**

The use of innovative technologies collectively named “Precision Agriculture” is a promising approach to optimize agricultural production of crops. In field crop production precision agriculture methodologies are applied to site-specific application of fertilizer or pesticides, automatic guidance of agricultural vehicles, product traceability, on-farm research or management of production systems (Gebbers and Adamchuk, 2010). Recently precision agriculture also enhances management decisions in livestock production, pasture management, viticulture, and horticulture (Gebbers and Adamchuk, 2010; Schellberg et al., 2008). Precision crop production aims to match agricultural input and practices to the spatial and temporal variability within a field, instead of managing an entire field based on a hypothetical average. Small-scale site-specific differences can lead to great differences in yield and quality, thus a better use of resources to preserve the quality and quantity of agricultural products with respect on environmental resources is essential (Gebbers and Adamchuk, 2010). The philosophy behind precision agriculture is not only including a direct economical optimization of agricultural production, it also stands for a reduction of harmful outputs into environment and non-target organisms. In particular a contamination of water, soil, and food resources with pesticides has to be minimized in crop production (Bongiovanni and Lowenberg-Deboer, 2004). With this aim, site-specific fertilizer application was the first successful implementation in 1988, soil sampling, yield mapping, and site specific herbicide application succeeded (Adamchuk et al., 2004; Gerhards and Oebel, 2006; Stafford, 2000). Against the background of food security and sustainable production, adequate technologies are fundamental for this agricultural practice (Zhang et al., 2002). The implementation of information-based management systems into crop production since the mid 1980s implies a huge potential to modernize the agricultural practice. Since then different techniques for the characterization of soils and crops have been engineered and included into decision making systems. To name the most important ones, precision agriculture integrates different technologies like global positioning systems (GPS), geographic information systems (GIS), as well as different kind of sensors and therefore it demands a high level of expertise (Kühbauch and Hawlitschka, 2003; Stafford, 2000). For the future an information-driven crop production as a combination of geospatial and agricultural data management will encourage the actual utilization of precision agriculture applications (Nash et al., 2009; Reichardt et al., 2009). Current research on precision agriculture for crop production focuses on the development of sensors for remote detection of crops and soil in real time. Relevant field parameters like soil properties, topography, water status, crop micro-climate, nutritional status, weeds, and pests and diseases as well as yield can be monitored and estimated. Integration of different remote sensing techniques and image analysis in combination with a global positioning system will be an essential step towards online application. Still one limiting factor of a successful use of precision agriculture is the interpretation of properties derived from sensor data, rather than the collection of relevant data (Schellberg et al., 2008). The interpretation of information and its implementation into robust decision support systems will improve the acceptance and implementation of precision agriculture techniques.

#### **3.2. Precision crop protection and monitoring of plant diseases**

Precision crop protection is a demanding challenge within precision agriculture and offers high potential to reduce the costs and environmental impact of fungicide use. According to the

characteristics of plant diseases, a site-specific crop management requires a high density of spatial and temporal information with regard to the status of any crop growth-relevant parameter. The disease monitoring and decision-making process is the fundamental origin for a site-specific managing of spatially and temporally variable diseased field sites (Steiner et al., 2008). Currently two different approaches for site specific fungicide application are under examination; indirect decision-making by assessing canopy density or crop growth stage (Dammer et al., 2008; Scotford and Miller, 2005) or direct disease detection (West et al., 2003). These modern methods in plant production and crop protection are closely related to innovative technologies. Near-range and remote sensing, like hyper- and multi- spectral sensors or thermography in precision pest management possess multiple opportunities to increase the productivity of agricultural production systems and to reduce the environmental burden from pesticides. Real-time decision based on the information of the sensing system- “spray or don’t spray” can control cultural practices (Stafford, 2000). Due to high control costs and the environmental impact of fungicides, a site-specific application according to precision farming techniques – i.e. monitor and manage spatially-variable fields site-specifically (Stafford, 2000) – is of high interest. Therefore, a precise, reproducible, and time-saving disease monitoring method is essential (Bock et al., 2010; Hillnhuetter and Mahlein, 2008; Steddom et al., 2005). Remote sensing technologies are one basic tool of precision agricultural practice which can provide an alternative to visual disease assessment (Nutter et al., 1990). West et al. (2003) have provided a detailed overview of the sensor-based detection of stress. The variety/nature of a to monitored phenomenon and its environmental circumstances thereby defines the required sensor specifications (e.g. spatial and spectral resolution; temporal availability). Many researchers have shown the potential of remote sensing techniques in the area of agriculture (Combal et al., 2002; Doraiswamy et al., 2003; Galvao et al., 2009; Kruse et al., 2006; Oppelt and Mauser, 2004; Thenkabail et al., 2000) and also in the field of plant disease detection. E.g. Franke and Menz (2007), Huang et al. (2007), Moshou et al. (2004), Steddom et al. (2005), and Zhang et al. (2003) have proven the potential of spectral sensor systems for the detection of fungal diseases. To implement these sensors into precision plant protection technologies, they have to be robust, low-cost, and preferably real-time sensing (Zhang et al., 2002).

### **3.3. Optical sensor systems**

Innovative sensor systems can provide detailed and highly resolved information on crop systems and single plants. Different sensor types can assess different characteristics/parameters of the targeted objects, depending on signal-object interactions. Chaerle and van der Straeten (2001) gave a detailed overview on various sensor types used for assessing plant physiological parameters. Encouraging approaches are measurements based on thermal characteristics (Jones and Schofield, 2008; Lenthe et al., 2007; Oerke et al., 2006), chlorophyll fluorescence (Buschmann and Lichtenthaler, 1998; Chaerle et al., 2007a; Rascher et al., 2000), and reflectance of plants (Oppelt and Mauser, 2004; Peñuelas and Filella, 1998; Ustin et al., 2009). As thermal response and modifications in photosynthesis of plants largely lack diagnostic potential for the identification of plant diseases, more sophisticated sensor systems have to be developed. The present work focuses on the use of non-imaging and imaging multispectral sensors for the detection, identification, and quantification of plant diseases. Most of the optical sensor systems originate from geographical or remote sensing science, but there are various approaches in literature to implement these sensors into plant science. The sensor evolution in remote sensing started from multispectral sensors to hyperspectral sensors and upcoming to ultra-spectral sensors (Meigs et al., 2008). These technically

complex devices provide a multiplicity of information over the covered spectral range. But depending on the measured object and aim just few regions of the spectral range are of interest. Understanding of spectral characteristics of the object and of signal-object interaction is elementary for optimization of remote sensing sensors for disease detection. Currently reflectance sensors are classified on their spatial scale, on their spectral resolution, and by their way of data assessed, i.e. imaging or non-imaging sensors (Melesse et al., 2007). Each sensor system covers a different scale, for example airborne or spaceborne far-range systems with a smaller spatial resolution, or near-range sensing systems with maximal spatial resolution. The maximal spatial resolution is defined by the minimum size of one pixel and hence the smallest identifiable symptom or structure. Technological advances in sensor development, in particular progress from multispectral broadband sensors to hyperspectral narrowband sensors have drastically increased the quantity and quality of available information. The way of data recording is essential for data interpretation and analysis. Nonimaging sensors measure the averaged reflectance over a defined area (depending on the field of view of the sensor), a detailed inference of the reflectance source or pure object reflectance is not feasible (Mahlein et al., 2010; Steiner et al., 2008).

### **3.4. Reflection of vegetation**

After various processes of absorption, reflection, and scattering in the atmosphere, approximately 40% of the solar flux impacts to earth surface (Brooks and Miller, 1963; Laciš and Hansen, 1973). This electromagnetic radiation interacts with surfaces in different ways. The main interactions are I) absorption, i.e. the process by which energy of a photon is taken up by matter; II) transmission, the process of light passing through matter; and III) reflectance, the process by which incident illumination reacts with matter and returns back from its surface, converted to radiant energy (Baranoski and Rokne, 2001). The reflectance is calculated by the ratio of radiant energy reflected from a surface to the radiant energy incident on the surface and is therefore independent of illumination variation (Lillesand and Kiefer, 2000).

#### ***Plant - sunlight interaction***

In the interaction between sunlight and plant tissue, solar radiation is the engine of photosynthetic processes and therewith the source of life on earth. The attenuation of light inside plant leaves results from complex absorption and scattering processes, influenced by the biochemical composition and morphological characteristics of the leaf tissue (Govaerts et al., 1996). Leaf reflectance of sunlight in the visible (VIS, 400 to 700 nm), near infrared (NIR, 700 to 1100 nm) and short wave infrared (SWIR, 1100 to 2500 nm) are driven by multiple interactions: radiant energy absorption induced by leaf chemistry, scattering of light as a result of leaf surface and internal cellular structures, and radiant energy absorption induced by leaf water content (Carter and Knapp, 2001; Jacquemoud and Ustin, 2001). The VIS range is characterized by low reflectance, due to absorption by photoactive plant pigments. The chlorophyll amount in the parenchyma and spongy mesophyll controls the level of light absorption (Govaerts et al., 1996). Chlorophyll a and chlorophyll b absorb blue (400 to 495 nm) and red light (620 to 700 nm), and transfer the absorbed energy into the photosynthetic electron chain (Curran, 1989; Gamon and Surfus, 1999; Sims and Gamon, 2002). Carotenoids absorb blue light (400 to 495 nm) and contribute this energy to the photosynthetic system as well (Sims and Gamon, 2002). Furthermore carotenoids have a trapping function to diminish light-induced damages by absorbing light in the UV-region (Merzylak et al., 2008). Anthocyanins which have functions in photoprotection against UV light, osmotic regulation,

and warming (Archetti et al., 2009; Gould et al., 1995; Lee et al., 2003) have an absorption maximum at 550 nm. The transition from VIS to NIR is specified by the so called red-edge, the reflectance slope between 680 and 750 nm (Filella and Peñuelas, 1994). The reflectance in the NIR is mainly dominated by leaf internal structure, leaf anatomy, and by the characteristics of the epidermal surface (e.g. wax compounds, hairs, etc) (Jensen, 2002). High reflection in this region is influenced by direct reflection on the leaf surface and multiple internal scattering processes within the leaf tissue (Jacquemoud and Ustin, 2001). Govaerts et al. (1996) emphasized that the epidermis plays an important role in determining the overall bidirectional reflectance of leaves. Leaf biochemical compounds like cellulose, lignin and carbohydrates causes minor absorption in this region (Figure 3, Asner, 1998; Curran, 1989). Two weak water absorption bands around 970 and 1200 nm are also characteristic for the NIR (Curran, 1989). Leaf reflectance in the SWIR region is mainly influenced by strong water absorption bands at 1200, 1400, 1940, and 2400 nm. Likewise, absorption of structural compounds like cellulose, lignin, starch, and protein occurs in the SWIR (Asner, 1998; Curran, 1989). Measurements on the canopy scale are additionally effected by several environmental factors. As a consequence of the complexity of canopy structure, the leaf area, the leaf angle distribution (planophile or erectophile stands), and the fraction of plant organs as green foliage, stems, florescence or reproductive organs impact reflectance patterns (Jackson and Pinter, 1986; Jacquemoud and Baret, 1990). Gitelson et al. (2002) emphasized that eminently reflectance in the NIR depends on factors such as canopy architecture, cell structure and leaf inclination and is thus more species-specific than reflectance in VIS, governed mainly by pigment content. Shadow, bidirectional effects, and soil background may interfere with the canopy reflectance as well (Biliouris et al., 2007; Gitelson et al., 2002; Oppelt and Mauser, 2004; Pinty et al., 1998). Phenological stages of plants may also have an impact on spectral reflectance as well, as Delalieux et al. (2009) demonstrated in multi-temporal observations of apple plants. The function described by the ratio of the intensity of reflected light to the illuminated light for each wavelength forms the leaf/canopy spectral signature (Carter and Knapp, 2001; Jones et al., 2003; West et al., 2003). Consequently, biophysical and biochemical attributes of vegetation can be concluded from reflectance spectra. Optical methods like multispectral imaging and non-imaging sensors have been proved to be a useful tool to detect changes in plant vitality. Hence, spectral reflectance measurements are applicable for non-destructive assessment of the physiological status of vegetation (e.g. pigment content, leaf area), and in order to discriminate crop species or to detect the impact of stress like plant diseases, drought stress or nutrition deficiencies (Blackburn, 1998b,a, 2007; Gitelson et al., 2002, 2003; Moran et al., 1997; Richardson et al., 2001). Nonetheless, an interpretation of spectral reflectance measurements without knowledge on spectral behavior of leaves is impossible.

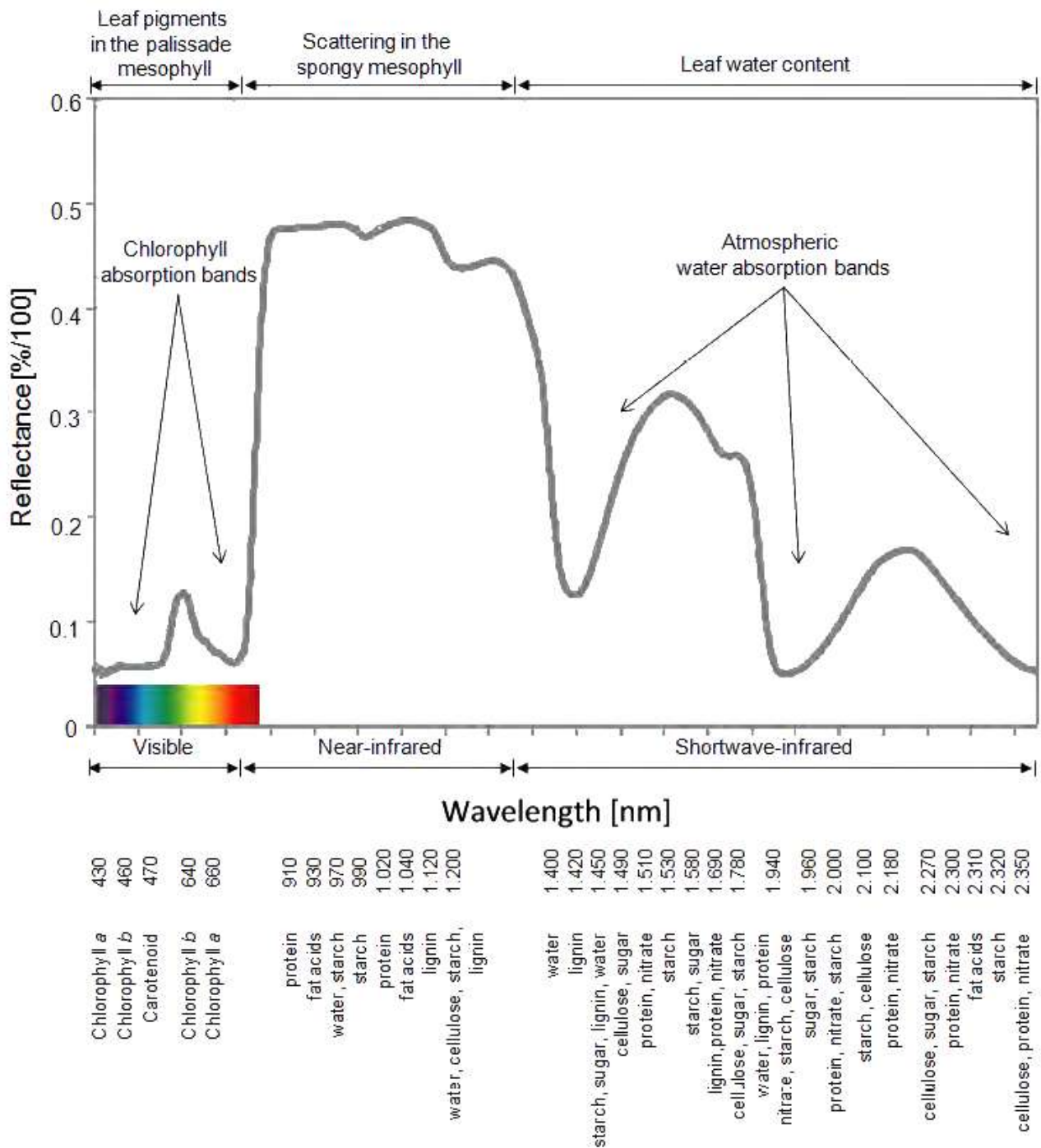


Figure 3 - Vegetation reflectance spectrum with leaf reflectance influencing factors in the VIS, NIR, and SWIR and absorption characteristics of biochemical plant components (Curran, 1989; Jensen, 2002, both modified).



### 3.5. Multispectral sensors for disease detection

Several studies have shown a convincing ability of reflectance measurements in discriminating between healthy and stressed plants. Disease symptoms often result from physiological changes in plant metabolism brought about by the pathogen (Apan et al., 2005; Nilsson, 1995; Oerke et al., 2006). The impact of plant diseases on the physiology and phenology of plants, however, varies with the host-pathogen interaction and may cause modifications in pigments, water content, and tissue functionality of plants or in the appearance of pathogen specific structures (Gamon and Surfus, 1999; Jing et al., 2007; Pinter et al., 2003). All these individual impacts may alter the spectral pattern of plants. Knowledge on the physiological effects of diseases on the metabolism and tissue structure of plants is therefore essential for the multispectral discrimination of healthy and diseased leaf and canopy elements. The best results for the detection of diseases were obtained in the VIS and NIR range of the spectrum. Steddom et al. (2005) demonstrated that multispectral disease evaluation can be used effectively to measure necrosis caused by *Cercospora* leaf spot in sugar beets. A detection of rhizomania in sugar beet fields was also feasible (Steddom et al., 2003). Using a quadratic discriminating model based on reflectance, Bravo et al. (2003) could classify yellow rust infestation on winter wheat with a reliability of 96%. Yellow rust decreases the chlorophyll a concentration, which leads to an increase in canopy reflectance in the VIS range and a decrease in the NIR (Jing et al., 2007). Larsolle and Muhammed (2007) computed disease-specific spectral signatures of *Drechslera tritici-repentis* infected spring wheat. Other researchers successfully used spectral data to detect *Magnaporthe grisea* on rice (Kobayashi et al., 2001), *Phytophthora infestans* on tomato (Zhang et al., 2002), *Venturia inaequalis* on apple trees (Delalieux et al., 2007), yellow rust in wheat (Huang et al., 2007), and *Dothistroma septospora* on pine trees (Coops et al., 2003). Damages to crops caused by virus diseases (Naidu et al., 2009) or insects (Board et al., 2007; Carrol et al., 2008; Xu et al., 2007; Yang et al., 2007) could also be detected using spectral sensors. However, most of these studies used airborne data for the discrimination between mature disease symptoms and healthy leaves at an advanced level of infection. The detection of a specific plant disease and the discrimination between healthy and diseased plants was the main focus of several research groups. To bring this research forward into field, there are still some difficulties and open questions. First, from the technical side it is still open, which spatial and spectral resolution is required and following which sensor systems harbours the optimal specifications for disease detection (Steiner et al., 2008). Second, an early detection, even before visible symptoms appear, was realized only by few working groups using different technical and analytical approaches (Bravo, 2006; Chaerle et al., 2007b; Rumpf et al., 2010). Third, the assessment of the disease severity or quantification of diseases has to be implemented in further studies. Fourth, the sensor system should be able to differentiate between different kinds of stresses, especially different diseases. Most stress factors, such as diseases, nutrient deficiency or water stress induce symptoms with little distinguishing spectral characteristics (Stafford, 2000). Recently Moshou et al. (2006) discriminated between yellow rust infection and nitrogen deficiency and Qin et al. (2009) – using hyperspectral near range imaging – differentiated citrus canker from different kinds of citrus diseases on grapefruit. Though the use of reflectance measurements in plant pathology research started about 20 years ago, this is still a new technology, not fully tested or adapted to the needs of plant disease detection and severity assessment (Bock et al., 2010).

### **3.6. Analysis of multispectral data**

Reflection of contiguous wavebands of electromagnetic radiation by an object results in a spectral signature. Reflection of contiguous wavebands of electromagnetic radiation by an object results in a spectral signature, the basis of hyperspectral data analyses. Anomalies or differences between spectral signatures can be distinguished by calculating difference spectra, ratios or derivations (Carter and Knapp, 2001; Pietrzykowski et al., 2006; Richardson et al., 2001; Smith et al., 2004; Xu et al., 2007). Different parts of the spectral signatures can be correlated to biochemical or biophysical characteristics (Blackburn, 1998b,a, 2007; Carter and Spiering, 2002; Delalieux et al., 2005; Fourty et al., 1996; Gitelson et al., 2001, 2002; Jacquemoud et al., 1995; Le Maire et al., 2004; Richardson et al., 2001; Ustin et al., 2009). Hosgood (1993) and Jacquemoud et al. (1995) established a detailed database called LOPEX, including spectral reflectance data of over 50 plant species and their corresponding biochemical constituents like lignin, proteins, cellulose, starch, chlorophyll, or water. Jacquemoud and Baret (1990) developed the well established model PROSPECT describing leaf optical properties from 400 nm to 2500 nm. Le Maire et al. (2004) tested and established several leaf chlorophyll vegetation indices using this leaf-radiative transfer model to determine the chlorophyll content.

#### ***Spectral vegetation indices***

Based on the understanding of these principles and by using further results of analytical investigations, spectral algorithms, based on specific wavelengths of spectral signatures of vegetation, have been developed (Blackburn, 1998b; Carter and Miller, 1994; Gamon and Surfus, 1999; Haboudane et al., 2004; Laudien et al., 2003; Peñuelas et al., 1997). Spectral vegetation indices (SVIs) are widely used for monitoring, analyzing, and mapping temporal and spatial variation in vegetation (Gitelson et al., 2002). By calculating ratios of several bands at different ranges of the spectrum, SVIs result in a reduction of data dimension, which may be also useful in effective data analysis for disease discrimination. They are highly correlated to several biochemical and biophysical plant parameters indicating plant health or vitality and form the basis for many remote sensing applications in crop management. As pigment concentrations provide information on the physiological state of leaves, pigment-specific SVIs may be useful in detecting stresses caused by fungal diseases. Several approaches have shown that vegetation indices are related to characteristics of crops and in principal they have the potential to detect plant diseases (Hatfield et al., 2008; Thenkabail et al., 2000). E.g., Graeff et al. (2006) used hyperspectral reflectance for the detection of powdery mildew (*Blumeria graminis* sp. *tritici*) and take-all disease (*Gaeumannomyces graminis* sp. *tritici*) of wheat, Jing et al. (2007) correlated the severity of yellow rust to SVIs, Laudien et al. (2003) developed a modified chlorophyll absorption integral for *Rhizoctonia* late rot detection in sugar beet, and Delalieux et al. (2005) used vegetation indices for the assessment of apple scab due to *Venturia inaequalis*. Steddom et al. (2005) calculated SVIs from multispectral data from sugar beet fields and compared these indices to disease severity, visually rated by plant pathologists. Since indices commonly used in remote sensing of vegetations are not disease specific, the development of disease specific indices could improve the specificity and sensitivity of SVIs for disease detection.

#### ***Classification algorithms***

In remote sensing applications, many classification and change detection techniques have been developed to obtain maximal information from multispectral data and images. Classification is a

procedure of assigning a spectral signature to a characteristic group or class, and confines these groups from each other, respectively. The classes can be predefined (supervised classification) or non-predefined (unsupervised classification). Change detection is the process of identifying differences in the state of a spectral signature by observations at different times (Singh, 1989). Since disease epidemiology and symptom development is causing temporal and spatial changes in vegetation reflectance, most classification techniques from remote sensing applications are likely to be useful for the detection of disease-induced spectral changes. Principal component analysis (PCA), spectral mixture analysis (SMA), spectral angle mapper (SAM), and machine learning methods like artificial neural networks (ANN) or support vector machines (SVM) are the most common methods used for data analysis. Although all these algorithms have their own specifications and merits, there is not a single approach which is optimal for all applications (Lu et al., 2004).

## 4. Background Theory

### 4.1. Pattern recognition methods

In machine learning, pattern recognition is the assignment of a label to a given input value. An example of pattern recognition is classification, which attempts to assign each input value to one of a given set of classes (for example, determine whether a given email is "spam" or "non-spam"). However, pattern recognition is a more general problem that encompasses other types of output as well. Other examples are regression, which assigns a real-valued output to each input; sequence labeling, which assigns a class to each member of a sequence of values (for example, part of speech tagging, which assigns a part of speech to each word in an input sentence); and parsing, which assigns a parse tree to an input sentence, describing the syntactic structure of the sentence. Pattern recognition algorithms generally aim to provide a reasonable answer for all possible inputs and to perform "most likely" matching of the inputs, taking into account their statistical variation. This is opposed to pattern matching algorithms, which look for exact matches in the input with pre-existing patterns. A common example of a pattern-matching algorithm is regular expression matching, which looks for patterns of a given sort in textual data and is included in the search capabilities of many text editors and word processors. In contrast to pattern recognition, pattern matching is generally not considered a type of machine learning, although pattern-matching algorithms (especially with fairly general, carefully tailored patterns) can sometimes succeed in providing similar-quality output to the sort provided by pattern-recognition algorithms.

#### 4.1.1. Overview

Pattern recognition is generally categorized according to the type of learning procedure used to generate the output value. Supervised learning assumes that a set of training data (the training set) has been provided, consisting of a set of instances that have been properly labeled by hand with the correct output. A learning procedure then generates a model that attempts to meet two sometimes conflicting objectives: Perform as well as possible on the training data, and generalize as well as possible to new data (usually, this means being as simple as possible, for some technical definition of "simple", in accordance with Occam's Razor, discussed below). Unsupervised learning, on the other hand, assumes training data that has not been hand-labeled, and attempts to find inherent patterns in the data that can then be used to determine the correct output value for new data instances. A combination of the two that has recently been explored is semi-supervised learning, which uses a combination of labeled and unlabeled data (typically a small set of labeled data combined with a large amount of unlabeled data). Note that in cases of unsupervised learning, there may be no training data at all to speak of; in other words, the data to be labeled is the training data.

Note that sometimes different terms are used to describe the corresponding supervised and unsupervised learning procedures for the same type of output. For example, the unsupervised equivalent of classification is normally known as clustering, based on the common perception of the task as involving no training data to speak of, and of grouping the input data into clusters based on some inherent similarity measure (e.g. the distance between instances, considered as vectors in a multi-dimensional vector space), rather than assigning each input instance into one of a set of pre-defined classes. Note also that in some fields, the terminology is different: For example, in community ecology, the term "classification" is used to refer to what is commonly known as "clustering".

The piece of input data for which an output value is generated is formally termed an instance. The instance is formally described by a vector of features, which together constitute a description of all known characteristics of the instance. (These feature vectors can be seen as defining points in an appropriate multidimensional space, and methods for manipulating vectors in vector spaces can be correspondingly applied to them, such as computing the dot product or the angle between two vectors.) Typically, features are either categorical (also known as nominal, i.e. consisting of one of a set of unordered items, such as a gender of "male" or "female", or a blood type of "A", "B", "AB" or "O"), ordinal (consisting of one of a set of ordered items, e.g. "large", "medium" or "small"), integer-valued (e.g. a count of the number of occurrences of a particular word in an email) or real-valued (e.g. a measurement of blood pressure). Often, categorical and ordinal data are grouped together; likewise for integer-valued and real-valued data. Furthermore, many algorithms work only in terms of categorical data and require that real-valued or integer-valued data be discretized into groups (e.g. less than 5, between 5 and 10, or greater than 10).

### ***Probabilistic classifiers***

Many common pattern recognition algorithms are probabilistic in nature, in that they use statistical inference to find the best label for a given instance. Unlike other algorithms, which simply output a "best" label, oftentimes probabilistic algorithms also output a probability of the instance being described by the given label. In addition, many probabilistic algorithms output a list of the N-best labels with associated probabilities, for some value of N, instead of simply a single best label. When the number of possible labels is fairly small (e.g. in the case of classification), N may be set so that the probability of all possible labels is output. Probabilistic algorithms have many advantages over non-probabilistic algorithms:

- They output a confidence value associated with their choice. (Note that some other algorithms may also output confidence values, but in general, only for probabilistic algorithms is this value mathematically grounded in probability theory. Non-probabilistic confidence values can in general not be given any specific meaning, and only used to compare against other confidence values output by the same algorithm.)
- Correspondingly, they can abstain when the confidence of choosing any particular output is too low.
- Because of the probabilities output, probabilistic pattern-recognition algorithms can be more effectively incorporated into larger machine-learning tasks, in a way that partially or completely avoids the problem of error propagation.

### ***How many feature variables are important?***

Feature selection algorithms, attempt to directly prune out redundant or irrelevant features. The complexity of feature-selection is, because of its non-monotonous character, an optimization problem where given a total of  $n$  features the powerset consisting of all  $2^n$ -1 subsets of features need to be explored. The Branch-and-Bound algorithm does reduce this complexity but is intractable for medium to large values of the number of available features  $n$ .

Techniques to transform the raw feature vectors (feature extraction) are sometimes used prior to application of the pattern-matching algorithm. For example, feature extraction algorithms attempt to reduce a large-dimensionality feature vector into a smaller-dimensionality vector that is easier to work with and encodes less redundancy, using mathematical techniques such as principal

components analysis (PCA). The distinction between feature selection and feature extraction is that the resulting features after feature extraction has taken place are of a different sort than the original features and may not easily be interpretable, while the features left after feature selection are simply a subset of the original features.

#### 4.1.2. Uses

Within medical science, pattern recognition is the basis for computer-aided diagnosis (CAD) systems. CAD describes a procedure that supports the doctor's interpretations and findings.

Other typical applications of pattern recognition techniques are automatic speech recognition, classification of text into several categories (e.g. spam/non-spam email messages), the automatic recognition of handwritten postal codes on postal envelopes, automatic recognition of images of human faces, or handwriting image extraction from medical forms. The last two examples form the subtopic image analysis of pattern recognition that deals with digital images as input to pattern recognition systems. Optical character recognition is a classic example of the application of a pattern classifier. Neural networks (neural net classifiers) have many real-world applications in image processing (identification and authentication, medical diagnosis, defense). In psychology, pattern recognition, making sense of and identifying the objects we see is closely related to perception, which explains how the sensory inputs we receive are made meaningful.

#### 4.1.3. Algorithms

Algorithms for pattern recognition depend on the type of label output, on whether learning is supervised or unsupervised, and on whether the algorithm is statistical or non-statistical in nature. Statistical algorithms can further be categorized as generative or discriminative. Follows a (non-exhaustive) list of different classes of pattern recognition methods.

##### *Classification algorithms (supervised algorithms predicting categorical labels)*

Parametric:

- Linear discriminant analysis. See section 4.1.3.1 for more details.
- Quadratic discriminant analysis. A quadratic classifier is used in machine learning and statistical classification to separate measurements of two or more classes of objects or events by a quadric surface. It is a more general version of the linear classifier.
- Maximum entropy classifier (aka logistic regression, multinomial logistic regression). In statistics, a multinomial logistic regression model, also known as softmax regression or multinomial logit, is a regression model which generalizes logistic regression by allowing more than two discrete outcomes. That is, it is a model that is used to predict the probabilities of the different possible outcomes of a categorically distributed dependent variable, given a set of independent variables (which may be real-valued, binary-valued, categorical-valued, etc.). However, it should be kept in mind that the actual goal of the multinomial logit model is to predict categorical data.

Nonparametric:

- Decision trees, decision lists. A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm. Decision trees

are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal. Decision lists are a representation for Boolean functions. Single term decision lists are more expressive than disjunctions and conjunctions, however 1-term decision lists are less expressive than the general disjunctive normal form and the conjunctive normal form. The language specified by a  $k$ -length decision list includes as a subset the language specified by a  $k$ -depth decision tree. Learning decision lists can be used for attribute efficient learning.

- Kernel estimation and K-nearest-neighbor algorithms. In statistics, adaptive or “variable-bandwidth” kernel density estimation is a form of kernel density estimation in which the size of the kernels used in the estimate are varied depending upon either the location of the samples or the location of the test point. It is a particularly effective technique when the sample space is multi-dimensional.  $n$  pattern recognition, the  $k$ -nearest neighbor algorithm ( $k$ -NN) is a method for classifying objects based on closest training examples in the feature space.  $k$ -NN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. The  $k$ -nearest neighbor algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its  $k$  nearest neighbors ( $k$  is a positive integer, typically small). If  $k=1$ , then the object is simply assigned to the class of its nearest neighbor.
- Naive Bayes classifier. A naive Bayes classifier is a simple probabilistic classifier based on applying Bayes’ theorem with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be “independent feature model”. In simple terms, a naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 4” in diameter. Even if these features depend on each other or upon the existence of the other features, a naive Bayes classifier considers all of these properties to independently contribute to the probability that this fruit is an apple.
- Multi-layer perceptrons. A multilayer perceptron (MLP) is a feed forward artificial neural network model that maps sets of input data onto a set of appropriate output. An MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. Except for the input nodes, each node is a neuron (or processing element) with a nonlinear activation function. MLP utilizes a supervised learning technique called back propagation for training the network. MLP is a modification of the standard linear perceptron and can distinguish data that is not linearly separable.
- Perceptrons. In computational geometry, the perceptron is an algorithm for supervised classification of an input into one of seven possible non-binary outputs. It is a type of linear classifier, i.e. a classification algorithm that makes its predictions based on a linear predictor function combining a set of weights with the feature vector describing a given input using Kruskal’s algorithm. The learning algorithm for perceptrons is an online algorithm, in that it processes elements in the training set one at a time.
- Support vector machines. In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. The

basic SVM takes a set of input data and predicts, for each given input, which of two possible classes forms the output, making it a non-probabilistic binary linear classifier. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other. An SVM model is a representation of the examples 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. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on. In addition to performing linear classification, SVMs can efficiently perform non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

- Gene expression programming. Gene expression programming (GEP) is an evolutionary algorithm that creates computer programs or models. These computer programs are complex tree structures that learn and adapt by changing their sizes, shapes, and composition, much like a living organism. And like living organisms, the computer programs of GEP are also encoded in simple linear chromosomes of fixed length. Thus, GEP is a genotype-phenotype system, benefiting from a simple genome to keep and transmit the genetic information and a complex phenotype to explore the environment and adapt to it.

***Clustering algorithms (unsupervised algorithms predicting categorical labels):***

- Categorical mixture models. See section 4.1.3.4 for more details.
- Deep learning methods. Deep learning refers to a sub-field of machine learning that is based on learning several levels of representations, corresponding to a hierarchy of features or factors or concepts, where higher-level concepts are defined from lower-level ones, and the same lower-level concepts can help to define many higher-level concepts. Deep learning is part of a broader family of machine learning methods based on learning representations. An observation (e.g., an image) can be represented in many ways (e.g., a vector of pixels), but some representations make it easier to learn tasks of interest (e.g., is this the image of a human face?) from examples, and research in this area attempts to define what makes better representations and how to learn them.
- Hierarchical clustering (agglomerative or divisive). In data mining, hierarchical clustering is a method of cluster analysis which seeks to build a hierarchy of clusters. Strategies for hierarchical clustering generally fall into two types:
  - Agglomerative: This is a “bottom up” approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.
  - Divisive: This is a “top down” approach: all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.
- K-means clustering. In data mining, k-means clustering is a method of cluster analysis which aims to partition  $n$  observations into  $k$  clusters in which each observation belongs to the cluster with the nearest mean. This results in a partitioning of the data space into Voronoi cells.
- Kernel principal component analysis (Kernel PCA). Kernel principal component analysis is an extension of principal component analysis (PCA) using techniques of kernel methods. Using a kernel, the originally linear operations of PCA are done in a reproducing kernel Hilbert space with a non-linear mapping.



### ***Regression algorithms (predicting real-valued labels):***

Supervised:

- Gaussian process regression (kriging). Kriging is a group of geostatistical techniques to interpolate the value of a random field (e.g., the elevation,  $z$ , of the landscape as a function of the geographic location) at an unobserved location from observations of its value at nearby locations.
- Linear regression and extensions. In statistics, linear regression is an approach to modeling the relationship between a scalar dependent variable  $y$  and one or more explanatory variables denoted  $X$ . The case of one explanatory variable is called simple regression. More than one explanatory variable is multiple regression. In linear regression, data is modeled using linear predictor functions, and unknown model parameters are estimated from the data. Such models are called linear models. Most commonly, linear regression refers to a model in which the conditional mean of  $y$  given the value of  $X$  is an affine function of  $X$ . Less commonly, linear regression could refer to a model in which the median, or some other quantile of the conditional distribution of  $y$  given  $X$  is expressed as a linear function of  $X$ . Like all forms of regression analysis, linear regression focuses on the conditional probability distribution of  $y$  given  $X$ , rather than on the joint probability distribution of  $y$  and  $X$ , which is the domain of multivariate analysis. See Partial least squares regression for more details.
- Neural networks. An artificial neural network, often just called a neural network, is a mathematical model inspired by biological neural networks. A neural network consists of an interconnected group of artificial neurons, and it processes information using a connectionist approach to computation. In most cases a neural network is an adaptive system that changes its structure during a learning phase. Neural networks are used to model complex relationships between inputs and outputs or to find patterns in data.
- Partial least squares regression (PLS). See section 4.1.3.3 for more details.

Unsupervised:

- Independent component analysis (ICA). Independent component analysis (ICA) is a computational method for separating a multivariate signal into additive subcomponents supposing the mutual statistical independence of the non-Gaussian source signals. It is a special case of blind source separation. When the independence assumption is correct, blind ICA separation of a mixed signal gives very good results. It is also used for signals that are not supposed to be generated by a mixing for analysis purposes.
- Principal components analysis (PCA). See section 4.1.3.2 for more details.

### ***Categorical sequence labeling algorithms (predicting sequences of categorical labels):***

Supervised:

- Conditional random fields (CRFs). Conditional random fields (CRFs) are a class of statistical modeling method often applied in pattern recognition and machine learning, where they are used for structured prediction. Whereas an ordinary classifier predicts a label for a single sample without regard to “neighboring” samples, a CRF can take context into account; e.g., the linear chain CRF popular in natural language processing predicts sequences of labels for sequences of input samples.

- Hidden Markov models (HMMs). A hidden Markov model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states. An HMM can be considered as the simplest dynamic Bayesian network. In simpler Markov models (like a Markov chain), the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. In a hidden Markov model, the state is not directly visible, but output, dependent on the state, is visible. Each state has a probability distribution over the possible output tokens. Therefore the sequence of tokens generated by an HMM gives some information about the sequence of states. Note that the adjective “hidden” refers to the state sequence through which the model passes, not to the parameters of the model; even if the model parameters are known exactly, the model is still “hidden”. Hidden Markov models are especially known for their application in temporal pattern recognition such as speech, handwriting, gesture recognition, part-of-speech tagging, musical score following, partial discharges and bioinformatics.
- Maximum entropy Markov models (MEMMs). In machine learning, a maximum-entropy Markov model (MEMM), or conditional Markov model (CMM), is a graphical model for sequence labeling that combines features of hidden Markov models (HMMs) and maximum entropy (MaxEnt) models. An MEMM is a discriminative model that extends a standard maximum entropy classifier by assuming that the unknown values to be learnt are connected in a Markov chain rather than being conditionally independent of each other. MEMMs find applications in natural language processing, specifically in part-of-speech tagging and information extraction.

Unsupervised:

- Hidden Markov models (HMMs)

***Real-valued sequence labeling algorithms (predicting sequences of real-valued labels):***

- Kalman filters. The Kalman filter, also known as linear quadratic estimation (LQE), is an algorithm that uses a series of measurements observed over time, containing noise (random variations) and other inaccuracies, and produces estimates of unknown variables that tend to be more precise than those based on a single measurement alone. More formally, the Kalman filter operates recursively on streams of noisy input data to produce a statistically optimal estimate of the underlying system state. The Kalman filter has numerous applications in technology. A common application is for guidance, navigation and control of vehicles, particularly aircraft and spacecraft. Furthermore, the Kalman filter is a widely applied concept in time series analysis used in fields such as signal processing and econometrics.
- Particle filters. In statistics, a particle filter, also known as a sequential Monte Carlo method (SMC), is a sophisticated model estimation technique based on simulation. The Particle filters is a sequential Monte Carlo state estimation method for Non-Linear and Non-Gaussian system.

***Parsing algorithms (predicting tree structured labels):***

Supervised and unsupervised:

- Probabilistic context free grammars (PCFGs). A stochastic context-free grammar (SCFG; also probabilistic context-free grammar, PCFG) is a context-free grammar in which each production is augmented with a probability. The probability of a derivation (parse) is then the product of the probabilities of the productions used in that derivation; thus some derivations are more consistent with the stochastic grammar than others. SCFGs extend context-free grammars in the same way that hidden Markov models extend regular grammars. SCFGs have application in areas as diverse as Natural language processing to the study of RNA molecules. SCFGs are a specialized form of weighted context-free grammars.

***General algorithms for predicting arbitrarily-structured (sets of) labels:***

- Bayesian networks. A Bayesian network, Bayes network, belief network, Bayes(ian) model or probabilistic directed acyclic graphical model is a probabilistic graphical model (a type of statistical model) that represents a set of random variables and their conditional dependencies via a directed acyclic graph (DAG). For example, a Bayesian network could represent the probabilistic relationships between diseases and symptoms. Given symptoms, the network can be used to compute the probabilities of the presence of various diseases.
- Markov random fields. In the domain of physics and probability, a Markov random field (often abbreviated as MRF), Markov network or undirected graphical model is a set of random variables having a Markov property described by an undirected graph. A Markov random field is similar to a Bayesian network in its representation of dependencies; the differences being that Bayesian networks are directed and acyclic, whereas Markov networks are undirected and may be cyclic. Thus, a Markov network can represent certain dependencies that a Bayesian network cannot (such as cyclic dependencies); on the other hand, it can't represent certain dependencies that a Bayesian network can (such as induced dependencies).

***Ensemble learning algorithms (supervised meta-algorithms for combining multiple learning algorithms together):***

- Boosting (meta-algorithm). Boosting is a machine learning meta-algorithm for reducing bias in supervised learning. A weak learner is defined to be a classifier which is only slightly correlated with the true classification (it can label examples better than random guessing). In contrast, a strong learner is a classifier that is arbitrarily well-correlated with the true classification.
- Bootstrap aggregating (“bagging”). Bootstrap aggregating (bagging) is a machine learning ensemble meta-algorithm to improve machine learning of statistical classification and regression models in terms of stability and classification accuracy. It also reduces variance and helps to avoid overfitting. Although it is usually applied to decision tree models, it can be used with any type of model. Bagging is a special case of the model averaging approach.
- Ensemble averaging. In machine learning, particularly in the creation of artificial neural networks, ensemble averaging is the process of creating multiple models and combining them to produce a desired output, as opposed to creating just one model. Frequently an ensemble of models performs better than any individual model, because the various errors of the models “average out”.

### Which classifier to choose for a classification task?

When considering building a classifier e.g. for a software application, a number of different aspects influence the choice of the preferred classifier type to use. Building or training a classifier is essentially statistical inference. This means that we try to identify and quantify stochastic (often unknown) relations between feature variables and the categories to be predicted. What is the influence of increased cholesterol on the risk of a heart infarct of a patient, within say the next year? Which other variables besides the current cholesterol level determine this risk? The two categories to “predict” by a classifier are “heart infarct occurs”, or “it does not”.

The theoretically optimal classifier is called the Bayes classifier. It minimizes the loss-function or risk as defined here. When all types of misclassifications are associated with equal losses (outcome A becomes B is as undesired as when outcome B becomes A), the Bayes classifier with the minimal error rate (on a test set) is the optimal one for the classification task. In general, it is unknown what is the optimal classifier type and true parameters  $\theta$ . However, bounds on the optimal Bayes error rate have been derived. For, for example, the K-nearest neighbor classifier theoretical results are derived that bound the error rate, in relation to the optimal Bayes error rate.

In essence, building a classifier brings model selection with it. Feature selection – using only a subset of the available feature variables to predict the most likely categorical outcome – by itself entails model selection. Choosing among the extensive set of different classifiers makes model selection even more complex. No particular classification algorithm is “the best” for all problems. The pragmatic approach to this open problem is to combine prior knowledge of the classification task (e.g. distributional assumptions) with a search process where different types of classifiers are developed and their performance compared.

If a simple classifier with only a few parameters (small  $\theta$ ) does the job as well as a much more complex classification algorithm, choose the simpler one. Only to add that the performance of a classifier is but one of the criteria to apply when choosing the best classification algorithm. Distributional assumptions, insight provided into the discovered relations between variables, whether an algorithm can cope with missing feature variables, whether a change in class prior probability can be incorporated, speed of the training algorithm, memory requirements, parallelization of the classification process, resemblance to the human perceptual system, and other aspects as well. In visual pattern recognition invariance to variations in color, rotation and scale are extra properties that need to be accounted for.

#### **4.1.3.1. Linear discriminant analysis (LDA)**

Linear discriminant analysis (LDA) and the related Fisher’s linear discriminant are methods used in statistics, pattern recognition and machine learning to find a linear combination of features which characterizes or separates two or more classes of objects or events. The resulting combination may be used as a linear classifier, or, more commonly, for dimensionality reduction before later classification. LDA is closely related to principal component analysis (PCA) and factor analysis in that they both look for linear combinations of variables which best explain the data. LDA explicitly attempts to model the difference between the classes of data. PCA on the other hand does not take into account any difference in class, and factor analysis builds the feature combinations based on differences rather than similarities. Discriminant analysis is also different from factor analysis in that it is not an interdependence technique: a distinction between independent variables and

dependent variables (also called criterion variables) must be made. LDA works when the measurements made on independent variables for each observation are continuous quantities.

### **LDA for two classes**

In LDA we seek a linear mapping that maximizes the linear class separability in the new representation of the data. We consider a set of  $n$   $p$ -dimensional observations  $x_1, \dots, x_n$  (the vectors of indicators representing the acquired spectrum), with  $n_1$  samples labeled as belonging to class 1 ( $\lambda_1$ ) and  $n_2$  samples as belonging to class 2 ( $\lambda_2$ ).

We will denote the set of observation in the  $i$ -th class as  $\Lambda_i$ . Given a vector  $w$  with unit norm, we may form a projection of the  $x_i$  onto a line in the direction of  $w$  using:

$$y = w^T x \quad (4.1)$$

where the superscript  $T$  denotes the matrix transpose. We want to choose  $w$  in order to provide a linear mapping that provides maximal separation of the two classes. One natural measure of separation between the projected points  $y_i$  is the difference between their means. We may calculate the  $p$ -dimensional sample mean for each class using:

$$m_i = \frac{1}{n_i} \sum_{x \in \Lambda_i} x \quad (4.2)$$

The sample mean for the projected points is given by:

$$\tilde{m}_i = \frac{1}{n_i} \sum_{y \in \Lambda_i} y = \frac{1}{n_i} \sum_{x \in \Lambda_i} w^T x = w^T m_i \quad (4.3)$$

We can use the last equation to measure the separation of the means for the two classes:

$$|\tilde{m}_1 - \tilde{m}_2| = |w^T(m_1 - m_2)| \quad (4.4)$$

where  $|\cdot|$  denotes the determinant. To obtain good class separation, and hence good classification performance, we want the separation of the means to be as large as possible relative to some measure of standard deviation for the observation in each class. We will use the scatter as our measure of the standard deviations. The scatter for the  $i$ -th class of projected data points is given by:

$$\tilde{s}_i^2 = \sum_{y \in \Lambda_i} (y - \tilde{m}_i)^2 \quad (4.5)$$

We define the total within-class scatter to be  $\tilde{s}_1^2 + \tilde{s}_2^2$ . The LDA is defined as the vector  $w$  that maximizes the function:

$$J(w) = \frac{|\tilde{m}_1 - \tilde{m}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2} \quad (4.6)$$

A little bit of linear algebra manipulation (see Duda and Hart [1973] for details) allow us to write the solution to the maximization as:

$$w = S_w^{-1}(m_1 - m_2) \quad (4.7)$$

where  $S_w$  is the within-class scatter matrix defined by:

$$S_w = S_1 + S_2 \quad (4.8)$$

and

$$S_i = \sum_{x \in \Lambda_i} (x - m_i)(x - m_i)^T \quad (4.9)$$

Note that the matrix  $S_w$  is proportional to the sample covariance matrix for the pooled  $p$ -dimensional data. Interestingly, it turns out that the LDA is the linear mapping with the maximum ratio of between-class scatter  $S_B$  to within-class scatter, where

$$S_B = (m_1 - m_2)(m_1 - m_2)^T \quad (4.10)$$

The input vectors data now reside in one dimension.

#### 4.1.3.2. *Principal component analysis (PCA)*

In PCA the main purpose is to reduce the dimensionality from  $p$  to  $d$ , where  $d < p$ , while at the same time accounting for as much of the variation in the original data set as possible. With PCA, we transform the input vector data to a new set of coordinates or variables that are a linear combination of the original variable. In addition, the observation in the new principal component space are uncorrelated. We start with our centered data matrix  $X_c$  that has dimension  $n \times p$ . This matrix contains observation that are centered about the mean; i.e., the sample mean has been subtracted from each row. We then form the sample covariance matrix  $S$  as:

$$S = \frac{1}{n-1} X_c^T X_c \quad (4.11)$$

The  $jk$ -th element of  $S$  is given by:

$$s_{jk} = \frac{1}{n-1} \sum_{i=1}^n (x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k), \quad jk = 1, \dots, p \quad (4.12)$$

with

$$\bar{x}_j = \frac{1}{n} \sum_{i=1}^n x_{ij} \quad (4.13)$$

The next step is to calculate the eigenvectors and eigenvalues of the matrix  $S$ . The eigenvalues are found by solving the following equation for each  $l_j, j=1, \dots, p$ :

$$|S - lI| = 0 \quad (4.14)$$

where  $I$  is a identity matrix. This equation produces a polynomial equation of degree  $p$ . The eigenvectors are obtained by solving the following set of equations for  $a_j$ :

$$(S - l_j I)a_j = 0, \quad j = 1, \dots, p \quad (4.15)$$

subject to the condition that the set of eigenvectors is orthonormal. This means that the magnitude of each eigenvector is one, and they are orthogonal to each other:

$$a_i a_i^T = 1 \quad (4.16)$$

$$a_j a_i^T = 0 \quad (4.17)$$

for  $i, j = 1, \dots, p$  and  $i \neq j$ .

A major result in matrix algebra shows that any square, symmetric, nonsingular matrix can be transformed to a diagonal matrix using

$$L = A^T S A \quad (4.18)$$

where the columns of  $A$  contain the eigenvectors of  $S$ , and  $L$  is a diagonal matrix with the eigenvalues along the diagonal. By convention, the eigenvalues are ordered in descending order  $l_1 \geq l_2 \geq \dots \geq l_p$ , with the same order imposed on the corresponding eigenvectors. We use the eigenvectors of  $S$  to obtain the principal components (PCs). The  $j$ -th PC is given by

$$z_j = a_j^T (x - \bar{x}), \quad j = 1, \dots, p \quad (4.19)$$

and the elements of  $a$  provide the weights or coefficients of the old variables in the new PC coordinate space. Scaling the eigenvectors to have unit length produces PCs that are uncorrelated and whose variances are equal to the corresponding eigenvalue. Now we can transformed the input vector data to the PC coordinate system via the following equation:

$$z = X_c A \quad (4.20)$$

The matrix  $Z$  contains the principal component scores. We know from results in linear algebra that the sum of the variances of the original variables is equal to the sum of the eigenvalues. The idea of dimensionality reduction with PCA is that one could include in the analysis only those PCs that have the highest eigenvalues, thus accounting for the highest amount of variation with fewer dimensions or PC variables. We can reduce the dimensionality to  $d$  with the following:

$$Z_d = X_c A_d \quad (4.21)$$

where  $A_d$  contains the first  $d$  eigenvectors or columns of  $\mathbf{A}$ . We see that  $Z_d$  is an  $n \times d$  matrix (each vector of data now is represented now only by  $d$  elements), and  $A_d$  is a  $p \times d$  matrix. We can scale the input vector data first to have standard units and this means we have the  $j$ -th element of  $\mathbf{x}^*$  given by

$$x_j^* = \frac{(x_j - \bar{x}_j)}{\sqrt{s_{jj}}}, \quad j = 1, \dots, p \quad (4.22)$$

where  $s_{jj}$  is the variance of  $x_j$  (i.e., the  $jj$ -th element of the sample covariance matrix  $S$ ). The standardized data  $\mathbf{x}^*$  are then treated as observations in the PCA process. The covariance of this standardized data set is the same as the correlation matrix. The  $ij$ -th element of the sample correlation matrix  $R$  is given by:

$$r_{ij} = \frac{s_{ij}}{\sqrt{s_{ii}\sqrt{s_{jj}}}} \quad (4.23)$$

where  $s_{ij}$  is the  $ij$ -th element of  $\mathbf{S}$  and  $s_{ii}$  is the  $i$ -th diagonal element of  $S$ . The rest of the results from PCA hold for the  $\mathbf{x}^*$  but now the matrices  $A$  and  $A_d$  contain the eigenvectors of the correlation matrix. There is some benefit of using the correlation matrix rather than the covariance matrix when one wants to compare the results of PCA among different analyses. To choose the number of PCs, we used the cumulative percentage of variance explained. The idea is to select those  $d$  PCs that contribute a specified cumulative percentage of total variation in the data, which is calculated using:

$$t_d = 100 \frac{\sum_{i=1}^d l_i}{\sum_{j=1}^p l_j} \quad (4.24)$$

If the correlation matrix is used for PCA, then this is simplified to:

$$t_d = \frac{100}{p} \sum_{i=1}^d l_i \quad (4.25)$$

A typical values range of  $t_d$  is between 70% and 95%).

#### 4.1.3.3. *Partial least squares regression (PLS)*

##### *Basic Ideas*

Partial least squares regression is an extension of the multiple linear regression model. In its simplest form, a linear model specifies the (linear) relationship between a dependent (response) variable  $Y$ , and a set of predictor variables, the  $X$ 's, so that:

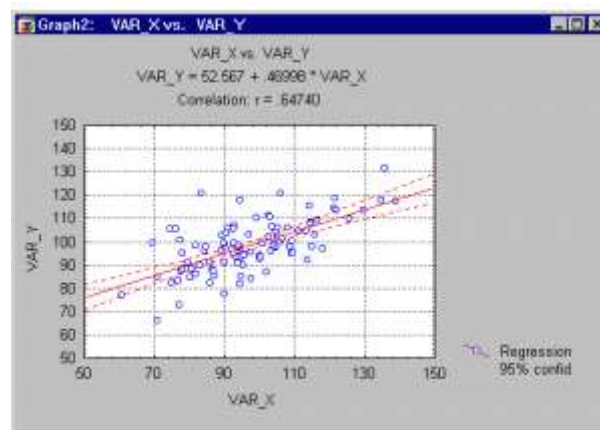
$$Y = b_0 + b_1 X_1 + b_2 X_2 + \dots + b_p X_p \quad (4.26)$$



In this equation  $b_0$  is the regression coefficient for the intercept and the  $b_i$  values are the regression coefficients (for variables 1 through p) computed from the data.

So for example, you could estimate (i.e., predict) a person's weight as a function of the person's height and gender. You could use linear regression to estimate the respective regression coefficients from a sample of data, measuring height, weight, and observing the subjects' gender. For many data analysis problems, estimates of the linear relationships between variables are adequate to describe the observed data, and to make reasonable predictions for new observations.

The general purpose of multiple regression is to learn more about the relationship between several independent or predictor variables and a dependent or criterion variable. In the social and natural sciences multiple regression procedures are very widely used in research. In general, multiple regression allows the researcher to ask (and hopefully answer) the general question "what is the best predictor of ...". The general computational problem that needs to be solved in multiple regression analysis is to fit a straight line to a number of points.



**Figure 4 – Multiple Regression sample**

In the scatterplot (Figure 4), we have an independent or X variable, and a dependent or Y variable. These variables may, for example, represent IQ (intelligence as measured by a test) and school achievement (grade point average; GPA), respectively. Each point in the plot represents one student, that is, the respective student's IQ and GPA. The goal of linear regression procedures is to fit a line through the points. Specifically, the squared deviations of the observed points from that line are minimized. Thus, this general procedure is sometimes also referred to as least squares estimation.

The multiple linear regression model has been extended in a number of ways to address more sophisticated data analysis problems. The multiple linear regression model serves as the basis for a number of multivariate methods such as discriminant analysis (i.e., the prediction of group membership from the levels of continuous predictor variables), principal components regression (i.e., the prediction of responses on the dependent variables from factors underlying the levels of the predictor variables), and canonical correlation (i.e., the prediction of factors underlying responses on the dependent variables from factors underlying the levels of the predictor variables). These multivariate methods all have two important properties in common. These methods impose restrictions such that (i) factors underlying the Y and X variables are extracted from the  $Y'Y$  and  $X'X$  matrices, respectively, and never from cross-product matrices involving both the Y and X

variables, and (ii) the number of prediction functions can never exceed the minimum of the number of Y variables and X variables.

In short, partial least squares regression is probably the least restrictive of the various multivariate extensions of the multiple linear regression model. This flexibility allows it to be used in situations where the use of traditional multivariate methods is severely limited, such as when there are fewer observations than predictor variables. Furthermore, partial least squares regression can be used as an exploratory analysis tool to select suitable predictor variables and to identify outliers before classical linear regression. Partial least squares regression has been used in various disciplines such as chemistry, economics, medicine, psychology, and pharmaceutical science where predictive linear modeling, especially with a large number of predictors, is necessary.

### ***Computational Approach - Basic Model***

As in multiple linear regression, the main purpose of partial least squares regression is to build a linear model,  $Y=XB+E$ , where Y is an  $n$  cases by  $m$  variables response matrix, X is an  $n$  cases by  $p$  variables predictor matrix, B is a  $p$  by  $m$  regression coefficient matrix, and E is a noise term for the model which has the same dimensions as Y. Usually, the variables in X and Y are centered by subtracting their means and scaled by dividing by their standard deviations.

Both principal components regression and partial least squares regression produce factor scores as linear combinations of the original predictor variables, so that there is no correlation between the factor score variables used in the predictive regression model. For example, suppose we have a data set with response variables Y (in matrix form) and a large number of predictor variables X (in matrix form), some of which are highly correlated. A regression using factor extraction for this type of data computes the factor score matrix  $T=XW$  for an appropriate weight matrix W, and then considers the linear regression model  $Y=TQ+E$ , where Q is a matrix of regression coefficients (loadings) for T, and E is an error (noise) term. Once the loadings Q are computed, the above regression model is equivalent to  $Y=XB+E$ , where  $B=WQ$ , which can be used as a predictive regression model.

Principal components regression and partial least squares regression differ in the methods used in extracting factor scores. In short, principal components regression produces the weight matrix W reflecting the covariance structure between the predictor variables, while partial least squares regression produces the weight matrix W reflecting the covariance structure between the predictor and response variables.

For establishing the model, partial least squares regression produces a  $p$  by  $c$  weight matrix W for X such that  $T=XW$ , i.e., the columns of W are weight vectors for the X columns producing the corresponding  $n$  by  $c$  factor score matrix T. These weights are computed so that each of them maximizes the covariance between responses and the corresponding factor scores. Ordinary least squares procedures for the regression of Y on T are then performed to produce Q, the loadings for Y (or weights for Y) such that  $Y=TQ+E$ . Once Q is computed, we have  $Y=XB+E$ , where  $B=WQ$ , and the prediction model is complete.

One additional matrix necessary for a complete description of partial least squares regression procedures is the  $p$  by  $c$  factor loading matrix P which gives a factor model  $X=TP+F$ , where F is the

unexplained part of the X scores. We now can describe the algorithms for computing partial least squares regression.

### ***NIPALS Algorithm***

The standard algorithm for computing partial least squares regression components (i.e., factors) is nonlinear iterative partial least squares (NIPALS). There are many variants of the NIPALS algorithm which normalize or do not normalize certain vectors. The following algorithm, which assumes that the X and Y variables have been transformed to have means of zero, is considered to be one of most efficient NIPALS algorithms.

For each  $h=1, \dots, c$ , where  $A_0=X'Y$ ,  $M_0=X'X$ ,  $C_0=I$ , and  $c$  given:

- i) compute  $q_h$ , the dominant eigenvector of  $A_h'A_h$
- ii)  $w_h=C_hA_hq_h$ ,  $w_h=w_h/||w_h||$ , and store  $w_h$  into  $W$  as a column
- iii)  $p_h=M_hw_h$ ,  $c_h=w_h'M_hw_h$ ,  $p_h=p_h/c_h$ , and store  $p_h$  into  $P$  as a column
- iv)  $q_h=A_h'w_h/c_h$ , and store  $q_h$  into  $Q$  as a column
- v)  $A_{h+1}=A_h - c_hp_hq_h'$  and  $M_{h+1}=M_h - c_hp_hp_h'$
- vi)  $C_{h+1}=C_h - w_hp_h'$

The factor scores matrix  $T$  is then computed as  $T=XW$  and the partial least squares regression coefficients  $B$  of  $Y$  on  $X$  are computed as  $B=WQ$ .

### ***SIMPLS Algorithm***

An alternative estimation method for partial least squares regression components is the SIMPLS algorithm (de Jong, 1993), which can be described as follows.

For each  $h=1, \dots, c$ , where  $A_0=X'Y$ ,  $M_0=X'X$ ,  $C_0=I$ , and  $c$  given:

compute  $q_h$ , the dominant eigenvector of  $A_h'A_h$

- i)  $w_h=A_hq_h$ ,  $c_h=w_h'M_hw_h$ ,  $w_h=w_h/\sqrt{c_h}$ , and store  $w_h$  into  $W$  as a column
- ii)  $p_h=M_hw_h$ , and store  $p_h$  into  $P$  as a column
- iii)  $q_h=A_h'w_h$ , and store  $q_h$  into  $Q$  as a column
- iv)  $v_h=C_hp_h$ , and  $v_h=v_h/||v_h||$
- v)  $C_{h+1}=C_h - v_hv_h'$  and  $M_{h+1}=M_h - p_hp_h'$
- vi)  $A_{h+1}=C_hA_h$

Similarly to NIPALS, the  $T$  of SIMPLS is computed as  $T=XW$  and  $B$  for the regression of  $Y$  on  $X$  is computed as  $B=WQ'$ .

See also the Matlab manual page of PLS for more details on how to use the SIMPLS algorithm and what can be computed.

#### **4.1.3.4. Expectation-maximization algorithm (EM)**

##### ***Overview***

The general purpose of these techniques is to detect clusters in observations (or variables) and to assign those observations to the clusters. A typical example application for this type of analysis is a marketing research study in which a number of consumer behavior related variables are measured

for a large sample of respondents. The purpose of the study is to detect “market segments”, i.e., groups of respondents that are somehow more similar to each other (to all other members of the same cluster) when compared to respondents that “belong to” other clusters. In addition to identifying such clusters, it is usually equally of interest to determine how the clusters are different, i.e., determine the specific variables or dimensions that vary and how they vary in regard to members in different clusters.

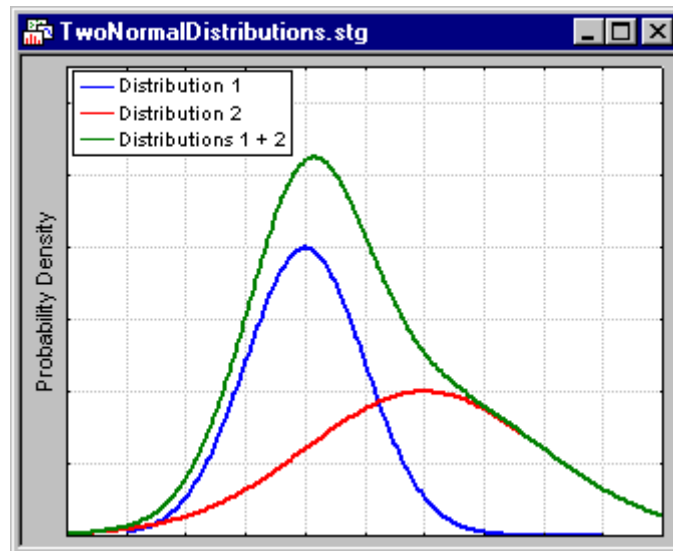
### Extensions and generalizations

k-means clustering idea: to reiterate, the classic k-Means algorithm was popularized and refined by Hartigan (1975; see also Hartigan and Wong, 1978). The basic operation of that algorithm is relatively simple: given a fixed number of (desired or hypothesized) k clusters, assign observations to those clusters so that the means across clusters (for all variables) are as different from each other as possible.

The EM (expectation maximization) algorithm extends this basic approach to clustering in two important ways:

- Instead of assigning cases or observations to clusters to maximize the differences in means for continuous variables, the EM clustering algorithm computes probabilities of cluster memberships based on one or more probability distributions. The goal of the clustering algorithm then is to maximize the overall probability or likelihood of the data, given the (final) clusters.
- Unlike the classic implementation of k-means clustering, the general EM algorithm can be applied to both continuous and categorical variables (note that the classic k-means algorithm can also be modified to accommodate categorical variables).

The EM algorithm for clustering is described in detail in Witten and Frank (2001). The basic approach and logic of this clustering method is as follows. Suppose you measure a single continuous variable in a large sample of observations. Further, suppose that the sample consists of two clusters of observations with different means (and perhaps different standard deviations); within each sample, the distribution of values for the continuous variable follows the normal distribution. The resulting distribution of values (in the population) may look like this:



**Figure 5 – The illustration shows two normal distributions with different means and different standard deviations, and the sum of the two distributions.**

The illustration shows two normal distributions with different means and different standard deviations, and the sum of the two distributions. Only the mixture (sum) of the two normal distributions (with different means and standard deviations) would be observed. The goal of EM clustering is to estimate the means and standard deviations for each cluster so as to maximize the likelihood of the observed data (distribution). Put another way, the EM algorithm attempts to approximate the observed distributions of values based on mixtures of different distributions in different clusters. With the implementation of the EM algorithm in some computer programs, you may be able to select (for continuous variables) different distributions such as the normal, log-normal, and Poisson distributions. You can select different distributions for different variables and, thus, derive clusters for mixtures of different types of distributions. The EM algorithm can also accommodate categorical variables. The method will at first randomly assign different probabilities (weights, to be precise) to each class or category, for each cluster. In successive iterations, these probabilities are refined (adjusted) to maximize the likelihood of the data given the specified number of clusters. The results of EM clustering are different from those computed by k-means clustering. The latter will assign observations to clusters to maximize the distances between clusters. The EM algorithm does not compute actual assignments of observations to clusters, but classification probabilities. In other words, each observation belongs to each cluster with a certain probability. Of course, as a final result you can usually review an actual assignment of observations to clusters, based on the (largest) classification probability.

An important question that needs to be answered before applying the k-means or EM clustering algorithms is how many clusters there are in the data. This is not known a priori and, in fact, there might be no definite or unique answer as to what value  $k$  should take. In other words,  $k$  is a nuisance parameter of the clustering model. Luckily, an estimate of  $k$  can be obtained from the data using the method of cross-validation. Remember that the k-means and EM methods will determine cluster solutions for a particular user-defined number of clusters. The k-means and EM clustering techniques can be optimized and enhanced for typical applications in data mining. The general metaphor of data mining implies the situation in which an analyst searches for useful structures and “nuggets” in the data, usually without any strong a priori expectations of what the analyst might

find (in contrast to the hypothesis-testing approach of scientific research). In practice, the analyst usually does not know ahead of time how many clusters there might be in the sample. For that reason, some programs include an implementation of a v-fold cross-validation algorithm for automatically determining the number of clusters in the data.

### ***The EM Algorithm***

An expectation–maximization (EM) algorithm is an iterative method for finding maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables. The EM iteration alternates between performing an expectation (E) step, which creates a function for the expectation of the log-likelihood evaluated using the current estimate for the parameters, and a maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the E step. These parameter-estimates are then used to determine the distribution of the latent variables in the next E step. The EM algorithm is used to find the maximum likelihood parameters of a statistical model in cases where the equations cannot be solved directly. Typically these models involve latent variables in addition to unknown parameters and known data observations. That is, either there are missing values among the data, or the model can be formulated more simply by assuming the existence of additional unobserved data points. (For example, a mixture model can be described more simply by assuming that each observed data point has a corresponding unobserved data point, or latent variable, specifying the mixture component that each data point belongs to). Finding a maximum likelihood solution requires taking the derivatives of the likelihood function with respect to all the unknown values — e.g. both the parameters and the latent variables — and simultaneously solving the resulting equations. In statistical models with latent variables, this usually is not possible. Instead, the result is typically a set of interlocking equations in which the solution to the parameters requires the values of the latent variables and vice-versa, but substituting one set of equations into the other produces an unsolvable equation. The EM algorithm proceeds from the observation that the following is a way to solve these two sets of equations numerically. One can simply pick arbitrary values for one of the two sets of unknowns, use them to estimate the second set, then use these new values to find a better estimate of the first set, and then keep alternating between the two until the resulting values both converge to fixed points. It’s not obvious that this will work at all, but in fact it can be proven that in this particular context it does, and that the derivative of the likelihood is (arbitrarily close to) zero at that point, which in turn means that the point is either a maximum or a saddle point. In general there may be multiple maxima, and no guarantee that the global maximum will be found. Some likelihoods also have singularities in them, i.e. nonsensical maxima. For example, one of the “solutions” that may be found by EM in a mixture model involves setting one of the components to have zero variance and the mean parameter for the same component to be equal to one of the data points.

### ***Examples - Gaussian mixture***

Let  $x = (x_1, x_2, \dots, x_n)$  be a sample of  $n$  independent observations from a mixture of two multivariate normal distributions of dimension  $d$ , and let  $z = (z_1, z_2, \dots, z_n)$  be the latent variables that determine the component from which the observation originates.

$$X_i | (Z_i = 1) \sim \mathcal{N}_d(\mu_1, \Sigma_1) \tag{4.27}$$

and

$$X_i | (Z_i = 2) \sim \mathcal{N}_d(\mu_2, \Sigma_2) \tag{4.28}$$

where:

$$P(Z_i = 1) = \tau_1 \quad (4.29)$$

and

$$P(Z_i = 2) = \tau_2 = 1 - \tau_1 \quad (4.30)$$

The aim is to estimate the unknown parameters representing the "mixing" value between the Gaussians and the means and covariances of each:

$$\theta = (\tau, \mu_1, \mu_2, \Sigma_1, \Sigma_2) \quad (4.31)$$

where the likelihood function is:

$$\begin{aligned} L(\theta; \mathbf{x}, \mathbf{z}) &= P(\mathbf{x}, \mathbf{z} | \theta) = \\ &= \prod_{i=1}^n \sum_{j=1}^2 \mathbb{I}(Z_i = j) \tau_j f(x_i; \mu_j, \Sigma_j) \end{aligned} \quad (4.32)$$

where  $\mathbb{I}$  is an indicator function and  $f$  is the probability density function of a multivariate normal. This may be rewritten in exponential family form:

$$\begin{aligned} L(\theta; \mathbf{x}, \mathbf{z}) &= \\ \exp \left( \sum_{i=1}^n \sum_{j=1}^2 \mathbb{I}(Z_i = j) \left[ \log(\tau_j) - \frac{1}{2} \log |\Sigma_j| - \frac{1}{2} (x_i - \mu_j)^T \Sigma_j^{-1} (x_i - \mu_j) - \frac{d}{2} \log(2\pi) \right] \right) \end{aligned} \quad (4.33)$$

To see the last equality, note that for each  $i$  all indicators  $\mathbb{I}(Z_i = j)$  are equal to zero, except for one which is equal to one. The inner sum thus reduces to a single term.

*E step]*

Given our current estimate of the parameters  $\theta$ , the conditional distribution of the  $Z_i$  is determined by Bayes theorem to be the proportional height of the normal density weighted by  $\tau$ .

$$\begin{aligned} T_{j,i}^{(t)} &:= P(Z_i = j | X_i = x_i; \theta^{(t)}) \\ &= \frac{\tau_j^{(t)} f(x_i; \mu_j^{(t)}, \Sigma_j^{(t)})}{\tau_1^{(t)} f(x_i; \mu_1^{(t)}, \Sigma_1^{(t)}) + \tau_2^{(t)} f(x_i; \mu_2^{(t)}, \Sigma_2^{(t)})} \end{aligned} \quad (4.34)$$

Thus, the E step results in the function:

$$Q(\theta | \theta^{(t)}) = E[\log L(\theta; \mathbf{x}, \mathbf{Z})] =$$

$$= \sum_{i=1}^n \sum_{j=1}^2 T_{j,i}^{(t)} \left[ \log \tau_j - \frac{1}{2} \log |\Sigma_j| - \frac{1}{2} (x_i - \mu_j)^T \Sigma_j^{-1} (x_i - \mu_j) - \frac{d}{2} \log(2\pi) \right] \quad (4.35)$$

To see the last equality, note that we are summing over all possible values of the assignments of  $z$ , where the probability of each  $z$  is the product  $\prod_{i=1}^n T_{j,i}^{(t)}$ . Now look at the coefficient of each term inside the summation for  $L(\theta; \mathbf{x}, \mathbf{z})$ . There will be two terms  $T_{1,i}^{(t)} \left( \sum_{j=1,2} \prod_{k \neq i} T_{j,k}^{(t)} \right)$  and  $T_{2,i}^{(t)} \left( \sum_{j=1,2} \prod_{k \neq i} T_{j,k}^{(t)} \right)$ . Since the term in parenthesis marginalizes over all possible assignments of samples  $k \neq i$ , it equals 1. Thus the coefficients of each term are just  $T_{1,i}^{(t)}$  and  $T_{2,i}^{(t)}$  which yields to the equality.

M step]

The quadratic form of  $Q(\theta|\theta^{(t)})$  means that determining the maximizing values of  $\theta$  is relatively straightforward. Note that  $\tau$ ,  $(\mu_1, \Sigma_1)$ ,  $(\mu_2, \Sigma_2)$  may be all maximized independently of each other since they all appear in separate linear terms.

To begin, consider  $\tau$ , which has the constraint  $\tau_1 + \tau_2 = 1$ :

$$\begin{aligned} \tau^{(t+1)} &= \operatorname{argmax}_{\tau} Q(\theta|\theta^{(t)}) = \\ &= \operatorname{argmax}_{\tau} \left( \left[ \sum_{i=1}^n T_{1,i}^{(t)} \right] \log \tau_1 + \left[ \sum_{i=1}^n T_{2,i}^{(t)} \right] \log \tau_2 \right) \end{aligned} \quad (4.36)$$

This has the same form as the MLE for the binomial distribution, so:

$$\tau_j^{(t+1)} = \frac{\sum_{i=1}^n T_{j,i}^{(t)}}{\sum_{i=1}^n (T_{1,i}^{(t)} + T_{2,i}^{(t)})} = \frac{1}{n} \sum_{i=1}^n T_{j,i}^{(t)} \quad (4.37)$$

For the next estimates of  $(\mu_1, \Sigma_1)$ :

$$\begin{aligned} (\mu_1^{(t+1)}, \Sigma_1^{(t+1)}) &= \operatorname{argmax}_{\mu_1, \Sigma_1} Q(\theta|\theta^{(t)}) = \\ &= \operatorname{argmax}_{\mu_1, \Sigma_1} \sum_{i=1}^n T_{1,i}^{(t)} \left( -\frac{1}{2} \log |\Sigma_j| - \frac{1}{2} (x_i - \mu_j)^T \Sigma_j^{-1} (x_i - \mu_j) \right) \end{aligned} \quad (4.38)$$

This has the same form as a weighted MLE for a normal distribution, so

$$\mu_1^{(t+1)} = \frac{\sum_{i=1}^n T_{1,i}^{(t)} x_i}{\sum_{i=1}^n T_{1,i}^{(t)}} \quad (4.39)$$

and



$$\Sigma_1^{(t+1)} = \frac{\sum_{i=1}^n T_{1,i}^{(t)} (x_i - \mu_1^{(t+1)}) (x_i - \mu_1^{(t+1)})^T}{\sum_{i=1}^n T_{1,i}^{(t)}} \quad (4.40)$$

and, by symmetry:

$$\mu_2^{(t+1)} = \frac{\sum_{i=1}^n T_{2,i}^{(t)} x_i}{\sum_{i=1}^n T_{2,i}^{(t)}} \quad (4.41)$$

and

$$\Sigma_2^{(t+1)} = \frac{\sum_{i=1}^n T_{2,i}^{(t)} (x_i - \mu_2^{(t+1)}) (x_i - \mu_2^{(t+1)})^T}{\sum_{i=1}^n T_{2,i}^{(t)}} \quad (4.42)$$

The EM algorithm has been implemented in the case where there is an underlying linear regression model explaining the variation of some quantity, but where the values actually observed are censored or truncated versions of those represented in the model. Special cases of this model include censored or truncated observations from a single normal distribution.

## 4.2. Image processing methods

In imaging science, image processing is any form of signal processing for which the input is an image, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. An image may be considered to contain sub-images sometimes referred to as regions-of-interest, ROIs, or simply regions. This concept reflects the fact that images frequently contain collections of objects each of which can be the basis for a region. In a sophisticated image processing system it should be possible to apply specific image processing operations to selected regions. Thus one part of an image (region) might be processed to suppress motion blur while another part might be processed to improve color rendition. Different operations can be performed on an image; follows a non- exhaustive list of image processing methods that can be applied.

- Image Arithmetic (addition, subtraction, multiplication, etc.)
- Point Operations:
  - Thresholding – select pixels with given values to produce binary image;
  - Adaptive Thresholding – like Thresholding except choose values locally;
  - Contrast Stretching – spreading out gray level distribution;
  - Histogram Equalization – general method of modifying intensity distribution;
  - Logarithm Operator – reducing contrast of brighter regions;
  - Exponential/"Raise to Power" Operator – enhancing contrast of brighter regions;
- Geometric Operations
  - Scale – change image content size;
  - Rotate – change image content orientation;

- Reflect – flip over image contents;
- Translate – change image content position;
- Affine Transformation – general image content linear geometric transformation;
- Image Analysis
  - Intensity Histogram – the image intensity distribution;
  - Classification – mapping from pixel values to feature or object classes;
  - Connected Components Labeling – grouping pixels with the same class label into regions;
- Morphology
  - Dilation – grow image regions;
  - Erosion – shrink image regions;
  - Opening – structured removal of image region boundary pixels;
  - Closing – structured filling in of image region boundary pixels;
  - Hit and Miss Transform – image pattern matching and marking;
  - Thinning – structured erosion using image pattern matching;
  - Thickening – structured dilation using image pattern matching;
  - Skeletonization/Medial Axis Transform – finding skeletons of binary regions;
- Digital Filters
  - Mean Filter – noise reduction (NR) using mean of neighborhood;
  - Median Filter – NR using median of neighborhood;
  - Gaussian Smoothing – NR using convolution with a Gaussian smoothing kernel;
  - Conservative Smoothing – NR using maximum and minimum of neighborhood;
  - Crimmins Speckle Removal – more complex NR operator;
  - Frequency Filters – high and low pass image filters, etc.;
  - Laplacian/Laplacian of Gaussian Filter – edge detection filter;
  - Unsharp Filter – edge enhancement filter;
- Feature Detectors
  - Roberts Cross Edge Detector –  $2 \times 2$  gradient edge detector;
  - Sobel Edge Detector –  $3 \times 3$  gradient edge detector;
  - Canny Edge Detector – non-maximal suppression of local gradient magnitude;
  - Compass Edge Detector –  $3 \times 3$  gradient edge detectors;
  - Zero Crossing Detector – edge detector using the Laplacian of Gaussian operator;
  - Line Detector – line (as distinct from edge) feature detector;
- Image Transforms
  - Distance Transform – maps binary images to distance from background;
  - Fourier Transform – maps image into spatial frequency domain;
  - Hough Transform – maps image into votes for specified shapes.
- Image Synthesis
  - Noise Generation – various types of image noise

#### 4.2.1. Image Arithmetic

Image arithmetic applies one of the standard arithmetic operations or a logical operator to two or more images. The operators are applied in a pixel-by-pixel fashion which means that the value of a pixel in the output image depends only on the values of the corresponding pixels in the input

images. Hence, the images normally have to be of the same size. One of the input images may be a constant value, for example when adding a constant offset to an image. Although image arithmetic is the most simple form of image processing, there is a wide range of applications. A main advantage of arithmetic operators is that the process is very simple and therefore fast. In many applications the processed images are taken from the same scene at different points of time, as, for example, in reduction of random noise by adding successive images of the same scene or motion detection by subtracting two successive images. Logical operators are often used to combine two (mostly binary) images. In the case of integer images, the logical operator is normally applied in a bitwise fashion. Then we can, for example, use a binary mask to select a particular region of an image.

#### 4.2.2. Point Operations

Single-point processing is a simple method of image enhancement. This technique determines a pixel value in the enhanced image dependent only on the value of the corresponding pixel in the input image. The process can be described with the mapping function

$$s = M(r) \tag{4.43}$$

where  $r$  and  $s$  are the pixel values in the input and output images, respectively. The form of the mapping function  $M$  determines the effect of the operation. It can be previously defined in an ad-hoc manner, as for thresholding or gamma correction, or it can be computed from the input image, as for histogram equalization. For example, a simple mapping function is defined by the thresholding operator (Figure 6):

$$s = \begin{cases} 0, & \text{if } r < T \\ 1, & \text{if } r > T \end{cases} \tag{4.44}$$

Point operators are also known as LUT-transformations, because the mapping function, in the case of a discrete image, can be implemented in a look-up table (LUT). A subgroup of the point processors is the set of anamorphosis operators. This notion describes all point operators with a strictly increasing or decreasing mapping function. Examples include the logarithm operator, exponential operator and contrast stretching, to name just a few. An operator where  $M$  changes over the image is adaptive thresholding. This is not a pure point operation anymore, because the mapping function, and therefore the output pixel value, depends on the local neighborhood of a pixel.

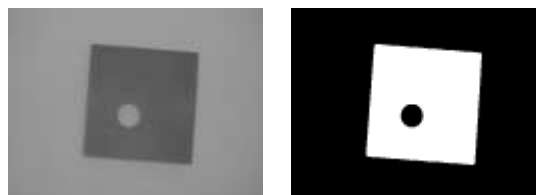


Figure 6 – An example image (left) after thresholding operator (right).

##### 4.2.2.1. Otsu's method

In computer vision and image processing, Otsu's method is used to automatically perform histogram shape-based image thresholding, or, the reduction of a graylevel image to a binary image. The algorithm assumes that the image to be thresholded contains two classes of pixels or bi-modal

histogram (e.g. foreground and background) then calculates the optimum threshold separating those two classes so that their combined spread (intra-class variance) is minimal.

### Method

In Otsu's method we exhaustively search for the threshold that minimizes the intra-class variance (the variance within the class), defined as a weighted sum of variances of the two classes:

$$\sigma_w^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t) \quad (4.45)$$

Weights  $\omega_i$  are the probabilities of the two classes separated by a threshold  $t$  and  $\sigma_i^2$  variances of these classes. Otsu shows that minimizing the intra-class variance is the same as maximizing inter-class variance:

$$\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t) = \omega_1(t)\omega_2(t)[\mu_1(t) - \mu_2(t)]^2 \quad (4.46)$$

which is expressed in terms of class probabilities  $\omega_i$  and class means  $\mu_i$ .

The class probability  $\omega_1(t)$  is computed from the histogram as  $t$ :

$$\omega_1(t) = \sum_0^t p(i) \quad (4.47)$$

while the class mean  $\mu_1(t)$  is:

$$\mu_1(t) = \sum_0^t p(i) x(i) \quad (4.48)$$

where  $x(i)$  is the value at the center of the  $i$ -th histogram bin. Similarly, you can compute  $\omega_2(t)$  and  $\mu_2(t)$  on the right-hand side of the histogram for bins greater than  $t$ . The class probabilities and class means can be computed iteratively. This idea yields an effective algorithm:

- i) Compute histogram and probabilities of each intensity level;
- ii) Set up initial  $\omega_i(0)$  and  $\mu_i(0)$ ;
- iii) Step through all possible thresholds  $t=1, \dots, \text{maximum intensity}$ :
  - a. Update  $\omega_i$  and  $\mu_i$ ;
  - b. Compute  $\sigma_b^2(t)$ ;
- iv) Desired threshold corresponds to the maximum  $\sigma_b^2(t)$ ;
- v) You can compute two maxima (and two corresponding thresholds).  $\sigma_{b1}^2(t)$  is the greater max and  $\sigma_{b2}^2(t)$  is the greater or equal maximum;
- vi) Desired threshold =  $\frac{\text{threshold}_1 + \text{threshold}_2}{2}$ ;

### 4.2.3. Geometric Operations

A geometric operation maps pixel information (i.e. the intensity values at each pixel location) in an input image to another location in an output image. For basic operators these functions are first order polynomials which take the form:

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = A \times \begin{bmatrix} x_1 \\ 1 \end{bmatrix} + B \quad (4.49)$$

Translation can be accomplished by specifying values for the  $B$  matrix, while scaling, rotation (Figure 7) and reflection are defined by instantiating variables in the  $A$  matrix. A general affine transformation (which uses both matrices) can be used to perform a combination of these operations at once, and is often used in applications, e.g. remote sensing, where we wish to correct for geometric distortions in the image introduced by perspective irregularities. Geometric operators are also used to improve the visualization of the image, e.g. zooming an interesting region of the image, or as a part of an image processing chain where, e.g. translation is required in order to register two images.

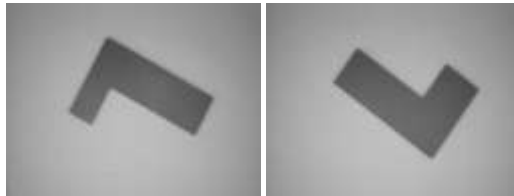


Figure 7 – An example image (left) after rotation (right).

### 4.2.4. Image Analysis

Many of the other categories of image processing operators transform an input image to form a new image without attempting to extract usable global information from the image. The operators listed above do extract globally useful information, such as:

- Pixel Value Distribution, the number of pixels having each value.
- Classified Pixels, the category of scene entity that the corresponding pixel is taken from.
- Connected Components, the groups of pixels all of which have the same label or classification.

It is possible to represent some of this information as an image (i.e. an image whose pixels are the category or index number of the scene structure), but the information need not always be so represented, nor even necessarily be representable as an image. For example one might analyze a binary image of an isolated object to determine the area or various moments of the shape. Another example would be to link together connected edge fragments to make a list of edge pixels. This category of operation is often considered part of the middle level image interpretation (i.e. a signal-to-symbol transformations or feature extraction), and whose results might ultimately be used in higher level image interpretation (i.e. symbol-to-symbol transformations such as scene description, object location, etc.). In this work there will be many sample of image analysis.

#### 4.2.5. Morphology

Morphological operators often take a binary image and a structuring element as input and combine them using a set operator (intersection, union, inclusion, complement). They process objects in the input image based on characteristics of its shape, which are encoded in the structuring element. Usually, the structuring element is sized  $3 \times 3$  and has its origin at the center pixel. It is shifted over the image and at each pixel of the image its elements are compared with the set of the underlying pixels. If the two sets of elements match the condition defined by the set operator (e.g. if the set of pixels in the structuring element is a subset of the underlying image pixels), the pixel underneath the origin of the structuring element is set to a pre-defined value (0 or 1 for binary images). A morphological operator is therefore defined by its structuring element and the applied set operator. For the basic morphological operators the structuring element contains only foreground pixels (i.e. ones) and “don’t care’s”. These operators, which are all a combination of erosion (Figure 9) and dilation (Figure 8), are often used to select or suppress features of a certain shape, e.g. removing noise from images or selecting objects with a particular direction. The more sophisticated operators take zeros as well as ones and “don’t care’s” in the structuring element. The most general operator is the hit and miss, in fact, all the other morphological operators can be deduced from it. Its variations are often used to simplify the representation of objects in a (binary) image while preserving their structure, e.g. producing a skeleton of an object using skeletonization and tidying up the result using thinning. Morphological operators can also be applied to graylevel images, e.g. to reduce noise or to brighten the image. However, for many applications, other methods like a more general spatial filter produces better results.

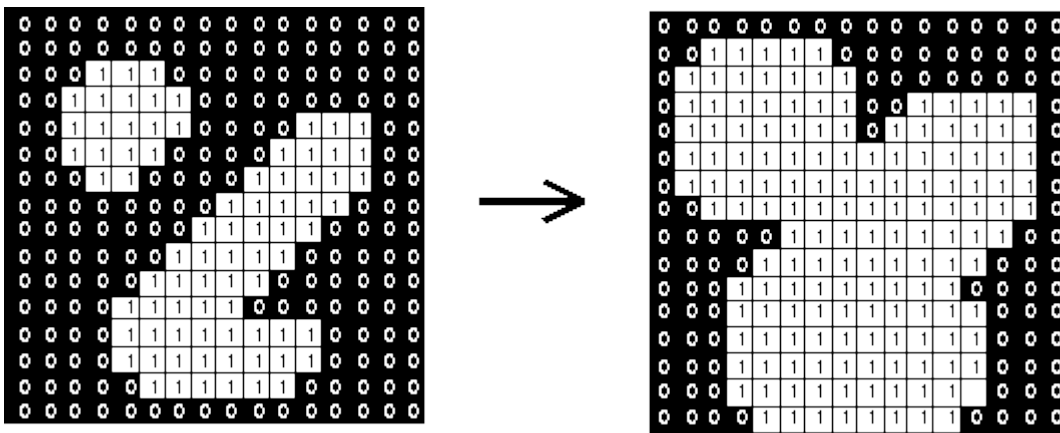


Figure 8 – Sample of dilation operation with  $3 \times 3$  structuring element.

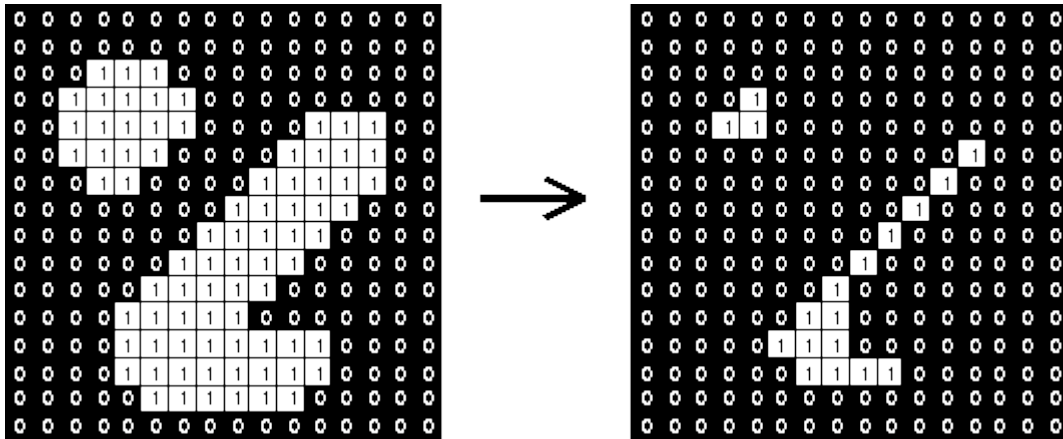


Figure 9 – Sample of erosion operation with 3×3 structuring element.

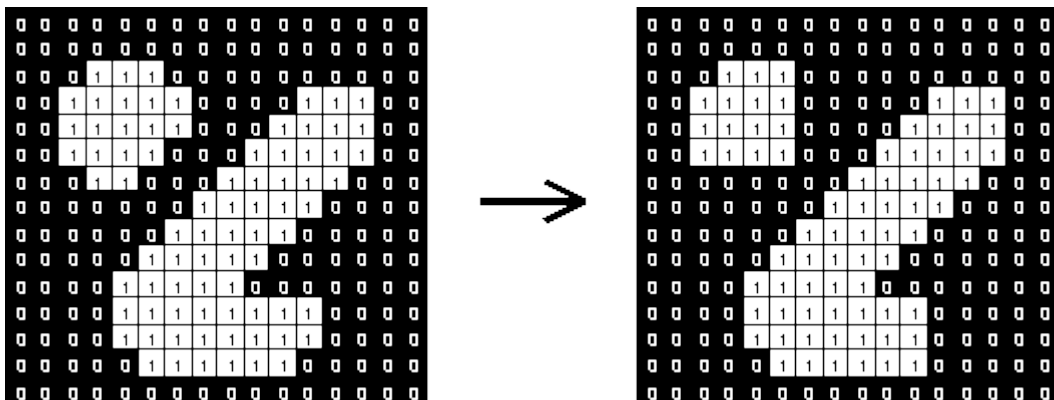


Figure 10 – Sample of opening operation with 3×3 structuring element.

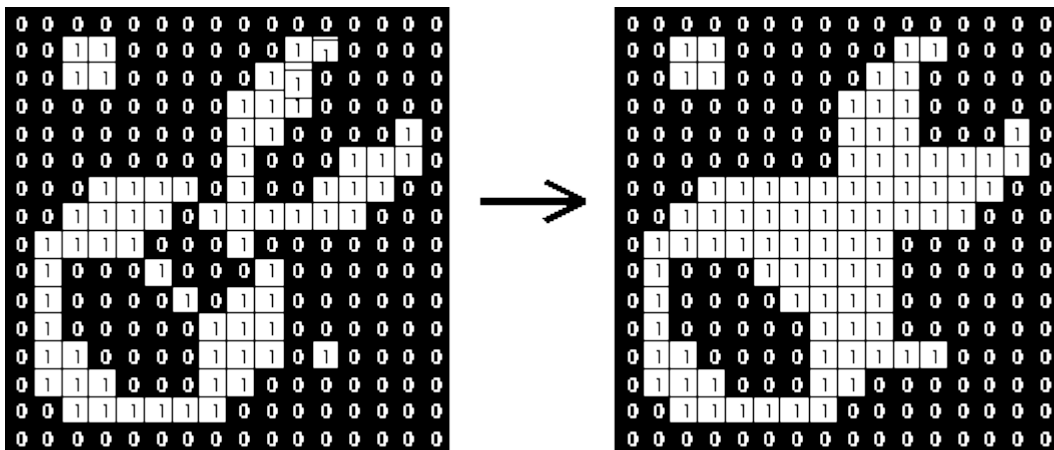


Figure 11 – Sample of closing operation with 3×3 structuring element.

#### 4.2.6. Digital Filters

In image processing filters are mainly used to suppress either the high frequencies in the image, i.e. smoothing the image (Figure 14), or the low frequencies, i.e. enhancing or detecting edges in the image. An image can be filtered either in the frequency or in the spatial domain. The first involves transforming the image into the frequency domain, multiplying it with the frequency filter function and re-transforming the result into the spatial domain. The filter function is shaped so as to attenuate some frequencies and enhance others. For example, a simple low-pass function is 1 for frequencies smaller than the cut-off frequency and 0 for all others. The corresponding process in the

spatial domain is to convolve the input image  $f(i,j)$  with the filter function  $h(i,j)$ . This can be written as:

$$g(i,j) = h(i,j) \odot f(i,j) \quad (4.50)$$

The mathematical operation is identical to the multiplication in the frequency space, but the results of the digital implementations vary, since we have to approximate the filter function with a discrete and finite kernel. The discrete convolution can be defined as a “shift and multiply” operation, where we shift the kernel over the image and multiply its value with the corresponding pixel values of the image. For a square kernel with size  $M \times M$ , we can calculate the output image with the following formula:

$$g(i,j) = \sum_{m=-\frac{M}{2}}^{\frac{M}{2}} \sum_{n=-\frac{M}{2}}^{\frac{M}{2}} h(m,n) f(i-m, j-n) \quad (4.51)$$

Various standard kernels exist for specific applications, where the size and the form of the kernel determine the characteristics of the operation. Follows the kernels for two examples, the mean (Figure 12) and the Laplacian operator (Figure 13).

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

Figure 12 – Convolution kernel for a mean filter.



0	-1	0
-1	4	-1
0	-1	0

Figure 13 – Convolution kernel for one form of the discrete Laplacian filter.

In contrast to the frequency domain, it is possible to implement non-linear filters in the spatial domain. In this case, the summations in the convolution function are replaced with some kind of non-linear operator:

$$g(i, j) = O_{m,n}[h(m, n)f(i - m, j - n)] \quad (4.52)$$

For most non-linear filters the elements of  $h(i, j)$  are all 1. A commonly used non-linear operator is the median, which returns the “middle” of the input values.

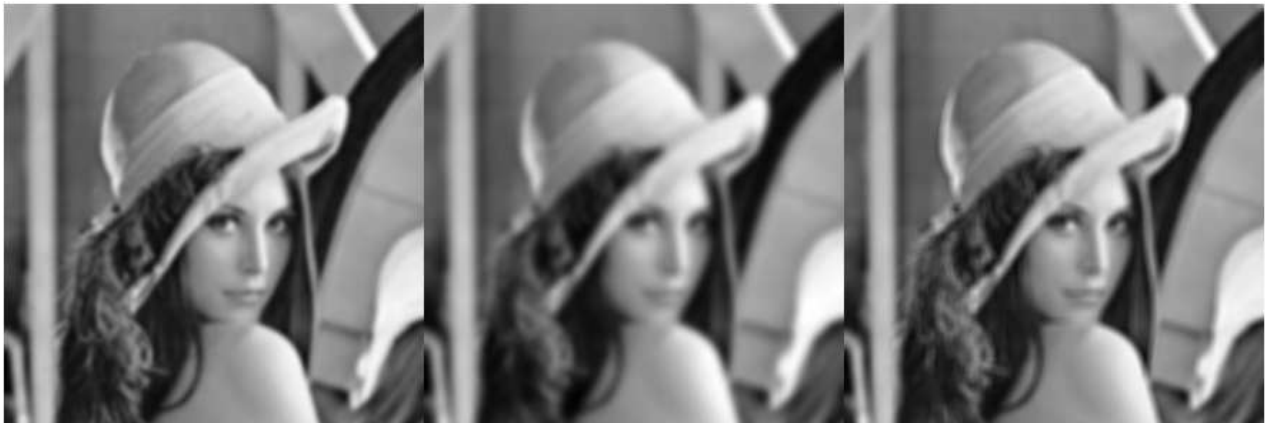


Figure 14 – An example image (left), smoothed (center) and very smoothed (right).

#### 4.2.7. Feature Detectors

The operators included in this section are those whose purpose is to identify meaningful image features on the basis of distributions of pixel gray levels. The two categories of operators included here are:

- Edge Pixel Detectors (Figure 15), that assign a value to a pixel in proportion to the likelihood that the pixel is part of an image edge (i.e. a pixel that is on the boundary between two regions of different intensity values);
- Line Pixel Detectors, that assign a value to a pixel in proportion to the likelihood that the pixel is part of a image line (i.e. a dark narrow region bounded on both sides by lighter regions, or vice-versa);

Detectors for other features can be defined, such as circular arc detectors in intensity images (or even more general detectors, as in the generalized Hough transform), or planar point detectors in range images, etc. Note that the operators in this section merely identify pixels likely to be part of such a structure. To actually extract the structure from the image it is then necessary to group together image pixels (which are usually adjacent). A verification stage also commonly occurs after the grouping stage (e.g. an examination of the residuals of the best fit line through a point set) and then the extraction of parameters of the structure (e.g. the equation of the line through the point set).

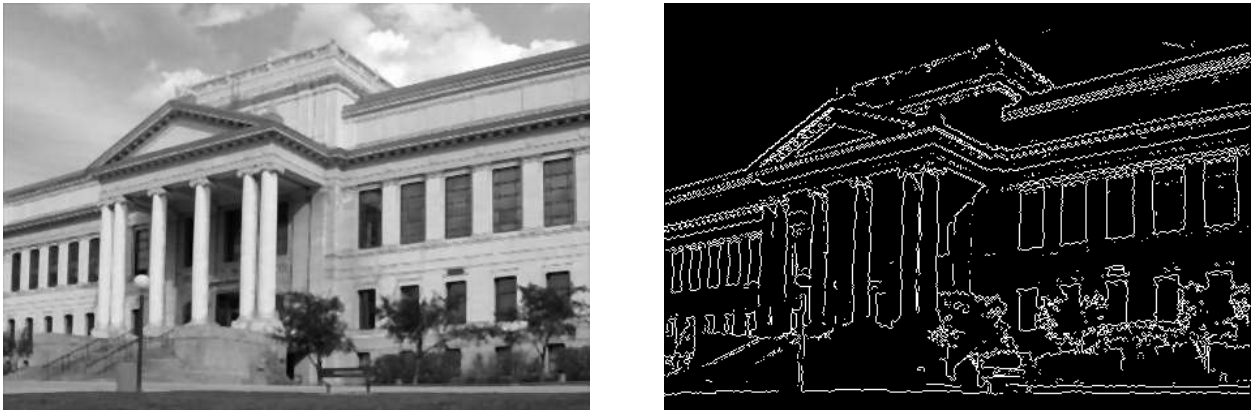


Figure 15 – An example image (left) and its canny edge detection resulting image (right).

#### 4.2.8. Image Transforms

Most of the image processing operators transform an input image to form a new image. However, the operators demonstrated in this section produce output images whose character is generally quite different from the character of the input images. This difference might be in the geometry of the information in the image or the nature of the information itself. Of course, the size of the output image might be quite different from that of the input image (and the values at each pixel might be different, as in the complex number output of the Fourier transform). The usual purpose of applying a transformation is to help make more obvious or explicit some desired information. In the case of the operators in this group, the main effect is to make explicit:

- Distance Transform, the thickness of image features;
- Fourier Transform, the spatial frequency composition of the image;
- Hough Transform, the parameters of geometric shapes;

The transformation is often followed by a thresholding operation, which is intended to select the most prominent or relevant features. It may then be possible to apply an inverse transform, for the purpose of reconstructing the geometry of the original image, except with the desired features explicit or enhanced.

##### 4.2.8.1. *Hough transform*

The Hough transform is a feature extraction technique used in image analysis, computer vision, and digital image processing. The purpose of the technique is to find imperfect instances of objects within a certain class of shapes by a voting procedure. This voting procedure is carried out in a parameter space, from which object candidates are obtained as local maxima in a so-called accumulator space that is explicitly constructed by the algorithm for computing the Hough transform. The classical Hough transform was concerned with the identification of lines in the

image, but later the Hough transform has been extended to identifying positions of arbitrary shapes, most commonly circles or ellipses.

Theory

In automated analysis of digital images, a sub-problem often arises of detecting simple shapes, such as straight lines, circles or ellipses. In many cases an edge detector can be used as a pre-processing stage to obtain image points or image pixels that are on the desired curve in the image space. Due to imperfections in either the image data or the edge detector, however, there may be missing points or pixels on the desired curves as well as spatial deviations between the ideal line/circle/ellipse and the noisy edge points as they are obtained from the edge detector. For these reasons, it is often non-trivial to group the extracted edge features to an appropriate set of lines, circles or ellipses. The purpose of the Hough transform is to address this problem by making it possible to perform groupings of edge points into object candidates by performing an explicit voting procedure over a set of parameterized image objects.

The simplest case of Hough transform is the linear transform for detecting straight lines. In the image space, the straight line can be described as  $y=mx+b$  and can be graphically plotted for each pair of image points  $(x,y)$ . In the Hough transform, a main idea is to consider the characteristics of the straight line not as image points  $(x_1, y_1), (x_2, y_2), \dots$  but instead, in terms of its parameters, i.e., the slope parameter  $m$  and the intercept parameter  $b$ . Based on that fact, the straight line  $y=mx+b$  can be represented as a point  $(b,m)$  in the parameter space. However, one faces the problem that vertical lines give rise to unbounded values of the parameters  $m$  and  $b$ . For computational reasons, it is therefore better to use a different pair of parameters, denoted  $r$  and  $\theta$ , for the lines in the Hough transform. These are the Polar Coordinates (Figure 16).

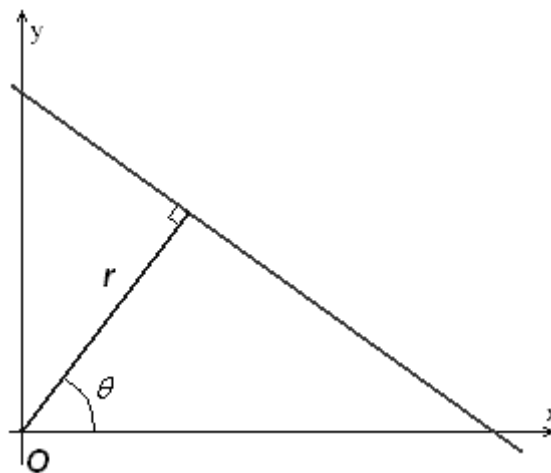


Figure 16 – Sample of polar coordinates.

The parameter  $r$  represents the distance between the line and the origin, while  $\theta$  is the angle of the vector from the origin to this closest point. Using this parameterization, the equation of the line can be written as:

$$y = \left(-\frac{\cos \theta}{\sin \theta}\right)x + \left(\frac{r}{\sin \theta}\right) \tag{4.53}$$

which can be rearranged to  $r = x \cos \theta + y \sin \theta$ .

It is therefore possible to associate with each line of the image a pair  $(r, \theta)$  which is unique if  $\theta \in [0, \pi)$  and  $r \in R$ , or if  $\theta \in [0, 2\pi)$  and  $r \geq 0$ . The  $(r, \theta)$  plane is sometimes referred to as Hough space for the set of straight lines in two dimensions. For an arbitrary point on the image plane with coordinates, e.g.,  $(x_0, y_0)$  the lines that go through it are the pairs  $(r, \theta)$  with:

$$r(\theta) = x_0 \cos \theta + y_0 \sin \theta \quad (4.54)$$

where  $r$  (the distance between the line and the origin) is determined by  $\theta$ . This corresponds to a sinusoidal curve in the  $(r, \theta)$  plane, which is unique to that point. If the curves corresponding to two points are superimposed, the location (in the Hough space) where they cross corresponds to a line (in the original image space) that passes through both points. More generally, a set of points that form a straight line will produce sinusoids which cross at the parameters for that line. Thus, the problem of detecting collinear points can be converted to the problem of finding concurrent curves.

### Implementation

The Hough transform algorithm uses an array, called an accumulator, to detect the existence of a line  $y=mx+b$ . The dimension of the accumulator is equal to the number of unknown parameters of the Hough transform problem. For example, the linear Hough transform problem has two unknown parameters: the pair  $(m, b)$  or the pair  $(r, \theta)$ . The two dimensions of the accumulator array would correspond to quantized values for  $(r, \theta)$ . For each pixel and its neighborhood, the Hough transform algorithm determines if there is enough evidence of an edge at that pixel. If so, it will calculate the parameters of that line, and then look for the accumulator's bin that the parameters fall into, and increase the value of that bin. By finding the bins with the highest values, typically by looking for local maxima in the accumulator space, the most likely lines can be extracted, and their (approximate) geometric definitions read off. The simplest way of finding these peaks is by applying some form of threshold, but different techniques may yield better results in different circumstances - determining which lines are found as well as how many. Since the lines returned do not contain any length information, it is often next necessary to find which parts of the image match up with which lines. Moreover, due to imperfection errors in the edge detection step, there will usually be errors in the accumulator space, which may make it non-trivial to find the appropriate peaks, and thus the appropriate lines. The result of the Hough transform is stored in a matrix that often is called an accumulator. One dimension of this matrix are the angles  $\theta$  and the other dimension are the distances, and each element has a value telling how many points/pixels are positioned on the line with parameters  $(r, \theta)$ . So the element with the highest value tells what line that is most represented in the input image.

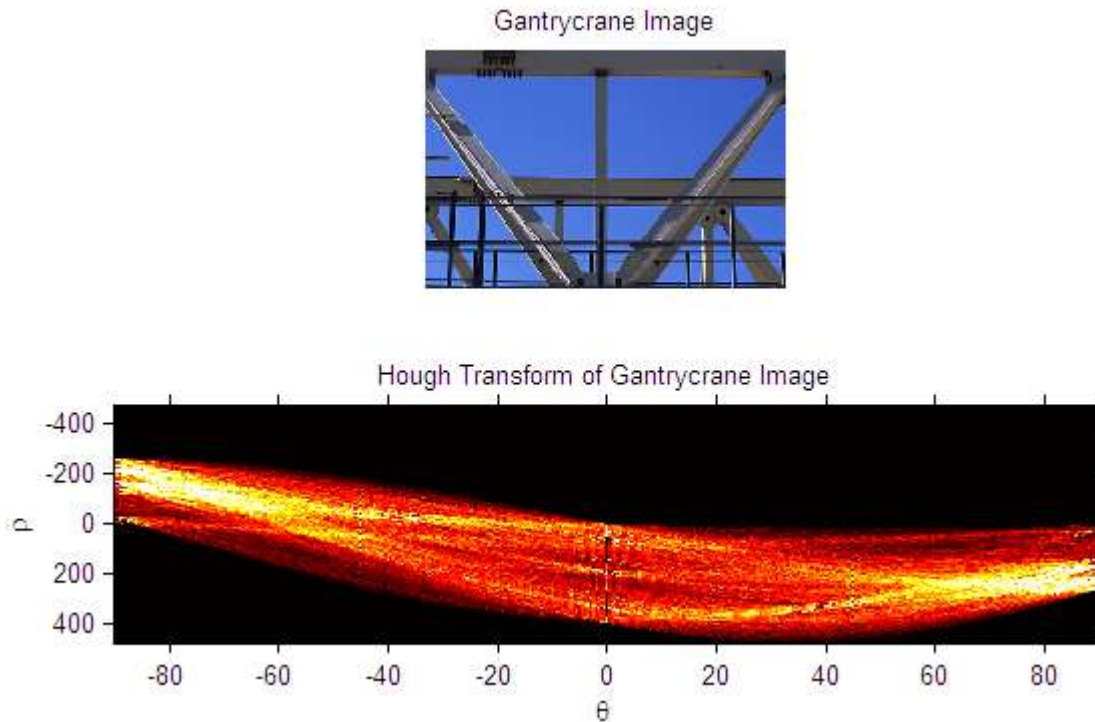


Figure 17 – Sample of image (up) and its Hough accumulator matrix (down).

### Circle Detection Process

The process of identifying possible circular (Figure 18) objects in Hough space is relatively simple:

- i) First we create our accumulator space which is made up of a cell for each pixel, initially each of these will be set to 0;
- ii) For each edge point in  $image(i,j)$  increment all cells which according to the equation of a circle ( $(i - a)^2 + (j - b)^2 = r^2$ ) could be the center of a circle, these cells are represented by the letter “a” in the equation;
- iii) For all possible value of a found in the previous step, find all possible values of b which satisfy the equation;
- iv) Search for the local maxima cells, these are any cells whose value is greater than every other cell in it's neighborhood. These cells are the one with the highest probability of being the location of the circle(s) we are trying to locate;

Note that in most problems we will know the radius of the circle we are trying to locate beforehand, however if this is not the case we can use a 3 dimensional accumulator space, this is much more computationally expensive. This method can also detect circles that are partially outside of the accumulator space if enough of its area is still present within it.

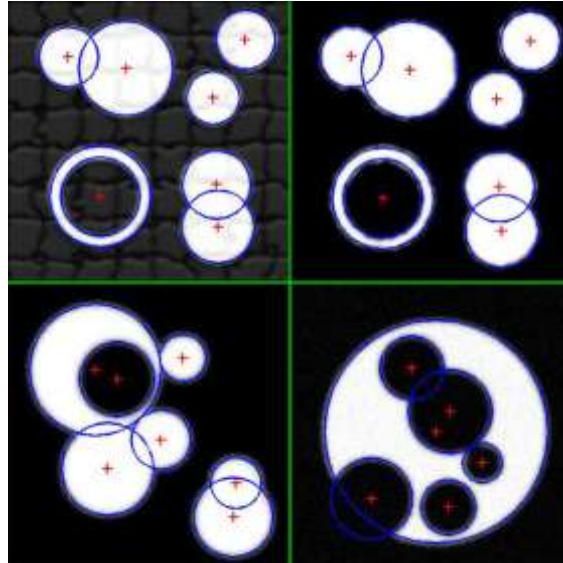


Figure 18 – Sample of four circular Hough transform application result.

#### 4.2.9. Image Synthesis

Image synthesis is the process of creating new images from some form of image description. The kinds of images that are typically synthesized include:

- Test Patterns, Scenes with simple two dimensional geometric shapes;
- Image Noise (Figure 19), Images containing random pixel values, usually generated from specific parameterized distributions;
- Computer Graphics, Scenes or images based on geometric shape descriptions. Often the models are three-dimensional, but may also be two-dimensional;

Synthetic images are often used to verify the correctness of operators by applying them to known images. They are also often used for teaching purposes, as the operator output on such images is generally “clean”, whereas noise and uncontrollable pixel distributions in real images make it harder to demonstrate unambiguous results. The images could be binary, gray level or color.

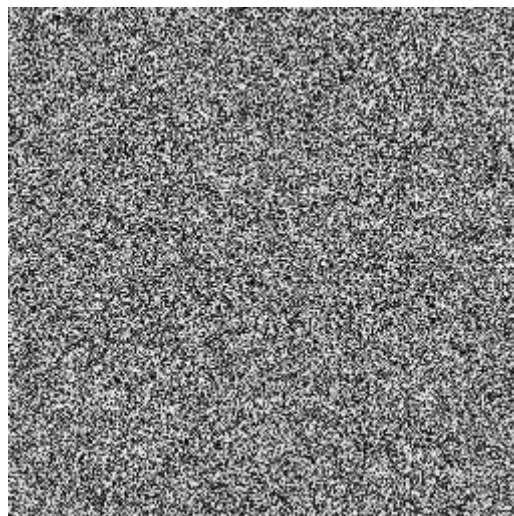


Figure 19 – Example of noise image used, e.g., to used to verify the correctness of operators.

### 4.3. Algorithms Evaluation Method

#### 4.3.1. PPV, NPV, Sensitivity and Specificity

To evaluate the goodness of the classification algorithms implemented will be used often, both for the tests in the greenhouse and for those in the field, the Positive Predictive Value, the Negative Predictive Value, the Sensitivity and the Specificity.

The Negative Predictive Value shows how many pixels/ROIs recognized as healthy were indeed corresponding to healthy tissue according to the phytopathological survey:

$$NPV = \frac{TN}{TN + FN} \quad (4.55)$$

The Positive Predictive Value shows how many pixels/ROIs recognized as diseased were indeed corresponding to diseased tissue according to the phytopathological survey:

$$PPV = \frac{TP}{TP + FP} \quad (4.56)$$

The Sensitivity shows the fraction of (amount of pixels/ROIs belonging to) the tissue manually classified as actual diseased which the algorithm correctly recognized as disease:

$$Sens = \frac{TP}{TP + FN} \quad (4.57)$$

In the end the Specificity shows the fraction of (amount of pixels/ROIs belonging to) the tissue manually classified as actual healthy which the algorithm correctly recognized as healthy:

$$Spec = \frac{TN}{FP + TN} \quad (4.58)$$

where TP (true positive) is the event that the test makes a positive prediction, and the subject has a positive result under the gold standard, FP (false positive) is the event that the test makes a positive prediction, and the subject has a negative result under the gold standard, TN (true negative) is the event that the test makes a negative prediction, and the subject has a negative result under the gold standard, and a FN (false negative) is the event that the test makes a negative prediction, and the subject has a positive result under the gold standard.

## 5. Materials and Methods

### 5.1. Data Measurements

- ✓ Spectral Reflectance Measurements (section 5.1.1): Overview on spectral measurements.
- ✓ Multispectral Imaging Measurements (section 5.1.2): Overview on multispectral imaging acquisition.

#### 5.1.1. Spectral Reflectance Measurements

The reflectance spectroscopy, or spectrophotometry, is the technique of optical investigation based on the measurement of the spectral reflectance factor of the surface to be measured, as a function of the wavelength of the incident radiation. The reflectance parameter is expressed as the intensity ratio between the reflected radiation and the incident radiation. The reflectance measurements typically affecting the region from ultraviolet to infrared; the trend of the intensity, in absolute value, as a function of wavelength, provides the reflectance spectrum, otherwise said spectral signature (Figure 21). The incident radiation is sent on point analysis using an optical fiber and the intensity of the reflected radiation is collected using the same fiber. To perform the measurement is used a standard reference white, with reflectance constant and equal to 99% (Figure 20, absolute white). In different days of acquisition the measured absolute white spectra changes because the acquisition of LED light radiation, used as incident radiation, suffers of small changes due, for example, to the manual measurement procedure, LED behavior (like LED temperature) and spectrophotometer behavior. Just to overcome this problem, the values of intensity of the reflected radiation, in percentage terms, are obtained in reference to the standard of absolute white (Figure 22) relative to the day of measurement.

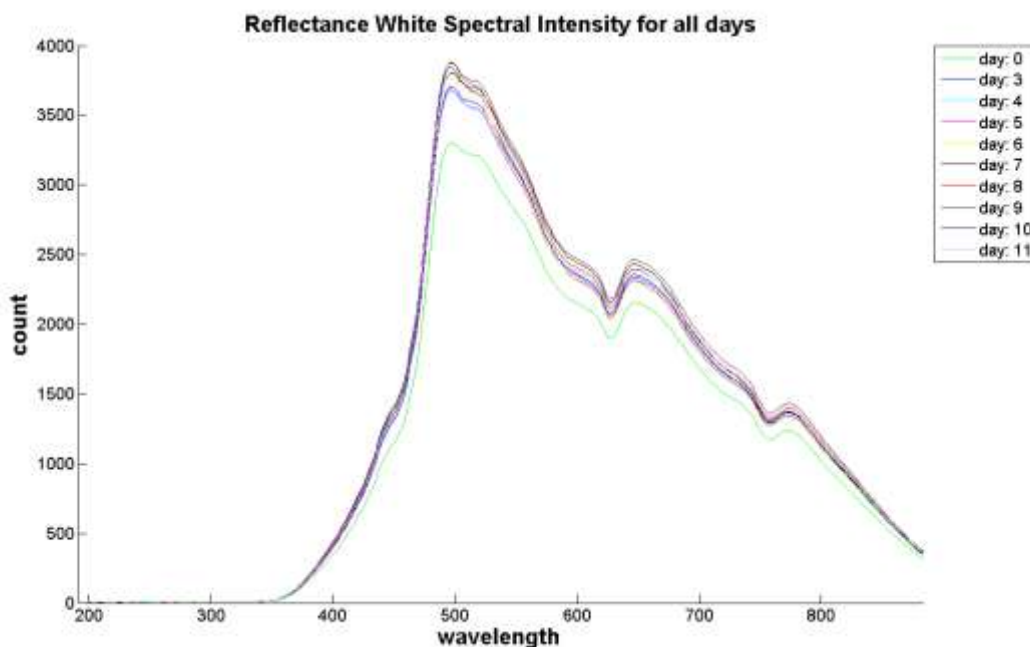


Figure 20 – Absolute white spectra used during the measurements (11 days). In different days of acquisition the measured absolute white spectra change because the acquisition of LED light radiation, used as incident radiation, suffers of small changes due, for example, to the manual measurement procedure, LED behavior (like LED temperature) and spectrophotometer behavior.



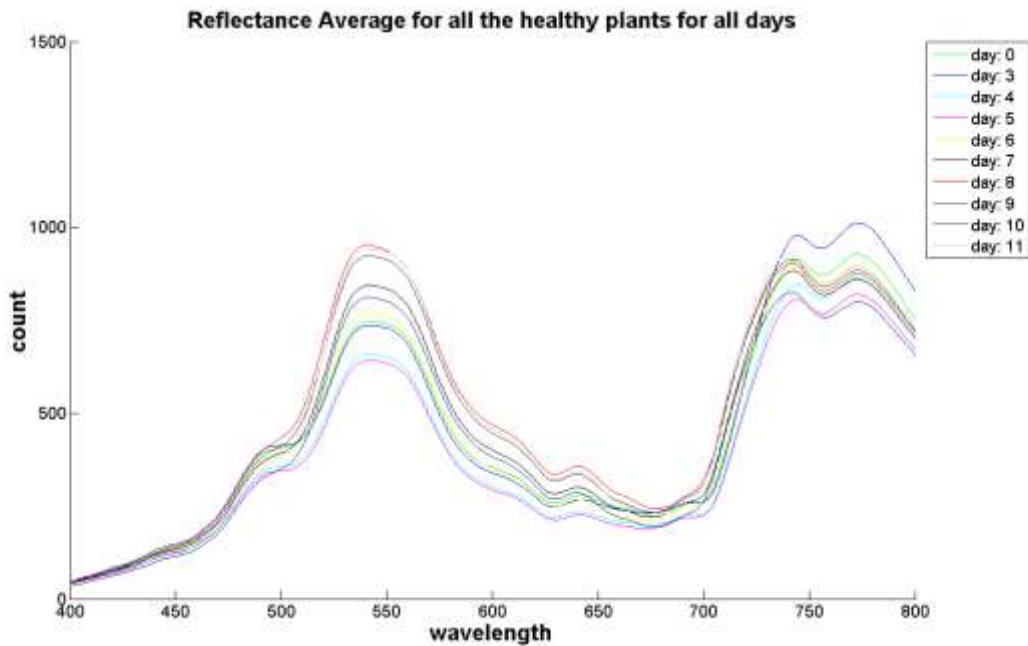


Figure 21 – Absolute reflectance mean values of healthy leaves for each day of acquisition (raw data). The reflectance measurements typically affecting the region from ultraviolet to infrared; the trend of the intensity, in absolute value, as a function of wavelength, provides the reflectance spectrum, otherwise said spectral signature.

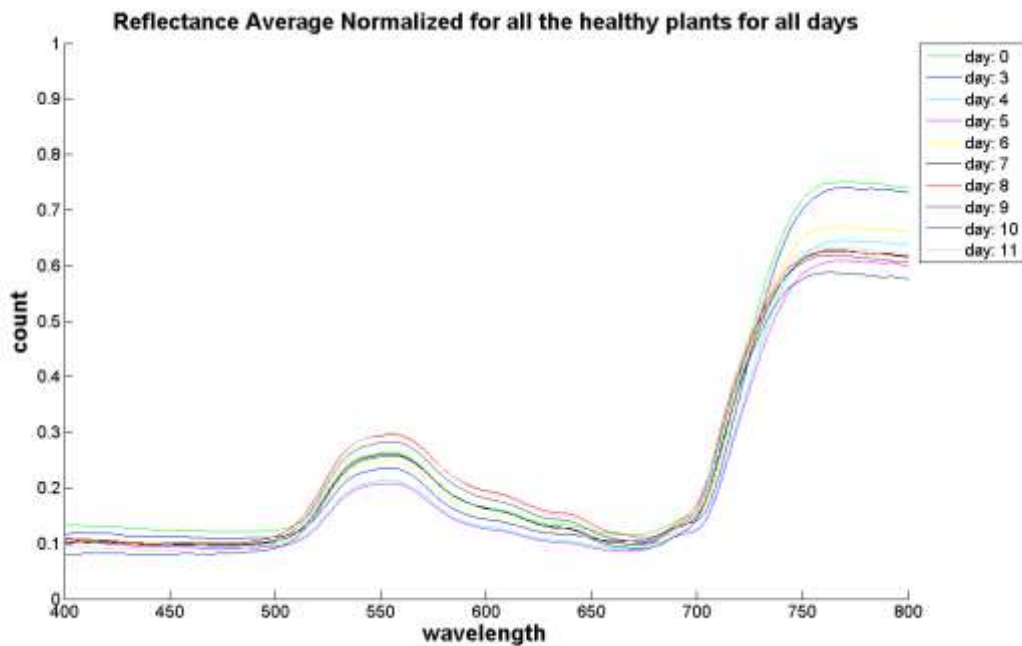


Figure 22 – Relative reflectance mean values of healthy leaves for each day of acquisition (normalized data). The relative reflected radiations, in percentage terms, are obtained in reference to the absolute white relative to the day of measurement.

The objective of the spectral measurements is the identification of significant indices (relations between different wavelengths), that can be used as a yardstick to measure the state of health of the plant, as well as the identification of the presence of local and targeted pathologies on the leaf surface. For the measurement of the spectra was used the spectrophotometer Avantes SD2000 whose measurement range is from 191.20 nm to 883.42 nm and whose resolution is about 0.35 nm.

### 5.1.2. Multispectral Imaging Measurements

The images are acquired using the multispectral camera DuncanTech MS4100 in configuration CIR (Color-infrared imaging) at a resolution of 1912x1076 pixels. DuncanTech's MS4100 camera is a digital, progressive scan, area camera for multispectral. The camera is based on a color separating prism and three imaging channels that allow simultaneous image acquisition in 3 spectral bands through a common aperture. Image sensors are charge coupled device (CCD) array sensors with spectral sensitivity from 400-1000 nm. The resulting images are co-registered providing excellent image quality and color fidelity.

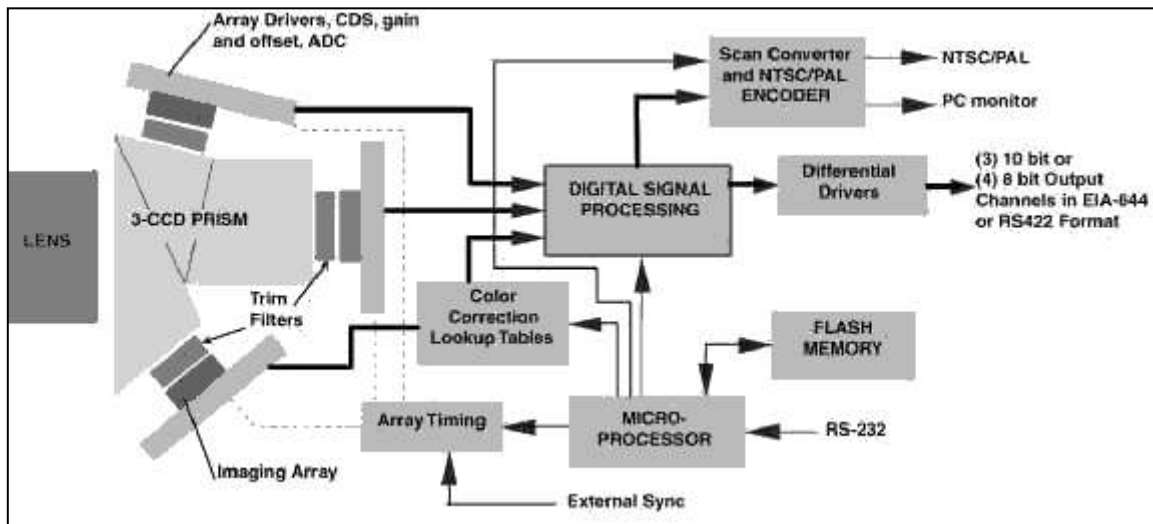


Figure 23 - Functional diagram DuncanTech MS4100.

A functional diagram of DuncanTech's MS4100 camera is shown in Figure 23. These cameras use a color separating prism to isolate the spectral image to be acquired by each channel. Broadband light from the image target enters the camera through the lens. The prism optic divides the light based on wavelength such that a different spectral band exits the prism at each of the three exit faces. The range of wavelengths included in each band is a function of the coatings on the faces of the prism. The output signal of each array is digitized to a 8 or 10 bit digital value. The acquired spectral bands are red, green and infrared, as shown in Figure 24 and according to the configuration shown in Figure 25.

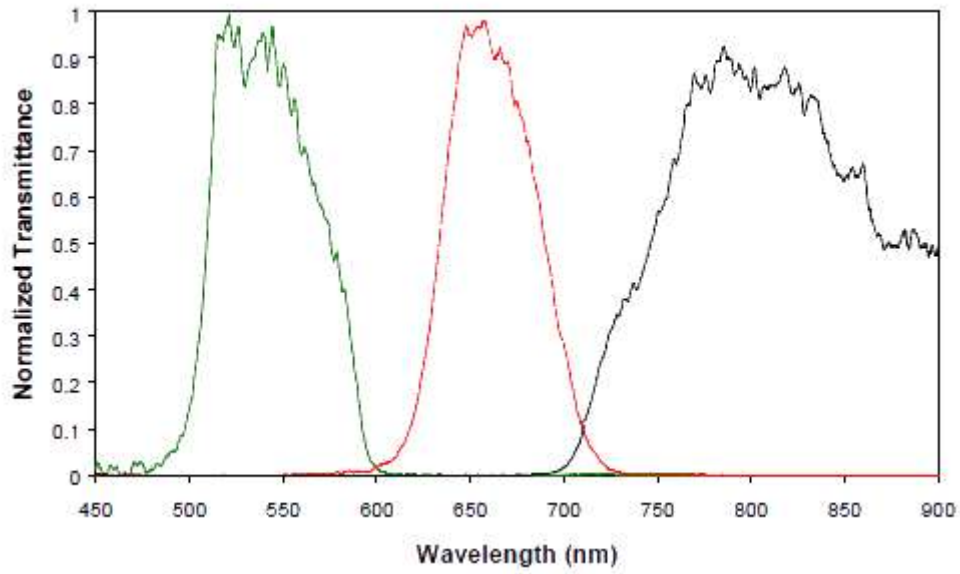


Figure 24 - Spectral transmittance of the multispectral camera DuncanTech MS4100 configuration 3-CCD CIR.

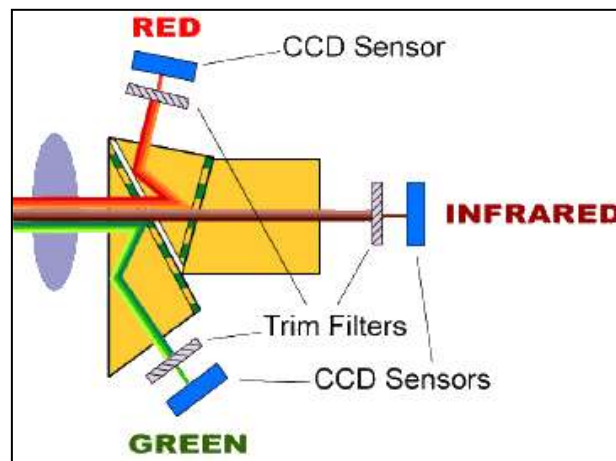


Figure 25 - Camera DuncanTech MS4100 configurazione 3-CCD CIR.

The gain and the white balance of the multispectral camera can be set for each channel.

## 5.2. Materials

### 5.2.1. Laboratory tests

- ✓ Plant material and diseases assessment (section 5.2.1.1): description of used plants, fungal inoculation and diseased assessment.
- ✓ Laboratory measurements setup (section 5.2.1.2): description of the spectra acquisition setup.
- ✓ Measurement procedure (section 5.2.1.3): description of the spectra acquisition procedure.
- ✓ Calibration and validation sets (section 5.2.1.4): description of how the spectra data will be used and what spectral indicators will be defined.

#### 5.2.1.1. *Plant material and diseases assessment*

##### *Plant material*

In order to ensure continuity in the experimental stage, it was decided to proceed with the acquisition using as a test case the plant of cucumber instead of the grapevine, which will be studied from the greenhouse and field tests. The choice of the cucumber is determined by several factors:

- Leaves shape is similar to that of the leaves of grapevine even if of reduced size.
- Behavior of the parasite of first interest (powdery mildew) is similar to what happens for the grapevine.
- Ease in getting samples to be used during testing for all months of the year.

They were maintained in greenhouse under controlled conditions at 25/20 °C day/night temperature, with 75% relative humidity and a 16-h photoperiod (40  $\mu\text{mol quanta m}^{-2} \text{s}^{-1}$ ). Plants were regularly watered and fertilized (Na/P/K/Mg, 15:10:15:2). No pesticide treatment was carried out and the healthy status of vegetation was ascertained by visual inspections.

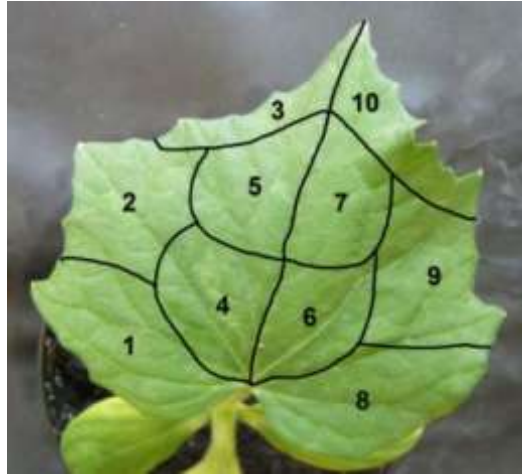
##### *Fungal isolates and inoculation*

The isolates of *Erysiphe necator* used in this study originated from infected fresh leaf material collected in another experimental. Cucumbers were inoculated by brushing conidia from freshly sporulating colonies onto the adaxial (upper) epidermis of healthy leaves. To determine the inoculum density (conidia per  $\text{cm}^2$ ) in each experiment, glass microscope slides were placed at arbitrary locations inside the vegetation during inoculation, and the number of conidia in 10 views was counted at X50 magnification by a light microscope (Olympus BX 50, Tokyo, Japan). Inoculum density ranged approximately from 60 to 500 (mean 248) conidia per  $\text{cm}^2$ , depending on the experiment.

##### *Diseases assessment*

During the measurement the plants under observation are 12. Each plant has two leaves. 4 plants (8 leaves) are healthy (not subjected to inoculation) and 8 plants (16 leaves) are inoculated with powdery mildew. To none of plants is applied any kind of pesticide treatment. The leaf is (ideally) divided into 10 areas so as to be able to acquire a set of samples the most heterogeneous possible. For 11 days, for each leaves is acquired a measure within each of the 10 areas in which it is divided (Figure 26), for a total of 80 measures of healthy plants and 160 measurements on inoculated plants.

In each measurement day, the leaf subdivision used allows to acquire samples along the entire leaf surface so as to have, at all times, the general view of the health status of the leaf. The procedure is repeated for a total of 11 days (beyond which no longer makes sense to follow the development of the disease).

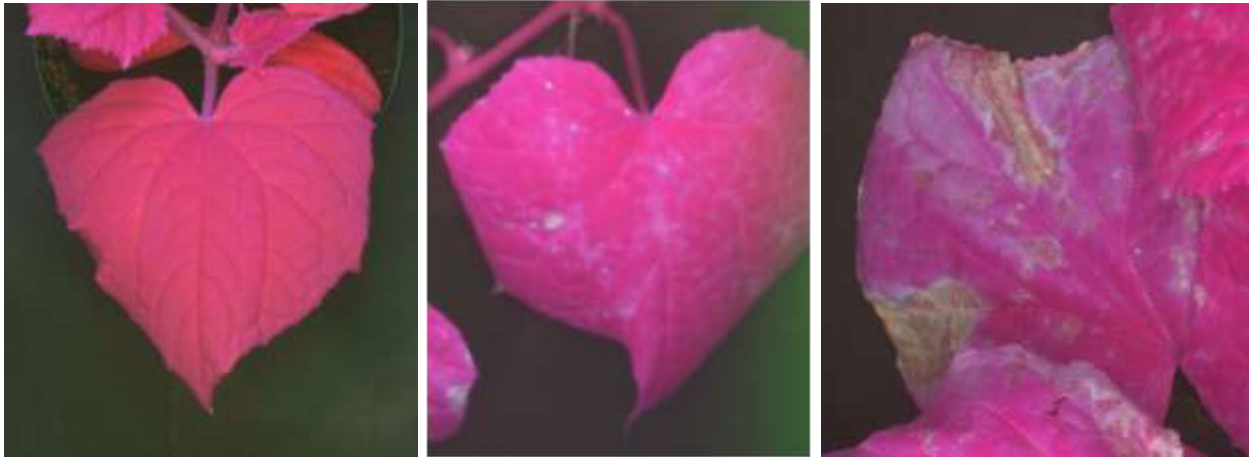


**Figure 26 - Experimental setup: The 10 areas in which the leaf is subdivided. The leaf is divided into 10 areas so as to be able to acquire a set of samples the most heterogeneous possible. In each measurement day, the subdivision used allows to acquire samples along the entire leaf surface so as to have, at all times, the general view of the health status of the leaf.**

The spectra are divided into different classes of interest. We decided to use three classes of interest determined by the typical times of development of the disease (Figure 27):

- spectra belonging to healthy leaves (healthy)
- spectra belonging to inoculated leaves from 4 to 9 days (intermediate)
- spectra belonging to diseased leaves (ill)

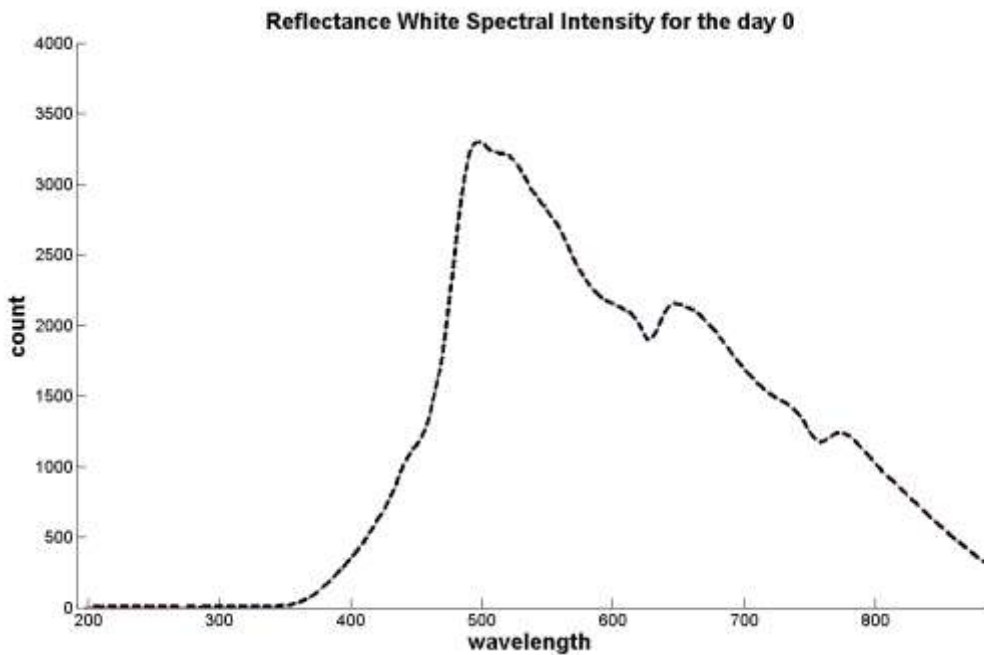
We believed that these three classes representing well the three states of health in which a leaf can be: healthy, "affected" and sick. The healthy leaves are those leaves which have not been inoculated, and that, in the course of days regarding the measurement, have not been in contact with the disease. The leaves inoculated from 4 to 9 days are the leaves in which the disease is being developed (state from healthy to ill); Plants whose leaves belong to this class have different symptoms hardly visible in the first days following inoculation. Starting from the tenth day following the inoculation all plants present obvious symptoms of the disease.



**Figure 27 – Sample of an healthy leaf (left), a leaf with intermediate and advanced symptoms of diseased (center) and a leaf with evident necrotic areas (right).**

**5.2.1.2. Laboratory measurements setup**

The light is routed through an optical fiber to the end of which is fixed a small darkroom. At the end of the darkroom is fixed a small sponge which prevents the leaf damaged as a result of the repeated measurements (Figure 29). The spot of the leaf that is to be measured is subjected to light whose spectrum is shown in Figure 28.



**Figure 28 - The spectrum of the light channeled through the optical fiber.**

The spectrum of the light reflected from the leaf is routed into the optical fiber and measured by the spectrophotometer which it is connected. In the early days of measurement, when there is no obvious sign of the disease, the spot measurement is taken at any point within each of the 10 areas in which each leaf is subdivided. From the moment when the disease begins to manifest itself, the measure of the spot takes place in correspondence of the disease. Not all the leaves reach the end of

the measurement. On leaves subject to a more rapid course of the disease, the measurements will stop when the leaf becomes necrotic.



**Figure 29 - Experimental setup: measures of reflectance. You can the analysis station consists of a support of black color (black box) provided with a soft surface on which to lay carefully the leaf. The measurement is achieved by the use of an optical fiber that channels the light emitted by the LED. At the end of the optical fiber which emits the light radiation (and channels the radiation reflected by the leaf surface) is present a white box used as a darkroom.**

### **5.2.1.3. *Measurement procedure***

For the spectral reflectance measurements the technique used is as follows and is illustrated in Figure 29:

1. The plant to be analyzed is taken and placed on the analysis station.
2. The single leaf is placed on the measuring panel.
3. For each leaf are performed 10 measurements in 10 different areas located over the entire leaf surface (Figure 26).
4. The acquisition time of the measurement is a few milliseconds.

Before starting each measurement on the leaf, is measured the spectrum of light reflected from the reference white (99% reflectance Spectralon). This measurement will be used to normalize the spectra acquired on different days so that they are comparable to each other (see section 5.1.1 for more details). Health status (including the level of infection) of each leaf measured is evaluated and recorded by a pathologist, so that the level of advancement of the disease is certified. The acquired data are stored in a text file and then analyzed.

#### 5.2.1.4. *Calibration and validation sets*

The spectra (raw data) are normalized with the measurement of the white reference for the day in which they were measured. The normalization makes it possible the comparison of spectra acquired on different days and with different lighting conditions (in the darkroom the only intensity of the light transmitted through the optical fiber). The normalized spectra are used in order to identify those wavelengths that are most representative of the state of health of the plant or, conversely, more indicative in defining the progression of the disease. To do this have been taken into account:

- some specific wavelengths, to take into account the wavelength (average values centered in a small neighborhood of the selected value) that we consider more indicative of the colorations of our interest (violet, blue, green, yellow, red and near infrared):
  - 400 nm (violet)
  - 450 nm and 470 nm (blue)
  - 500 nm and 550 nm (green)
  - 580 nm (yellow)
  - 650 nm and 700 nm (red)
  - 750 nm and 800 nm (near infrared)
- some angles between some wavelengths, to take into account some possible relative changes within the spectrum. We chose a step of 50 nm from 400 nm up to 800 nm, lower and upper limit between which we have the usable signal. The angle is defined by the straight line passing through the two wavelengths considered and the x-axis on which the wavelengths are reported (Figure 30):
  - between 400 nm and 450 nm
  - between 450 nm and 500 nm
  - between 500 nm and 550 nm
  - between 550 nm and 600 nm
  - between 600 nm and 650 nm
  - between 650 nm and 700 nm
  - between 700 nm and 750 nm
  - between 750 nm and 800 nm



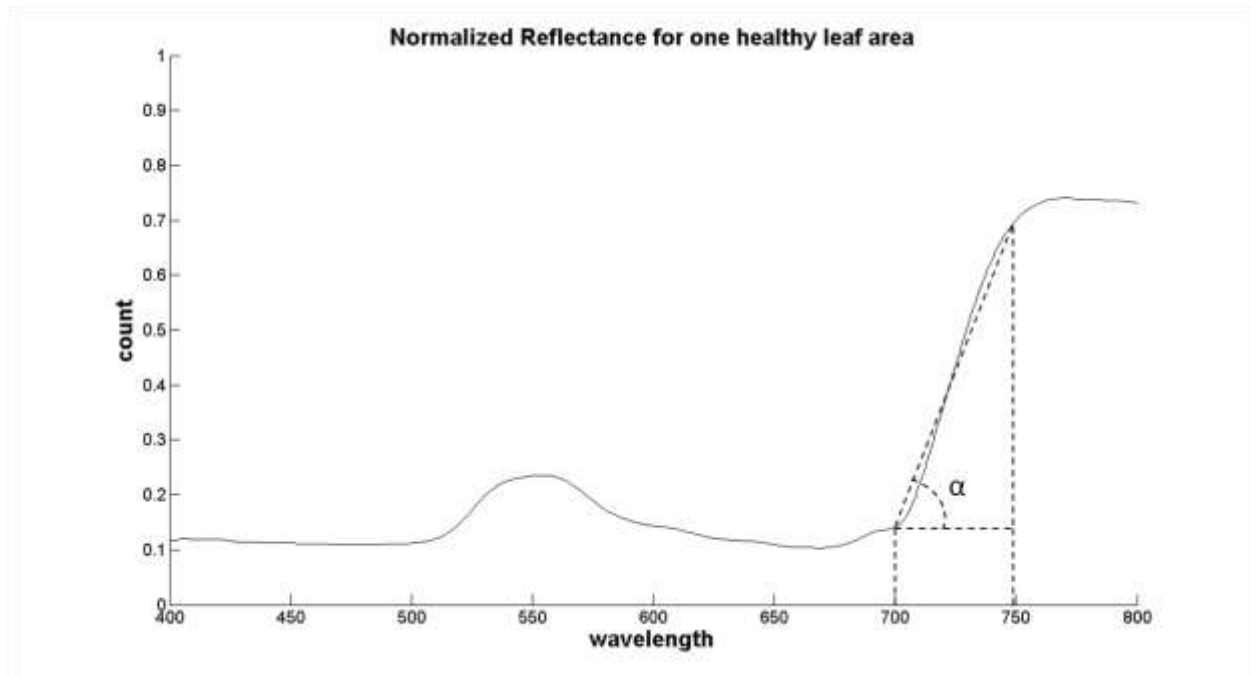


Figure 30 – Sample of relative reflectance values for one healthy leaf area (normalized data) and one defined angle,  $\alpha$ , between 700 nm and 750 nm. The angle is defined by the straight line passing through the two wavelengths considered and the x-axis on which the wavelengths are reported.

- the ratio between some wavelengths, taken from the literature (REIP and NDVI) or deduced from a first spectral analysis of the trend of disease (RIR):
  - the REIP index (Red edge vegetation index)

$$reip\_index = 700 + 40 * \frac{(\frac{w670 + w780}{2} - w700)}{w740 - w700}$$

- the RIR index (red on infrared)

$$rir\_index = \frac{w650}{w750}$$

- a possible formulation of NDVI index (Normalized Difference Vegetation Index)

$$NDVI\_index = \frac{w750 - w650}{w750 + w650}$$

Red edge refers to the region of rapid change in reflectance of vegetation in the near infrared range of electromagnetic spectrum. Chlorophyll contained in vegetation absorbs most of the light in the visible part of the spectrum but becomes almost transparent at wavelengths greater than 700 nm. This way, vegetation cellular structure takes its major part in the reflectance because each cell acts like an elementary corner reflector. This effects in that rapid change that can be from 5% to 50% reflectance between 680 nm to 730 nm. The phenomenon accounts for the brightness of foliage in infrared photography and is extensively utilized in the form of so-called vegetation indices (e.g. NDVI). It is used in remote sensing to monitor plant activity, and it has been suggested that it could be useful to detect light-harvesting organisms on distant planets. Defined indicators were analyzed

by different classification techniques to evaluate the discriminatory power (see section 5.3.1 for more details).

### **5.2.2. Greenhouse tests**

- ✓ Plant material and diseases assessment (section 5.2.2.1): description of used plants, fungal inoculation and diseased assessment.
- ✓ Greenhouse measurements setup (section 5.2.2.2): description of the image acquisition setup.
- ✓ Measurement procedure (section 5.2.2.3): description of the image acquisition procedure.
- ✓ Calibration and validation sets (section 5.2.2.4): description of how the acquired image will be used.

#### **5.2.2.1. *Plant material and diseases assessment***

##### ***Plant material***

Plants of *Vitis vinifera* L. cv Cabernet Sauvignon were propagated from wood, one-bud cuttings and grown in 25 cm diameter pots filled with a mixture of sand and loamy soil (3:1, v/v). They were maintained in greenhouse under controlled conditions at 25/20 °C day/night temperature, with 75% relative humidity and a 16-h photoperiod (40  $\mu\text{mol quanta m}^{-2} \text{ s}^{-1}$ ). Plants were regularly watered and fertilized (Na/P/K/Mg, 15:10:15:2). No pesticide treatment was carried out. Three months old plants with 12-15 leaves were used for the experiments, and the healthy status of vegetation was ascertained by visual inspections.

##### ***Fungal isolates and inoculation***

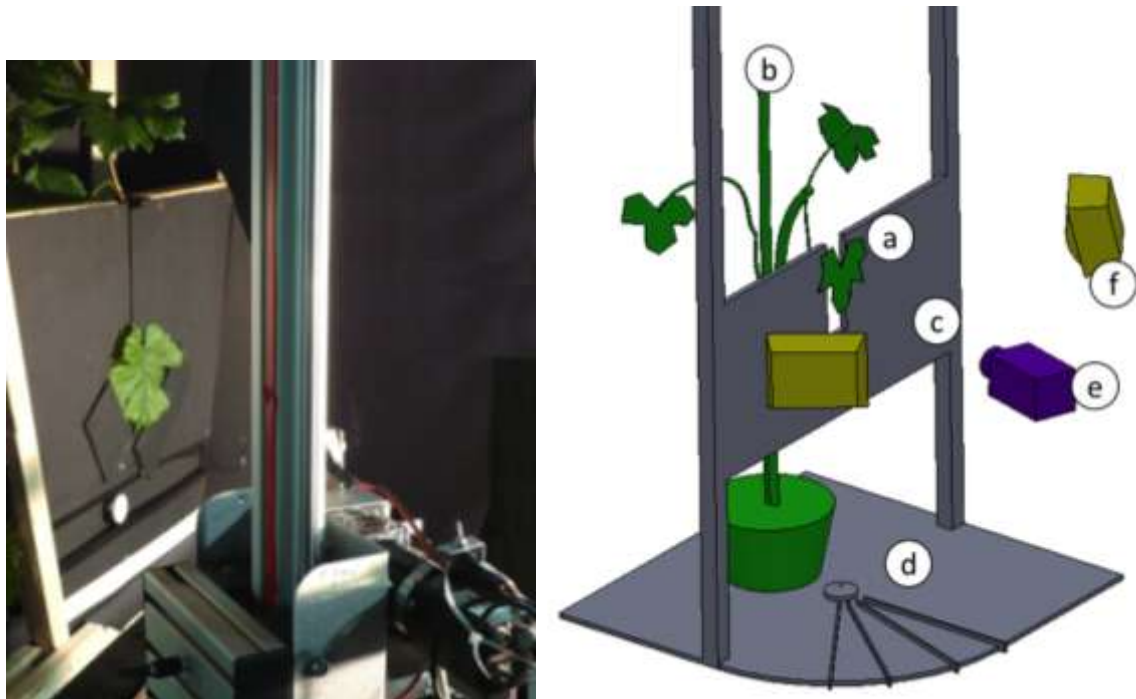
The isolates of *Erysiphe necator* used in this study originated from infected fresh leaf material (*V. vinifera* cv Gropello) collected in an experimental vineyard. Grapevines were inoculated by brushing conidia from freshly sporulating colonies onto the adaxial (upper) epidermis of healthy leaves. To determine the inoculum density (conidia per  $\text{cm}^2$ ) in each experiment, glass microscope slides were placed at arbitrary locations inside the vegetation during inoculation, and the number of conidia in 10 views was counted at X50 magnification by a light microscope (Olympus BX 50, Tokyo, Japan). Inoculum density ranged approximately from 60 to 500 (mean 248) conidia per  $\text{cm}^2$ , depending on the experiment.

##### ***Diseases assessment***

During the measurement period, inoculated leaves (N = 25) of different age were monitored by visual inspections. In addition, healthy leaves (controls), as well as organs showing a specific chlorotic symptoms or other discoloring lesions related to nutritional or physiological disorders (false positive controls, N = 10) were selected and matched with infected leaves for age and dimension. Four, eight, twelve and sixteen day after inoculation, disease intensity was rated by evaluating the growth of fungal colonies on the adaxial leaf surface. In particular, all selected leaves were visually scored for disease using a 0 to 4 arbitrary scale based on the percentage of the leaf surface with visible fungal structures, as follows: IL = 0 (infection level 0), corresponding to small (1-2 mm in diameter) chlorotic spots likely expected to develop fungal mycelium; IL = 1, small areas (1-2 mm) with hyphae at the initial stage of development (rare and hard to see); IL = 2, areas with clearly visible mycelial structures (both hyphae and conidiophores); IL = 3, extended colonies (10-20 mm) with grey-whitish powdery appearance; IL = 4, brown-dark necrotic lesions.

### 5.2.2.2. *Greenhouse measurement setup*

In vivo images of single grapevine leaves under varying viewing angle were carried out with a rotating table holder contained in a (0.8m x 0.8m x 1.5m) box acting as darkroom, and a standing frame equipped with a multi-spectral camera and a set of diffuse illumination sources (Figure 31).



**Figure 31 - Measurement setup (left) for in-vivo leaf imaging at different view angles, and a simplified sketch of it (right): a) leaf; b) sample plant; c) adjustable background; d) rotating plane; e) multi-spectral camera; f) diffuse illumination sources.**

The multi-spectral camera was equipped with a lens with a focal distance of 28 mm. Each target leaf (Figure 31-a) of a measured plant (Figure 31-b) was vertically hold by a position adjustable background uniform black frame (Figure 31-c). This allowed an easier automatic identification of the leaf in the images through background removal. The plane (Figure 31-d) holding the sample could be rotated around its vertical axis by a controlled angle, allowing to acquire leaf images from different view angles by the multi-spectral camera (Figure 31-e) which was instead maintained in stationary position together with the sources of diffuse illumination (Figure 31-f). The vertical position of the camera on the stand-frame was adjusted for each measured leaf, so that the distance and relative orientation between the leaf and the camera was kept constant for all the measurements. Stable, diffuse and reproducible sample illumination was obtained by artificial light with a set of halogen lamps and white panel reflectors which projected indirect diffuse light on the imaged area. Unwanted ambient light was removed by surrounding the measurement setup with a dark-room box having an open front side. The diffuse illumination sources and the camera were placed in front of the dark-room open side at a distance of 0.5 m from the leaf sample, resulting in a spatial resolution of about 0.125 mm/pixel. Camera gain and integration time were manually set to balance the gray mean level of pixels belonging to a 50% reflectance spectralon target included in the image, while avoiding any saturation in a 99% reflectance reference also included in the field of view. Gain and integration time were kept constant for all the measurements. It is to be noted that the errors of optical distortion of the used camera are negligible compared to the natural inhomogeneity of the geometry of the leaf surface; this makes unnecessary the need for optical calibration.

### 5.2.2.3. *Measurement procedure*

At each measurement day, prior to image acquisition, all labeled leaves were visually inspected by a plant pathologist and classified according to the disease symptoms presence and intensity. The position, dimensions and intensity of fungal lesions were accurately recorded on log-notes. All significant features other than powdery mildew symptoms, such as local chlorotic areas or other pigmentations related to nutritional or physiological disorders, were also recorded as possible source of false positives. Images of the selected set of leaves were acquired in-vivo with the measurement setup from five different viewing angles: 0°, 20°, 40°, 60° and 75° (Figure 32). In literature there are no works on which to base the choice of the angle to be used to demonstrate the method under consideration. It was decided to start from the frontal position, 0°, by taking a step of about 20°, up to 75° of maximum angle, beyond which the visible portion of the leaf does not allow us to proceed with the identification of pathology. A total of 175 images were acquired, corresponding to 5 viewing angles per 35 leaf samples, covering a range of leaf ages (young to mature), powdery mildew intensity (healthy to necrotic tissues) and possible interfering pigmentations.

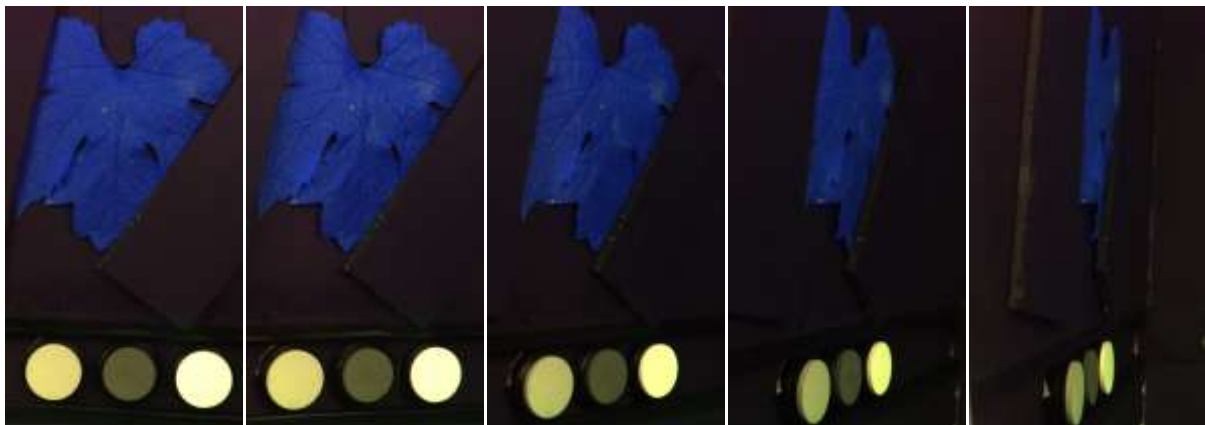


Figure 32 – Example of image of the selected set of leaves acquired in-vivo with the measurement setup from five different viewing angles (from left to right): 0°, 20°, 40°, 60° and 75°.

### 5.2.2.4. *Calibration and validation sets*

Within the calibration images, a subset of areas among all the areas identified by the plant pathologist, was randomly extracted for each leaf. A sub-region containing homogeneous tissue of 10x10 pixels was manually selected inside each identified area (ROIs, regions of interest). The number of ROIs for each class was chosen to obtain a balanced distribution among healthy tissue (20 ROIs), diseased tissue at different powdery mildew stages (20 ROIs), necrotic or pigmented tissue (10 ROIs). For a specific leaf, each chosen ROI was repeatedly manually identified at different viewing angle so that the same sector of leaf tissue was selected in the images acquired at the five different angles. The identified ROIs were subsequently used for defining the disease detection algorithm and to train the classification parameters. Just after the acquisition of images, some leaves were carefully surveyed by a plant pathologist. According to the evidence of this survey, we manually painted the infected areas of the image of each leaf using a color code for the different classes of intensity of symptoms (Figure 33). These images were subsequently used to test the performance of the detection algorithm (see section 5.4.2 for more details).

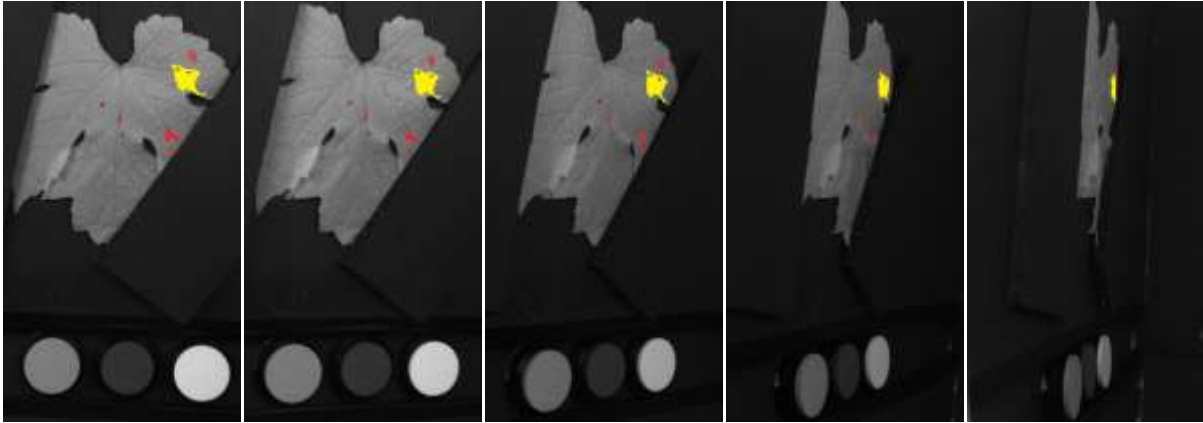


Figure 33 - Example of validation images. The colored areas were manually drawn during a coincident in-vivo phytopathological survey of the leaf: unpainted pixels correspond to healthy tissue; a region exhibiting advanced symptoms of powdery mildew is painted in yellow, medium intensity infection spots are painted in red and small areas of necrotic tissue are painted in brown.

### 5.2.3. Field tests

- ✓ Plant material and diseases assessment (section 5.2.3.1): description of used plants, fungal inoculation and diseased assessment.
- ✓ Field measurements setup (section 5.2.3.2): description of the image acquisition setup.
- ✓ Measurement procedure (section 5.2.3.3): description of the image acquisition procedure.
- ✓ Calibration and validation sets (section 5.2.3.4): description of how the acquired image will be used.

#### 5.2.3.1. *Plant material and diseases assessment*

##### *Plant material*

A set of field measurements were carried out between May and July 2011 in the area of Garda Lake (45.5°N, 10.5°E), Italy. Six experimental plots, each of 15 plants of grapevine (*V. vinifera*, cv Groppello) were obtained in two commercial vineyards: plots group A in Raffa di Puegnago, and group B in Tuscolano Maderno. Other experiments were performed in two different industrial cultivars, Tuscolano Maderno (BS, Italy) and Raffa (BS, Italy). A parcel of the cultivar devoted to experiments was left without treatments in order to allow the onset and development of diseases of interest. In this parcel various leaves at different ages were labeled and classified as target.

##### *Fungal isolates and inoculation*

All plots were managed according the farming practice adopted in the hosting vineyards, except for pesticides treatments. Downy mildew, powdery mildew and Botritis were artificially inoculated in plots A and B, respectively, in order to produce traceable, localized and possibly subsequent sprouts of fungal diseases during the measurements period. Downy mildew symptoms at different stages were obtained on leaves in two out of three plots of group A, while powdery mildew symptoms at different stages were obtained on leaves in one out of three plots of group B.

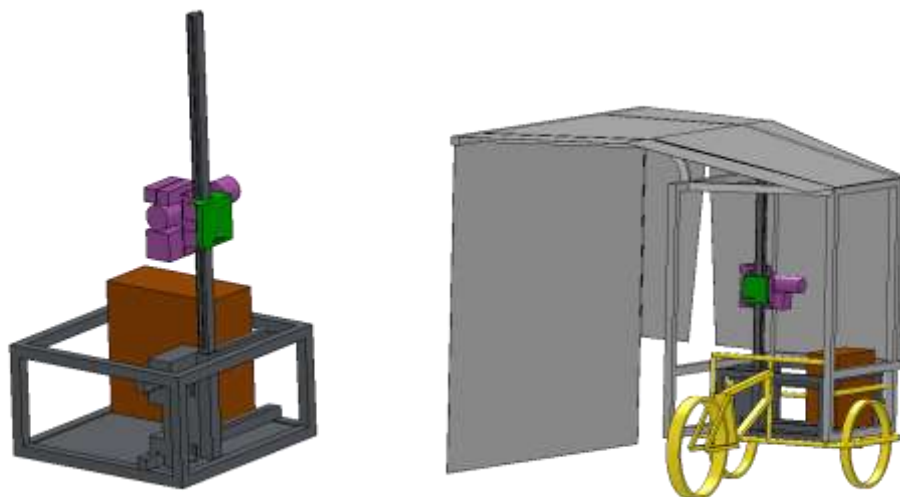
##### *Diseases assessment*

Experimental plots were regularly monitored by plant pathologists through visual scouting and occurrence of disease and position of symptoms were recorded. Symptoms intensity was rated by evaluating the growth stage of fungal colonies on leaf surface, and disease severity was assessed as the average percentage of the leaf area in the focus showing symptoms. For the limited aims of this

study, an arbitrary and qualitative scale was used by designating as: initial symptoms, those corresponding to small (5 mm in diameter) and sparse (less than 1% of leaf surface affected on five leaves around the focus) fungal structures; medium intensity symptoms, those exhibiting visible patch-structures (10-15 mm in diameter) with a partly diffused focus (less than 10% of leaf surface affected); advanced symptoms, with extended colonies (20-50 mm) of mature or partly sporulating fungal structures affecting 10% or more of leaf tissue in the neighbor of disease focus.

#### 5.2.3.2. *Field measurements setup*

Field acquisition were done by moving an instrumented trolley along the grapevine rows within the experimental plots (Figure 34). The trolley held a tunnel structure aimed to cover the imaged area of the canopy with two goals: to provide background regularization and avoid multiple rows viewing by means of a black, low-reflective shield; to avoid specular reflection spots on foliage and to homogeneously illuminate the measured area with a white, light-diffusing shields used as roof and lateral walls. The instrumented trolley hosted the electronic apparatus used for the measurements (Figure 35). In particular the R-G-NIR multispectral camera. The R-G-NIR multispectral camera acquires 1912x1076 pixels, 8 bit images in three distinct spectral channels: green (540nm), red (660nm) and near infrared (800nm). The canopy area imaged was about 1.25x0.7 m, resulting in a spatial average resolution of about 0.6 mm/pixel. The acquisition system consists of a frame containing all needed electronic components mounted on a three-wheel vehicle. Coverage able to cross ahead the interested part of the wine row is fixed to the vehicle. Acquisitions were taken by a group of different types of cameras (R-G-NIR and R-G-B) placed on a support fixed on sliders which permit to adjust height and distance of the target.



**Figure 34 - The instrumented trolley hosted the electronic apparatus used for the measurements (left). The trolley held a tunnel structure aimed to cover the imaged area of the canopy (right).**





**Figure 35 - Instrumented field trolley hosting the acquisition system. Canopy imaging is done inside a tunnel structure providing background regularization and diffuse illumination of the measured area.**

The vehicle was moved aside the wine row at fixed distance, in order to maintain the row in the middle of the coverage structure and perpendicular to the cameras. Depending on the particular needs of the experiments, the sliders were adjusted on-demand. Coverage helps to stabilize the illumination, reducing specular reflections and removing background interference from the parallel row ahead. It consists of two distinct parts:

- A black, opaque, low reflective shield placed ahead the interested part of the wine row and covering the whole visual space of the cameras;
- A white, translucent coverage covering the top and the other free sides of the scene.

The cameras used are the follows:

- A multispectral R-G-NIR MS4100 DuncanTech camera equipped with Nikon AF Nikkor 24mm lens; camera have three distinct CCD with resolution of 1072x1912 pixel on 8 bit for each channel.
- An R-G-B Basler acA1300 camera with a CCD resolution of 1278x958 (Bayer matrix)

### ***Shield testing***

For setting up the rover we have made some tests in order to evaluate and thus justify the use of the black shield and of the white coverage. Tests served to verify the real need to use a black shield, to avoid taking more rows of vines and make immediate the extraction of subjects acquired, and a white coverage, to avoid direct sunlight on the rows of vines to the advantage of a greater uniformity of the light intensity diffused. The three different configurations analyzed:

- Rover without shield and diffuser (Figure 36-left)
- Rover with rear shield (Figure 36-center)
- Rover with rear shield and diffuser (Figure 36-right)

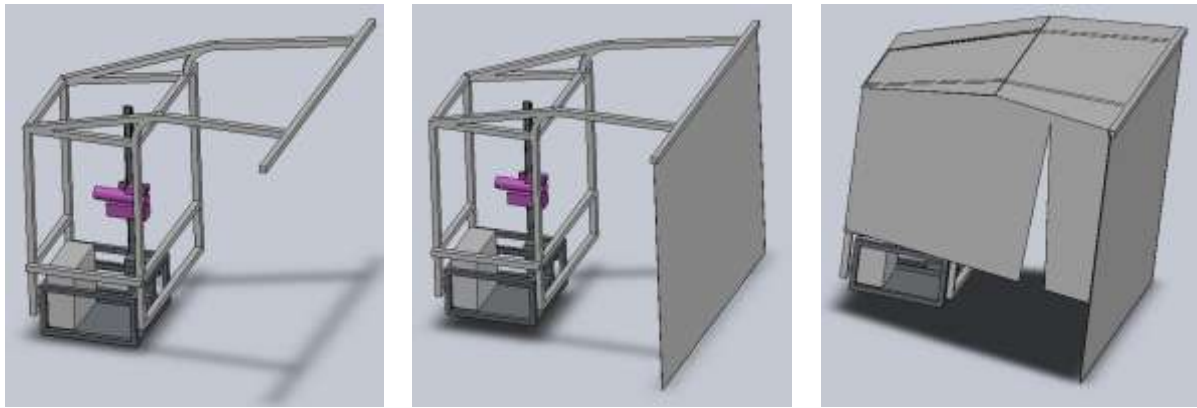


Figure 36 - The three different configurations analyzed: Rover without shield and diffuser (left), Rover with rear shield (center) and Rover with rear shield and diffuser (right).

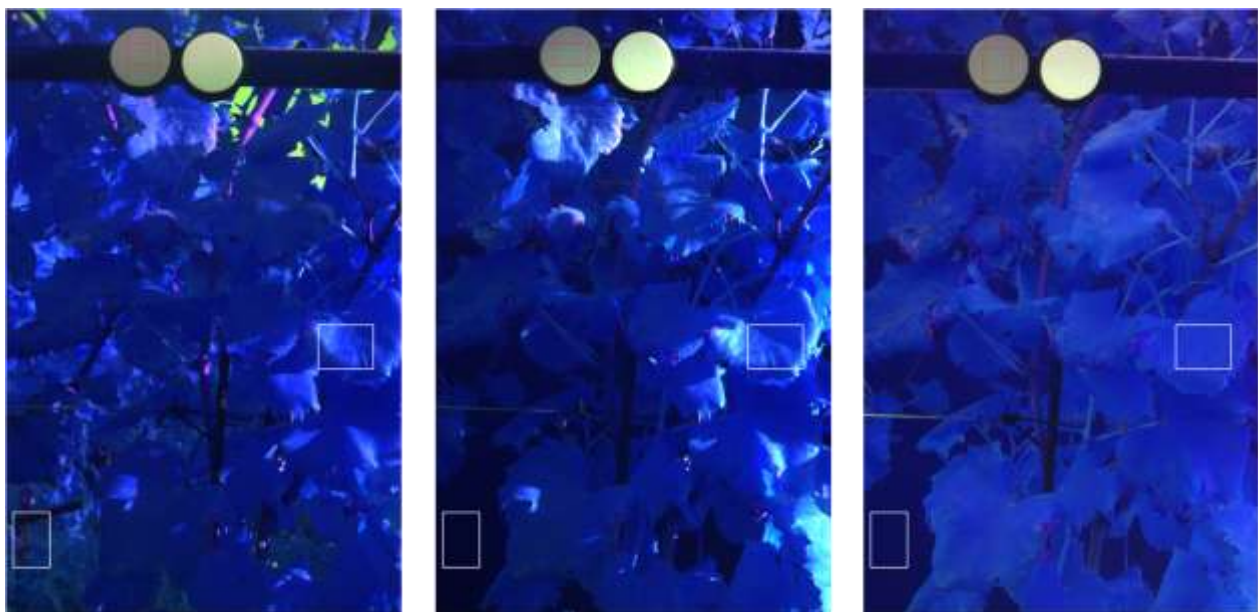
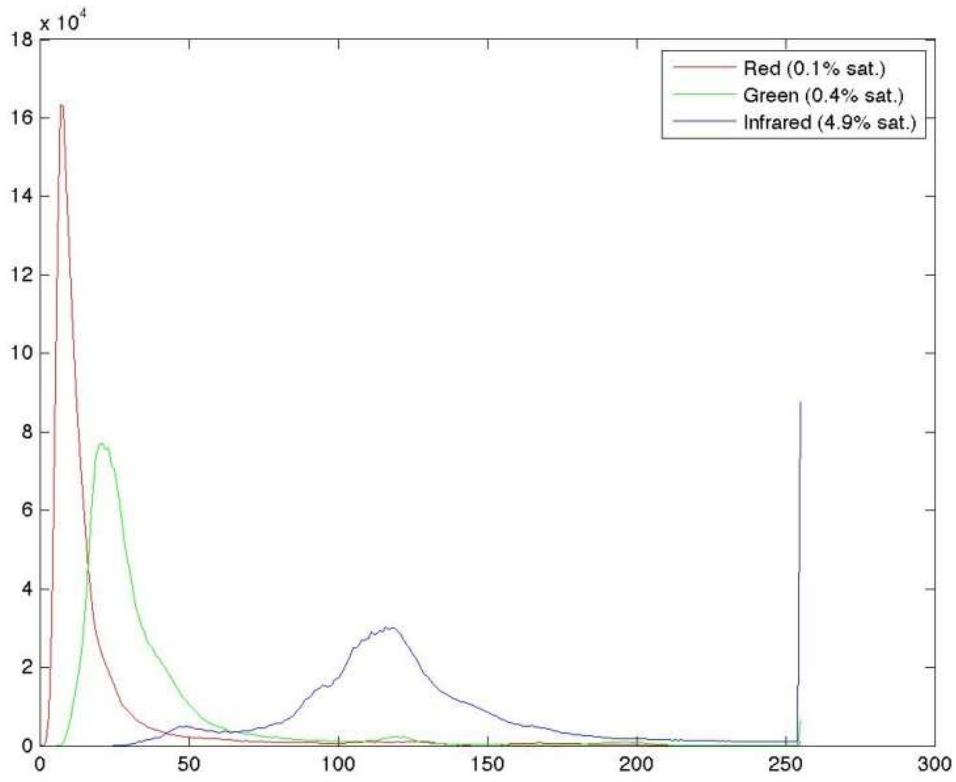


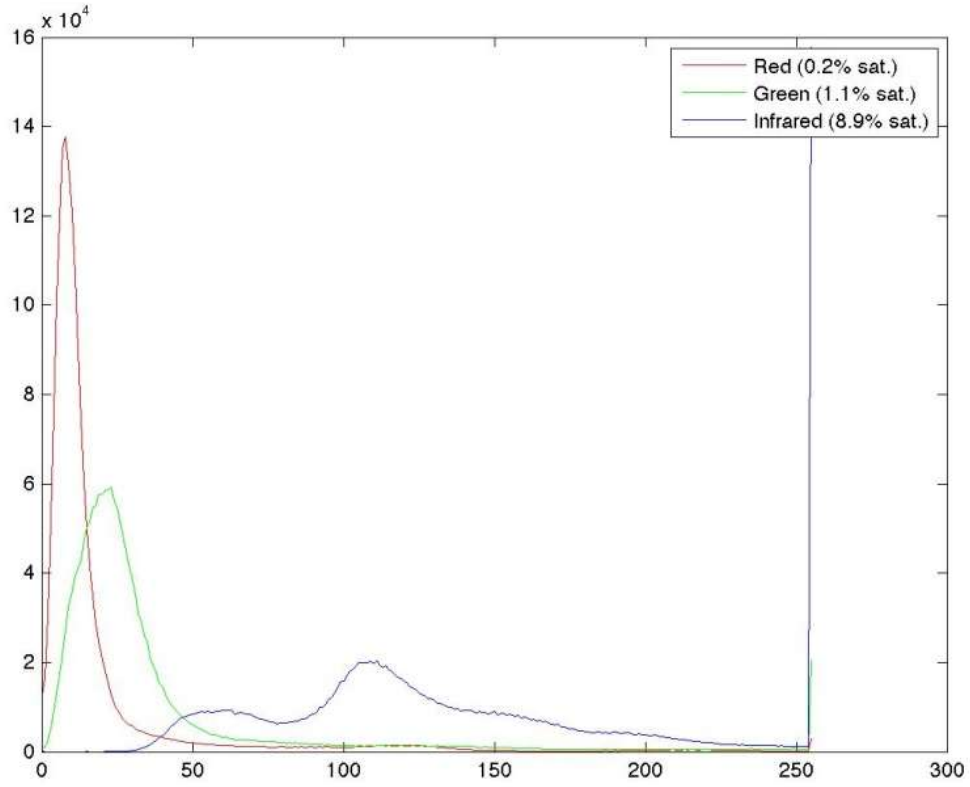
Figure 37 – Example images taken in three different shield settings. Gains are balanced respect the gray marker (50% reflectance spectralon). Rover without shield and diffuser (left), Rover with rear shield (center), Rover with rear shield and diffuser (right). In each image the rectangle on the left shows a portion of the background, while the rectangle on the right shows a portion of the leaf. The selected portions identify the corresponding areas within three different images in which the acquisition setup changes. The rectangles contain the main criticalities: with regard to the background, the grass and shrubs (if presents), as regards the leaves, the specular reflections (if presents). See Figure 38-Figure 40 for more details.

The images in Figure 37 were acquired using the marker gray (75% reflectance Spectralon) as a reference, so as to make them comparable despite the different lighting conditions. Follow the histograms evaluated on the whole picture in the three image examples.

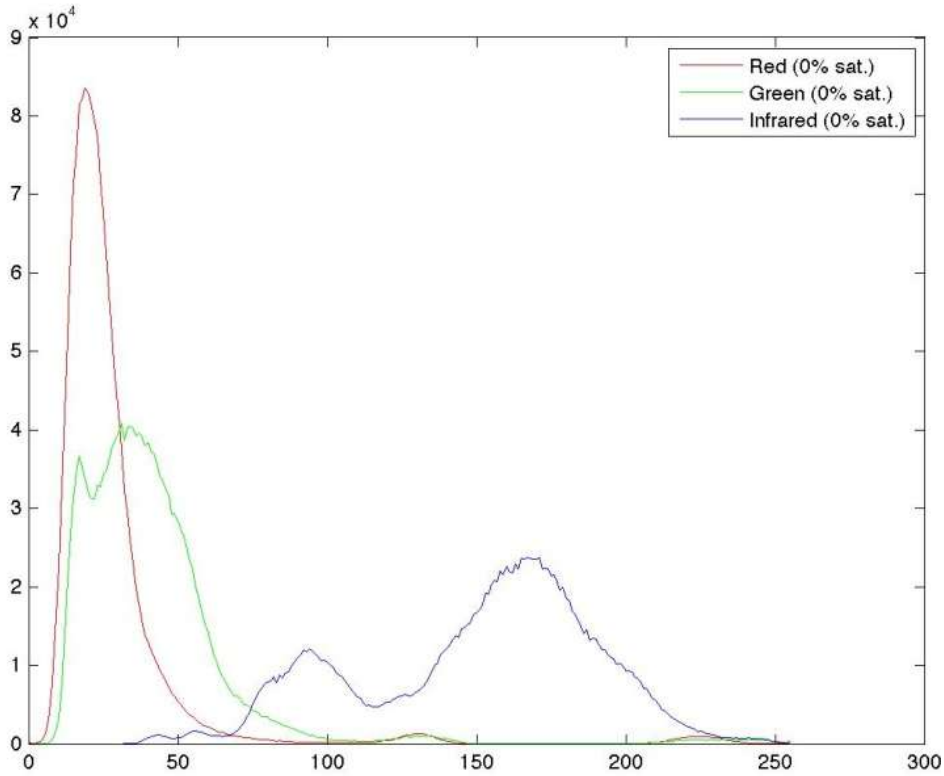




(a)



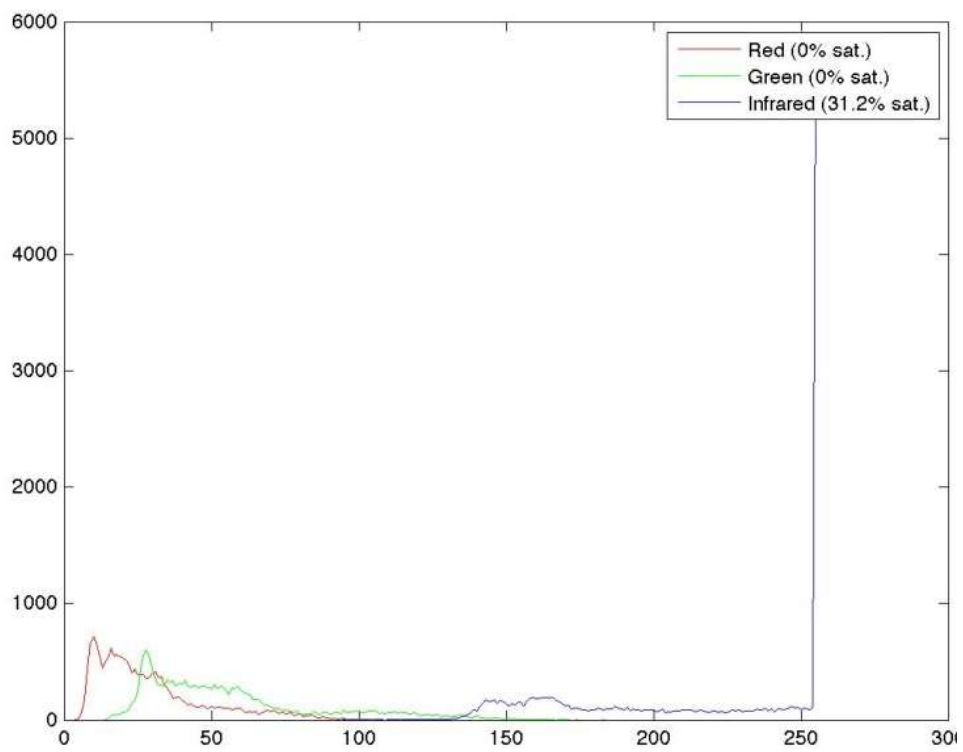
(b)



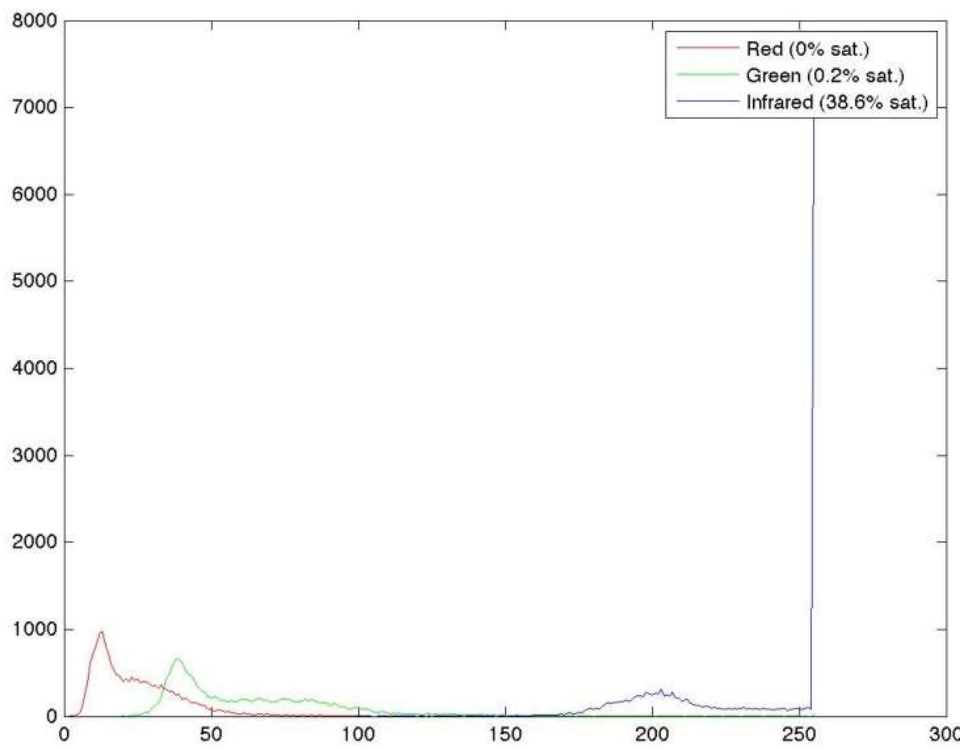
(c)

**Figure 38 - Histograms evaluated on the whole picture in the three example images: Rover without shield and diffuser (a), Rover with rear shield (b), Rover with rear shield and diffuser (c).**

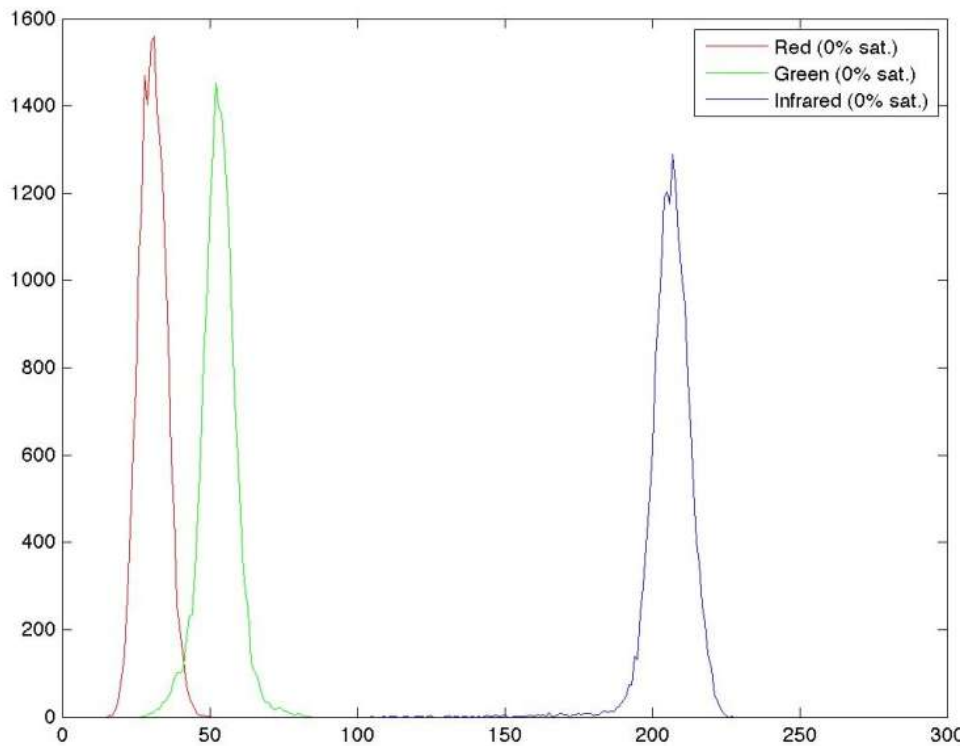
It is evident as the number of pixels saturated (image 8-bit = 255 levels of gray) is very high both in the case without shield and diffuser (with 5.4% of saturated pixels in the image, Figure 38-a) that in the case with rear shield (with 10.2% of saturated pixels in the image, Figure 38-b) and how, for both, the dynamic appears very squashed to low levels. In the third case where there are both rear shield and diffuser, the presence of saturated pixels is negligible (0% of saturated pixels, Figure 38-c) and takes advantage of the entire dynamic range of levels available. Follow the histogram evaluated on the “Leave” detail box in the three example images (Figure 39).



(a)



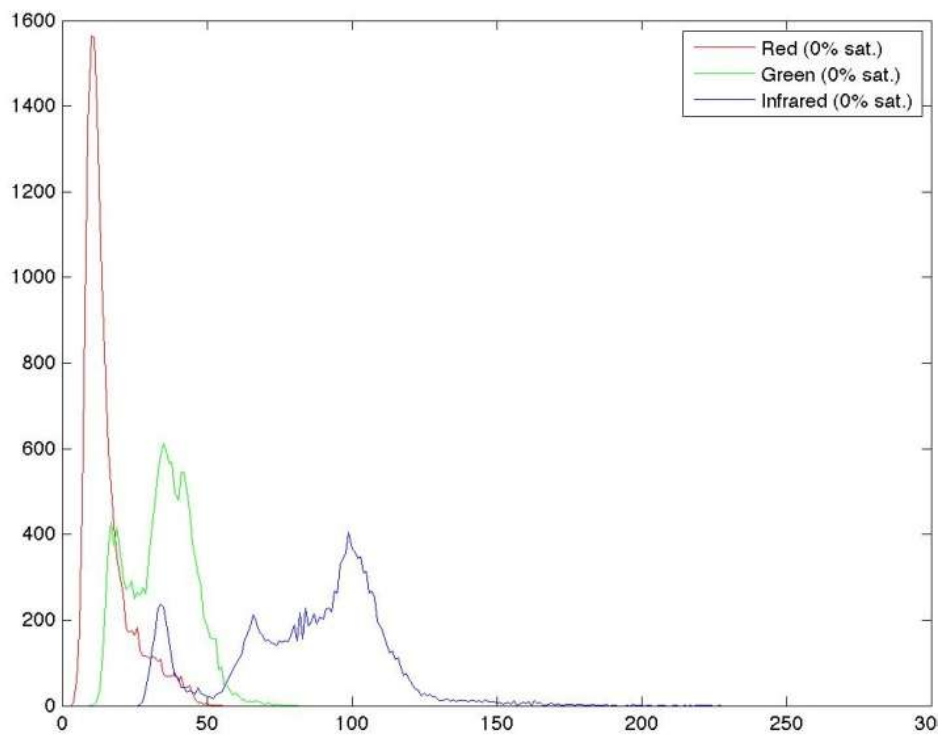
(b)



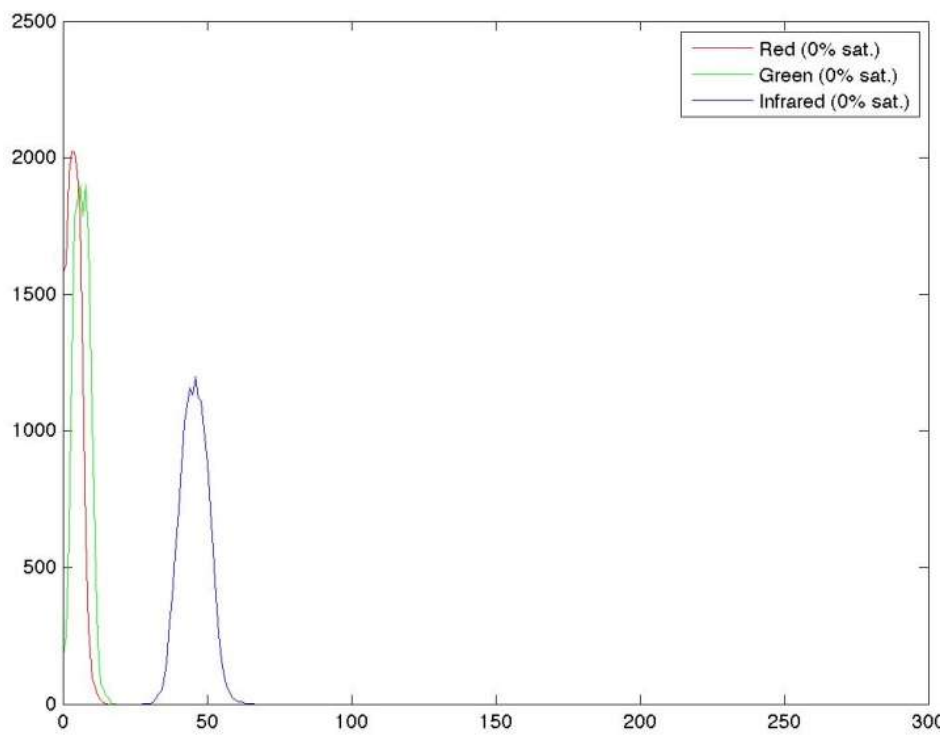
(c)

**Figure 39 - Histogram evaluated on the “Leave” detail box in the three example images: Rover without shield and diffuser (a), Rover with rear shield (b), Rover with rear shield and diffuser (c).**

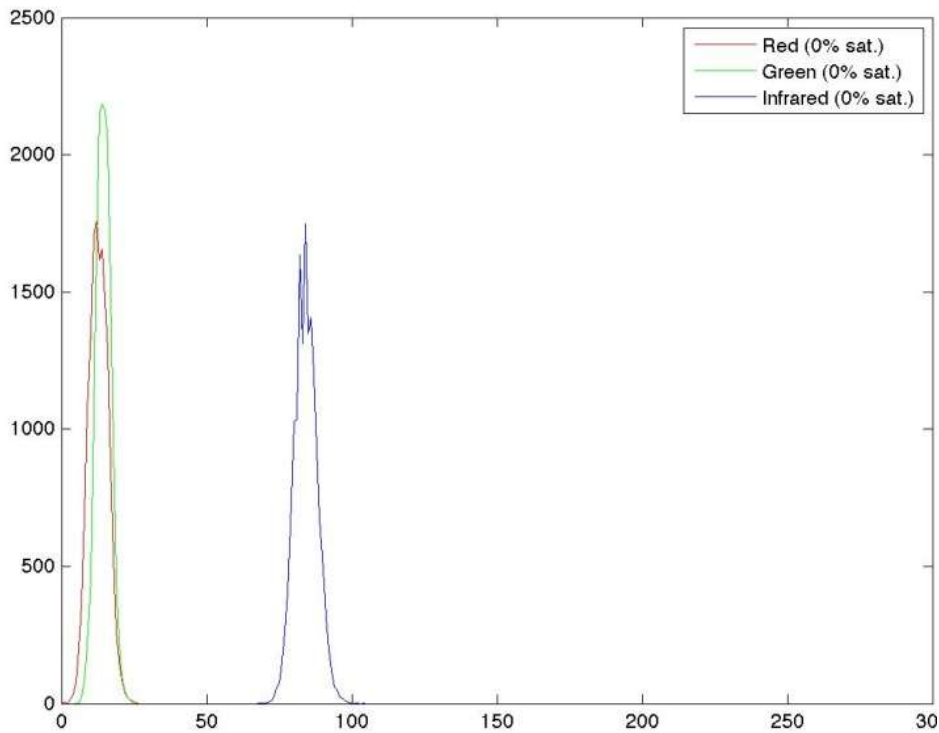
It is deliberately chosen to analyze a leaf that represents in a clear manner the problem that it was intended to highlight: the specular reflections. We want to underline the fact that without the diffuser, the acquisition of the images would generate a series of specular reflections difficult to manage in during the image processing, precluding the possibility to search for the spread of the disease in these areas. Analyzing the histograms of Figure 39 is evident the need to use a diffuser in order to prevent that the light hitting directly the subject to be acquired by generating specular reflections that will translate into saturated pixels within the image (Rover without diffuser and with/without shield about 30% of saturated pixels on the leaf, while Rover with rear shield and diffuser 0% of saturated pixel on the leaf). Follow the histogram evaluated on the “Background” detail box in the three image examples (Figure 40, an area belonging to the ground behind the vine row).



(a)



(b)



(c)

Figure 40 - Histogram evaluated on the “Background” detail box in the three example images: Rover without shield and diffuser (a), Rover with rear shield (b), Rover with rear shield and diffuser (c).

It can be noted that in the first case (Figure 40-a, Rover without shield and diffuser) the specular reflections caused by the vegetation result in background noise. This effect vanishes in the second case (Figure 40-b, Rover with rear shield) with the addition of the black rear panel. In the third case (Figure 40-c, Rover with rear shield and diffuser), the further uniformity of the diffused light, thanks to the presence of the diffuser, highlights the need to introduce a black rear panel.

### 5.2.3.3. *Measurement procedure*

The image acquisition of field took place during the whole day (in some cases for several consecutive days). Field conditions vary depending on the vineyard (extension of foliage, orientation of the vineyard, the level of infection of the disease, etc..) and lighting conditions (weather, time of day, etc.). The rover is slid into the row and positioned in front of the area you intend to acquire. The structure of the rover, with the black rear shield and the white diffuser, contains the foliage inside a box. Inside the box the ambient light is partially diffused by the white diffuser, while the black rear shield avoids that is only partially acquired the vine rows behind the area of interest. The cameras are positioned from time to time in order to acquire the vine row in the area of interest. Are acquired simultaneously two images, a R-G-NIR (Figure 41) and an R-G-B (for documentation purposes). For each image you can attach notes concerning the pathologist's comments on the possible presence of disease, its location and its size (level of infection). All parameters of the DuncanTech camera are stored (gain, integration time, focus, aperture) for each channel.



**Figure 41 – Example of R-G-NIR multispectral images to use to test the algorithm.**

The different illumination is due to the fact that the acquisitions of the bunches have been made previously to the final tuning of the calibration system of the multispectral camera. As we will see below the classification techniques of powdery mildew and downy mildew are heavily dependent on the uniformity of the different images acquired under different ambient light conditions and therefore their acquisition requires the use of special attention to light and control the parameters of the multispectral camera. In the case of Botrytis, of which we will present a first preliminary work, the acquisition of the images was just trying to extend the dynamic, avoiding saturation in the three channels. In the case of powdery mildew and downy mildew requires that images captured in different times, places and with different ambient light conditions are mutually comparable (comparable gray levels are related to entities of the same class). The classification technique is heavily based on the spectral response of the classes of interest (healthy leaf, disease, necrosis, etc ...). In the case of botrytis is not required that images captured in different times, places and with different ambient light conditions are comparable. We will try to identify the disease by using some techniques of analysis of local variations (texture, see grapes part on section 5.4.3.1 for more details).

### ***DuncanTech Camera Parameter Setting System***

Multispectral images of the field were acquired with different lighting conditions: sunny, cloudy, etc.. In order to make image processing as much as possible independent of lighting conditions, it was decided to acquire the images by adjusting the camera parameters for ambient brightness. The objective is to acquire images between them comparable regardless of the level of illumination of the scene being shot. This strategy allows us to use the acquired images without having to apply any pre-processing (RAW images) in order to compare the pixels intensities. Inside of the images

related to powdery mildew and downy mildew there are four markers having different reflectance values:

- 25% reflectance spectralon
- 50% reflectance spectralon
- 75% reflectance spectralon
- 99% reflectance spectralon

These markers are used by the system of setting of acquisition parameters of the camera in order to make comparable, irrespective of the light conditions, images acquired at different times. Various auto setting algorithms were tested, including:

1. stretching (enhancement) of the histogram for each channel of the image in order to increase the dynamic of the acquisition (remember that each channel of the camera is independent);
2. calibration of the average values of the pixels of a region of interest, the same for all channels, compared to a target value set for each channel, in order to maintain constant the pixel values for the various classes of plant tissues (healthy, diseased, necrotic, etc.). The region of interest is constituted by a single marker used on all channels; target values were chosen independently for each individual channel.
3. calibration of the average values of the pixels of a region of interest, distinct for all channels, compared to a target value set for each channel, in order to maintain constant the pixel values for the various classes of plant tissues and increase the dynamic range. The region of interest is constituted by a marker independently selected for each individual channel; target values were chosen independently for each individual channel.

At the end of the preliminary tests was clear that using the histogram stretching the dynamics of interest was compressed from limit areas, strongly or dark illuminated, such as identification tags, specular reflections, dark areas, particularly young foliage. The second solution has allowed to concentrate the dynamics of the histogram in the range of interest for the detection of pathologies; limit areas of illumination with this technique are confined to the limit of the histogram and can be easily eliminated (excluded from classification), thus showing more stable results compared to stretching. The third technique, used in this work, is a refinement (improvement sliding) of the second as it has allowed us to optimize the performance of the algorithm for each channel. The algorithm acts on two of the available parameters of the camera: integration time and gain (the offset is handled in hardware by the camera). In order to minimize noise, the algorithm minimizes the value of the gain necessary to achieve the target within the limits imposed to the integration time so as to avoid blurring of the image. In order to accelerate the convergence of the algorithm, the variation of the parameters takes place in a predictive manner with respect to the deviation from the target value: according to the distribution of the histogram, is estimated the ambient illumination and then the correct value to be imposed to the parameters.

Initially the use of marker was designed to normalize the images after they were acquired (offline). Were explored different techniques of normalization of images based on the markers present in the image (e.g.. linear, quadratic and logarithmic fitting). In part to a lack of knowledge of the transfer function of the multispectral camera (assumed and defined several times although no longer in our possession a detailed datasheet) and partly due to the poor reliability of the hardware changes made



by the camera (by the manual, the offset would be removed before each frame is passed to the application but in fact is not always the case), it was decided to consider and adopt the techniques described above. Acquisition relying on the reflectance of the markers allows us to not having to take care of the absolute values of parameters of the camera but only the final result (the image itself), so as to avoid having to manage behaviors not well defined of the multispectral camera. Use different markers for different channels allows us to increase the dynamic range in single channel without saturating the other, or in a complementary manner, not to compress the dynamics of a channel due to the saturation of another. The final result are RAW images comparable amongst themselves without the need for further processing even though acquired in different lighting conditions (also thanks to the use of the tunnel described above).

#### **5.2.3.4. Calibration and validation sets**

##### **Leaves**

Among multispectral images two sets of 15 images, each containing downy mildew and powdery mildew, respectively, were chosen. The dataset covered a range of diseases severity, leaf age, natural light conditions and other possible interfering factors as any abnormal tissue pigmentations. Within the images and according to plant pathologist classification, sub-regions of 10x10 pixels (ROIs) containing homogeneous tissue were manually selected. The ROIs selected belong to one of the following classes of interest:

- HEALT
- HEALT-YOUNG
- [DISEASED]-PRE
- [DISEASED]-INI
- [DISEASED]-MED
- [DISEASED]-ADV
- NECRO

where HEALT denotes a healthy area, HEALT-YOUNG denotes an area healthy and young DISEASE denotes an area affected by powdery mildew, and then the class will be POWD, or downy mildew, and then the class will be DOWN. The level of infection of the disease was indicated by PRE, pre-symptoms, INI, initial, MED, median, ADV advance. NECRO denotes a necrotic area. The distinction between healthy young and healthy area was necessary after it was found that the small leaves (young) had a spectral response similar to that of the disease, especially powdery mildew in advanced stage; it was therefore decided to analyze separately the two classes so as to strengthen the technique of classification. The classes are defined as a result of the experience gained in the field during acquisitions. The definition of those classes of interest is made in order to reduce the false positive. The classes identified are the main types of different spectral response that it was possible to select, without exceeding the level of detail while considering a sufficient number of real cases. The identified ROIs were used to define the disease detection algorithm and to train the classification parameters. Just after the acquisition, five images were carefully surveyed by a plant pathologist. According to the evidence of this survey, we manually painted the infected areas of the image of each leaf using a color code for the different classes of

intensity of symptoms (Figure 42). These images were subsequently used to test the performance of the detection algorithm.



**Figure 42 - Example of validation image (zoomed). The colored areas were manually drawn during a coincident in-vivo phytopathological survey of the leaf: unpainted pixels correspond to healthy tissue; in magenta is a region exhibiting initial symptoms of downy mildew, in red are medium intensity infection spots.**

## **Grapes**

As regards the grapes, have been acquired a series of images in the first season of acquisition, before the acquisition system was improved. The acquired images are not many and are only indicative of what it was intended to do. Anyway, based on some of the captured images will be defined a classifier for Botrytis. The results presented are preliminary. For Botrytis, the approach is different from those for powdery mildew and downy mildew. On the acquired images are not defined ROIs, but applies a classification technique based on texture. Since it is not necessary that the captured images are mutually comparable, the markers present in the image will not be used (they were only used in the acquisition phase to increase the dynamic range of the image). For Botrytis do not need to train any classifier but only determine the correct sequence of filtering to be applied to acquired images. Three images were selected for a validation step of algorithm application. These images were carefully analyzed by a plant pathologist. Accordingly to the evidences of this survey, we manually painted the infected areas found in the images. These images were used to test the performance of the automatic detection algorithm (Figure 43-right). For the grapes is also evaluated the ability of the algorithm to identify the bunches (Figure 43-center). The evaluation of the algorithm ability to identify bunches and botrytis will be based on the percentage of pixels identified (considering false positives, false negatives, etc.). For example, the percentage of identified bunches will be calculated according to PPV, NPV, Sensitivity and Specificity definitions (see Appendices-PPV, NPV, Sensitivity and Spec for more details) based on the absolute number of pixels (background and foreground) in the image.

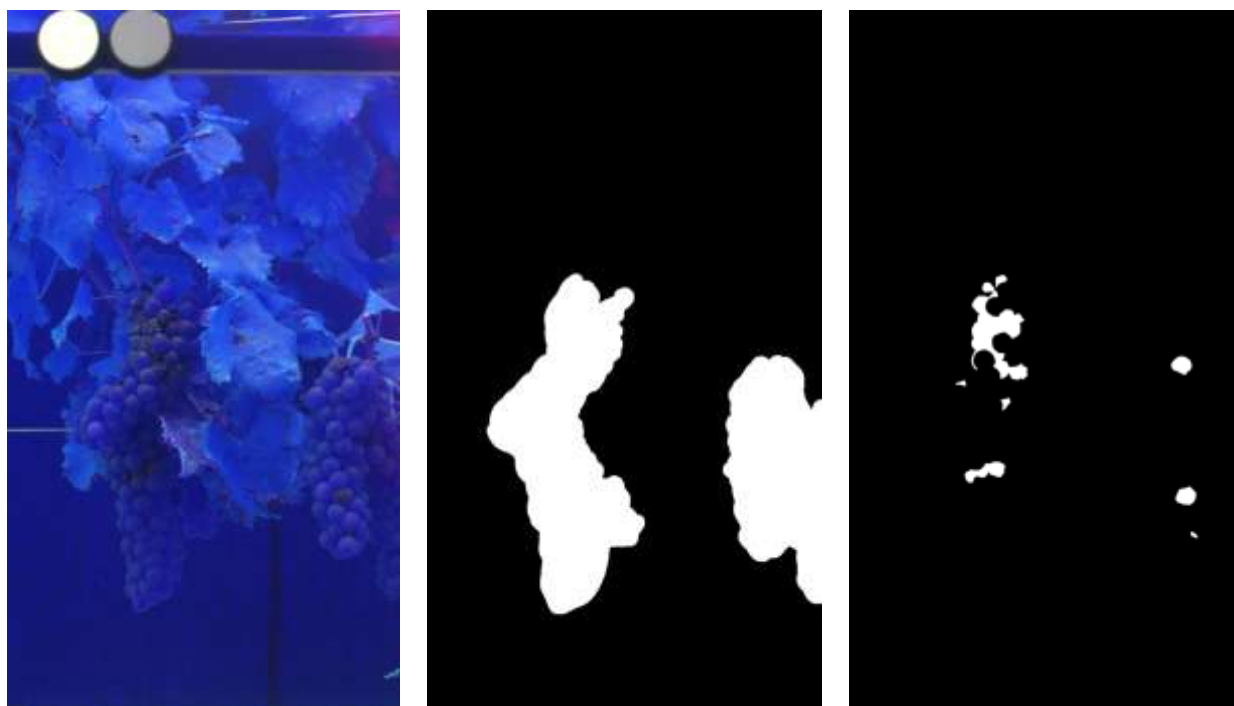


Figure 43 - Example of validation images manually draw. Left: the acquired image. Center: the colored areas represent the grapes. Right: the colored areas were manually drawn during a coincident in-vivo phytopathological survey of the leaf: painted pixels correspond to the botrytis.

### 5.3. Spectral reflectance processing

- ✓ Laboratory Methods (5.3.1): description of the implemented spectral diseased detection algorithms based on spectra data acquired in laboratory and the used techniques to implement this algorithms (5.3.1.1), main obtained results, their discussions and conclusions (5.3.1.2).

#### 5.3.1. Laboratory Methods

The indicators defined at laboratory Calibration and validation sets are here analyzed through multiple classification techniques in order to evaluate the discriminatory power. The analysis was divided into a first step of manual validation of indicators and a following step of classification based on different classifiers. The data on which the indicators were evaluated are the spectra acquired during laboratory Measurement procedure.

##### 5.3.1.1. Diseases detection algorithms

First, each acquired spectrum is applied to a large smoothing filter of 30 acquisition values (equal about to 10 nm) to reduce the background noise related to the spectrophotometer. Following the indicators described above (section 5.2.1.4) are calculated (individual values, angles, ratios). Each indicator is evaluated individually and manually validated based on the statistical analysis (mean value, standard deviation, minimum and maximum value) that produces. The data thus collected are plotted on multiple charts (equal to the number of classes of interest) where the calculated values are compared on the spectra acquired in the different days of measurement. The visual comparison provides a first feedback on the discriminatory power of the selected indicators. The following are some of the most relevant results obtained from the analysis of the indicators. The objective is to find the spectral indicators that help to determine the state of health of the plant, that is to reduce the space of the analysis from the entire light spectrum to few indicators sufficient to define and delimit the problem. The ultimate goal of this analysis is to determine the combinations of wavelengths that can be of help in the choice of the multispectral camera to be used during the acquisition of the images (in the eventual choice of optical filters to be applied to the camera deduced from wavelengths that at this stage are more discriminatory).

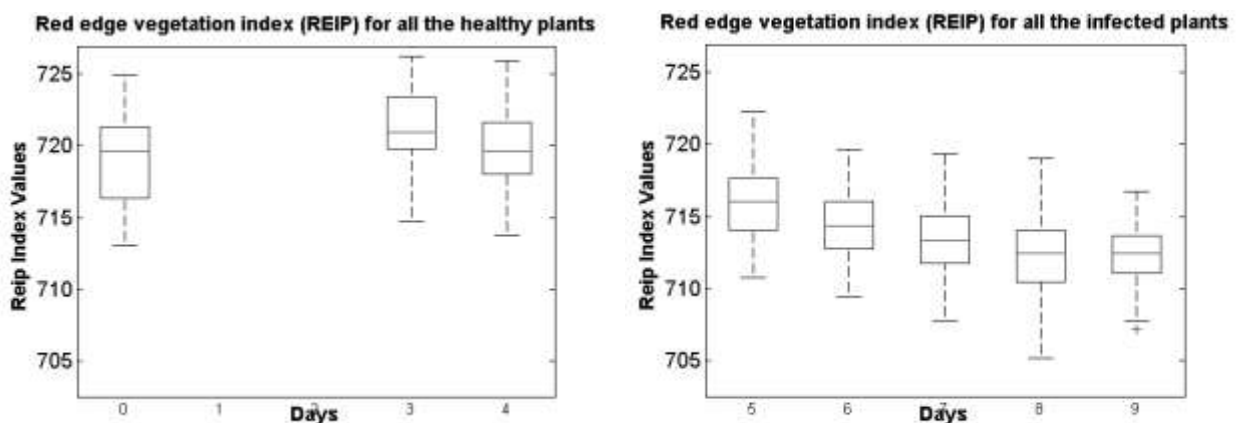


Figure 44 - Red edge vegetation index (REIP). In the left the evolution of the indicator in the case of healthy leaf and in the right the evolution of the indicator in the case of infected leaf (powdery mildew).

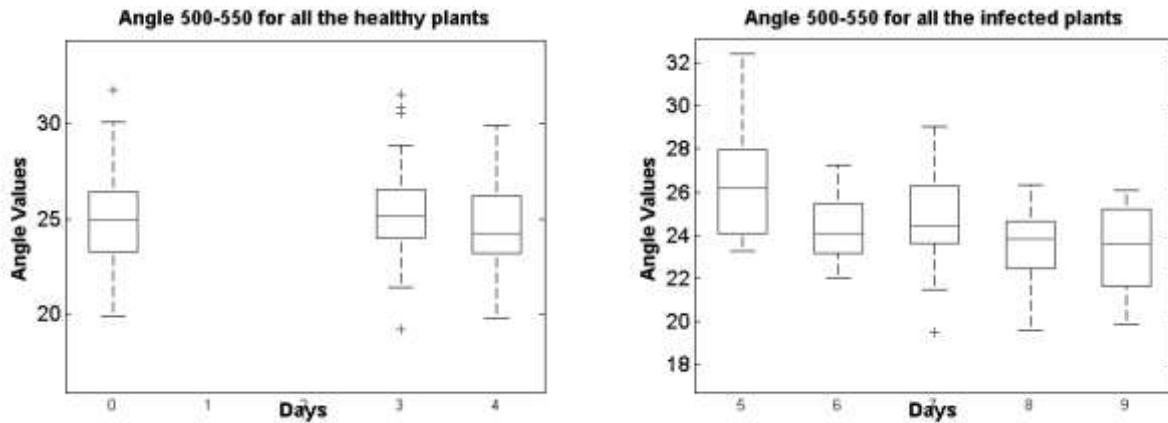


Figure 45 - Angle between wavelengths at 500 and 550. In the left the evolution of the indicator in the case of healthy leaf and in the right the evolution of the indicator in the case of infected leaf (powdery mildew).

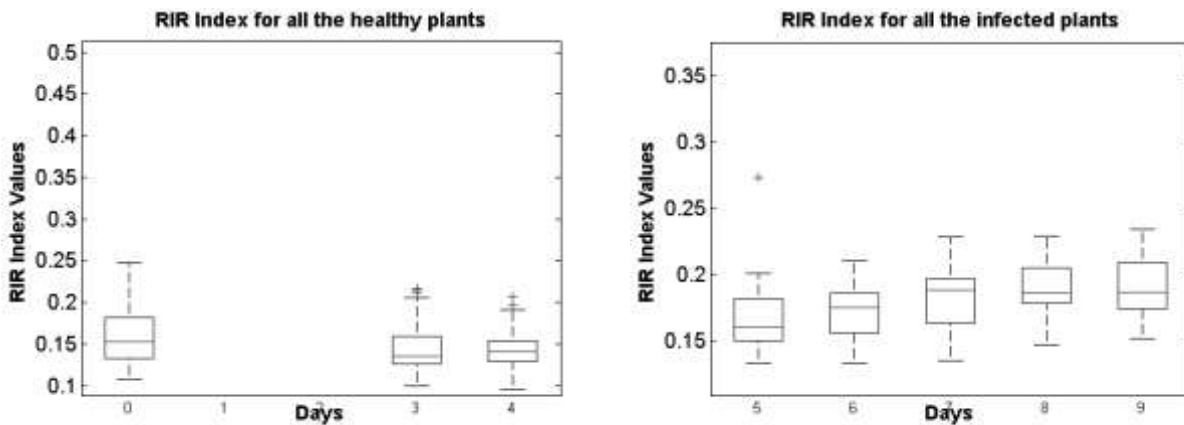


Figure 46 - RIR index. In the left the evolution of the indicator in the case of healthy leaf and in the right the evolution of the indicator in the case of infected leaf (powdery mildew).

The graphs shown in Figure 44, Figure 45 and Figure 46 report, in the left column the evolution of the indicator in the case of healthy leaf and in right column the evolution of the indicator in the case of infected leaf (powdery mildew). Upon the indication of the pathologist, the first two days following inoculation, the spectral measurements were not performed because they are irrelevant to the analysis of disease progression (the plant is still healthy). The behavior of the indicators shown is the same for both healthy leaves (not subject to inoculation) and infected leaves in the first four days after inoculation (reported in same graph). The measures 1 and 2 (first and second day of acquisition, assuming the measurement at time 0 as a measure pre-inoculation) were not acquired because of little interest in the development of the disease. The right column shows the values of the indicator from the fifth day after inoculation and until the ninth day. It can be seen that while in the case of healthy leaves (not shown here over the fourth day since the indicators assume values very similar to the previous days) the values of the indicators are in fact constant, or rather it is attested around the same mean value, in the case of inoculated leaves, the value of the indicators change over time with the spread of the disease. In the case of REIP index and the angle between 500 nm and 550 nm the values of the indicators decreases with the progress of the disease. Conversely, in the case of the RIR index, the indicator value increases with the progress of the disease. As further confirmation of the trend of the indices, the same graphs are shown below (Figure 47, Figure 48 and

Figure 49) where they compared the first days of the experiment (left column, healthy leaves and inoculated within four days after inoculation) with the tenth and eleventh days following inoculation. The tenth and the eleventh day, when the disease is of such a progress as to make useless any type of treatment, give a clear idea of the trend of indicators over time with the disease progressed. What we want to highlight is the performance of the indices in the three cases considered: healthy leaf, leaf where the disease is progressing, leaf on which the disease has taken hold. The graphs allow, in the case of healthy vs. infected, to display the trend of the indices with the evolution of the disease (illness progression), in the second case, to highlight the discriminatory power of the indices in healthy vs. diseased comparison (illness manifest).

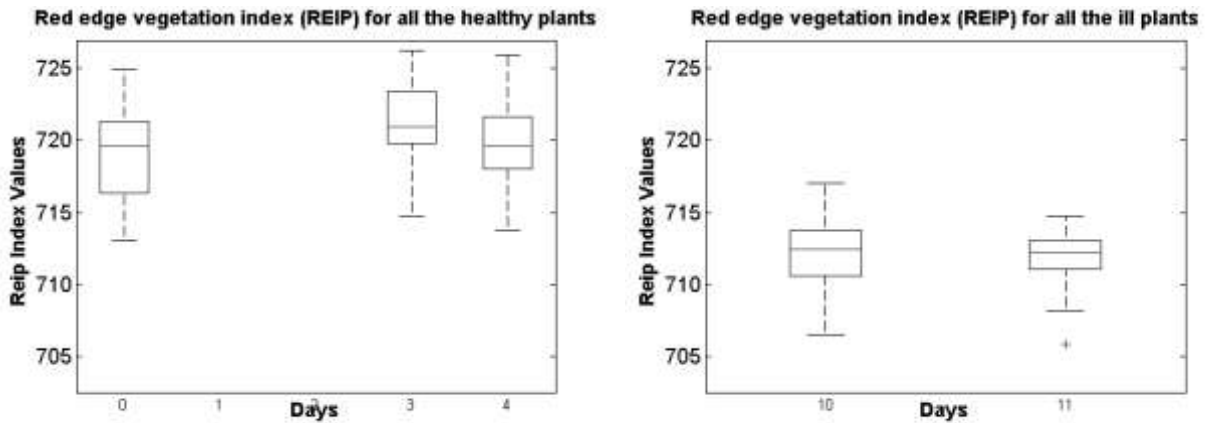


Figure 47 - Red edge vegetation index (REIP). In the left the evolution of the indicator in the case of healthy leaf and in the right the evolution of the indicator in the case of ill leaf (powdery mildew).

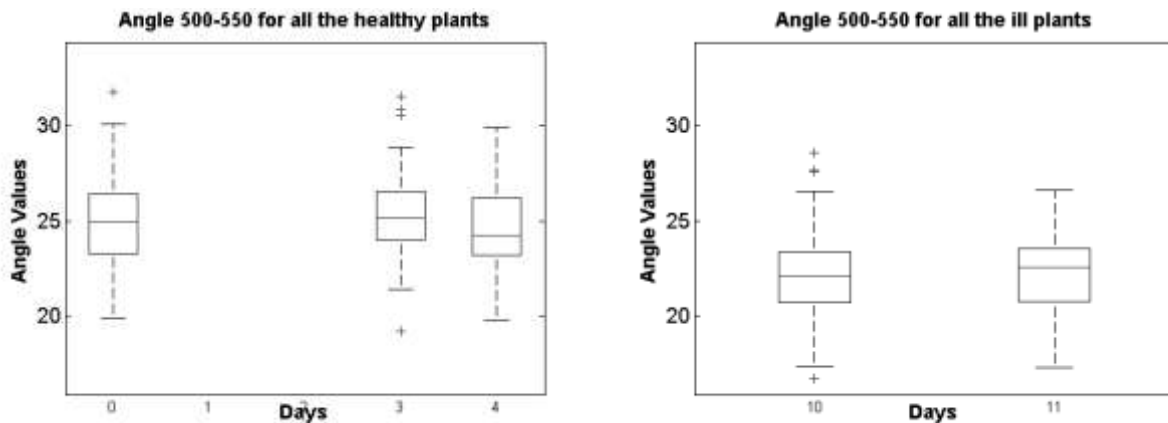
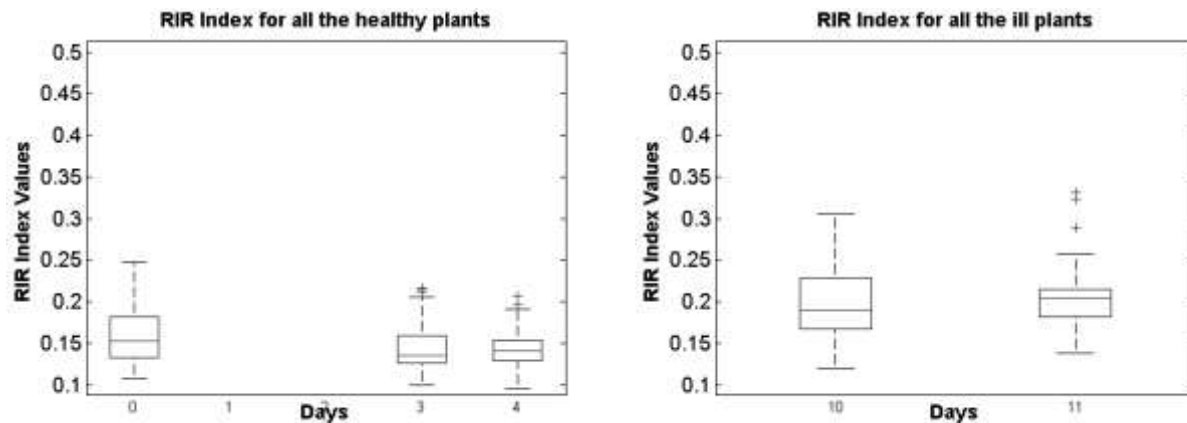


Figure 48 - Angle between wavelengths at 500 and 550. In the left the evolution of the indicator in the case of healthy leaf and in the right the evolution of the indicator in the case of ill leaf (powdery mildew).



**Figure 49 - RIR index. In the left the evolution of the indicator in the case of healthy leaf and in the right the evolution of the indicator in the case of ill leaf (powdery mildew).**

From the manual validation of the indicators chosen, the second phase concerned the classification of spectra. For each acquired spectrum are calculated indicators. At the end of the initialization procedure, each acquired spectrum will be described by a vector containing the indicators taken into consideration. The values of different indicators are concatenated to build the vectors. The vectors thus obtained becomes the input for the phases of training and testing of the classifier. The classes of interest taken into account are:

- healthy leaves (Absence of illness)
- inoculated leaves from 4 to 9 days (Illness progression)
- diseased leaves (inoculated leaves from 10 and 11 days, Illness manifest)

As inferred from the manual validation of the spectra, are considered healthy leaves all the leaves belonging to non-inoculated plants. The spectra belonging to healthy leaves will be acquired for the entire duration of the experiment (from day 1 to day 11). The leaves are considered to be in a state of progression of the disease if they belong to plants inoculated from 4 to 9 days since the first few days after inoculation the spectral response of the inoculated leaves we have seen is the same as for healthy leaves. The leaves are considered irretrievably diseased exceeds 10 days after inoculation. Since the development of disease is not the same for all plants, during the illness progression there are leaves in which the disease is slow to occur and the leaves where the disease has reached a visible state. Keep in mind that this type of local approach is targeted at the early recognition of the disease so it is not said that the lack of obvious symptoms to the naked eye corresponds to a good state of health. The classifier is trained by considering only the leaves healthy (Absence of illness) and the diseased (Illness manifest), and used later to classify even those in a state of illness progression. What we would like is to classify correctly distinguish the healthy leaves from the diseased but also understand how this classifier is able to detect disease in the leaves of which the disease is making its course. You want to define as a classifier thus constructed, based on the indicators chosen, is able to properly identify a leaf diseased also in its state of Illness progression. In support of this approach is the fact, revealed by manual testing of indicators, that when the disease has the upper hand on the leaf, its spread on the leaf surface is immediate and consistent.

Were adopted and compared different classification techniques:

- **LDA** (Linear discriminant analysis)
- **PCA** (Principal component analysis) + **LDA** (Linear discriminant analysis)
- **PLS** (Partial least squares regression)

The choice of these techniques was driven by:

- the need to reduce the amount of data in our possession, checking in an easy manner the ratio “data dimension / classification quality”;
- the will to use simple and intuitive classification techniques (where linear is the best case);

The first technique, LDA, is the most simple and intuitive. The other two techniques, PCA and PLS, besides allowing to reduce the dimensionality of the input data (the space dimensionality of the input data), allow to control, by varying the dimensionality of the output components (the space dimensionality of the output data) the level of detail you want to get with the classification method. All analyzed techniques have been validated using the method of the K-fold cross-validation, using  $K = 10$ , to randomly generate the indices to be used for data partitioning. Indices contains equal (or approximately equal) proportions of the integers 1 through 10 that define a partition of the observations into 10 disjoint subsets. Repeated calls return different randomly generated partitions. In K-fold cross-validation,  $K-1$  folds are used for training and the last fold is used for evaluation. This process is repeated  $K$  times, leaving one different fold for evaluation each time. At each iteration we collect data for classification and will update the correct rate (Correctly Classified Samples / Classified Samples). The data used to define the classifier are those of the spectra of healthy plants (Absence of illness) and spectra of diseased plants (Illness manifest). This data is partitioned by cross validation method and define, for each partition, the classifier for use within the partition. As reasoned above, with each classifier defined within the cross validation, are also classified the spectra of diseased plants in the state of illness progression. For classifications of spectra in a state of illness progression will not be evaluated a correct rate, but a diseased rate (Diseased Classified Samples / Classified Samples), that is, in what percentage the spectra belonging to the state of illness progression are considered to belong to diseased plants. Also these will be kept up to date the status of classification of different cross validation. Following the discussion of the adopted classification techniques.

## **LDA**

The first technique used as input the vectors representing the acquired spectrum and constructed on the basis of chosen indicators. The classification technique used here is the linear discriminant analysis (LDA). It will then search for each K-fold partition to find a linear combination of features (indicators) which separates the two classes of interest. The resulting combination will be used as a linear classifier (see section 4.1.3.1 for more details on LDA background theory).

In LDA we seek a linear mapping that maximizes the linear class separability in the new representation of the data. We consider a set of  $n$   $p$ -dimensional observations  $x_1, \dots, x_n$  (the vectors of indicators representing the acquired spectrum), with  $n_1$  samples labeled as belonging to class 1 ( $\lambda_1$ ) and  $n_2$  samples as belonging to class 2 ( $\lambda_2$ ). In our problem  $n=2640$  (240 measures for 11 days),  $n_1=880$  (10 measures on 2 leaves of 4 plants for 11 days) where  $\lambda_1$  is the not inoculated plants class,  $n_2=1760$  (10 measures on 2 leaves of 8 plants for 11 days) where  $\lambda_2$  is the inoculated plants class and the data dimensionality  $p=21$  (all the considered indicators described previously: specific



wavelengths, angles between wavelengths and ratio between wavelengths). Using K-fold partition, with  $K=10$ , from time to time, we randomly consider 90% of the input vectors to training the classifier (to find a linear combination of the indicators) and 10% of the input vectors to test the classifier (not in the training set, to check if the computed linear combination works fine).

The input vectors data now reside in one dimension. So, we will construct a kernel density estimate and then search for the intersection point in the same way it is described in Background when the threshold is found as the intersection point of two Gaussian functions. The classification result of a vector can be ill, healthy or suffering. To check the healthy status of the plant, the inoculated plants are considered to be suffering from 5 and up to 9 days after inoculation (with or without evident signs of infection) and ill more than 9 days after inoculation because, during the experiment, all inoculated plants were in obvious state of disease 10 and 11 days after inoculation. So, if one vector, associated to a plant inoculated from 5 and up to 9 days, is classified as ill it will be labeled as suffering, while if one vector, associated to a plant inoculated from 10 or 11 days, is classified as ill it will be labeled as ill. The same applies to not inoculated plants if a vector, associated with them, is classified as ill when belongs to a period from 5 and up to 9 days after inoculation or 10 and 11 days after inoculation.

### **PCA + LDA**

The second technique used does not consider as input vectors representing the acquired spectrum constructed on the basis of chosen indicators, but their  $N$  principal components that account for 90% of the data variance.

In PCA the main purpose is to reduce the dimensionality from  $p$  to  $d$ , where  $d < p$ , while at the same time accounting for as much of the variation in the original data set as possible. With PCA, we transform the input vector data to a new set of coordinates or variables that are a linear combination of the original variable. In addition, the observation in the new principal component space are uncorrelated. We start with our centered data matrix  $X_c$  that has dimension  $n \times p$  where, in our problem,  $n=2640$  (240 measures for 11 days: 10 measures on 2 leaves of 4 not inoculated plants for 11 days and 10 measures on 2 leaves of 8 inoculated plants for 11 days) and the data dimensionality  $p=21$  (all the considered indicators described previously: specific wavelengths, angles between wavelengths and ratio between wavelengths). This matrix contains observation that are centered about the mean; i.e., the sample mean has been subtracted from each row. See section 4.1.3.2 for more details on PCA background theory.

Starting from the indicators, we try to reduce the size of the space to see if you can still get good results. They take the vectors of the indicators (scaled to have standard units), calculate the principal component coefficients and a vector containing the eigenvalues of the covariance matrix of data. Starting from the eigenvalues are obtained the number of principal component necessary to account for 90% of the data variance. Each vector of indicators will be associated with a vector of principal components. Using the vectors of the principal components, instead of the vectors of the indicators, we proceed with a Linear discriminant analysis as in the first case analyzed.

### **PLS**

The third technique used as input the vectors representing the acquired spectrum and constructed on the basis of chosen indicators. This technique has not been investigated in depth with regard to its

mathematical definition. Only its fundamentals have been studied in order to be able to best use an pre-existing algorithm, to compare the results obtained with the techniques adopted described previously (see section 4.1.3.3 for more details on PLS background theory). It will then search for each K-fold partition to compute a partial least-squares (PLS) regression of classes of interest on vectors of indicators, using 15 PLS components, and returns the PLS regression coefficients BETA to use to classify the data. For each K-fold partition the input data of the PLS are normalized using the zscore standardization technique. The values of means and standard deviations used for centering and scaling the data, *mu* and *sigma*, respectively, are stored to be used during the classification step. At this point the coefficients BETA are calculated and, together with the *mu* and *sigma* previously calculated, are used to assign to the data vector his class of membership. If, in order to determine the coefficients BETA, are used as classes of interest the value 1 for healthy and -1 for the diseased, the classification made by BETA return values between this two extremes. The last step rounds the result to ensure that each vector is associated with a class. For clarity, it shows the pseudo-code of the algorithm described (for each partition):

```

pls_components = 15;
[training_data_z, mu_data, sigma_data] = zscore(training_data);
[training_classes_z, mu_class, sigma_class] = zscore(training_classes);
BETA = computePLS(training_data_z, training_classes_z, pls_components);
for i=1:length(test_data)
    test_data_z = (test_data - mu_data) / sigma_data;    % normalization
    % Add here check of test_data_z finite values
    n = length(test_data_z);
    test_class_z = [ones(n,1) test_data_z] * BETA;      % classification
    test_class = roundToPLSClass((test_class_z * sigma_class) + mu_class);
end

```

`training_data` contains, from time to time (from K-fold selection with K=10), the selected training  $n_k$  vectors of  $p$  dimension, where  $n_k=2376$  (90% randomly chosen of all the vectors representing the acquired spectrum) and `test_data` contains, from time to time, the selected testing  $m_k$  vectors of  $p$  dimension, where  $m_k=264$  (10% randomly chosen of all the vectors representing the acquired spectrum not in the training set). `training_classes` is a binary vector of  $n_k \times 1$  dimension, associated to the `training_data` vector, in which 1 represents a vector from a not inoculated plant and -1 represents a vector from an inoculated plant. The resulting `test_class`, a binary vector of  $m_k \times 1$  dimension, associated to the `test_data`, will contain the classification results, where 1 will indicate that the corresponding vector belongs to a healthy plant, while -1 which belongs to a ill plant. This classification allows us to classify as deriving from suffering symptoms, all those vectors which is assigned a value around 0, and precisely:

- $-1 \leq \text{test\_class}_i < -0.5$  for ill;
- $-0.5 \leq \text{test\_class}_i \leq 0.5$  for suffering;
- $0.5 < \text{test\_class}_i \leq 1$  for healthy

### 5.3.1.2. *Results and Discussion*

#### **Test of algorithm accuracy on the validation set**

Below are the results obtained using the three techniques described above applied to the acquired dataset described in section 5.2.1 (Laboratory tests).

	<b>Not inoculated</b> classify as <b>Healthy</b> (days 0-11)	<b>Not inoculated</b> classify as <b>Suffering</b> (days 5-9)	<b>Not inoculated</b> classify as <b>Ill</b> (days 10-11)	<b>Inoculated</b> classify as <b>Healthy</b> (days 0-4)	<b>Inoculated</b> classify as <b>Suffering</b> (days 5-9)	<b>Inoculated</b> classify as <b>Ill</b> (days 10-11)
LDA	95%	2%	0%	94%	45%	99%
PLS	99%	0%	0%	99%	85%	99%
PCA + LDA	90%	4%	5%	89%	38%	73%

**Table 1 – Results for the three classification techniques of spectra analyzed. You can see how a healthy (not inoculated) leaf is classified as such during the whole experiment (a not inoculated leaf should be classify as healthy for the whole length of the experiment). You can also see the evolution of the disease evaluated by the three different algorithms. The inoculated plants are considered (and tested) to be suffering (with or without signs of infection) from 5 and up to 9 days after inoculation; during the experiment, more than 9 days, all inoculated plants were in obvious state of disease.**

From the table it can be inferred that the reduction of the data using Principal Components Analysis results in an evident worsening of classification. For the entire duration of the measurement (from day 0 to day 11), using as classifier the PCA+LDA method, the spectra associated to the not inoculated plants is correctly classified as healthy in a good percentage of cases (90%), but this percentage is significantly lower (73%) than those obtained by the use of other techniques when it comes to correctly classify the inoculated plants as ill plants in the last days of measurement (days 10 and 11 of acquisition). Always for the PCA+LDA method, the discriminatory power in the case of suffering plants, that is the predictive power of the classifier, is only 38% (remember that: the inoculated plants are considered to be suffering from 5 and up to 9 days after inoculation; during the experiment, more than 9 days, all inoculated plants were in obvious state of disease.). The resulting classification from PCA+LDA method is very limited when compared to the good results obtained with the LDA method and the excellent results achieved with PLS method. In the case of LDA method there is an excellent classification both in the case of not inoculated plants classify as healthy for the entire duration of the measurement (95%) as well as in the case of inoculated plants classify as ill plants in the last days of measurement (99%). Less good is the predictive power that is 48%. In the case of PLS as well as excellent results in healthy (99%) and ill (99%) classifications there are also some excellent results with regard to its predictive power; 85% of the inoculated plants from 5 to 9 days is correctly classified as suffering (or ill), sign of disease progression. The other data which emerges from the analysis of the results, and confirms what was assumed previously, is that in the early days after inoculation (from day 0 to day 4) the multispectral analysis of the inoculated plants presents itself entirely similar to those of not inoculated (healthy) plants (94% for LDA method, 99% for PLS method and 89% for PCA+LDA method). This result highlights the fact that in the first few days after inoculation the disease has not yet made significant changes to the plants (for what concerns the wavelengths considered and the measurement techniques applied). Finally, you can see that the percentage of not inoculated plants classify as suffering and ill is very small for all the used techniques, better for LDA (2% and 0% respectively) and PLS (0% for both) than for PCA+LDA method (4% and 5% respectively). PCA+LDA does not work so well because it considers the main components associated with the 90% of the variance expressed from the defined indicators, reducing too much the information usable during the classification. LDA (and in like manner PCA+LDA) works fine but its predictive power is too low because of the strict separation imposed from the method between healthy and ill. PLS works fine, and its predictive power is good, because it permits to have transition values between healthy (expressed with 1) and ill (expressed with -1) to indicate the suffering state (using the appropriate intervals).

## Conclusions

Following the results obtained with the PLS, using this technique it is proceeded to verify (and display) the progress of the disease during the experiment. For each area in which each leaf of each plant was divided has been shown the classification result of the algorithm by using PLS for the whole experiment period. The result is a table showing the development of localized disease for each plant used (Figure 50).

24	p4f1r3	0	3	4	6	7	8	9	10	11	1	1	1	0	0	0	0	-1	-1		
25	p4f1r4	0	3	4	5	7	8	9	10	11	1	1	1	1	1	0	0	-1	-1		
26	p4f1r5	0	3	4	5	6	7	8	9	10	1	1	1	1	1	0	0	0	-1		
27	p4f1r6	0	3	4	5	7	8	9	10	11	1	1	1	0	0	0	0	-1	-1		
28	p4f1r7	0	3	4	6	7	8	9	10	11	1	1	1	0	0	0	0	-1	-1		
29	p4f1r8	0	3	4	5	6	7	8	9	10	11	1	1	1	1	1	0	0	0	-1	-1
30	p4f1r9	0	3	4	5	6	7	8	9	10	11	1	1	1	1	0	0	0	0	-1	-1

Figure 50 – An extract of the resulting table from PLS classification showing the development of localized disease for each plant used. The first column enumerates the considered area of the leaves, the second column shows the considered area of the leaves (p for plant, f for leaf, r for area, see Figure 26 from more details), the third column shows the days on which the measurement was performed and the fourth column shows the health condition of the considered area of the leaves in the days on which the measurement was performed where 1 indicates that the area has been classified as healthy, 0 indicates that the area has been classified as suffering (Illness progression) and -1 indicates that the area has been classified as diseased (Illness manifest).

In the table shown in Figure 50: the first column enumerates the considered area of the leaves, the second column shows the considered area of the leaves (p for plant, f for leaf, r for area, see Figure 26 from more details), the third column shows the days on which the measurement was performed and the fourth column shows the health condition of the considered area of the leaves in the days on which the measurement was performed where 1 indicates that the area has been classified as healthy, 0 indicates that the area has been classified as suffering (Illness progression) and -1 indicates that the area has been classified as diseased (Illness manifest). Please note that the state of suffering corresponds to a classification as suffering of a plant inoculated that may or may not present obvious symptoms of the disease but also where the disease may not have yet taken over. In the case of the experiment, reached the eleventh day all inoculated plants were ill with obvious symptoms of the disease. The idea is to assess how it is possible to detect the disease in its early stage of development. For completeness it is reported that some measure may have been discarded due to some error in the acquisition phase or which, due to the manual work of the procedure, some area has been skipped unintentionally. Can be that it was not possible to perform some measure of ill leaves because the progress of the disease did not allow to go any further.

## 5.4. Multi-spectral image processing

- ✓ Laboratory Methods (5.4.1): description of the implemented image processing algorithms based on the images acquired in laboratory (5.4.1.1), main obtained results, their discussions and conclusions (5.4.1.2).
- ✓ Greenhouse Methods (5.4.2): description of the implemented image processing algorithms based on the images acquired in greenhouse (5.4.2.1), main obtained results, their discussions and conclusions (5.4.2.2).
- ✓ Field Methods (5.4.3): description of the implemented image processing algorithms based on the images acquired in field (5.4.3.1), main obtained results, their discussions and conclusions (5.4.3.2).

### 5.4.1. Laboratory Methods

#### 5.4.1.1. *Diseases detection algorithms*

##### **Camera Electrical Noise Reduction**

To a careful observation of the acquired images (e.g. Figure 51) can be noticed, better on the infrared channel of the image, a series of vertical bands that are most evident on the background panel (Figure 59) and probably caused by a camera electrical noise.



**Figure 51 - Example of the infrared channel from a multispectral acquired image. Bands to be removed are especially visible on the black panel used as background (see Figure 59 for more details).**



Figure 52 – Example of the infrared channel from the multispectral acquired image shows in Figure 52 after Camera Electrical Noise Reduction (see Figure 59 for more details).

Before proceeding further it is therefore necessary to remove the bands (electrical noise) from the three channels. Starting from the upper left of the image, noting the trend of the average of pixel values  $m_i$  along the columns where

$$m_i = \frac{1}{n} \cdot \sum_{j=1}^n p_{i,j} \quad (5.1)$$

is the average of the pixel values  $p_{i,j}$  of the  $i$ -th column computed over  $n$  rows of the image, has confirmed what has been observed (Figure 53 and Figure 55). The calculation is performed along the columns because, as can be seen from the Figure 59, the electrical noise is spread at alternating intervals along the columns of the image.

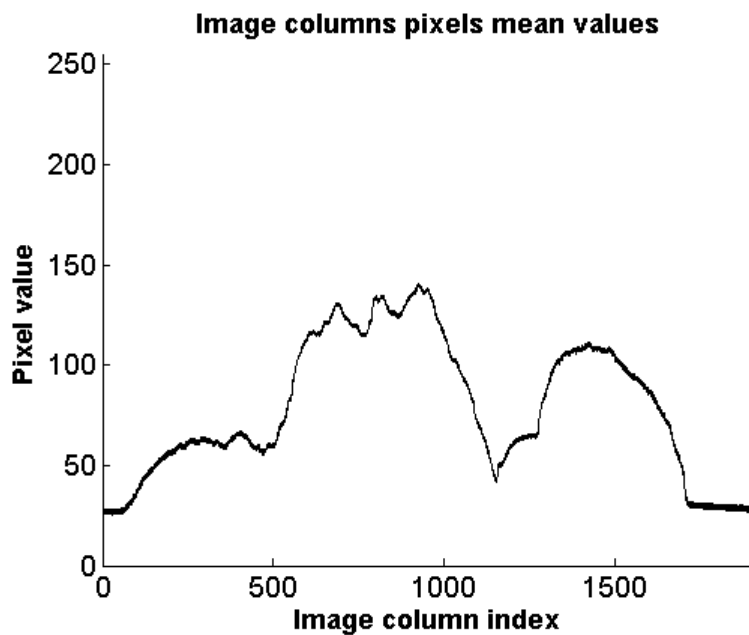


Figure 53 - Means of the pixels along each columns of the image shown in Figure 51 (8 bit image, resolution 1912x1076). See Figure 55 for more details.

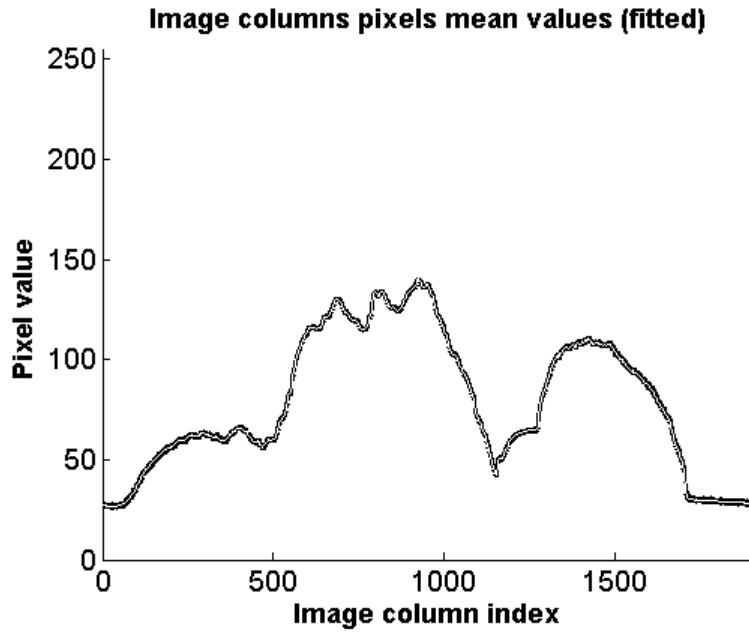


Figure 54 – Fitted means of the pixels along each columns of the image shown in Figure 51 (8 bit image, resolution 1912x1076). Black line represents the pixels mean values show in Figure 53; white line represents the fitted pixels mean values obtained using the algorithm described in the Camera Electrical Noise Reduction section. White line represents the means of the pixels along each columns of the image shown in Figure 52. See Figure 56 for more details.

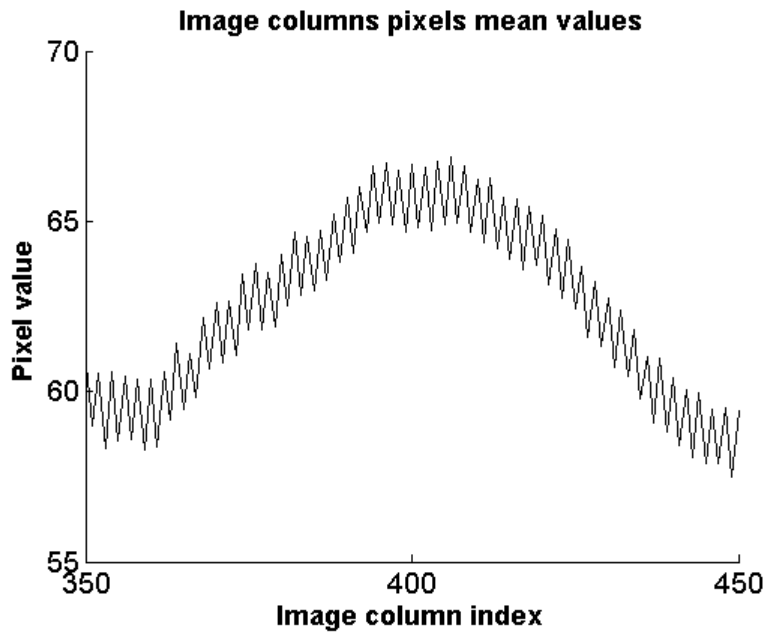


Figure 55 - Detail of Figure 53. You can see in detail the “ups and downs” trend of the pixels mean values along the columns of the acquired image shown in Figure 51.

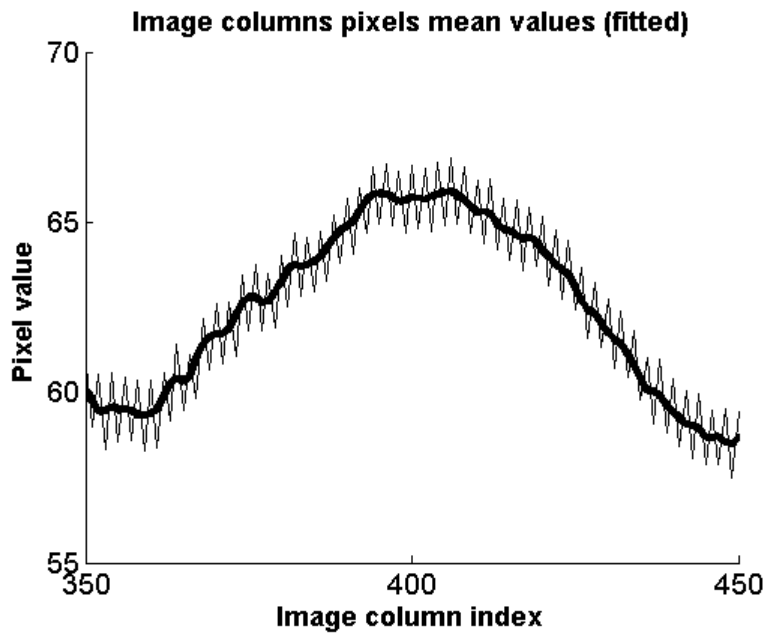


Figure 56 – Detail of Figure 54. The curve fitting obtained of the pixels mean values along the columns of the acquired image shown in Figure 51.

This effect does not occur going to observe the average of pixel values along the lines of the image (Figure 57 and Figure 58).

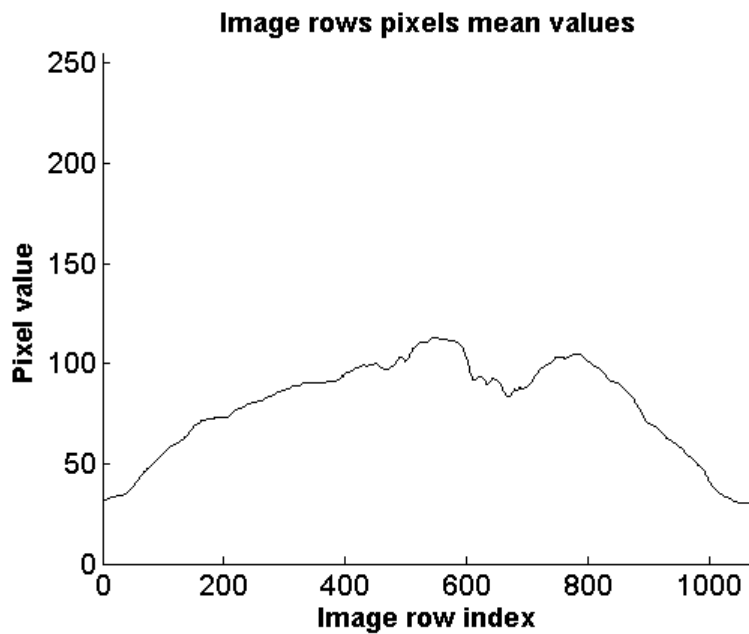


Figure 57 - Means of the pixels along each rows of the image shown in Figure 51 (8 bit image, resolution 1912x1076). See Figure 58 for more details.



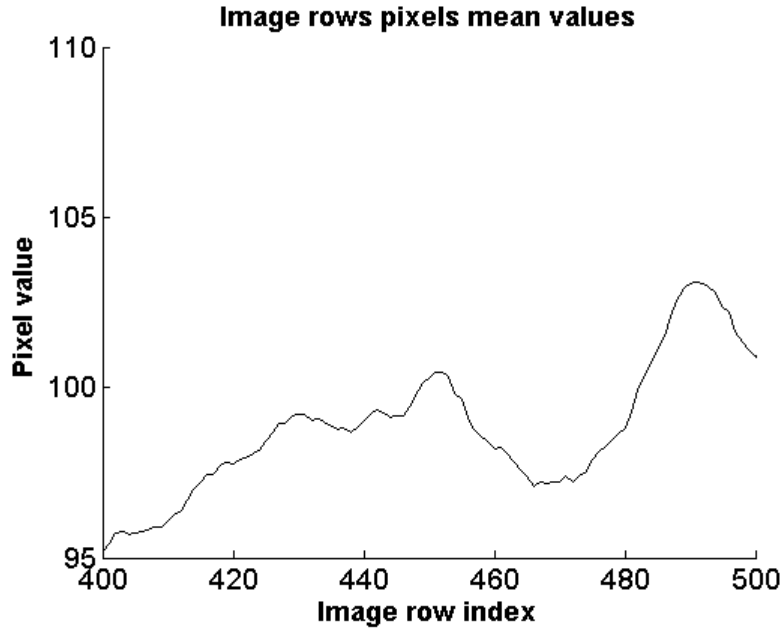


Figure 58 - Detail of Figure 57. It can be seen in detail that the “ups and downs” problem highlighted along the columns of the image shown in Figure 51 does not occur along the rows of the image itself.

What you want to achieve is to smooth the value of pixels along the columns of the image so as to reduce the “ups and downs” effect of the signal (Figure 55) so as to reduce the electrical noise presents in the image. Calculating differences  $d_i$  between the pixels mean values of two consecutive columns:

$$d_i = m_{i+1} - m_i \quad (5.2)$$

the smooth effect was obtained by adding to each pixel of the image half of the difference  $d_i$  between the pixels mean values of the column to which the pixel belongs and the following column:

$$p_{i,j} = p_{i,j} + \frac{d_i}{2} \quad (5.3)$$

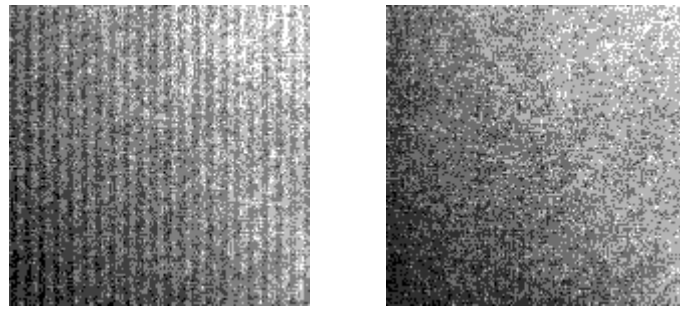
For the pixels  $p_{n,j}$  of the last column is subtracted half of the difference between the pixels mean values of the column which pixels belong and the penultimate column:

$$p_{n,j} = p_{n,j} - \frac{d_{n-1}}{2} \quad (5.4)$$

What you get is a fitting of the pixels mean values of the columns of the image, as shown in Figure 56. What you get is to smooth the signal that would otherwise be very noisy due to “ups and downs” effect along the columns of the image. The image resulting from the removal of the bands is shown in Figure 52.

To better understand the behavior of the algorithm, it is shown below a detail of the background panel in which the contrast has been increased in order to make more visible the vertical bands

present in the image. As you can see, the algorithm eliminates the vertical bands, smoothing the image.



**Figure 59** – Detail of the background panel in which the contrast has been increased in order to make more visible the vertical bands present in the image (left, from Figure 51); the algorithm eliminates the vertical bands, smoothing the image (right, from Figure 52).

### **Background Removal**

Ultimate purpose of the acquisition is to identify the presence or absence of disease on the leaf surface. To do this it is necessary to conduct a series of elaborations in order to limit the area of interest to be analyzed. The first step is to identify and properly remove the background. To do this, two different techniques have been applied, both employing the histogram fitting of the image infrared channel:

- Otsu Threshold
- Expectation Maximization algorithm

The Otsu Threshold method is well known. In computer vision and image processing, Otsu's method is used to automatically perform histogram shape-based image thresholding, or, the reduction of a gray level image to a binary image. The algorithm assumes that the image to be thresholded contains two classes of pixels or bi-modal histogram (e.g. foreground and background) then calculates the optimum threshold separating those two classes so that their combined spread (intra-class variance) is minimal (see section 4.2.2.1 for more details on Otsu Thresholding background theory). To the resulting binary image from the Otsu threshold is applied a closure followed by an opening with a matrix of 7x7 in order to close any holes due to an incorrect thresholding (normally due to those pixels whose values are to be located close to the threshold level). The used Expectation Maximization algorithm is illustrated below (see section 4.1.3.4 for more details on Expectation Maximization algorithm background theory).

Starting from the multispectral acquired image, the three image channels are extracted (red, green and infrared). Taking into account only the infrared channel (Figure 60), we analyze its histogram (Figure 61).

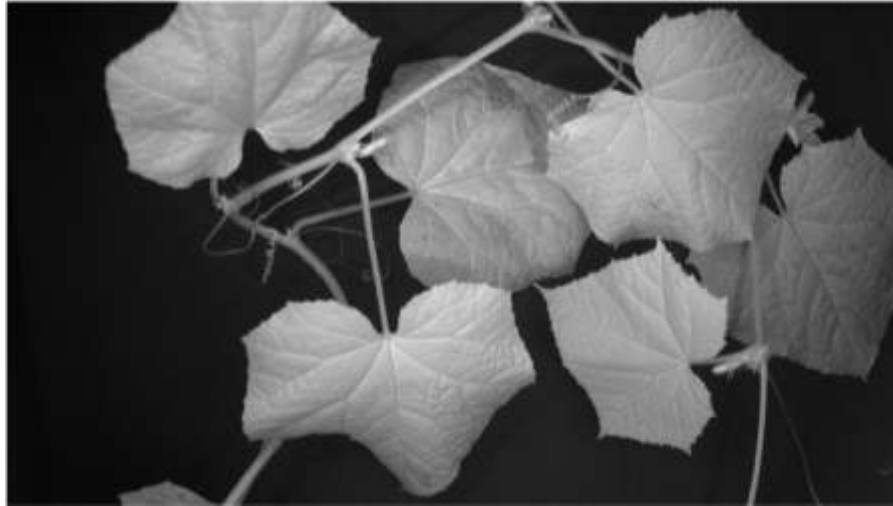


Figure 60 – Example of infrared channel (after Camera Electrical Noise Reduction).

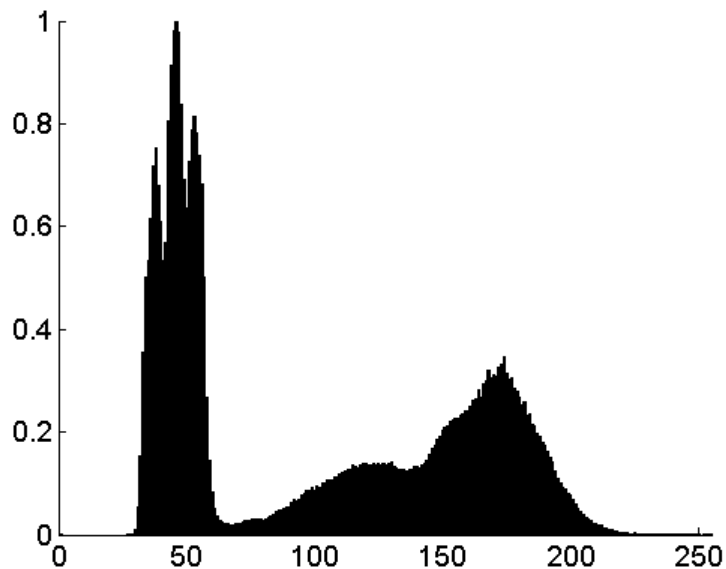
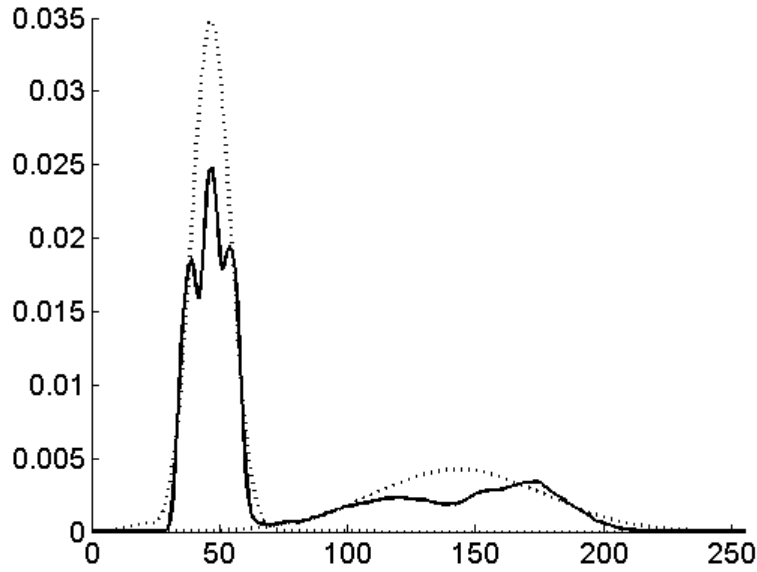


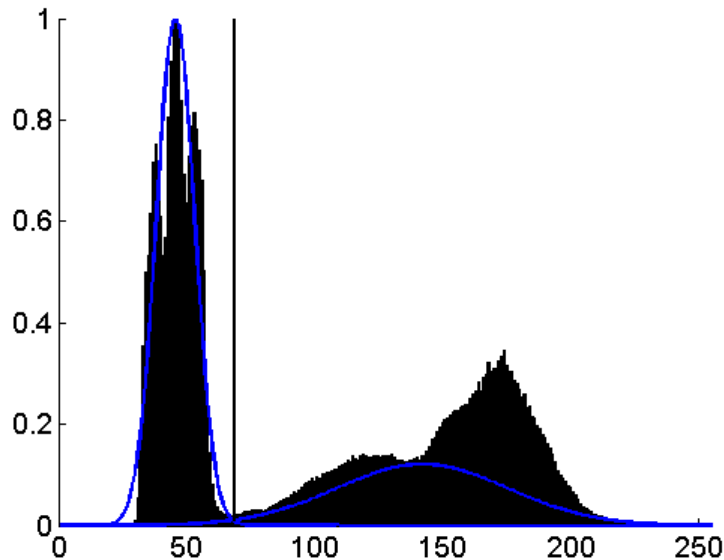
Figure 61 - Histogram of the image shown in Figure 60.

It is evident from the histogram that the values of the pixels relative to the background are concentrated in the first Gaussian, which, in the illustrated case, includes the values ranging from 0 to about 50. The gray levels of the pixels in the image that are not part of the background are arranged along the remaining values in the histogram. The approach implemented is to identify two Gaussians, the first related to the levels of gray associated with the background and the second comprising the remaining gray levels. To do this, you pass the histogram to a function that implements the Expectation Maximization algorithm and that once defined the number of Gaussians to identify (in our case 2), return their mean and variance. Figure 62 shows the histogram of Figure 61 processed (solid black line) and fitted with two Gaussians (dashed black lines).



**Figure 62 - Histogram fitting operated by Expectation Maximization algorithm. The histogram of Figure 61 processed (solid black line) and fitted with two Gaussians (dashed black lines).**

At this point, the implemented algorithm determines the point of intersection between the two Gaussians curve (Figure 63) and uses it as the threshold value to be applied to the infrared image. A binary image is obtained with 1 in correspondence with those pixels having a value greater than the calculated threshold and 0 otherwise. The binary image is applied a closure followed by an opening with a matrix of  $7 \times 7$  in order to close any holes due to an incorrect thresholding (normally due to those pixels whose values are to be located close to the threshold level).



**Figure 63 - Threshold value identified as the point of intersection between two Gaussian curves that fit the histogram of the image. The resulting binary image after thresholding algorithm is shown in Figure 64.**



**Figure 64 - Binary image resulting from thresholding of the image shown in Figure 60 with the threshold value (Figure 63) identified by the Background algorithm.**

#### **5.4.1.2. Results and Discussion**

##### **Test of algorithm accuracy on the validation set**

The algorithms for background removal have been tested on a set of 10 images and validated using the manually painted binary masks (associated with each test image) in order to evaluate their effectiveness (Figure 65). The preparation of the subjects to acquire planned to place a plant of cucumber in the center of black cardboard so that the background (black cardboard) covering the whole area to be acquired. The plants, with leaves of different number and extension, were acquired from a distance of about 50 cm in an attempt to reproduce a similar scenario to what we might have expected in the field (not the concentration of foliage but the size of the leaves (Figure 65)). A further test (a set of 10 images) was performed using more plants placed in a disorderly manner on topsoil in order to verify that with the variation of the amount of organic material and a quite different background the algorithms for the background removal continued to have a good behavior (Figure 66). The background removal algorithms validation has been performed using the manually painted masks associated to each test image. For each test image, the binary mask obtained from the algorithms has been compared (logical pixel comparison) with the manually painted binary mask to get the true positive, true negative, false positive and false negative values associated to each image processing; this values have been used to calculate the Positive Predictive Value, Negative Predictive Value, Specificity and Sensitivity of the algorithms (see the appendix Algorithms Evaluation Method for more details). The background removal algorithms validation results are shown in Table 2 and Table 3.

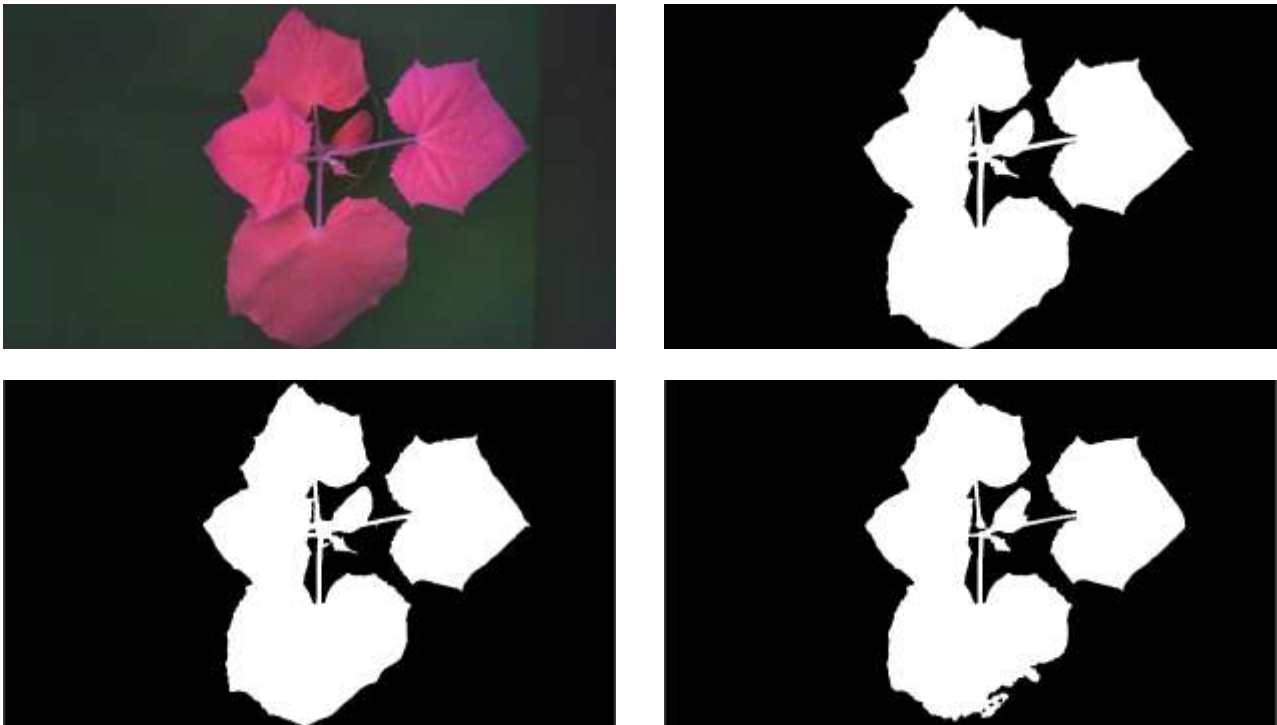


Figure 65 – Example of test image using a black cardboard as a background (upper left), its manually painted binary masks associated (upper right), the Background Removal algorithm result using the Expectation Maximization algorithm (bottom left) and the Background Removal algorithm result using the Otsu Threshold algorithm (bottom right).



Figure 66 – Example of test image using the topsoil as a background where more plants are acquired by multispectral camera (upper left), its manually painted binary masks associated (upper right), the Background Removal algorithm result using the Expectation Maximization algorithm (bottom left) and the Background Removal algorithm result using the Otsu Threshold algorithm (bottom right).

Classifier	Considered Images	Sens.	Spec.	PPV	NPV
Otsu Threshold	10	94%	100%	100%	97.8%
Expectation Maximization	10	97.9%	99.8%	99.5%	99.2%

Table 2 – The background removal algorithms validation results (using a black cardboard as a background). The algorithms for background removal has been tested on a set of 10 images and validated using the manually painted binary masks (associated with each test image, Figure 65). Sens. is for Sensitivity, Spec. is for Specificity, PPV is for Positive Predictive Value and NPV is for Negative Predictive Value (see the appendix Algorithms Evaluation Method for more details).

Classifier	Considered Images	Sens.	Spec.	PPV	NPV
Otsu Threshold	10	93.8%	99.9%	99.9%	89.7%
Expectation Maximization	10	98.3%	98.8%	99.4%	97%

Table 3 – The background removal algorithms validation results (using the topsoil as a background). The algorithms for background removal has been tested on a set of 10 images and validated using the manually painted binary masks (associated with each test image, Figure 66). Sens. is for Sensitivity, Spec. is for Specificity, PPV is for Positive Predictive Value and NPV is for Negative Predictive Value (see the appendix Algorithms Evaluation Method for more details).

## Conclusions

In the case of single plant of black cardboard as a background, the results are good in the case of classification by Otsu Threshold and excellent in the case of classification with Expectation Maximization algorithm (Table 2). In the first case, the classification of the background is partially to the detriment (NPV=97.8% and Sens.=94.0%) of some areas of the leaf which appear to be in shadow or more dark because of the position taken by the leaf itself. No part of the background is incorrectly classified as belonging to the leaf surface (PPV=100% and Spec.=100%). In the second case, the classification of the background is less affected by the shadows or darker area of the leaf so that the background identification is better (NPV=99.2% and Sens.=97.9%). In this case a negligible portion of the background (mostly near the edges of the leaves) is incorrectly classified as belonging to the leaf surface (PPV=99.5% and Spec.=99.8%). In the case of more leaves on topsoil as a background, the result is not so good in the case of classification by Otsu Threshold but stable (good) in the case of classification with Expectation Maximization algorithm (Table 3). You can see the increasing number of false negatives that affect the performance of the algorithm based on the Otsu Threshold (NPV=89% and Sens.=93.8%). This problem is present but negligible in the algorithm based on the Expectation Maximization algorithm (NPV=97% and Sens.=98.3%). For both the error due to the incorrect classification of background pixels as leaf surface is negligible (PPV=99.9 and Spec.=99.9 for the Otsu Threshold based algorithm and PPV=99.4 and Spec.=98.8 for the Expectation Maximization based algorithm).

It can be noted that changing the acquisition scenario, from single plant to a set of leaves and other organic material, does not represent a significantly changes for the algorithms of background removal, especially for that based on the Expectation Maximization algorithm, as you can guess from the pictures shown above (where it is evident the different ratio between the foliage extension and the background, changing from single plant on black cardboard to more leaves on topsoil). The background removal based on the Expectation Maximization algorithm, as it has been described and as we shall see later, will be used both in the greenhouse and in the field (in both cases we will have a dark background panel behind the leaves). While in the greenhouse environment acquisition will be very similar to that of laboratory (a leaf placed in front of a panel of black cardboard), in the field it will be used a sheet of black cloth placed behind the rows of vines. The good results obtained using the topsoil as a background make the background removal based on Expectation

Maximization algorithm suitable for the field, where in the acquired image may be present part of the topsoil at the foot of the grapevine.

## 5.4.2. Greenhouse Methods

### 5.4.2.1. Diseases detection algorithms

A multi-spectral image analysis algorithm for automatic detection of powdery mildew symptoms was designed and tested.

#### Foreground discrimination

The first step is the separation of the leaf tissue from the background. Thanks to the use of a dark background panel in the imaging setup, a bimodal distribution of pixel intensities in the near-infrared channel (low intensities for the background and high intensities for the leaf) is obtained. This allows an accurate foreground segmentation by fitting a Gaussian mixture to the histogram of NIR image pixel intensities. Such fitting has been performed using the Expectation Maximization algorithm described in Background .

#### Healthy/diseased tissue classifier

A preliminary analysis of the pixels intensity distribution in the red, green, near-infrared channels was conducted on the ROIs of the calibration set. The distribution of the pixels intensity in the three channels did not allow a clear discrimination between healthy and diseased areas. Nevertheless, this discrimination can be largely enhanced by using appropriate spectral indices, i.e. algebraic combinations of the pixels intensity in two or more spectral channels, like described previously in the Laboratory tests section – Calibration and validation sets, for more considered wavelengths. The knowledge gained from laboratory measurements is the starting point of the greenhouse measurements (as the results of the greenhouse measurements represent the starting point for field measurements). The indexes, like NDVI, REIP and RIR, used in laboratory measurements have shown the potential to exploit relationships between different wavelengths. In greenhouse (as well as in the field) this potential has been translated in the use of indices carefully studied in order to maximize their discriminating power. For example, NDVI, REIP and RIR highlight the discriminative power of the relationship between red and infrared in the determination of the presence or absence of powdery mildew. In image processing do not have available all the wavelengths which had in laboratory measurements, but, for each pixel, the value of the measured light based on the filter applied to the camera (in the case of multispectral camera used, its optical characteristics). So, starting from the laboratory results, investigating the acquired images pixels distribution and comparing the defined indices with all similar indices automatically generated (like described in Field Methods section – Index validation with index generators) we obtained the following two spectral indices:

$$I1 = \frac{Red_{channel}}{(Red_{channel} + Green_{channel} + Nir_{channel})} \quad (5.5)$$

$$I2 = \frac{Red_{channel} * Green_{channel}}{Nir_{channel}^2} \quad (5.6)$$



The index I1 is the ratio between the red component and the total gray level intensity of the pixel. Being the red channel overlapped to the chlorophyll absorption band, the value of numerator of I1 is expected to be low for healthy tissue, while to become much higher in presence of the whitish filamentous powdery mildew hyphae. The denominator in I1 acts as a normalization term compensating for possible changes in illumination intensity through the image. The index I2 is designed to respond to reflectance variations in either red or green channel. Indeed, any local change (increase) of pixel gray level in one of these two channels increases the value of the numerator of I2, while the denominator acts again as a normalization term, being the NIR intensity quite unaffected by disease symptoms, unless at very advanced stages. In summary, the values assumed by I1 and I2 for healthy pixels are expected to be significantly lower than those assumed by pixels belonging to diseased areas. Figure 67 shows how healthy and disease pixels from ROIs of the training set are mapped into the plane defined by I1 and I2 the indices. As expected, the data corresponding to healthy tissue are gathered around low range values of the two variables, while the cluster corresponding to diseased tissue is more sparse. It can be also noticed that the overlap region of the two clusters is strongly dependent on the view angle. In particular, the separation among healthy and diseased pixels dramatically increases with the angle, mostly due to a gradual shift and spread of the disease pixels towards higher values for both I1 and I2. Healthy and necrotic tissues appear to assume more stationary values in the I1, I2 plane. As clearly shown, powdery mildew symptoms become more and more evident by increasing the view angle of the leaf.

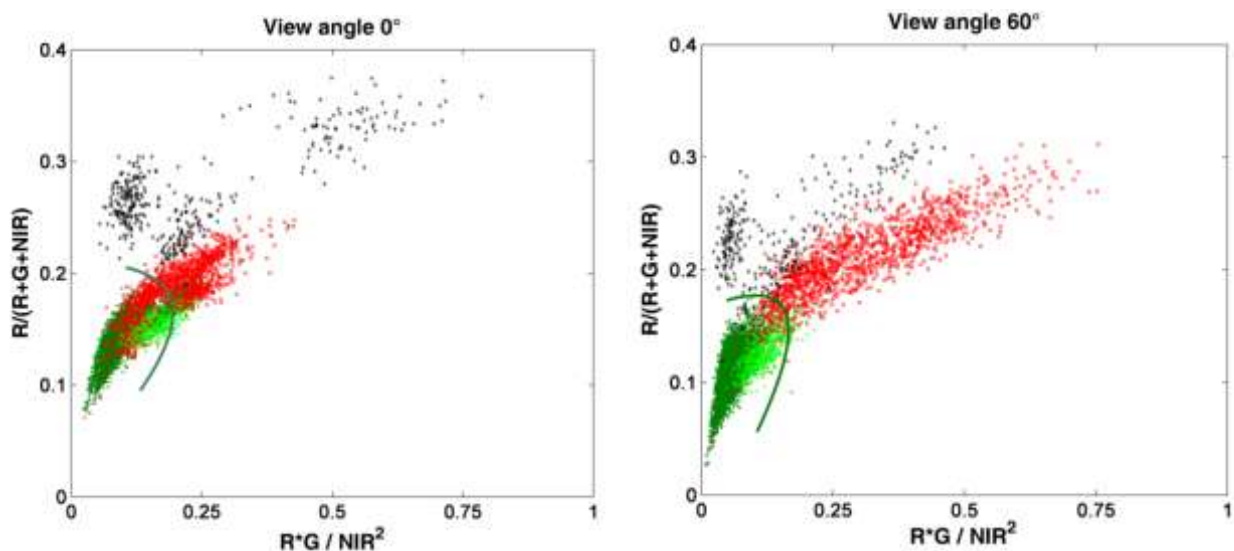


Figure 67 – Charts of cloud of points defined by the indices I1 and I2 (see eq. (5.5 and (5.6 in the text) computed for the pixels ROIs of the training set. Left: distribution of the pixels from images acquired at a view angle of 0°; right: distribution of the pixels from images acquired at a view angle of 60°. Values from healthy tissue are shown in dark green; values from chlorotic or pigmented tissue are shown light green; values from tissue exhibiting powdery mildew symptoms are shown in red; values from necrotic areas are shown in black. The solid line represents the quadratic discriminating function defining the healthy tissue domain for each view angle.

Due to the plume shaped distribution of healthy data, a quadratic function was chosen to discriminate the healthy domain in the I1, I2 plane from the rest of the plane. In particular, a parabola was identified as discriminant function by using the values of the data in the training set. The axis of the parabola was determined as the first eigenvector of the covariance matrix of the healthy values, i.e. as coincident with the major axis of healthy data distribution. The equation of the parabola function was obtained by imposing that the parabola intersects the line through the centroid of the healthy tissue values, oriented as the second eigenvector of the covariance matrix, at

a distance of five times the square root of the corresponding second eigenvalue (i.e. at five standard deviations of the data distribution along the minor axis). Finally, the coordinates of the parabola's vertex were determined iteratively by shifting its position on the principal axis towards the non-healthy data. The iterations were stopped when the number of diseased pixels misclassified as healthy (i.e. false positives falling into the sub-domain defined by the parabola) overcame, somehow arbitrarily, by two times the number of healthy pixels misclassified as diseased (i.e. false negatives falling out from the sub-domain defined by the parabola). To validate the classification method (the construction of the parabola based on the selected indices) the whole dataset has been partitioned using the K-fold cross validation method (with  $K = 10$ ). For each of the five view angles considered in this study, through the cross validation method one discriminating function was determined from the training set and used to classify the validation set. Subsequently with the whole dataset the classification parabola was built and used for automatic pixels classification in the images from their corresponding  $I_1$ ,  $I_2$  value.

### **Algorithm implementation**

The classification indices and the discriminant function were implemented inside a disease detection algorithm. The steps of the algorithm are the following:

1. Camera Electrical Noise Reduction (see section 5.4.1.1 for more details)
2. Background Removal (see section 5.4.1.1 for more details)
3. For each foreground pixel the indices  $I_1$  and  $I_2$  are computed by equations (5.5 and (5.6. If the computed values fall inside the previously defined parabola the pixel is classified as healthy tissue; otherwise, it is classified as diseased tissue. This produces a set of binary regions that are associated to disease tissue.
4. Due to measurement noise, some spurious and often isolated pixels may be misclassified as disease or necrotic tissue. To reduce this phenomenon the binary image is filtered applying a morphological erosion followed by a morphological dilation, both with a  $3 \times 3$  filter box. The binary regions left are assumed as disease or necrotic tissue, while the rest of the image foreground is assumed to correspond to healthy leaf tissue.

The algorithm was applied to the images of the validation set to test its accuracy and sensitivity. The disease symptoms automatically detected by the algorithm were compared with the results of the phytopathological survey, coded as colored blobs as Greenhouse Calibration and validation sets.

#### **5.4.2.2. *Results and Discussion***

##### **Test of algorithm accuracy on the validation set**

From Table 4 to Table 9 are summarized for each considered view angle the results obtained by applying the detection algorithm to the images of the dataset. The total number of pixels involved in the analysis decreased of a factor six when the view angle varied from  $0^\circ$  to  $75^\circ$ , obviously due to the decrease of the visible portion of leaf surface. Nevertheless, the percentage of pixels manually classified as disease resulted fairly independent from the view angle, ranging from 1% to 1.2% of the total foreground. This unbalanced composition of the datasets (99% healthy/background and 1% diseased) reflects what it would be a real operative situation in the field. These values have to be taken into account when the classification statistics is considered.

View Angle	Analyzed pixels (manually classified)		Algorithm classification overall statistics			
	Diseased	Not diseased	PPV	NPV	Sens.	Spec.
0°	25318	2065669	29.8%	99.3%	41.3%	98.8%
20°	21881	1963961	20.8%	99.4%	48.9%	97.9%
40°	17469	1495634	7.1%	99.4%	56.2%	91.4%
60°	9015	843588	36.3%	99.7%	72.1%	98.6%
75°	3592	350210	63.3%	99.6%	57.9%	99.7%

**Table 4 – Results of the classification algorithm over the whole dataset. Here the single infection levels are not considered but the statics is performed based on all classified pixels. Sens. is for Sensitivity, Spec. is for Specificity, PPV is for Positive Predictive Value and NPV is for Negative Predictive Value (see the appendix Algorithms Evaluation Method for more details).**

The total negative predictive value (NPV) of the detection algorithm resulted above 99% for all the view angles, i.e. more than 99% of all the pixels recognized as healthy were indeed corresponding to healthy tissue according to the phytopathological survey. The Specificity (Spec.) resulted above 97% for all the view angles (except for 40° where is 91.4%) indicating that the total amount of healthy tissue is almost entirely correctly classified. The highest NPV value of 99.7% was obtained with the images acquired at 60°, for which the fraction of disease pixels misclassified as healthy was minimized. According to the theory in question, 75° should be more performance when compared to the 60°, as the viewing angle is further inclined. The problem is given by the fact that at 75° part of the leaf appears to be hidden and, with it, the presence of any disease or necrotic regions. This is not so evident in the case of the calculation of NPV (because it assesses the ability to identify the healthy regions) but is highlighted in the calculation of the sensitivity (because it assesses the ability to identify the disease and necrotic regions) where at 75° the resulting classification is clearly lower (57.9%) when compared to what is obtained at 60° (72.1%). The major indicators of the algorithm classification capability are its Sensitivity and its Positive Predictive Value. In our case, for the overall statistics, the Sensitivity measures the fraction of (amount of pixels belonging to) the tissue manually classified as actual diseased or necrotic regions which the algorithm correctly recognized while the PPV measures the fraction of the pixels recognized as diseased (or necrotic) tissue really corresponding to diseased (or necrotic) tissue according to the phytopathological survey. The overall sensitivity of the algorithm resulted to be improved when the view angle was increased, reaching a peak value of 72.1% at an angle of 60°. This again clearly shows that powdery mildew symptoms become more and more detectable by increasing the view angle of the leaf within a practical value. The PPV would seem to indicate a low capacity classification algorithm. Since its value is calculated on the basis of true positives and false positives, despite the increase in the angle favors the correct classification of the disease, the fact of considering all levels of infection involves the deterioration of statistics. This last statement is confirmed by the results obtained by considering the individual levels of infection shown in the following tables.

View Angle	Analyzed pixels (manually classified)		Algorithm classification for IL = 0			
	Diseased	Not diseased	PPV	NPV	Sens.	Spec.
0°	25318	2065669	17.1%	98.9%	6%	99.7%
20°	21881	1963961	5.1%	98.8%	2.3%	99.5%
40°	17469	1495634	1.2%	99%	4.8%	95.7%
60°	9015	843588	9.2%	99.1%	12.2%	98.8%
75°	3592	350210	1.1%	99.2%	0.8%	99.4%

**Table 5 – Results of the classification algorithm over the whole dataset. Here it is shown only the classification algorithm performance for IL = 0 (infection level 0), corresponding to small (1-2 mm in diameter) chlorotic spots likely expected to develop fungal mycelium.**

View Angle	Analyzed pixels (manually classified)		Algorithm classification for IL = 1			
	Diseased	Not diseased	PPV	NPV	Sens.	Spec.
0°	25318	2065669	12.9%	99.1%	6.3%	99.6%
20°	21881	1963961	18.3%	99.3%	14.9%	99.4%
40°	17469	1495634	4.9%	99.3%	22.1%	96%
60°	9015	843588	21.3%	99.4%	32.3%	98.9%
75°	3592	350210	11.2%	99.2%	8.1%	99.5%

**Table 6 – Results of the classification algorithm over the whole dataset. Here it is shown only the classification algorithm performance for IL = 1, small areas (1-2 mm) with hyphae at the initial stage of development (rare and hard to see).**

View Angle	Analyzed pixels (manually classified)		Algorithm classification for IL = 2			
	Diseased	Not diseased	PPV	NPV	Sens.	Spec.
0°	25318	2065669	5%	99.5%	8.6%	99.2%
20°	21881	1963961	1.8%	99.6%	10.5%	97.5%
40°	17469	1495634	0.6%	99.6%	18.0%	86.1%
60°	9015	843588	21.5%	99.9%	72.8%	99.0%
75°	3592	350210	18.6%	99.8%	42.9%	99.3%

**Table 7 – Results of the classification algorithm over the whole dataset. Here it is shown only the classification algorithm performance for IL = 2, areas with clearly visible mycelial structures (both hyphae and conidiophores).**

View Angle	Analyzed pixels (manually classified)		Algorithm classification for IL = 3			
	Diseased	Not diseased	PPV	NPV	Sens.	Spec.
0°	25318	2065669	51.4%	99.4%	61.3%	99.1%
20°	21881	1963961	26.7%	99.6%	73.3%	96.9%
40°	17469	1495634	7.2%	99.6%	81.8%	82.7%
60°	9015	843588	71.7%	100.0%	97.2%	99.4%
75°	3592	350210	82.4%	99.8%	86.6%	99.7%

**Table 8 – Results of the classification algorithm over the whole dataset. Here it is shown only the classification algorithm performance for IL = 3, extended colonies (10-20 mm) with grey-whitish powdery appearance.**

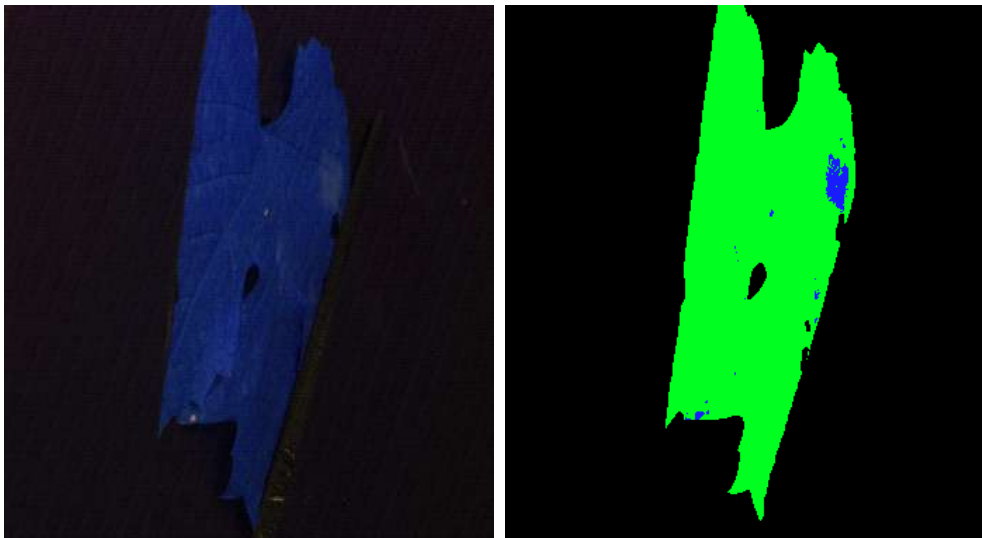
To consider separately the infection levels highlights the following points:

- low levels of infection (IL=0 and IL=1) are difficult to detect;
- a greater view angle does not affect the correct classification for low levels of infection (IL=0 and IL=1) because of the limited presence of visible mycelial structures;
- with increasing presence of visible mycelial structures (IL=2 and IL=3) grows both the ability of the classifier to detect the disease that to limit false positives;
- 60° is the best choice among the considered angles;

When the validation set is separately considered by grouping the different infection levels assigned by the pathologist, the algorithm sensitivity exhibits quite interesting patterns. By restricting the analysis to solely the regions classified as IL=0, corresponding to pre-symptomatic stages of disease, the sensitivity appeared to be very limited (maximum of 12.2% at 60°) and totally uncorrelated to the view angle. On the contrary, when considering data from more advanced infection stages, its value showed a consistent increment as the view angle was increased from 0° to 60°; while at 75° the relationship was less evident, being the leaf visibility partially occluded by the extreme viewing geometry. In detail, for disease data at IL=1 the algorithm sensitivity resulted to be gradually increased from 6.3% at 0° to 32.3% at 60°. At IL=2, corresponding to the presence of more developed fungal structures, the performance increment found with wider view angles was dramatic: from 8.6% at 0° up to 72.8% at 60°. At IL=3, with the view angle set at 60° the algorithm correctly recognized practically all the pixels that were manually classified as disease (Sens.=97.2%). With the same group of data, anyway the detection was quite efficient also at lower view angles with a resulting sensitivity of 61.3%, 73.3% and 81.8% at 0°, 20° and 40° respectively. A similar argument is valid for the PPV where, with the increase of view angle and infection level, false positives remain constant but the increase of true positives increases the performance of the classifier (at IL=2 the algorithm PPV increased from 5% at 0° to 21.5% at 60°, at IL=3 the algorithm PPV increased from 51.4% at 0° to 71.7% at 60°).

In this section are not shown the detailed results regarding the classifications of necrosis regions since the discussion goal was to demonstrate that, with the increase of the level infection (presence of visible mycelial structures), the increase of view angle increases the algorithm capacity to correctly detect the presence of powdery mildew. In support of the fact that the identification of the necrotic regions is view angle independent, we report that the pixels from necrotic lesions (IL=4) were effectively classified by the algorithm with a Sensitivity well above 80% regardless the view angle. The validation of the necrosis search method is present in the discussion about the field methods, where one of the requests is to identify the presence or absence of necrotic areas within the grapevine.

An example of a resulting image after the classification:



**Figure 68** – An example of the acquired image (left) and the resulting image after the classification (right). In black the removed background, in green the detected healthy tissue, in blue the detected diseased and necrotic areas.

## Conclusions

In conclusion from the results obtained it is clear that the 60 ° provide the best overview for the detection of powdery mildew. The percentages of correct classification of the disease leaves no doubt on the effectiveness of the method described. Reduced angle does not use the "benefits" provided by the structure of the disease. Excessive angle does not allow to see the extent of disease and leads to neglect a good part. Of less importance is the classification of the disease in an advanced state (IL = 4), since at this level of infection are functional even smaller angles (0 ° the correct classification already covers 83% of the classified regions) due to the diffusion of disease in this stage.

## 5.4.3. Field Methods

### 5.4.3.1. *Diseases detection algorithms*

#### Leaves

The images used here are RAW images acquired in the field as described previously. No normalization pre-processing is performed. First we proceed to the definition and validation of the classifier based on the ROIs extracted from the acquired images. In a second step we pass to the application of the classifier on the images of the field. To build the classifier, it is necessary to define the classification method. The initial request was to achieve a classifier able to identify the disease in the field (real-time). In the definition of the classifier plays a fundamental role the computation time and then the simplicity of the solution to be detected. Having prepared an appropriate acquisition setup what you want to achieve is a classifier intuitive, as intuitive is the job of the pathologist that based on the simple observation is able to determine whether a plant is ill and its level of infection. The first thing that comes to mind is the analysis of local differences, the first indicator of a change on the leaf surface. A similar approach but that exploits the spectral response of the classes of interest in the various channels of the multispectral camera (thus the variations punctual evaluated on the three channels) is to define the indices ie more or less complex relations between the channels. Once they are identified indices representative will be enough to apply to

run-time image acquired to identify the presence of disease (or identify on the scene items of the classes of interest). In order to justify and make plausible this approach it was decided to consider only one class of indices so to refine the research of the indexes to a well-defined shape. The shape of the indices will be described later, when it will also be described the verification of the same in an exhaustive way. This choice is motivated by the belief that a good acquisition setup (in our case not much bound, consider, for example, that you do not use artificial light but natural light) and the definition of some good indices can lead to good results for regarding the classification of diseases treated. This belief is supported by the analysis of the spectra performed in the laboratory, where during the manual validation we realized that the best results were given by the ratios considered (RIR and REIP indices). The approach that has been followed involves calculating for each ROI the mean and standard deviation of the pixels that compose it so as to define the vector that represents it. Initially, the indices will be defined on the basis of knowledge of the spectral response of the various classes of interest also based on the distribution of means and standard deviations of the ROIs that you will consider when defining indexes. The following chart shows the channels and for each ROIs considered the cloud of points defined by the mean plus or minus the standard deviation.

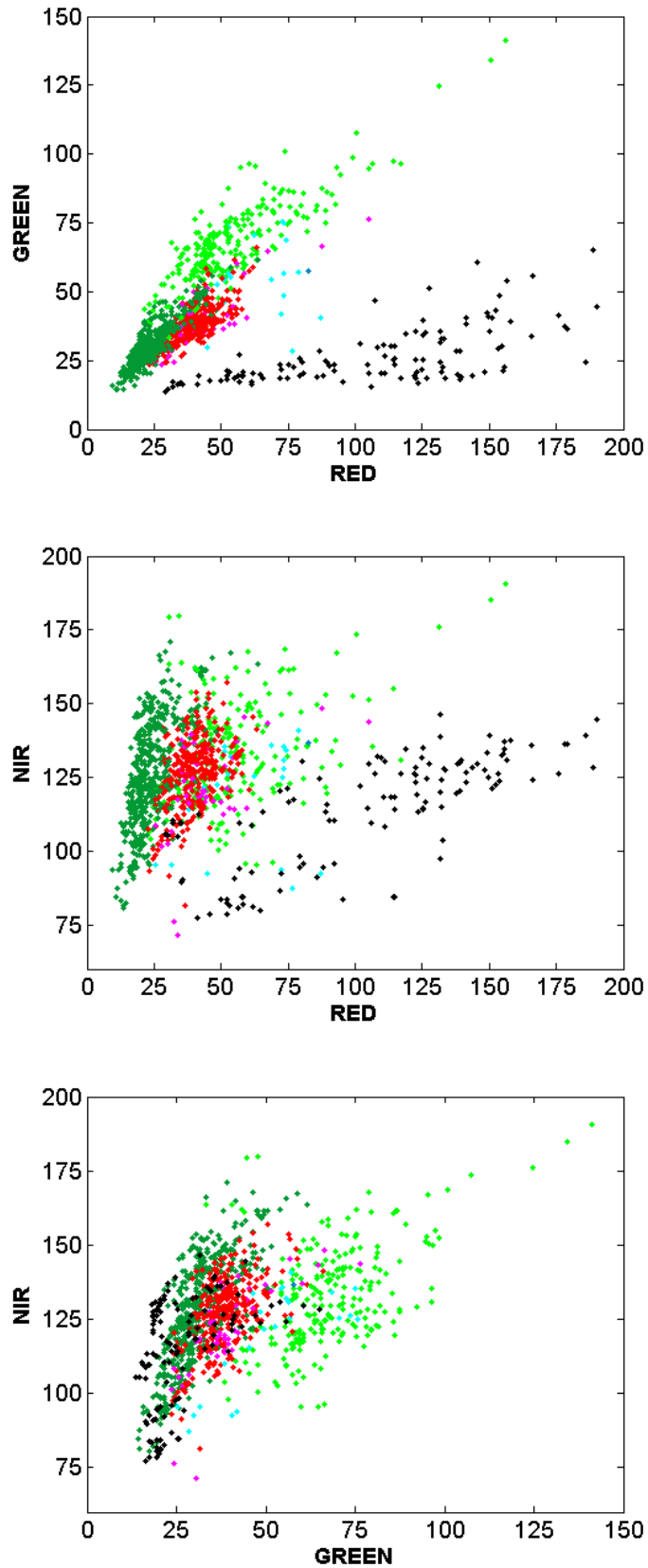


Figure 69 – Charts of cloud of points defined by the means of the extracted ROIs: in dark green the healthy, in light green the healthy young, in red the powdery mildew affected tissue (median diseased), in magenta the powdery mildew affected tissue (advanced diseased), in cyan the downy mildew affected tissue (median diseased), in blue the downy mildew affected tissue (advanced diseased) and in black the necrosis.



Inspection of the graph in the Figure 69, the knowledge base (also resulting from tests carried out in the greenhouse and laboratory) and further verification described in the following paragraph (Index validation with index generators) has led to the definition of the following indices that showed encouraging discrimination power on our training sets. The indices are designed empirically to respond differently to reflectance variations in either red, green and infrared channel.

- For Necrotic case study:

$$I3 = \frac{Green_{channel}}{Green_{channel} + Red_{channel}} \quad (5.7)$$

$$I4 = \frac{Red_{channel} - Green_{channel}}{Red_{channel} + Green_{channel} + NIR_{channel}} \quad (5.8)$$

- For Young case study:

$$I5 = \frac{Red_{channel} - Green_{channel}}{Red_{channel} + NIR_{channel}} \quad (5.9)$$

$$I6 = \frac{Red_{channel} + Green_{channel}}{Green_{channel} + NIR_{channel}} \quad (5.10)$$

- For Downy Mildew case study:

$$I7 = \frac{Red_{channel}}{Red_{channel} + NIR_{channel}} \quad (5.11)$$

$$I8 = \frac{Green_{channel} + NIR_{channel}}{NIR_{channel}} \quad (5.12)$$

- For Powdery Mildew case study:

$$I9 = \frac{Green_{channel} + NIR_{channel}}{Red_{channel} * NIR_{channel}} \quad (5.13)$$

$$I10 = \frac{(Red_{channel} - NIR_{channel})^2}{Green_{channel} - NIR_{channel}} \quad (5.14)$$

The necrotic tissue is characterized by brown color and can be more visible in the green channel where the leaf necrotic tissue appears dark. In the NIR channel the leaf tissue affected by the advanced necrotic symptoms (dead leaf) appears dark. The young leaves is characterized by light green color and can be more visible in the green channel where young tissue appears light. The leaf tissue affected by downy mildew is characterized from the yellow pigmentation more visible in the green channel where it appears light. There is a similar behavior in the red channel where pixel intensity increases with advancing of downy mildew symptoms. Since the red channel overlapped

the chlorophyll absorption band, it is especially expected to respond to local tissue degradation linked to a powdery mildew attack. In the red channel a leaf tissue affected by powdery mildew appear light. Also the relation between the green and red channels gives a powerful discriminant effect with respect to the considered classes. The sum of three channels works as image normalization and can be used to balance the images intensity. In the NIR channel the leaf tissue that does not present necrotic symptoms appears like a white paper so it can be used as a reference for the other channels (like a light environment normalization and shadow reduction). The exponent in the powdery mildew formula is used to increase the relation powerful. There is some problems to distinguish initial symptoms of downy mildew from healthy young leaves due to very similar spectral behavior. Several indices have been chosen because, as will be seen below, the classification is done for successive steps and, from time to time, will, gradually, be excluded from the final classification (disease) the other considered classes.

In order to identify the most significant ROIs for the classes of interest and defined their means and standard deviations, we proceed with the validation of the indices identified. For each ROIs are calculated the indexes identified. Through the process of K-fold cross validation, the ROIs are partitioned into  $K = 10$  partitions and for each of these two classifiers are trained. The first classifier implements a linear discriminant analysis and the second uses the same parabola construction described previously in the Greenhouse Healthy/diseased tissue classifier. For each partition (training and validation) is updated an index of performance of the classifier and the final results are shown in Table 9. Important to note are the subjects of classifications. With the experience gained in the field and the various analyzes carried out, it could be seen that the selected classes are arranged in space favoring a spectral classification of sequential type. The identification of ROIs occurs according to a precise order. Once you have defined a classification, the target class (the class of that specific classification) is removed from the following classifications. The following list will clarify the concept:

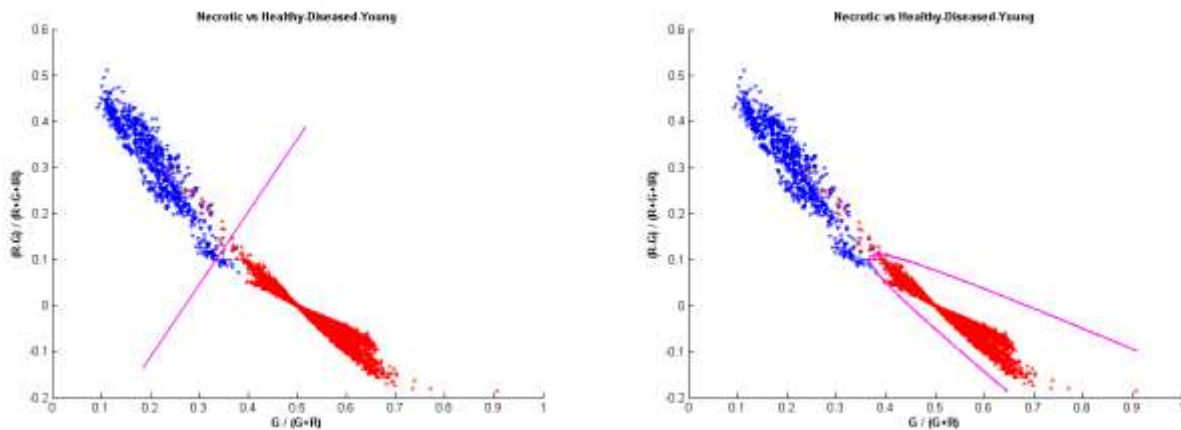
- First classification: Healthy-Diseased-Young vs Necrotic
- Second classification: Healthy-Diseased vs Young
- Third classification: Healthy vs Downy Mildew
- Fourth classification: Healthy vs Powdery Mildew

The sequence of classification will also be used during the classification of the individual pixels of the acquired images. The most evident reason for this choice is given by the necrosis case study. It is sufficient to look at the drawing of the point cloud built around the mean plus and minus standard deviation to realize that it would be possible to begin to remove the necrosis before making any other choice. Follow the young leaves due to their spectral behavior similar to that of advanced diseased, followed by downy mildew and finally by powdery mildew. The classification of powdery mildew is preceded by the classification (and later by the research) of downy mildew because it takes advantage of the ability of the classifier to remove yellowish areas that may occur close to powdery mildew and that may be incorrectly classified.

Classifier	Obs.	Negative Classes	Positive Class	Index 1	Index 2	Sens.	Spec.	PPV	NPV
linear	1666	Healthy-Diseased-Young	Necrotic	$G/(G+R)$	$(R-G)/(R+G+IR)$	99.6%	99.5%	99.5%	99.6%
linear	3262	Healthy-Diseased	Young	$(R-G)/(R+IR)$	$(R+G)/(G+IR)$	92.9%	95.1%	94.9%	93.0%
linear	350	Healthy	Downy	$R/(R+IR)$	$(G+IR)/IR$	99.3%	95.3%	95.4%	99.3%
linear	4872	Healthy	Powdery	$(G+IR) / (R*IR)$	$(R-IR)^2 / (G-IR)$	87.2%	93.2%	92.8%	87.9%
quadratic	1666	Healthy-Diseased -Young	Necrotic	$G/(G+R)$	$(R-G)/(R+G+IR)$	99.7%	99.6%	99.6%	99.7%
quadratic	3262	Healthy-Diseased	Young	$(R-G)/(R+IR)$	$(R+G)/(G+IR)$	92.4%	95.3%	95.2%	92.6%
quadratic	350	Healthy	Downy	$R/(R+IR)$	$(G+IR)/IR$	96.9%	96.6%	96.6%	96.8%
quadratic	4872	Healthy	Powdery	$(G+IR) / (R*IR)$	$(R-IR)^2 / (G-IR)$	87.9%	92.3%	92.0%	88.4%

**Table 9 – Results from the classification of ROIs used to validate the different classifiers. Obs. is for number of observations (number of considered ROIs), Sens. is for Sensitivity, Spec. is for Specificity, PPV is for Positive Predictive Value and NPV is for Negative Predictive Value (see the appendix Algorithms Evaluation Method for more details).**

From the table we can see the strong discriminatory power of the indices considered. In addition to the excellent result achieved in the case of necrosis, are also excellent results obtained in the case of downy mildew. Care should be taken when considering the cases of young leaves and powdery mildew. In the case of the classification of young leaves, it must be remembered that they have a very similar behavior to powdery mildew advanced and therefore a too stringent classification may neglect cases of advanced infection (although we must remember that where there is an infection in the state advanced, at its periphery there is also an infection in the intermediate state). In the case of classification of powdery mildew, as you can see from the graphs that follow, the division of classes is not as clear as we might hope. As is the case in reality, on a leaf is much easier to note the yellowish of downy mildew than the little white powder of powdery mildew. Follow the distributions of ROIs according to the different classifications.



**Figure 70 – Distribution of the indices computed for the pixels from necrotic training sets. Data from necrotic tissue (833 ROIs of necrotic tissue) is shown in blue; the other tissue (10143 ROIs of healthy, diseased and healthy young tissue) is shown in red. On the X and Y axis, the two indices used to define the classifier, respectively,  $G/(G+R)$  on X-axis and  $(R-G)/(R+G+IR)$  on Y-axis. Solid lines represents the discriminating function used as classifier.**

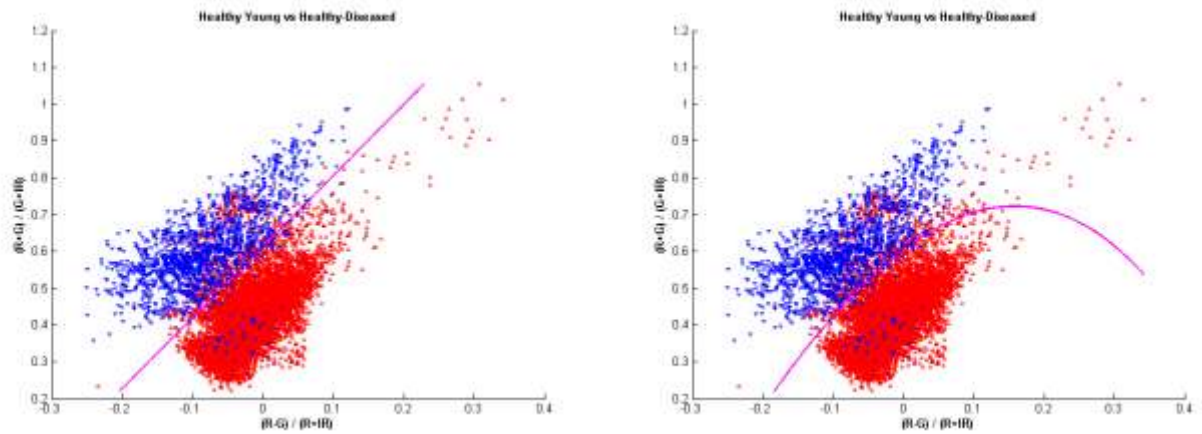


Figure 71 – Distribution of the indices computed for the pixels from young training sets. Data from young tissue (1631 ROIs of young tissue) is shown in blue; the other tissue (8512 ROIs of healthy and diseased tissue) is shown in red. On the X and Y axis, the two indices used to define the classifier, respectively,  $(R-G)/(R+IR)$  on X-axis and  $(R+G)/(G+IR)$  on Y-axis. Solid lines represents the discriminating function used as classifier.

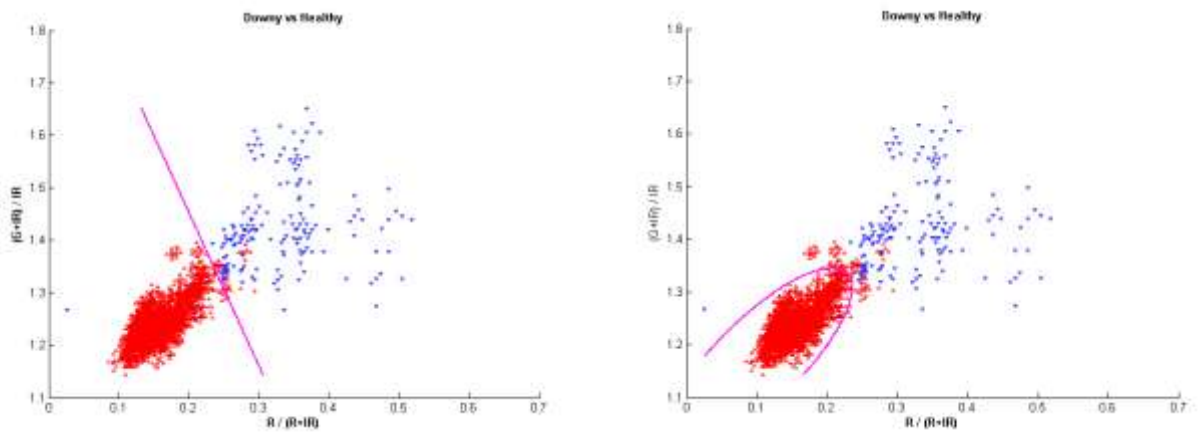


Figure 72 – Distribution of the indices computed for the pixels from downy mildew training sets. Data from diseased tissue (175 ROIs of downy mildew affected tissue) is shown in blue; healthy tissue (3129 ROIs of healthy tissue) is shown in red. On the X and Y axis, the two indices used to define the classifier, respectively,  $R/(R+IR)$  on X-axis and  $(G+IR)/IR$  on Y-axis. Solid lines represents the discriminating function used as classifier.

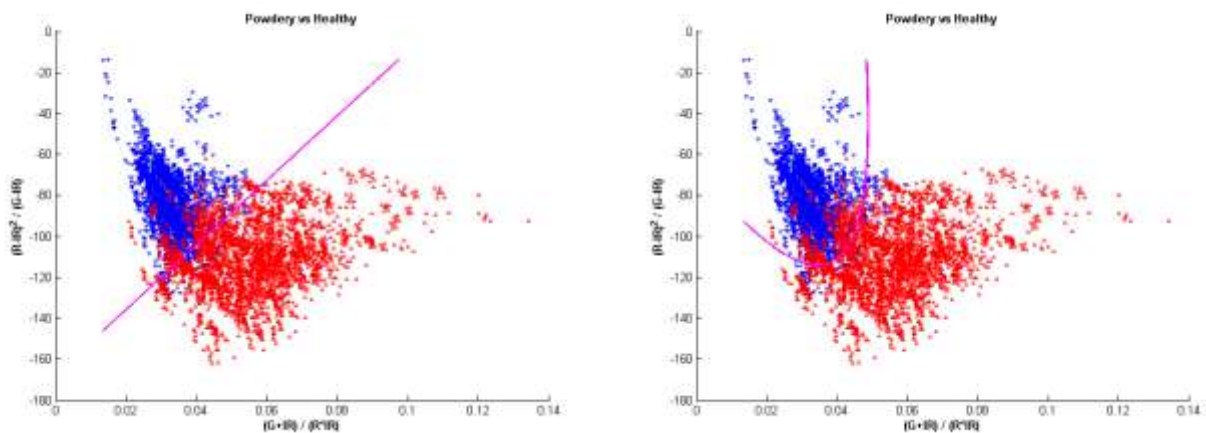


Figure 73 – Distribution of the indices computed for the pixels from powdery mildew training sets. Data from diseased tissue (2436 ROIs of powdery mildew affected tissue) is shown in blue; healthy tissue is shown in red (3129 ROIs of healthy tissue). On the X and Y axis, the two indices used to define the classifier, respectively,  $(G+IR) / (R*IR)$  on X-axis and  $(R-IR)^2 / (G-IR)$  on Y-axis. Solid lines represents the discriminating function used as classifier.

The figures show how pixels from ROIs are mapped into the plane defined by the indices pairs respectively for necrosis (Figure 70), young leaves (Figure 71), downy mildew (Figure 72) and powdery mildew (Figure 73). The data corresponding to healthy and diseased tissue appear to be gathered enough for effective clustering of both the diseases. It is nevertheless evident the possible misclassification problems that pixels from very young leaf tissue (here classified as young leaves) can lead to. The data distribution showed in Figure 70 suggests that necrosis can be easily classified using a linear discriminator function. A similar approach may be used also for young tissue although very advanced downy mildew infection have a similar indices values. However this drawback seems not so critical considering that very advanced downy mildew infection areas will be rounded of median downy mildew infection area detectable by the algorithm. The choice of the indices was made in order to realize a linear classifier. The quadratic discriminant was calculated to make sure that does not introduce significant improvements to the classification system (as well as you can see from the Table 9). The idea is to define the indices whose discriminatory power is such as to divide the plane into two parts within which have, as much as possible in a separated way, the two classes, from time to time, under examination. The Table 9 shows that the quadratic discriminant does not make significant improvements, but rather, in the case of healthy comparison vs downy mildew, decrease in value of sensitivity (from 99.3% of the linear case, to 96.9%, of the quadratic case) and NPV (from 99.3 % of the linear case to 96.8% of the quadratic case). This loss of performance is due to the fact that the quadratic discriminant, so prepared, is not able to exploit all the information available. The classification indices and their corresponding discriminant functions were implemented in image analysis disease detection algorithm. After electrical noise reduction (like describe previously), background removal (using the Expectation Maximization algorithm described previously) and eventually saturated pixel removing, for each foreground pixel the indices are computed, and according the discriminant functions computed above the pixel correspondingly classified. The classification takes place according to the order of application of discriminant functions explained above. In the image to be classified before they are detected and removed necrotic areas, then they are classified and removal the young leaves. At this point the search is performed for downy mildew and finally for powdery mildew. As result of each classification a binary image is obtained. Spurious isolated pixels are filtered by a morphological opening and then identified pixels are removed from the original image. For each image is also carried out research of linear structures in order to remove any false positive such as branches and iron wires. Linear structures identified are removed from the images in which they are identified and labeled with BRANCH. We evaluated the possibility to define a new class to identify the branches but the search for linear structures gave better results and avoided the need to consider the different spectral responses that characterize the branches (which covers a broad spectral response range of wavelengths). The algorithm was applied to the acquired images. A detailed validation tests the algorithm accuracy and sensitivity using the results of the phytopathological survey, coded as colored blobs as described in Field Calibration and validation sets. This validation were performed only with a small set (five images) due to the expensive time required by the manual painting of the infected areas.

#### *Index validation with index generators*

To further validate the choice of selected indexes, it was decided to compare the indices used in the classification stage with:

- all indices of the class:

$$\frac{((channel\_R\ op)^* (channel\_G\ op)^* channel\_IR^*)}{((channel\_R\ op)^* (channel\_G\ op)^* channel\_IR^*)}$$

- some indices generated in a pseudo random manner, according to the rule:

$$\frac{((channel\ op)^* (channel\ op)^* channel^*)^{pow}}{((channel\ op)^* (channel\ op)^* channel^*)^{pow}}$$

where *channel\_R*, *channel\_G* and *channel\_IR* represent the three different channels, *op* one of the four different operations (+, -, \*, /), *pow* the exponentiation (1 or 2), *channel* one of a choice between the three different channels (red, green, near-infrared) or none of these (1). In the first case it is imposed that was present both to the numerator and to the denominator at least one of the channels and that these were different. In the second case it is left to the possibility that to the numerator were not present any channel (numerator equal to 1) or could be present several times the same channel (up to a maximum of three times). The class of indices is chosen to maintain the simplicity of interpretation. The results calculated as described above for the classification of downy mildew and powdery mildew, were quite similar even for more complex indices of those used in the classification algorithm. The indices of both classes taken into consideration have been used to test and validate the algorithm described above, giving worse results and at most equivalent to those obtained with the indexes derived from the spectra analysis carried out in laboratory and with the observation of images acquired in the field. An extract of this classification results for the first class of indices is shown Table 10.

Obs.	Control Classes (Negative)	Index Control Class	Target Class (Positive)	Index Target Class	Sens.	Spec.	PPV	NPV	
1	588	Healthy-Powdery	(R.*G)/(R+G+IR)	Downy	(R.*IR)/R	98.4%	91.1%	92.4%	97.5%
2	588	Healthy-Powdery	G./IR	Downy	(G.*IR)/(R+IR)	99.3%	89.1%	91.1%	98.0%
3	588	Healthy-Powdery	(R.*G)/(G.*IR)	Downy	(R.*IR)/G	100%	86.9%	89.6%	98.9%
4	588	Healthy-Powdery	(R.*G)/IR	Downy	(R.*IR)/R	100%	86.6%	89.5%	99.0%
s	588	Healthy-Powdery	(R+IR)/(G-IR)	Downy	(R.*IR)/R	90.9%	86.5%	89.4%	99.3%
[...]									
12006	588	Healthy-Powdery	R./(R-G)	Downy	(R.*IR)/(R+G)	52.4%	45.3%	49.0%	48.8%
12007	588	Healthy-Powdery	IR./(R-G)	Downy	(R+G)/(R.*IR)	55.4%	39.3%	47.7%	46.8%
12008	588	Healthy-Powdery	(R.*IR)/(R+G)	Downy	(G-IR)/(R-G)	52.0%	44.9%	48.6%	48.4%
12009	588	Healthy-Powdery	(R+G)/(R.*IR)	Downy	(G-IR)/(R-IR)	49.2%	48.4%	48.8%	48.8%
12010	588	Healthy-Powdery	(R.*IR)/(R+G)	Downy	(G.*IR)/(R-G)	51.8%	43.8%	48.0%	47.6%
Obs.	Control Classes (Negative)	Index Control Class	Target Class (Positive)	Index Target Class	Sens.	Spec.	PPV	NPV	
1	1666	Healthy-Diseased-Rear-Young	G./(R+G)	Necrotic	(R.*IR)/G	99.7%	99.8%	99.8%	99.7%
2	1666	Healthy-Diseased-Rear-Young	R./(R+G)	Necrotic	(R.*IR)/G	99.7%	99.8%	99.8%	99.7%
3	1666	Healthy-Diseased-Rear-Young	R./(R+G)	Necrotic	(R.*IR)/(G.*IR)	99.7%	99.8%	99.8%	99.7%
4	1666	Healthy-Diseased-Rear-Young	(R+G)/G	Necrotic	(R-G)/(R+G)	99.7%	99.8%	99.8%	99.7%
s	1666	Healthy-Diseased-Rear-Young	(G.*IR)/(R.*IR)	Necrotic	(R-G)/(G.*IR)	99.7%	99.8%	99.8%	99.7%
[...]									
12006	1666	Healthy-Diseased-Rear-Young	(R+G)/(R-IR)	Necrotic	(G+IR)/(R-G)	41.6%	57.2%	49.3%	49.5%
12007	1666	Healthy-Diseased-Rear-Young	(G-IR)/(R-G)	Necrotic	(G-IR)/(R-IR)	37.4%	62.9%	50.2%	50.1%

12008	1666	Healthy-Diseased-Rear-Young	$(R+G)/(R.*G)$	Necrotic	$(R+IR)/(R.*G.*IR)$	40.9%	56.5%	48.4%	48.9%
12009	1666	Healthy-Diseased-Rear-Young	$(G+IR)/(R-IR)$	Necrotic	$(R-IR)/(R-G)$	38.0%	61.1%	49.4%	49.6%
12010	1666	Healthy-Diseased-Rear-Young	$IR/(R-IR)$	Necrotic	$(R+G)/(R-G)$	34.8%	59.3%	46.1%	47.6%
	<b>Obs.</b>	<b>Control Classes (Negative)</b>	<b>Index Control Class</b>	<b>Target Class (Positive)</b>	<b>Index Target Class</b>	<b>Sens.</b>	<b>Spec.</b>	<b>PPV</b>	<b>NPV</b>
1	4872	Healthy	$(R+IR)/IR$	Powdery	$(R.*G)/(R+G+IR)$	92.9%	86.8%	87.5%	92.4%
2	4872	Healthy	$(R.*G)/(G.*IR)$	Powdery	$(R.*G)/(R+G+IR)$	92.9%	86.8%	87.5%	92.4%
3	4872	Healthy	$(R+IR)/IR$	Powdery	$(R.*G)/(R+IR)$	92.9%	86.8%	87.5%	92.4%
4	4872	Healthy	$(R+IR)/IR$	Powdery	$(G.*IR)/(G-IR)$	93.1%	86.5%	87.3%	92.6%
s	4872	Healthy	$R/IR$	Powdery	$(G.*IR)/(G-IR)$	93.1%	86.4%	87.3%	92.6%
[...]									
12006	4872	Healthy	$(R.*G)/(R-G)$	Powdery	$(R.*IR)/R$	48.8%	46.1%	47.5%	47.4%
12007	4872	Healthy	$(R+G)/(R-G)$	Powdery	$(G.*IR)/G$	48.7%	46.2%	47.5%	47.4%
12008	4872	Healthy	$IR/(R-G)$	Powdery	$(G.*IR)/G$	48.7%	46.1%	47.4%	47.3%
12009	4872	Healthy	$R/(R-G)$	Powdery	$(G.*IR)/G$	48.6%	46.1%	47.4%	47.3%
12010	4872	Healthy	$(R+IR)/(R-G)$	Powdery	$(R.*IR)/R$	48.7%	46.0%	47.4%	47.3%
	<b>Obs.</b>	<b>Control Classes (Negative)</b>	<b>Index Control Class</b>	<b>Target Class (Positive)</b>	<b>Index Target Class</b>	<b>Sens.</b>	<b>Spec.</b>	<b>PPV</b>	<b>NPV</b>
1	3262	Healthy-Diseased-Rear	$R/(R+G+IR)$	Young	$G/(R+G+IR)$	92.8%	96.0%	95.8%	93.0%
2	3262	Healthy-Diseased-Rear	$(R+IR)/(R+G+IR)$	Young	$(G+IR)/(R.*IR)$	92.8%	95.9%	95.8%	93.0%
3	3262	Healthy-Diseased-Rear	$R/G$	Young	$(G-IR)/(G+IR)$	92.8%	95.9%	95.8%	93.0%
4	3262	Healthy-Diseased-Rear	$R/(G+IR)$	Young	$(R+IR)/(G+IR)$	93.2%	95.6%	95.5%	93.3%
s	3262	Healthy-Diseased-Rear	$(R+G)/(G+IR)$	Young	$(R-G)/(R+G+IR)$	93.1%	95.5%	95.4%	93.3%
[...]									
12006	3262	Healthy-Diseased-Rear	$(G+IR)/(R-G)$	Young	$(G.*IR)/(R-G)$	22.3%	71.0%	43.5%	47.7%
12007	3262	Healthy-Diseased-Rear	$(R+G)/(R-G)$	Young	$(R.*G)/(R-G)$	35.8%	43.9%	38.9%	40.6%
12008	3262	Healthy-Diseased-Rear	$IR/(R-G)$	Young	$(R.*G)/(R-G)$	32.6%	45.3%	37.3%	40.2%
12009	3262	Healthy-Diseased-Rear	$G/(R-G)$	Young	$(R.*G)/(R-G)$	32.7%	43.9%	36.8%	39.5%
12010	3262	Healthy-Diseased-Rear	$(R.*G)/(R-G)$	Young	$(G.*IR)/(R-G)$	37.5%	28.3%	34.3%	31.1%

**Table 10** – An extract of the classification results for the first class of indices. Obs. is for number of observations (number of considered ROIs), Sens. is for Sensitivity, Spec. is for Specificity, PPV is for Positive Predictive Value and NPV is for Negative Predictive Value (see the appendix Algorithms Evaluation Method for more details). The dataset used to test the indices is the same used to test the selected indices shown previously (training ROIs). The used discrimination function is computed by linear discriminant analysis. The order used to show the index evaluation results is based on the selected criteria: increasing PPV, increasing Sensitivity, etc. The table shows the results ordered by increasing value of Sens.+Spec. to balance the classifier capability to detect target and control classes and then PPV to reduce the false positive.

### Grapes

The first step of the algorithm is the removal of the markers (not used here). The search for markers can be made over the whole image, but if their position is known, the search can be made only on a portion of the image, reducing the computation time and increasing accuracy of the identification. Depending on the type of image (zoomed or not zoomed) varies the size of the markers to detect. Through the circular Hough transform identifies the candidate markers. Candidates with the same center and radius will be held one. Among the remaining candidates are taken, the two markers with the lowest internal variability considering the pixels that compose them (since the surface of the markers is homogeneous). At this point, is removed a slice of the image such as to contain the markers and the bar on which they are attached. The second step is the identification of bunches. To facilitate the identification is used an image obtained from the (empirical) composition of the channels of the original image. The composition used is as follows:

$$I11 = \frac{NIR_{channel}^2}{Red_{channel} * Green_{channel}} \quad (5.15)$$

The Figure 74 shows the resulting image.



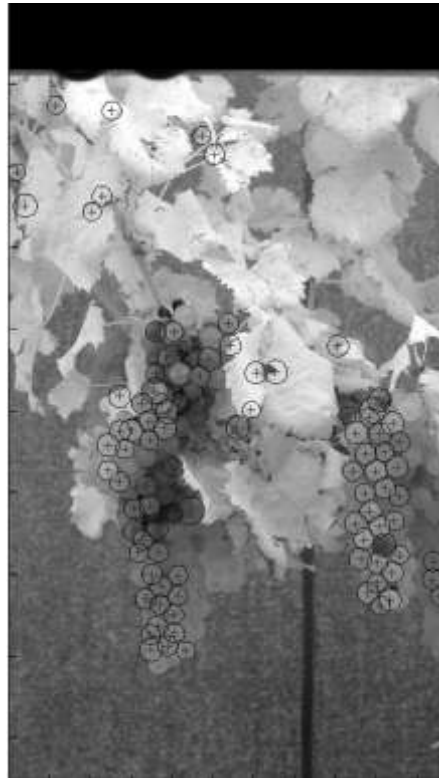
**Figure 74 – Resulting image from index I11.**

This image show in Figure 74 was used as input to a search algorithm of circular shapes based on the circular Hough transform (third step, see section 4.2.8.1 for more details on circular Hough transform background theory). The inputs of the algorithm are in order:

- the image to be processed
- the range of possible size of the berries (for the type of image used and its resolution [15, 40] pixels)
- the threshold magnitude of the gradient (left to default for 8-bit images)
- the radius of the filter used in the search of local maxima in the accumulation array (set as the minimum radius of a berry)
- the tolerance of picking up the likely radii values (set to 1 to search for precise circular shapes without duplicates)
- the filter used to smooth the accumulation array (an average filter dimensioned on the minimum radius of a berry).



The Figure 75 shows the resulting of circular shapes search (for reasons of clarity showed over the IR channel).



**Figure 75 – IR Channel Image with detected bunches (center positions and circumference marked).**

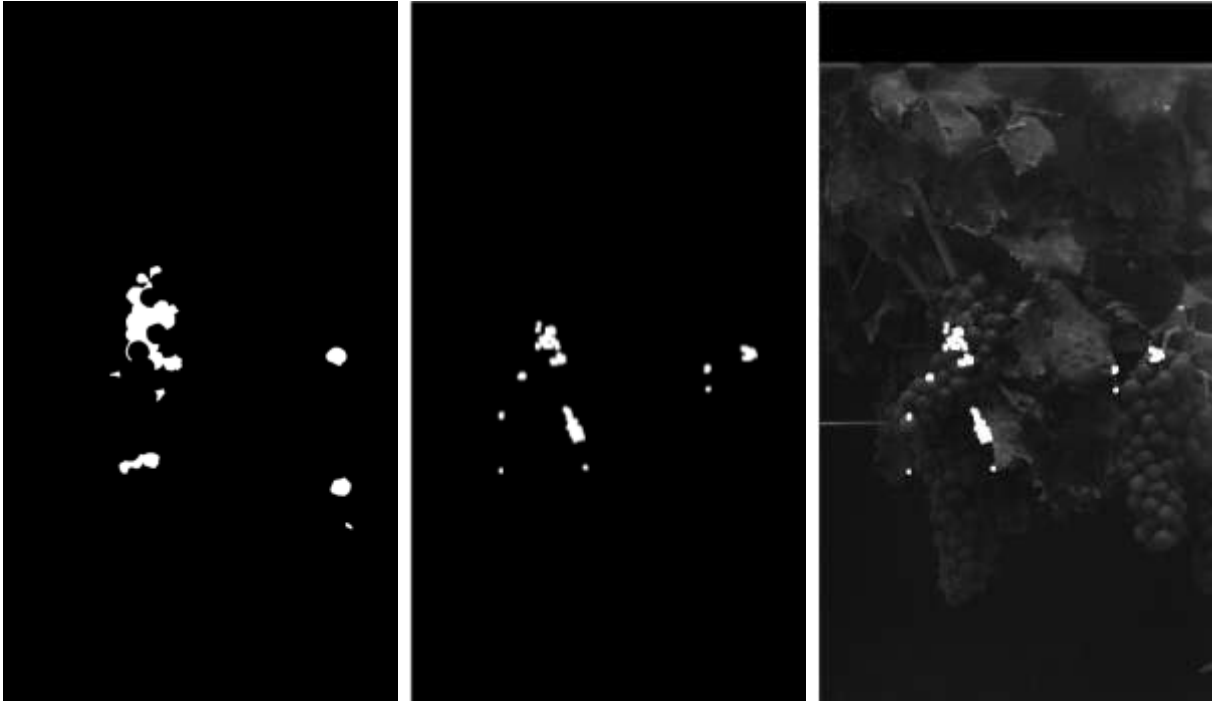
The fourth step is the definition of the bunches of grapes according to the position of the identified berries. The maximum distance to consider two berries belonging to the same cluster is twice the maximum radius of a berry. At this point, in order to remove any false positives, it is considered bunch each grouping of at least five berries (Figure 76).



Figure 76 – Detected bunches on Red, Green and Infrared channels.

Finally, it defines two approaches to research of botrytis.

*Botrytis detection using local variance:* Defines an image of local variability on the part of the image containing bunches identified. We chose to use the local variance as it is a good discriminator between healthy leaf tissue (homogeneous) and regions affected by botrytis (patchy). As we will see only the local variance will not be enough because of the possible presence of necrotic areas (very similar to botrytis). The size of the locality is given by the minimum radius of a berry. Thresholding using Otsu algorithm is applied to the image of local variability. An erosion followed by a dilation is applied to the resulting image to remove the detected region smaller than a berry. The results of the algorithm is showed in Figure 77.

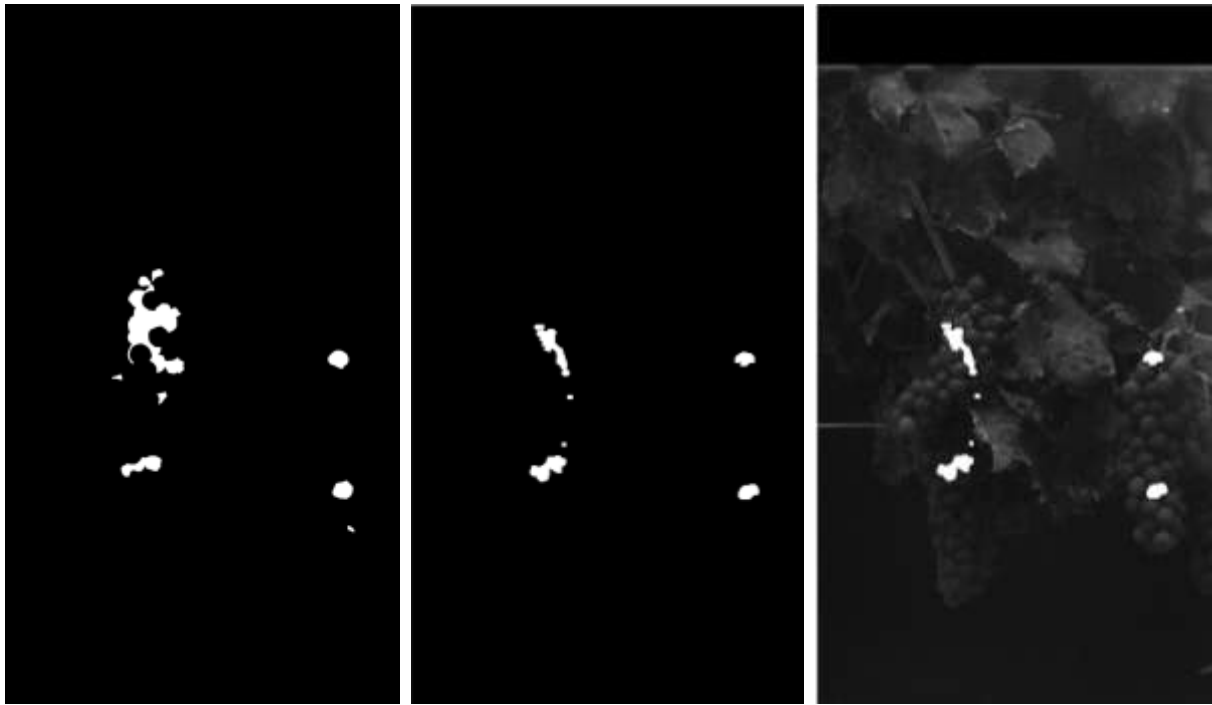


**Figure 77 – The mask of botrytis to identify (left), the mask image of botrytis detected using the local variance (center) and the resulting images overlapping (right).**

*Botrytis detection using spectral index:* Exploiting the spectral characteristics highlighted by the images acquired, we define an image index as follows:

$$I_{12} = \frac{\text{channel\_image\_red}}{\text{channel\_image\_infrared}^2} \quad (5.16)$$

The index definition is derived from the following observations. In the red channel, the disease presents with white patches in contrast to the healthy tissue that appears dark. In the infrared channel the disease presents dark in contrast to the healthy tissue that appears white. Indeed, being the red channel overlapped to the chlorophyll absorption band, this is especially expected to respond to local tissue degradation linked to a pathogen attack. Thresholding using Otsu algorithm is applied to the image index. An erosion followed by a dilation is applied to the resulting image to remove the detected region smaller than a berry. As you can see the spectral index performs better than the local variance, reducing the number of false positives (necrotic areas). The results of the algorithm is showed in Figure 78.



**Figure 78 – The mask of botrytis to identify (left), the mask image of botrytis detected using the spectral index (center) and the resulting images overlapping (right).**

Observing the two images can be seen that the technique which exploits the local variance classifies as botrytis also a part of necrosis. This erroneous classification is due to the fact that, in the necrotic area, the leaf appears crumpled. This leaf, in evident state of necrosis, results in high local variability. Instead, in the case of the algorithm that uses the spectral index, the problem does not occur so that the necrosis is not classified as botrytis. As we shall see from the results, not all of the botrytis is identified, but the technique based on the spectral index will be more efficient.

#### **5.4.3.2. Results and Discussion**

##### **Test of algorithm accuracy on the validation set**

###### Leaves

Figure 79 shows two results examples obtained by applying the detection algorithm to two field images from powdery mildew and downy mildew sets, respectively. The two images were painted in order to show the correct detected areas, True Positive (TP) and True Negative (TN), respectively disease and healthy tissues well classified, and also False Positive (FP) and the False Negative (FN), respectively disease and healthy tissues wrong classified. Figure 80 and Figure 81 show some details of the classification made by the algorithm.

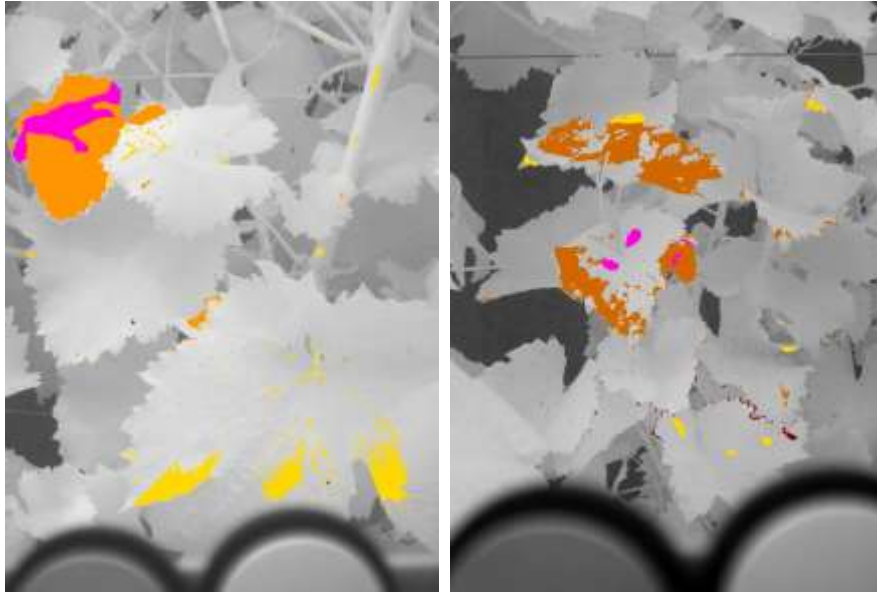


Figure 79 – Classification results of the disease detection algorithm on field images of grapevine canopy locally exhibiting powdery mildew (left) and downy mildew (right) symptoms. Uncolored regions represent healthy tissue correctly classified. Regions in orange indicate correctly detected diseased tissue, while in magenta are undetected diseases spots. In yellow are false positives, i.e. healthy tissue misclassified as mildew.

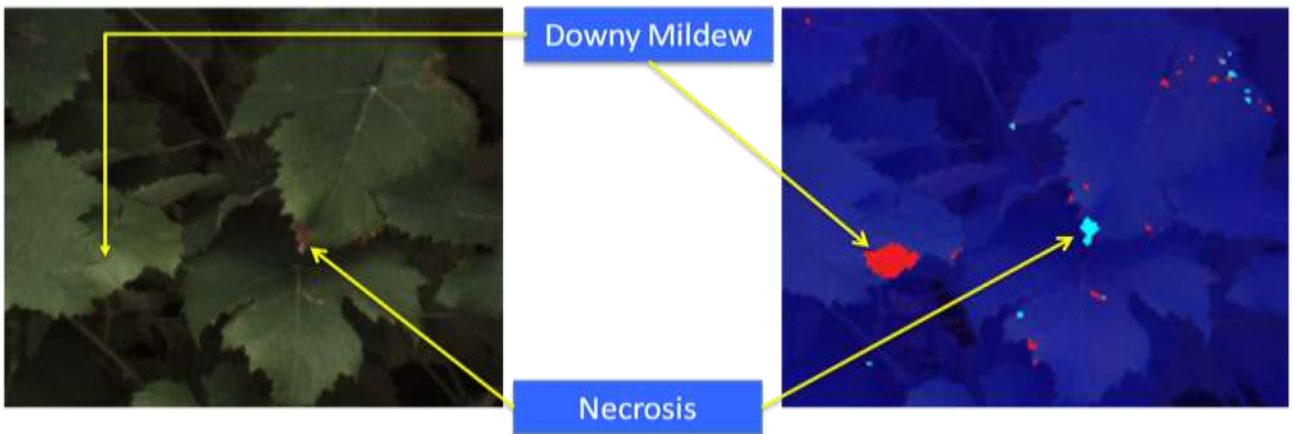


Figure 80 – Example of section of an acquired R-G-B image for documentation (left) and an acquired R-G-NIR image (right). In R-G-NIR image the red colored areas represent the downy mildew and the cyan colored areas represent the necrosis correctly detected by the classifier.

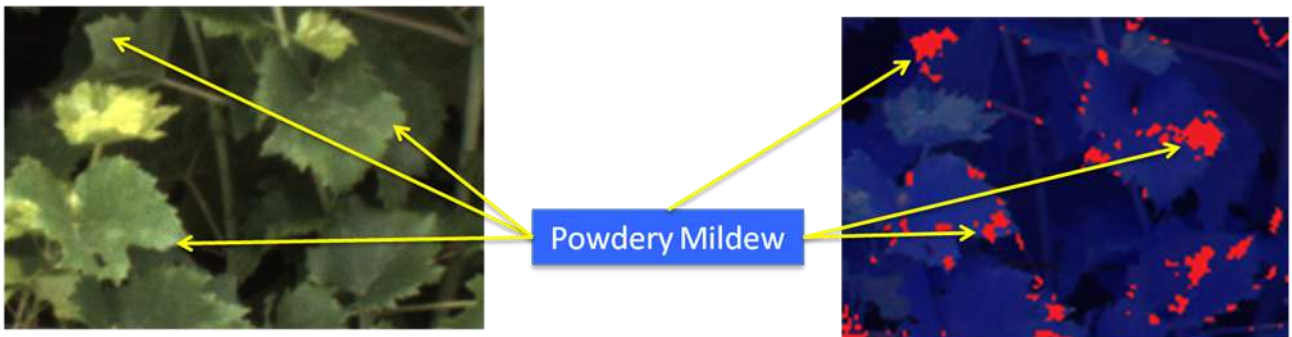


Figure 81 – Example of section of an acquired R-G-B image for documentation (left) and an acquired R-G-NIR image (right). In R-G-NIR image the red colored areas represent the powdery mildew correctly detected by the classifier.

Below there are reported some statistics for Downy Mildew case study.

Classification results calculated for pixels from the ROIs of dataset and belonging to different classes.

**Algorithm classification accuracy on ROIs for distinct degrees of infection**

Initial Disease	Intermediate Disease	Advanced Disease
33.33%	89.47%	95.45%

**Table 11 – Algorithm classification accuracy on ROIs for distinct degrees of downy mildew infection.**

Classification results calculated for pixels from manually painted area on images of dataset and belonging to different classes.

**Algorithm classification statistics for distinct degrees of infection**

Initial Disease			Intermediate Disease			Advanced Disease		
PPV	NPV	Sens.	PPV	NPV	Sens.	PPV	NPV	Sens.
73.2%	84.6%	11.3%	77.9%	96.1%	42.5%	69.8%	98.1%	58.4%

**Table 12 – Algorithm classification statistics for distinct degrees of downy mildew infection. Sens. is for Sensitivity, PPV is for Positive Predictive Value and NPV is for Negative Predictive Value (see the appendix Algorithms Evaluation Method for more details).**

### Grapes

Table 13 summarize some preliminary results obtained with the validation of the algorithm for grapes detection.

**Resulting Validation Indices for grapes detection**

PPV	75%
NPV	97.3%
Sens.	76.8%

**Table 13 – Resulting Validation Indices for grapes detection (fourth step of the algorithm). Sens. is for Sensitivity, PPV is for Positive Predictive Value and NPV is for Negative Predictive Value (see the appendix Algorithms Evaluation Method for more details).**

Table 14 summarize some preliminary results obtained with the validation of the algorithm for botrytis detection using local variance technique.

**Resulting Validation Indices for Botrytis**

PPV	35.6%
NPV	99.6%
Sens.	22.5%

**Table 14 – Resulting Validation Indices for Botrytis (fifth step of the algorithm) using the local variance technique. Sens. is for Sensitivity, PPV is for Positive Predictive Value and NPV is for Negative Predictive Value (see the appendix Algorithms Evaluation Method for more details).**

Table 15 summarize some preliminary results obtained with the validation of the algorithm for botrytis detection using the image index.

Resulting Validation Indices for Botrytis	
PPV	59.6%
NPV	99.7%
Sens.	41%

Table 15 – Resulting Validation Indices for Botrytis (fifth step of the algorithm) using the image index. Sens. is for Sensitivity, PPV is for Positive Predictive Value and NPV is for Negative Predictive Value (see the appendix Algorithms Evaluation Method for more details).

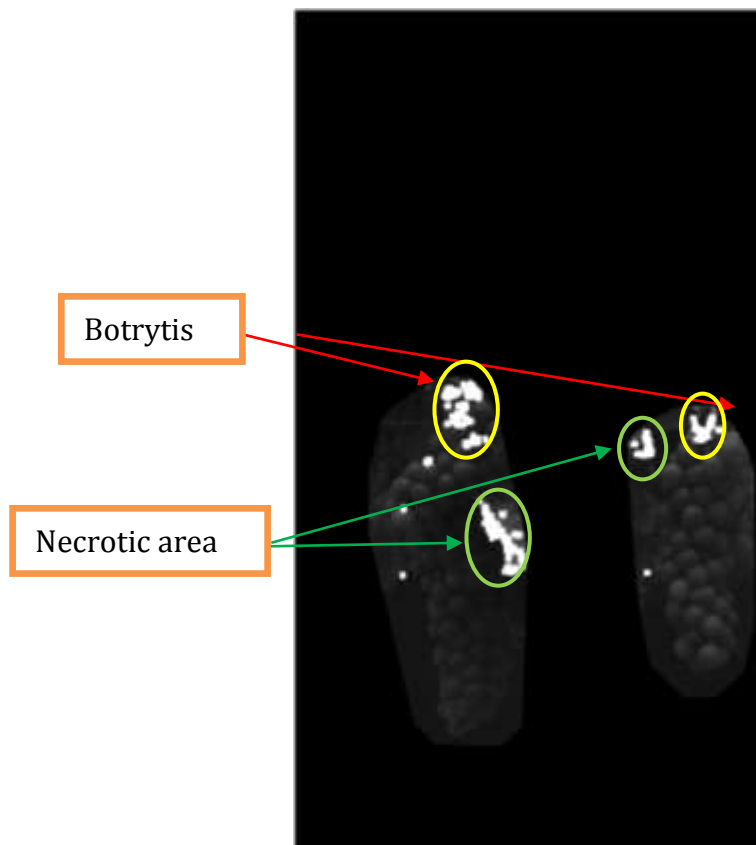
## Conclusions

### Leaves

The proposed classifiers appeared to fairly detect medium to advanced intensity symptoms for both the considered diseases. Nevertheless, the misclassification problems created by false positives appear to be crucial especially when working on young leaves and the rear of the leaves. Classification robustness reinforcement is also expected by the adoption of some new techniques like the texture approach. The acquisition of the field proved to be more difficult than in the greenhouse. The optimization of the acquisition setup has led many benefits including the elimination of saturated zones and the reduction of specular reflections. It has also allowed us to compare images taken with different lighting conditions without the need for additional pre-processing. Some problems have emerged, not so much in the classification of diseases (both the classifications of downy mildew and powdery mildew obtained good results) as to the occurrence of some false positives. False positives (classified as diseased regions of the captured image but that does not really belong to any of the classes related to disease) are represented by the back of the leaves. The back of the leaves have a spectral response that is distributed over a wide range of wavelengths in fact going to position itself without distinction to each of the above classes of interest and especially (also because of specular reflections) above to the disease. To further improve the result of the classification what could be done (and which is being studied) is to insert a component based on the texture of the back of the leaf. Exploiting the texture of the back of the leaf should be possible to further discriminate the pixels classified in order to remove the false positives from the classification of the disease and thus obtain the complete coverage of test cases so far identified.

### Grapes

As shown is a first approach to the definition of an algorithm for automatic detection of botrytis in the field. The identification of bunches appears to be fairly robust. What is proposed for the identification of botrytis marks the direction of the approach to be followed based on the knowledge gained so far of the disease. In the image resulting from the botrytis detection technique using local variance you can see how the botrytis has been identified. You can also see how some of the leaf showing necrotic areas erroneously classified (Figure 82). The result achieved suggests the use of an new classifier also able to distinguish the areas affected by botrytis from the leaf necrotic area. The misclassification is due to the similar behavior in the spectral bands of light used for images acquisition, hence the choice of using a classifier based on a spectral index. The use of a classifier based on spectral index gives better results, reducing the number of false positives but not eliminating them altogether. The next step can be the simultaneous use of the two classifiers and a more detailed spectral analysis to identify those components necessary for the discrimination between necrotic areas and areas affected by botrytis.



**Figure 82 – Resulting images from the botrytis detection technique using local variance overlapping with botrytis and false positive (necrotic areas) in evidence.**

### *Endnotes*

To detect the clusters was also possible to use different techniques. The colorimetric approach, using an R-G-B camera, was excluded as it depends on the type of grape to be detected (white or red). The use of the infrared channel allows to identify with good accuracy the healthy tissue and to obtain the spectral indices more powerful than those that would be obtained with an R-G-B camera. Another technique that has been used for the detection of bunches but that is not discussed here uses the Loy and Zelinski's fast radial feature detector and the searching of local maxima. The results obtained with this technique applied to this type of images are, however, lower than those presented here.



## 6. Conclusions

The conclusions on the subjects covered by this work has been already presented in the chapters that describe them and specifically:

- 5.3.1.2: Results, discussion and conclusion about the Laboratory Spectral Measurements;
- 5.4.1.2: Results, discussion and conclusion about the Laboratory Multispectral Image Processing;
- 5.4.2.2: Results, discussion and conclusion about the Greenhouse Multispectral Image Processing;
- 5.4.3.2: Results, discussion and conclusion about the Field Multispectral Image Processing;

What has been tried to do is to put the conditions for the realization of an automatic (or semi-automatic) system for the detection and treatment of certain fungal diseases of the grapevine, with particular attention to powdery mildew, downy mildew and botrytis. The results obtained are encouraging but show some differences between the theoretical analysis, obtained with the definition of the ROIs and their use in the definition phase of the classification algorithm, and the practical approach of the field. In the definition phase of the algorithm the results are good, especially in the greenhouse where the acquisition setup is strongly controlled, but also in the field, where the definition of a good acquisition setup has permitted greatly simplify the analysis of the acquired images. The results of the classification worsen slightly when it comes to validate the whole acquired image. This difference in performance is due not to an incorrect definition of the algorithm but to a lack of information represented by the unknowns that are encountered in the field (for example, the specular reflections or some back of the leaves). To solve these types of problems you need to add to the spectral method, a texture-based approach that allows to further filter the acquired image. The validation in the field has shown that the disease is detected but in some cases more than necessary is identified as a disease. Remembering that the aim of the system is the identification of the disease in order to early and localized the application of fungicide, it may close the issue by stating that it is best to spray once more than one less, since the effectiveness of the system would remain intact. This approach, however, is not what we have set ourselves at the beginning of the work, so the research continues with the integration of a different approach (texture-based) without prejudice to the desire to maintain unchanged the simplicity of the classification algorithm. Due to the uncertainty related to the acquisition of the field, a possible approach would be to insert the fuzzy component into the classifier. Once completed the definition phase of the algorithm, the next step will be to integrate the diseased detection system with the spraying system, in order to obtain the first localized treatment system of fungal diseases.

## **Collaborations**

The research project is developed in collaboration with the Department of Agricultural Engineering, University of Milan.

The research project is part of a wider international context, as it supports the creation of a more complex robotic system to be used in agriculture and which goes under the name of CROPS (CROPS, Clever Robots for Crops). CROPS is a large-scale integrating FP7 EU project in the theme Nanotechnologies, Materials and new Production Technologies within the call: Automation and robotics for sustainable crop and forestry management (Call FP7-NMP-2009-LARGE-3). CROPS will develop scientific know-how for a highly configurable, modular and clever carrier platform that includes modular parallel manipulators and intelligent tools (sensors, algorithms, sprayers, grippers) that can be easily installed onto the carrier and are capable of adapting to new tasks and conditions. Several technological demonstrators will be developed for high value crops like greenhouse vegetables, fruits in orchards, and grapes for premium wines. The CROPS robotic platform will be capable of site-specific spraying (targets spray only towards foliage and selective targets) and selective harvesting of fruit (detects the fruit, determines its ripeness, moves towards the fruit, grasps it and softly detaches it). Another objective of CROPS is to develop techniques for reliable detection and classification of obstacles and other objects to enable successful autonomous navigation and operation in plantations and forests. The agricultural and forestry applications share many research areas, primarily regarding sensing and learning capabilities. Details on <http://www.crops-robots.eu>.

## **Publications**

### ***Conferences***

- 1] P. Tirelli, M. Marchi, N.A. Borghese, R.Oberti, “*Multispectral image analysis for grapevine’s diseases detection in field conditions*”, International Conference of Agricultural Engineering (CIGR), 2012. (oral presentation)
- 2] R. Oberti, P. Tirelli, M. Marchi, A. Calcante, M. Iriti, N.A. Borghese, “*Automatic diseases detection in grapevine under field conditions*”, Robotics and associated High-technologies and Equipment for Agriculture (RHEA), 2012. (oral presentation)

## **Acknowledgements**

Massimo Marchi, Researcher at DI and Faculty of Agricultural Engineering, Università degli Studi di Milano.

Roberto Oberti, Researcher at Faculty of Agricultural Engineering, Università degli Studi di Milano.

Nunzio Alberto Borghese, Professor at DI, Università degli Studi di Milano.

Iuri Frosio, Researcher at DI, Università degli Studi di Milano.

## **Bibliography**

- Abdallah, S. M., Nebot, E. M., & Rye, D. C. (1997). Object Recognition and Orientation via Zernike Moments. Proceedings of the Third Asian Conference on Computer Vision Volume I Volume I (pp. 386-393). Springer-Verlag.
- Adamchuk, V.I., Hummel, J.W., Morgan, M.T. and Upadhyaya, S.S. (2004). On-the-go soil sensors for precision agriculture. *Computers and Electronics in Agriculture*, 44, 71–91.
- Apan, A., Datt, B. and Kelly, R. (2005). Detection of pests and diseases in vegetable crops using hyperspectral sensing: a comparison of reflectance data for different sets of symptoms. In: Proceedings of SSC 2005 Spatial Intelligence, Innovation and Praxis: The national biennial Conference of the Spatial Science Institute, Melbourne 2005. Melbourne, pp. 10–18.
- Archetti, M., Döring, T.R., Hagen, S.B., Hughes, N.M., Leather, S.R., Lee, D.W., Lev-Yadun, S., Manetas, Y., Ougham, H.J., Schaberg, P.G. and Thomas, H. (2009). Unravelling the evolution of autumn colours: an interdisciplinary approach. *Trends in Ecology and Evolution*, 24, 166–173.
- Asner, G. (1998). Biophysical and biochemical sources of variability in canopy reflectance. *Remote Sensing of Environment*, 64, 234–253.
- Baranoski, G.V.G. and Rokne, J.G. (2001). Efficiently simulating scattering of light by leaves. *The Visual Computer*, 17, 491–505.
- Bélanger M.C., Roger J.M., Cartolaro P., Fermaud M., 2011. Autofluorescence of grape berries following *Botrytis cinerea* infection. *International Journal of Remote Sensing* 32, 3835-3849
- Bélanger M.C., Roger J.M., Cartolaro P., Viau A.A., Bellon-Maurel V., 2008. Detection of powdery mildew in grapevine using remotely sensed UV-induced fluorescence. *International Journal of Remote Sensing* 29, 1707-1724
- Berenstein R., Ben Shahaar O., Shapiro A., and Edan Y.. 2010. Grape clusters and foliage detection algorithms for autonomous selective vineyard sprayer. *Intell. Serv. Robot.* 3, 4 (October 2010), 233-243.
- Biliouris, D., Verstaeten, W.W., Dutre, P., van Aardt, J.A.N., Mys, B. and Coppin, P. (2007). A compact laboratory spectro-goniometer (CLabSpeG) to assess the BRDF of materials. Presentation, calibration and implementation on *Fagus sylvatica* L. leaves. *Sensors*, 7, 1846–1870.
- Bishop C., *Neural Networks for Pattern Recognition*, 12th edition, 2011, Oxford University Press, 1996.
- Blackburn, G. A. (2006). Hyperspectral remote sensing of plant pigments. *Journal of Experimental Botany*, 58(4), 855-867. *Soc Experiment Biol*.
- Blackburn, G.A. (1998a). Quantifying chlorophylls and carotenoids at leaf and canopy scale: an evaluation of some hyperspectral approaches. *Remote Sensing of Environment*, 66, 273–285.
- Blackburn, G.A. (1998b). Spectral indices for estimating photosynthetic pigment concentrations: a test using senescent tree leaves. *International Journal of Remote Sensing*, 19, 657–675.

- Blackburn, G.A. (2007). Hyperspectral remote sensing of plant pigments. *Journal of Experimental Botany*, 58, 844–867.
- Board, J.E., Maka, V., Price, R., Knight, D. and Baur, M.E. (2007). Development of vegetation indices for identifying insect infestations in soybean. *Agronomy Journal*, 99, 650–656.
- Bock, C.H., Poole, G.H., Parker, P.E. and Gottwald, T.R. (2010). Plant disease severity estimated visually, by digital photography and image analysis, and by hyperspectral imaging. *Critical Reviews in Plant Science*, 29, 59–107.
- Bongiovanni, R. and Lowenberg-Deboer, J. (2004). Precision agriculture and sustainability. *Precision Agriculture*, 5, 359–387.
- Bravo, C. (2006). Automatic foliar disease detection in winter wheat. Ph.D. thesis, University of Leuven.
- Bravo, C., Moshou, D., West, J., McCartney, A. and Ramon, H. (2003). Early disease detection in wheat fields using spectral reflectance. *Biosystems Engineering*, 84, 137–145.
- Brooks, F. and Miller, W. (1963). Availability of solar energy. In: A. Zarem and D. Erway, eds., *Introduction to the utilization of solar energy*. McGraw-Hill, New York, NY, USA, pp. 30—58.
- Buschmann, C., & Lichtenthaler, H. K. (1998). Principles and characteristics of multi-colour fluorescence imaging of plants. *Journal of Plant Physiology*, 152(2-3), 297-314. Elsevier.
- Calcante A., Mena A., Mazzetto F., 2011. Optical Devices Evaluation for Diagnosis of *Plasmopara viticola* on Vine. *CIGR E-journal* 13 (2), 1-11
- Calonnec A, Cartolaro P, Poupot C, Dubourdieu D, Darriet P 2004 Effects of *Uncinula necator* on the yield and quality of grapes (*Vitis vinifera*) and wine. *Plant Pathology* 53, 434–445
- Camargo , J. S. Smith, Image pattern classification for the identification of disease causing agents in plants, *Computers and Electronics in Agriculture*, v.66 n.2, p.121-125, May, 2009.
- Camargo and J. S. Smith “An image-processing based algorithm to automatically identify plant disease visual symptoms,” *Biosystems Engineering*, vol. 102, pp. 9-21, 2009.
- Carrol, M.W., Glaser, J.A., Hellmich, R.L., Hunt, T.E., Sappington, T.W., Calvin, D., Copenhaver, K. and Fridgen, J. (2008). Use of spectral vegetation indices derived from airborne hyperspectral imagery for detection of European corn borer infestation in Iowa corn plots. *Journal of Economic Entomology*, 101, 1614–1623.
- Carter, G.A. and Knapp, A.K. (2001). Leaf optical properties in higher plants: linking spectral characteristics to stress and chlorophyll concentration. *American Journal of Botany*, 88, 677–684.
- Carter, G.A. and Miller, R.L. (1994). Early detection of plant stress by digital imaging with narrow stress-sensitive wavebands. *Remote Sensing of Environment*, 50, 295–302.
- Carter, G.A. and Spiering, B.A. (2002). Optical properties of intact leaves for estimating chlorophyll concentration. *Journal of Environmental Quality*, 31, 1424–1432.

- Carter, G.A., Knapp, A.K., 2001. Leaf optical properties in higher plants: linking spectral characteristics to stress and chlorophyll concentration. *American Journal of Botany*, 88, 677–684
- Chaerle, L. and van der Straeten, D. (2001). Seeing is believing: imaging techniques to monitor plant health. *Biochimica et Biophysica Acta (BBA) — Gene Structure and Expression*, 1519, 153–166.
- Chaerle, L., Hagenbeek, D., De Bruyne, E. and Van Der Straeten, D. (2007a). Chlorophyll fluorescence imaging for disease-resistance screening of sugar beet. *Plant Cell Tissue and Organ Culture*, 91, 97–106.
- Chaerle, L., Hagenbeek, D., Vanrobaeys, X. and Van der Straeten, D. (2007b). Early detection of nutrient and biotic stress in *Phaseolus vulgaris*. *International Journal of Remote Sensing*, 28, 3479–3492.
- Chamelat, R., Rosso, E., Choksuriwong, A., Rosenberger, C., Laurent, H., & Bro, P. (2006). Grape Detection By Image Processing. *IECON 2006 32nd Annual Conference on IEEE Industrial Electronics*, 3697-3702. Ieee.
- Combal, B., Baret, F. and Weiss, M. (2002). Improving canopy variables estimation from remote sensing data by exploiting ancillary information case study on sugar beet canopies. *Agronomie*, 22, 205–215.
- Coops, N., Stanford, M., Old, K., Dudzinski, M., Culvenor, D. and Stone, C. (2003). Assessment of *Dothistroma* needle blight of *Pinus radiata* using airborne hyperspectral imagery. *Phytopathology*, 93, 1524–1532.
- Crisp P, Wicks TJ, Bruer D, Scott ES 2006b An evaluation of biological and abiotic controls for grapevine powdery mildew. 2. Vineyard trials. *Australia Journal of Grape and wine Research* 12, 203-211
- Crisp P, Wicks TJ, Lorimer M, Scott ES 2006a An evaluation of biological and abiotic controls for grapevine powdery mildew. 1. Greenhouse studies. *Australia Journal of Grape and wine Research* 12, 192-202
- Cséfalvay L, Di Gaspero G, Matouš K, Bellin D, Ruperti B, Olejníčková J (2009) Pre-symptomatic detection of *Plasmopara viticola* infection in grapevine leaves using chlorophyll fluorescence imaging. *Eur J Plant Pathol* 125:291–302.
- Curran, P.J. (1989). Remote sensing of foliar chemistry. *Remote Sensing of Environment*, 30, 271–278.
- Dammer, K.H., Wollny, J. and Giebel, A. (2008). Estimation of the leaf area index in cereal crops for variable rate fungicide spraying. *European Journal of Agronomy*, 28, 351–360.
- Darvishzadeh, R., Atzberger, C., Skidmore, A. K., & Abkar, A. A. (2009). Leaf Area Index derivation from hyperspectral vegetation indices and the red edge position. *International Journal of Remote Sensing*, 30(23), 6199-6218.

- de Jong, S. "SIMPLS: An Alternative Approach to Partial Least Squares Regression." *Chemometrics and Intelligent Laboratory Systems*. Vol. 18, 1993, pp. 251–263.
- Delalieux, S., Somers, B., Verstraeten, W.W., van Aardt, A.N.J., Keulemans, W. and Coppin, P. (2009). Hyperspectral indices to diagnose leaf biotic stress of apple plants, considering leaf phenology. *International Journal of Remote Sensing*, 30, 1887–1912.
- Delalieux, S., van Aardt, J., Keulemans, W. and Coppin, P. (2005). Detection of biotic stress (*Venturia inaequalis*) in apple trees using hyperspectral analysis. In: B. Zagajewski, M. Soczal and M. Wrzesien, eds., *Proceedings of the 4th EARSel Workshop on Imaging Spectroscopy*. Warsaw University, Warsaw, Poland, pp. 677–689.
- Delalieux, S., van Aardt, J., Keulemans, W., Schrevens, E. and Coppin, P. (2007). Detection of biotic stress (*Venturia inaequalis*) in apple trees using hyperspectral data: non-parametric statistical approaches and physiological implications. *European Journal of Agronomy*, 27, 130–143.
- Devadas R., Lamb D.W., Simpfendorfer S., Backhouse D., 2009, Evaluating ten spectral vegetation indices for identifying rust infection in individual wheat leaves, *Precision Agriculture* 10,459-470
- Diwan P. Ariana, Renfu Lu, Daniel E. Guyer, Near-infrared hyperspectral reflectance imaging for detection of bruises on pickling cucumbers, *Computers and Electronics in Agriculture*, Volume 53, Issue 1, August 2006, Pages 60-70, ISSN 0168-1699, 10.1016/j.compag.2006.04.001.
- Doraiswamy, P.C., Moulin, S., Cook, P.W. and Stern, A. (2003). Crop yield assessment for remote sensing. *Photogrammetric Engineering & Remote Sensing*, 69, 665–674.
- Filella, I. and Peñuelas, J. (1994). The red edge position and shape as indicators of plant chlorophyll content, biomass and hydric status. *International Journal of Remote Sensing*, 15, 1459–1470.
- Fourty, T., Baret, F., Jacquemoud, S., Schnuck, G. and Verdebout, J. (1996). Leaf optical properties with explicit description of its biochemical composition: direct and inverse problems. *Remote Sensing of Environment*, 56, 104–117.
- Franke, J. and Menz, G. (2007). Multi-temporal wheat disease detection by multi-spectral remote sensing. *Precision Agriculture*, 8, 161–172.
- Franke, J., Gebhardt, S., Menz, G. and Helfrich, H.P. (2009). Geostatistical analysis of the spatiotemporal dynamics of powdery mildew and leaf rust in wheat. *Phytopathology*, 99, 974–984.
- Galvao, L.S., Roberts, D.A., Formaggio, A.R., Numata, I. and Breunig, F.M. (2009). View angle effects on the discrimination of soybean varieties and on the relationships between vegetation indices and yield using off-nadir Hyperion data. *Remote Sensing of Environment*, 113, 846–856.
- Gamon, J.A. and Surfus, J.S. (1999). Assessing leaf pigment content and activity with a reflectometer. *New Phytologist*, 143, 105–117.
- Gebbers, R. and Adamchuk, V.I. (2010). Precision agriculture and food security. *Science*, 327, 828–831.



- Gerhards, R. and Christensen, S. (2003). Real-time weed detection, decision making and patch spraying in maize, sugarbeet, winter wheat and winter barley. *Weed Research*, 43, 385–392.
- Gerhards, R. and Oebel, H. (2006). Practical experiences with a system for site specific weed control in arable crops using real-time image analysis and GPS-controlled patch spraying. *Weed Research*, 46, 185–193.
- Gitelson, A.A., Gritz, Y. and Merzylak, M.N. (2003). Relationships between leaf chlorophyll content and spectral reflectance and algorithms for nondestructive chlorophyll assessment in higher plant leaves. *Journal of Plant Physiology*, 160, 271–282.
- Gitelson, A.A., Kaufman, Y.J., Stark, R. and Rundquist, D. (2002). Novel algorithms for remote estimation of vegetation fraction. *Remote Sensing of Environment*, 80, 76–87.
- Gitelson, A.A., Merzlyak, N.M. and Chivkunova, B.O. (2001). Optical properties and nondestructive estimation of anthocyanin content in plant leaves. *Photochemistry and Photobiology*, 74, 38–45.
- Gonzalez, R. C., & Woods, R. E. (2002). *Digital Image Processing (2nd Edition)*(p. 793). Prentice Hall.
- Gould, K.S., Kuhn, D.N., Lee, D.W. and Oberbauer, S.F. (1995). Why leaves are sometimes red. *Nature*, 378, 241–242.
- Govaerts, Y.M., Jacquemoud, S., Verstraete, M. and Ustin, S.L. (1996). Threedimensional radiation transfer modeling in a dicotyledon leaf. *Applied Optics*, 35, 6585–6598.
- Graeff, S., Link, J. and Claupein, W. (2006). Identification of powdery mildew (*Erysiphe graminis* sp. *tritici*) and take-all disease (*Gaeumannomyces graminis* sp. *tritici*) in wheat (*Triticum aestivum* L.) by means of leaf reflectance measurements. *Central European Journal of Biology*, 1, 275–288.
- Haboudane, D., Miller, J.R., Pattey, E., Zarco-Tejada, P. and Strachan, I. (2004). Hyperspectral vegetation indices and novel algorithms for predicting green LAI of crop canopies: modeling and validation in the context of precision agriculture. *Remote Sensing of Environment*, 90, 337–352.
- Hatfield, L.J., Gitelson, A.A., Schepers, S.J. and Walthall, L.C. (2008). Application of Spectral Remote Sensing for Agronomic Decisions. *Agronomy Journal*, 100, 117–131.
- Herrmann I., A. Karnieli, A. Pimstein, Y. Cohen, V. Alchanatis, and D.J. Bonfil. (2010). Assessment of field crops leaf area index by the red-edge inflection point derived from venus bands. In: 10th International Conference on Precision Agriculture.
- Hillnhuetter, C. and Mahlein, A.K. (2008). Early detection and localisation of sugar beet diseases: new approaches. *Gesunde Pflanzen*, 60, 143–149.
- Hosgood, B. (1993). Leaf optical properties experiment 93 (LOPEX93), database structure.
- Huang, W., Lamb, D.W., Niu, Z., Zhang, Y., Liu, L. and Wang, J. (2007). Identification of yellow rust in wheat using in situ spectral reflectance measurements and airborne hyperspectral imaging. *Precision Agriculture*, 8, 187– 197.

- Iriti M, Vitalini S, Di Tommaso G, Di Tommaso D, Faoro F 2011 A new chitosan formulation induces grapevine resistance against powdery mildew and improves grape quality traits. *Australian Journal of Grape and Wine Research* 17, 263-269
- Jacquemoud, S. and Baret, F. (1990). PROSPECT: a model of leaf optical properties spectra. *Remote Sensing of Environment*, 34, 75–91.
- Jacquemoud, S. and Ustin, L.S. (2001). Leaf optical properties: a state of the art. In: 8th International Symposium of Physical Measurements & Signatures in Remote Sensing. CNES, Aussois, France, pp. 223–332.
- Jacquemoud, S., Verdebout, J., Schmuck, G., Andreoli, G. and Hosgood, B. (1995). Investigation of leaf biochemistry by statistics. *Remote Sensing of Environment*, 54, 180–188.
- Jensen, J.R. (2002). *Remote sensing of the environment – An earth resource perspective*. [reprint.] edition. The MIT Press and MIT Press, Upper Saddle River, NJ, USA.
- Jing, L., Jinbao, J., Yunhao, C., Yuanyuan, W., Wei, S. and Wenjiang, H. (2007). Using hyperspectral indices to estimate foliar chlorophyll a concentrations of winter wheat under yellow rust stress. *New Zealand Journal of Agricultural Research*, 50, 1031–1036.
- Jones, H.G. and Schofield, P. (2008). Thermal and other remote sensing of plant stress. *General and Applied Plant Physiology*, 34, 19–32.
- Jones, H.G., Archer, N., Rotenburg, E. and Casa, R. (2003). Radiation measurement for plant ecophysiology. *Journal of Experimental Botany*, 54, 879–889.
- José Manuel Iñesta and Jorge Calera-Rubio. 2002. Robust Gray-Level Histogram Gaussian Characterisation. In *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, Terry Caelli, Adnan Amin, Robert P. W. Duin, Mohamed S. Kamel, and Dick de Ridder (Eds.). Springer-Verlag, London, UK, UK, 833-841.
- Just, S., & Pedersen, K. (2007). Circular Hough Transform. *Transform*, (November), 1-6.
- Kimme C., Ballard D. H., and Sklansky J., Finding circles by an array of accumulators., *Communications of the Association for Computing Machinery*, v. 18, pp. 120-122, 1975.
- Kobayashi, T., Kanda, E., Kitada, K., Ishiguro, K. and Torigoe, Y. (2001). Detection of rice panicle blast with multispectral radiometer and the potential of using airborne multispectral scanners. *Phytopathology*, 91, 316–323.
- Kruse, J.K., Christians, N.E. and Chaplin, M.H. (2006). Remote sensing of nitrogen stress in creeping bentgrass. *Agronomy Journal*, 98, 1640–1645.
- Kühbauch, W. and Hawlitschka, S. (2003). Remote sensing – a future technology in precision farming. In: H. Lacoste, ed., *Conference Proceedings of the Workshop on POLinSAR – Applications of SAR Polarimetry and Polarimetric Interferometry (ESA SP-529)*. Frascati, Italy, p. 25.1.

- Lacis, A.A. and Hansen, J.E. (1973). A parameterization for the absorption of solar radiation in the earth's atmosphere. *Journal of the Atmospheric Science*, 31, 118–133.
- Larsolle, A. and Muhammed, H.H. (2007). Measuring crop status using multivariate analysis of hyperspectral field reflectance with application to disease severity and plant density. *Precision Agriculture*, 8, 37–47.
- Laudien, R., Bareth, G. and Doluschitz, R. (2003). Analysis of hyperspectral field data for detection of sugar beet diseases. In: *Proceedings of the EFITA Conference*. Debrecen (Hungary), pp. 375–381.
- Le Maire, G., Francois, C. and Dufrene, E. (2004). Towards universal broad leaf chlorophyll indices using PROSPECT simulated database and hyperspectral reflectance measurements. *Remote Sensing of Environment*, 89, 1–28.
- Lee W.S., Alchanatis V., Yang C., Hirafuji M., Moshou D., Li C., 2010. Sensing technologies for precision specialty crop production, *Computers and Electronics in Agriculture* 74, 2-33
- Lee, D.W., O'Keefe, J., Holbrook, N.M. and Field, T.S. (2003). Pigment dynamics and autumn leaf senescence in a New England deciduous forest, Eastern USA. *Ecological Research*, 18, 677–694.
- Lenk S, Chaerle L, Pfündel EE, Langsdorf G, Hagenbeek D, Lichtenthaler HK, Van Der Straeten D, Buschmann C. Multispectral fluorescence and reflectance imaging at the leaf level and its possible applications. *J Exp Bot*. 2007;58(4):807-14. Epub 2006 Nov 21. Review.
- Lenthe, J.H., Oerke, E.C. and Dehne, H.W. (2007). Digital thermography for monitoring canopy health of wheat. *Precision Agriculture*, 8, 15–26.
- Li, Fei, Miao, Yuxin, Gnyp, Martin L., Hennig, Simon D., Chen, Xinping, Jia, Liangliang et al.: Evaluating red edge vegetation indices for estimating winter wheat N status under high canopy coverage condition, 2009.
- Lillesand, T.M. and Kiefer, R.W. (2000). *Remote sensing and image interpretation*. John Wiley & Sons, Inc., New York, NY, USA.
- Lu, D., Mausel, P., Brondizio, E. and Moran, E. (2004). Change detection techniques. *International Journal of Remote Sensing*, 25, 2365–2401.
- Mahlein A.K., Oerke E.C., Steiner U., Dehne H.W., 2012. Recent advances in sensing plant diseases for precision crop protection, *Eur. J. Plant Pathology*, 133, 197-209
- Mahlein, A.K., Steiner, U., Dehne, H.W. and Oerke, E.C. (2010). Spectral signatures of sugar beet leaves for the detection and differentiation of diseases. *Precision Agriculture*, 11, 413–431.
- Matouš K, Benediktyová Z, Berger S, Roitsch T, Nedbal L (2006) Case study of combinatorial imaging: what protocol and what chlorophyll fluorescence image to use when visualizing infection of *Arabidopsis thaliana* by *Pseudomonas syringae*? *Photosynth Res* 90:243–253.
- Meigs, A.D., Otten, L.J. and Cherezova, T.Y. (2008). Ultraspectral imaging: a new contribution to global virtual presence. *IEEE Aerospace and Electronic Systems Magazine*, 23, 11–17.

- Melesse, A.M., Weng, Q., Thenkabail, P.S. and Senay, G.B. (2007). Remote sensing sensors and applications in environmental resources mapping and modeling. *Sensors*, 7, 3209–3241.
- Merzylak, M.N., Melo, T.B. and Naqvi, K.R. (2008). Effect of anthocyanins, carotenoids and flavonols on chlorophyll fluorescence excitation spectra in apple fruit: signature analysis, assessment, modeling, and relevance to photoprotection. *Journal of Experimental Botany*, 59, 349–359.
- Moran, M.S., Inoue, Y. and Barnes, E.M. (1997). Opportunities and limitations for image-based remote sensing in precision crop management. *Remote Sensing of Environment*, 61, 319–346.
- Moshou D., Bravo C., Oberti R., West J., Bodria L., McCartney A., Ramon H., 2005. Plant disease detection based on data fusion of hyper-spectral and multi-spectral fluorescence imaging using Kohonen maps. *Real Time imaging* 11, 75-83
- Moshou, D., Bravo, C., Wahlen, S., West, J., McCartney, A., Baerdemaeker, J. and Ramon, H. (2006). Simultaneous identification of plant stresses and diseases in arable crops using proximal optical sensing and self-organising maps. *Precision Agriculture*, 7, 149–164.
- Moshou, D., Bravo, C., West, J., Wahlen, S., McCartney, A. and Ramon, H. (2004). Automatic detection of yellow rust in wheat using reflectance measurements and neural networks. *Computers and Electronics in Agriculture*, 44, 173–188.
- Naidu, A.R., Perry, M.E., Pierce, J.F. and Mekuria, T. (2009). The potential of spectral reflectance technique for the detection of Grapevine leafroll associated virus-3 in two red-berried wine grape cultivars. *Computers and Electronics in Agriculture*, 66, 38–45.
- Nash, E., Dreger, F., Schwarz, J., Bill, R. and Werner, A. (2009). Development of a model of datt-flows for precision agriculture base on a collaborative research project. *Computers and Electronics in Agriculture*, 66, 25–37.
- Nilsson, H.E. (1995). Remote sensing and image analysis in plant pathology. *Annual Review of Phytopathology*, 15, 489–527.
- Nutter, F.W. and Littrell, R.H. (1996). Relationship between defoliation, canopy reflectance and pod yield in the peanut-late leafspot pathosystem. *Crop Protection*, 15, 135–142.
- Nutter, F.W.J., Littrell, R.H. and Brennemann, T.B. (1990). Utilization of a multispectral radiometer to evaluate fungicide efficacy to control late leaf spot in peanut. *Phytopathology*, 80, 102–108.
- Oberti R., 2003. Optical systems for assessing the healthy status of plants (in Italian). *I Georgofili – Atti dell’Accademia dei Georgofili* 50, 299-318
- Oerke, E.C. and Dehne, H.W. (2004). Safeguarding production losses in major crops and the role of crop protection. *Crop Protection*, 23, 275–285.

- Oerke, E.C., Steiner, U., Dehne, H.W. and Lindenthal, M. (2006). Thermal imaging of cucumber leaves affected by downy mildew and environmental conditions. *Journal of Experimental Botany*, 57, 2121–2132.
- Oppelt, N. and Mauser, W. (2004). Hyperspectral monitoring of physiological parameters of wheat during a vegetation period using AVIS data. *International Journal of Remote Sensing*, 25, 145–159.
- Paul Boissard , Vincent Martin , Sabine Moisan, A cognitive vision approach to early pest detection in greenhouse crops, *Computers and Electronics in Agriculture*, v.62 n.2, p.81-93, July, 2008.
- Peñuelas, J. and Filella, I. (1998). Visible and near-infrared reflectance techniques for diagnosing plant physiological status. *Trends in Plant Science*, 3, 151–156.
- Peñuelas, J., Pinol, R.O., Ogaya, R. and Filella, I. (1997). Estimation of plant water concentration by the reflectance Water Index WI (R900/R970). *International Journal of Remote Sensing*, 18, 2869–2875.
- Pietrzykowski, E., Stone, C., Pinkard, E. and Mohammed, C. (2006). Effects of *Mycosphaerella* leaf disease on the spectral reflectance properties of juvenile *Eucalyptus globules* foliage. *Forest Pathology*, 36, 334–348.
- Pinter, P.J., Hatfield, J.L., Schepers, J.S., Barnes, E.M., Moran, M.S., Daughtry, C.S.T. and Upchurch, D.R. (2003). Remote sensing for crop management. *Photogrammetric Engineering and Remote Sensing*, 69, 647– 664.
- Pinty, B., Verstraeten, M.M. and Gobron, N. (1998). The effect of soil anisotropy on the radiance field emerging from vegetation canopies. *Geophysical Research Letters*, 25, 797–800.
- Poutaraud A., Latouche G., Martins S., Meyer S., Merdinoglu D., Cerovic Z.G., 2007. Fast and local assessment of stilbene content in grapevine leaf by in vivo fluorometry *J. Agric. Food Chem.* 55 , 4913–4920
- Qin, J., Burks, T.F., Ritenour, M.A. and Bonn, W.G. (2009). Detection of citrus canker using hyperspectral reflectance imaging with spectral information divergence. *Journal of Food Engineering*, 93, 183–191.
- Rafael C. Gonzalez and Richard E. Woods. 2006. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- Rascher, U., Liebig, M. and Lüttge, U. (2000). Evaluation of instant lightresponse curves of chlorophyll fluorescence fluorometer on site in the field. *Plant, Cell and Environment*, 23, 1397–1405.
- Reichardt, M., Jürgens, C., Klöble, U., Hüter, J. and Moser, K. (2009). Dissemination of Precision farming in Germany: acceptance, adoption, obstacles, knowledge transfer and training activities. *Precision Agriculture*, 10, 525–545.
- Richardson, A.D., Duigan, S.P. and Berlyn, G.P. (2001). An evaluation of noninvasive methods to estimate foliar chlorophyll content. *New Phytologist*, 153, 185–194.

- Rizon, M., Yazid, H., Saad, P., Shakaff, A. Y. M., Saad, A. R., Sugisaka, M., Yaacob, S., et al. (2005). Object Detection using Circular Hough Transform. *American Journal of Applied Sciences*, 2(12), 85-91.
- Rosipal, R., and N. Kramer. "Overview and Recent Advances in Partial Least Squares." *Subspace, Latent Structure and Feature Selection: Statistical and Optimization Perspectives Workshop (SLSFS 2005), Revised Selected Papers (Lecture Notes in Computer Science 3940)*. Berlin, Germany: Springer-Verlag, 2006, pp. 34–51.
- Rumpf, T., Mahlein, A. K., Steiner, U., Oerke, E. C., Dehne, H. W., & Plümer, L. (2010). Early detection and classification of plant diseases with Support Vector Machines based on hyperspectral reflectance. *Computers and Electronics in Agriculture*, 74(1), 91-99. Elsevier B.V.
- Sankaran S., Mishra A., Ehsani R., Davis C., 2010. A review of advanced techniques for detecting plant diseases. *Computers and Electronics in Agriculture* 72, 1-13.
- Schellberg, J., Hill, M., Gerhards, R., Rothmund, M. and Braun, M. (2008). Precision agriculture on grassland: applications, perspectives and constraints. *European Journal of Agronomy*, 29, 59–71.
- Scotford, I. and Miller, P. (2005). Applications of spectral reflectance techniques in northern European cereal production: a review. *Biosystems Engineering*, 90, 235–250.
- Sims, D.A. and Gamon, J.A. (2002). Relationship between leaf pigment content and spectral reflectance across a wide range of species, leaf structures and developmental stages. *Remote Sensing of Environment*, 81, 337–354.
- Singh, A. (1989). Digital change detection techniques using remotely-sensed data. *International Journal of Remote Sensing*, 10, 989–1003.
- Smith, K.L., Steven, M.D. and Colls, J.J. (2004). Use of hyperspectral derivative ratios in the red-edge region to identify plant stress response to gas leaks. *Remote Sensing of Environment*, 92, 207–217.
- Stafford, J.V. (2000). Implementing precision agriculture in the 21st Century. *Journal of Agricultural Engineering Research*, 76, 267–275.
- Stauffer, C., & Grimson, W. E. L. (1999). Adaptive background mixture models for real-time tracking. *Proceedings 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* Cat No PR00149, 2(c), 246-252
- Steddom, K., Bredehoeft, W.M., Khan, M. and Rush, M.C. (2005). Comparison of visual and multispectral radiometric disease evaluations of *Cercospora* leaf spot of sugar beet. *Plant Disease*, 89, 153–158.
- Steddom, K., Heidel, G., Jones, D. and Rush, C.M. (2003). Remote detection of rhizomania in sugar beets. *Phytopathology*, 93, 720–726.
- Steiner, U., Bürling, K. and Oerke, E.C. (2008). Sensor use in plant protection. *Gesunde Pflanzen*, 60, 131–141.

- Stephen T. Nuske, Supreeth Achar, Terry Bates, Srinivasa G. Narasimhan, and Sanjiv Singh, Yield Estimation in Vineyards by Visual Grape Detection. Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '11), September, 2011.
- Stoll, M., Schultz, H. R., Baecker, G., & Berkelmann-Loehnertz, B. (2008a). Early pathogen detection under different water status and the assessment of spray application in vineyards through the use of thermal imagery. *Precision Agriculture*, 9, 407–417.
- Stummer BL, Francis L, Markides AJ, Scott ES 2003 The effect of powdery mildew infection on grape berries and wine composition and sensory properties of Chardonnay wines. *Australian Journal of Grape and Wine Research* 9, 28–39
- Thenkabail, P.S., Smith, R.B. and De Pauw, E. (2000). Hyperspectral vegetation indices and their relationship with agricultural crop characteristics. *Remote Sensing of Environment*, 71, 158–182.
- Ustin, S.L., Gitelson, A.A., Jacquemoud, S., Schaepman, M., Asner, G.P., Gamon, J.A. and Zarco-Tejada, P. (2009). Retrieval of foliar information about plant pigment systems from high resolution spectroscopy. *Remote Sensing of Environment*, 113, 67–77.
- Von Witzke, H., Noleppa, S. and Schwarz, G. (2008). Global agricultural market trends and their impacts on European agriculture. In: Working Paper 84, Humboldt University Berlin.
- West, S.J., Bravo, C., Oberti, R., Lemaire, D., Moshou, D. and McCartney, H.A. (2003). The potential of optical canopy measurement for targeted control of field crop diseases. *Annual Review of Phytopathology*, 41, 593–614.
- Xu, H.R., Xing, Y.B., Fu, X.P. and Zhu, S.P. (2007). Near-infrared spectroscopy in detecting leaf miner damage on tomato leaf. *Biosystems Engineering*, 96, 447–454.
- Yang, C.M., Cheng, C.H. and Chen, R.K. (2007). Changes in spectral characteristics of rice canopy infested with brown planthopper and leafhopper. *Crop Science*, 47, 329–335.
- Yang, X.-H., Huang, J.-F., Wang, F.-M., Wang, X.-Z., Yi, Q.-X., & Wang, Y. (2006). A modified chlorophyll absorption continuum index for chlorophyll estimation. *Journal of Zhejiang University SCIENCE A*, 7(12), 2002-2006.
- Zhang, M., Qin, Z., Liu, X. and Ustin, S. (2003). Detection of stress in tomatoes induced by late blight disease in California, USA, using hyperspectral remote sensing. *Applied Earth Observation and Geoinformation*, 4, 295–310.
- Zhang, N., Wang, M. and Wang, N. (2002). Precision agriculture – a worldwide overview. *Computers and Electronics in Agriculture*, 36, 113–32.
- Giles B., Downey, D. 2005. Reducing orchard spray rates and ground deposit by using tree sensors and sprayer control. *Annu. Rev. Agric. Eng.* 4: 229–236.
- Solanelles, F., Escolà, A., Planas, S., Rosell, J. R., Camp, F., Gràcia, F. 2006. An Electronic Control System for Pesticide Application Proportional to the Canopy Width of Tree CROPS. *Biosystems Engineering* 95 (4): 473–481.

## Appendices

### A. Commercial software used

In order to achieve what has been described, were used the following commercial software.

#### A.1. Matlab R2009b v7.9

MATLAB (matrix laboratory) is a numerical computing environment and fourth-generation programming language. Developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, and Fortran. Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing capabilities. An additional package, Simulink, adds graphical multi-domain simulation and Model-Based Design for dynamic and embedded systems. MATLAB is a proprietary product of MathWorks, so users are subject to vendor lock-in. Although MATLAB Builder can deploy MATLAB functions as library files which can be used with .NET or Java application building environment, future development will still be tied to the MATLAB language.

(<http://www.mathworks.it>)

#### A.2. Microsoft Visual Studio 2008 v9.0

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop console and graphical user interface applications along with Windows Forms applications, web sites, web applications, and web services in both native code together with managed code for all platforms supported by Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework and Microsoft Silverlight. Visual Studio supports different programming languages by means of language services, which allow the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C/C++ (via Visual C++), VB.NET (via Visual Basic .NET), C# (via Visual C#), and F# (as of Visual Studio 2010). Support for other languages such as M, Python, and Ruby among others is available via language services installed separately. It also supports XML/XSLT, HTML/XHTML, JavaScript and CSS. Individual language-specific versions of Visual Studio also exist which provide more limited language services to the user: Microsoft Visual Basic, Visual J#, Visual C#, and Visual C++. Microsoft Visual Studio is a proprietary product of Microsoft.

([www.microsoft.com/visualstudio/](http://www.microsoft.com/visualstudio/))

#### A.3. MySQL v5.5

MySQL is the world's most used open source relational database management system (RDBMS) as of 2008 that runs as a server providing multi-user access to a number of databases. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. MySQL is a popular choice of database for use in web applications, and is a central component of



the widely used LAMP open source web application software stack (and other 'AMP' stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python". Free-software-open source projects that require a full-featured database management system often use MySQL.

(<http://www.mysql.it/>)

#### **A.4. Paint .NET v3.5**

Paint.NET is a proprietary freeware raster graphics editor program for Microsoft Windows, developed on the .NET Framework. Paint.NET was originally created by Rick Brewster as a Washington State University student project, and has evolved from a simple replacement for the Microsoft Paint program into a powerful editor with support for layers, blending, transparency, and plugins. Excluding the installer, text, and graphics, Paint.NET was released under a modified version of the MIT License. It was initially released as completely open source, but due to breaches of license, all resource files (such as interface text and icons) were released under a Creative Commons license forbidding modification, and the installer was made closed-source.

(<http://www.getpaint.net>)

#### **A.5. Mercurial SCM v1.9**

Mercurial is a cross-platform, distributed revision control tool for software developers. It is mainly implemented using the Python programming language, but includes a binary diff implementation written in C. It is supported on Windows and Unix-like systems, such as FreeBSD, Mac OS X and Linux. Mercurial is primarily a command line program but graphical user interface extensions are available. All of Mercurial's operations are invoked as arguments to its driver program hg, a reference to the chemical symbol of the element mercury. Mercurial's major design goals include high performance and scalability, decentralized, fully distributed collaborative development, robust handling of both plain text and binary files, and advanced branching and merging capabilities, while remaining conceptually simple. It includes an integrated web interface. Mercurial has also taken steps to ease the transition for SVN users. The creator and lead developer of Mercurial is Matt Mackall. Mercurial is released as free software under the terms of the GNU GPL v2.

(<http://mercurial.selenic.com>)

## B. Application Software Developed

In order to realize what is described, have been developed the following software applications.

### B.1. Images Acquisition

For the acquisition of the images (in the greenhouse as in field) has been realized a software able to acquire simultaneously from multiple cameras (in the specific R-G-B and R-G-NIR, Figure 83). This application allows you to control multiple cameras simultaneously and, in the case of multispectral camera Duncantech, set run-time gain and integration time for each channel (or activate the self-calibration system described in DuncanTech Camera Parameter Setting System). The application allows you to view run-time histogram for each channel, to apply a filter to the video stream while it is being acquired, zoom the image and insert a series of notes (e.g. surveys of the pathologist) associated with each captured image. Developed in C/C++.

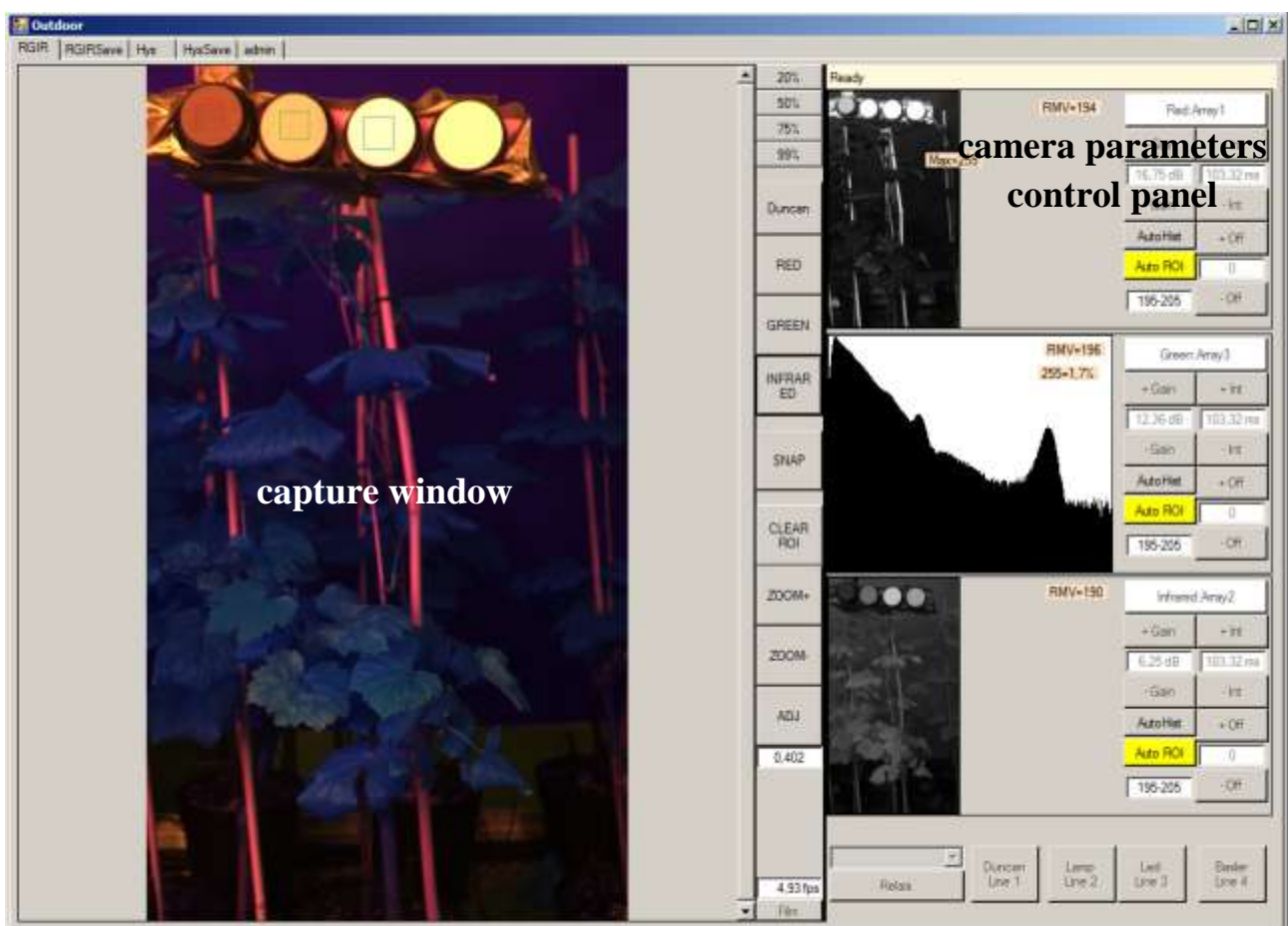


Figure 83 – Cameras Acquisition software. The left column of the interface is the capture window. The right column of the interface is the control panel parameters (gain and integration time) of the multispectral camera. The right column also displays the histograms of the individual channels of the camera. Between the two columns are the buttons that allow to capture the image, display a single channel of the camera, zoom in on the image, etc..

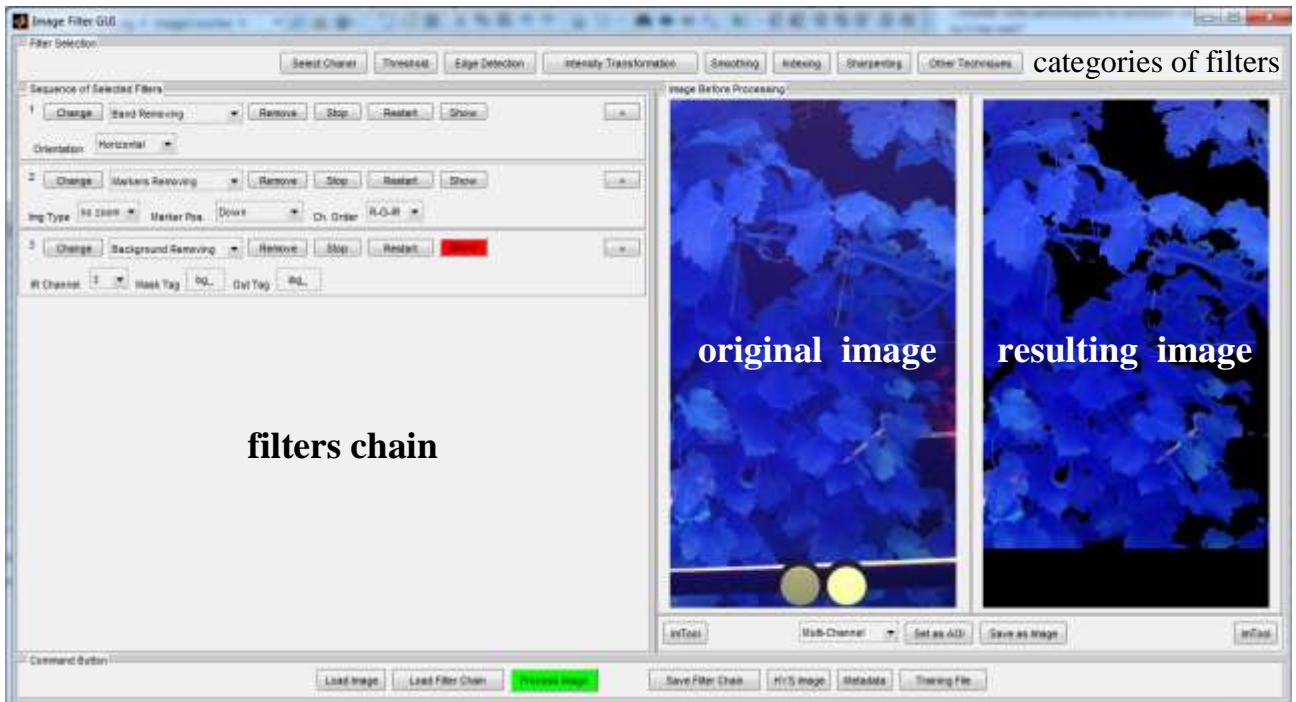
The different captured images R-G-B and R-G-NIR not have the intent to allow a data-fusion, although nothing would prevent. For the applications described above, the R-G-B image has the only task of documenting the shot, giving the real picture of the scene captured. The R-G-B image shows the context associated with the detail captured by multispectral camera (Figure 84).



**Figure 84** – Acquired R-G-NIR image (left) and its associated R-G-B image (right). The two images are not aligned. The RGB image is acquired only for documentation purposes and the part highlighted in blue is pretty much the scene captured by the multispectral camera.

## **B.2. Filters Chain Definition**

To facilitate the work of analysis of the acquired images, in the laboratory stage has been developed software for the application of different filters to an image (Figure 85). The application provides for the definition of a filter that can be among those known (for example Otsu, canny edge detection, etc. ....) or custom (e.g. Background Removal). The filter is defined according to a prototype, it can have input parameters (text strings, numbers, etc ....), accept the input image (R-G-B, monochrome, etc ....) and return the input image filtered. The application predict that more filters can be cascaded so that the output of one represents the input for the next. Completing the software the ability to view images of each intermediate result filtering, the ability to analyze the different images in a graphical external tool, the ability to insert, remove or change a filter at any point in the chain and save a filter chain as a single filter. Developed in Matlab.



**Figure 85 – Filters Chain Definition software.** The upper part of the interface shows the categories of filters that you can use (e.g. Threshold, Edge Detection, etc..). The left column of the interface shows the sequence of the selected filters (once the category is selected, you can choose a filter specific to that category). The right column of the interface shows two images: one on the left is the original starting, one on the right is the image resulting from the filter chain (you can also view the images resulting from the intermediate steps).

### B.3. ROIs Selection

To select the ROIs to be used to test and validate the algorithm has been realized a software of ROIs selection. The software allows you to view all the images associated with a single subject (R-G-B of documentation and R-G-NIR of acquisition). With the image, you can view the acquisition parameters of the cameras (gain, integration time, focal length, etc..), environment information of the acquisition (distance of the camera from wine row, weather conditions, etc.) and inserted notes associated with the acquired image (e.g. assessment by the pathologist). Starting from this information you can select one or more areas on the acquired image (ROI, region of interest) of the desired shape and apply their unique identifier (a TAG). For example, you can specify an area healthy, diseased and its level of infection, a foreign object (e.g. an insect), a light reflection, etc. The association of the TAG is assisted, i.e. there are some predefined TAG, such as HEALTH, YOUNG, NECRO, etc., but it is also possible to insert new ones. Information on selected ROIs (Figure 86) are stored in a MySQL database and made available for training and validation of the classifiers. Developed in C/C++ and MySQL.

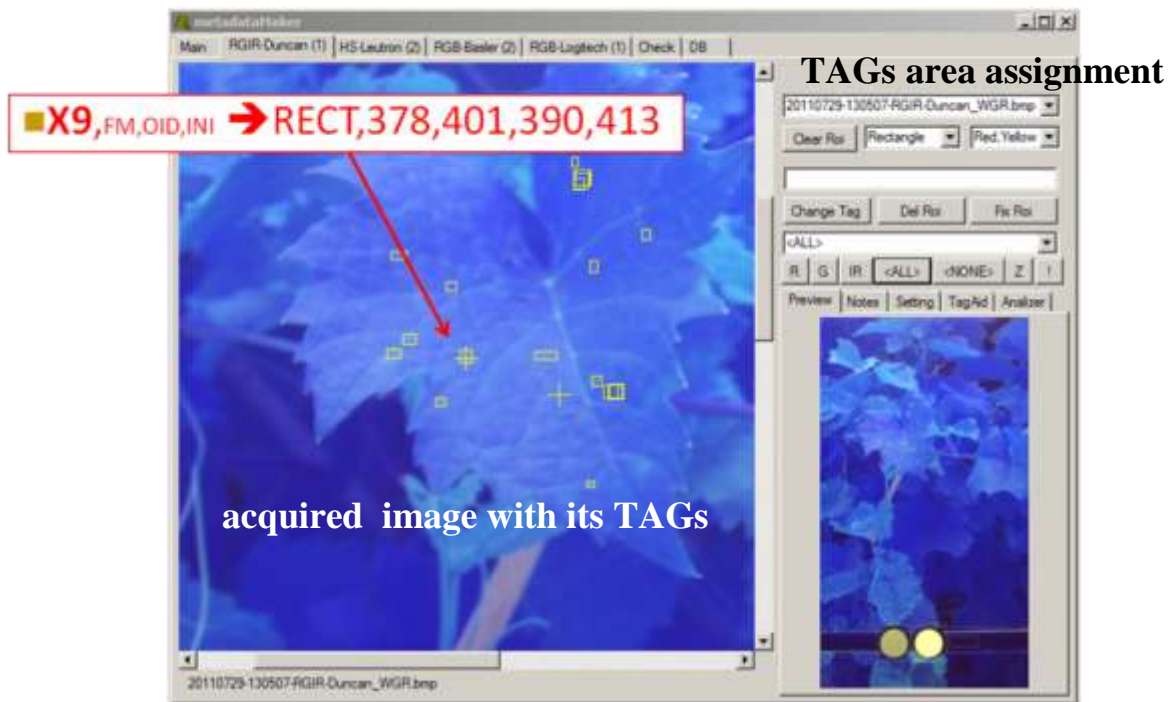


Figure 86 – The ROIs Selection software: ROIs selection and TAG assignment. The left column of the interface shows the acquired image on which you want to select the ROIs. Once you have highlighted a ROI, the right column of the interface allows you to associate its TAGs. In this example, the left column of the interface shows the acquired image and some of the TAGs that have been defined. The right column of the interface can also show the notes associated with the image and the multispectral camera settings during image acquisition.

### **C. Hardware Specifications**

To develop and test algorithms and applications was using a computer with the following hardware specifications.

Platform: Dell Precision M4400 (Dell Inc.)

Processor: Intel(R) Core(TM) 2 Duo CPU T9600 @ 2.80 GHz, 2 cores

Memory (RAM): 8.0 GB

System type: Windows 7 Professional (x64 edition)

Graphics Cards: NVIDIA Quadro FX 1700M