UNIVERSITÀ DEGLI STUDI DI CATANIA

FACOLTÀ DI AGRARIA

DIPARTIMENTO DI GESTIONE DEI SISTEMI AGROALIMENTARI E AMBIENTALI

SEZIONE MECCANICA

DOTTORATO DI RICERCA INTERNAZIONEALE IN

INGEGNERIA AGRARIA

XXIV CICLO

dott. ing. ALBA PENNISI

INTRODUCTION AND DEVELOPMENT OF INTELLIGENT SYSTEMS FOR PRECISION FARMING ACTIVITIES IN GREENHOUSE ENVIRONMENT

TESI PER IL CONSEGUIMENTO DEL TITOLO

TUTOR:

CHIAR.MO PROF. GIAMPAOLO SCHILLACI

COORDINATORE:

CHIAR.MA PROF.SSA CLAUDIA ARCIDIACONO

DICEMBRE 2011

Abstract

The main target of this research activity has been to develop an autonomous electrical vehicle able to perform precision farming tasks inside greenhouses. The vehicle can increase operations safety level for operators and quality of the chemicals distribution, allowing a lower environmental pollution and a better greenhouse product quality. Farmers thus obtain a return on their investment by using such a robot and technologies thanks to a waste reduction of phytosanitaries and fertilizers with a consequent cost lowering.

Moreover, due to constant reduction of electronic parts cost, like computer, sensors and power control system and to the presence on the market of new low cost devices for precision farming, the development of a dedicated autonomous vehicle is not as expensive as in the past. In this thesis new low-cost solution will be introduced and detailed.

Meanwhile, the robot has been equipped with an intelligent vision system in order to perform tomatoes detection in greenhouse and automatic guidance: the algorithm allows the tomato recognition and, as future development, once it will be integrated with the robot, it will be really useful to perform precision farming activities: fruit classification, harvesting, local chemicals treatment etc.

During a first step, the activity has been focused on the development and test of the electrical vehicle able to autonomously navigate along the greenhouse's rows. After the vehicle had been fully tested, great effort has been carried out on the development of the vision algorithm and sensor integration through the development of a versatile and high-modular software framework. Encouraging results will be shown and interesting perspective can rise from this analysis.

Acknowledgements

First of all, I want to express my gratitude to my professor, Giampaolo Schillaci who has been the ideal mentor. I will never forget the support that he gave to me when I had my job opportunity and it came to do the last part of the research in Florence. We both found the way to continue the research and carry it out. I would say that this thesis would not have been completed without his support.

Second of all, I am grateful to Prof. Giovanni Muscato who has provided me the opportunity of a contact for a PhD on such an interesting and challenging topic.

In addition, thanks to the thesis reviewer, Domenico Longo, for earnestly reading the thesis and his detailed comments on the manuscript in a really short time. Domenico has always played such helpful role in the project. He encouraged me to carry on with the research until the realization of Phd's.

I would like to thank all my friends and colleagues from DIEEI, GeSa and MICC for making laboratories, either in Catania or in Florence, fun and interesting environmental workplaces. My special thanks to Donato Melita (DIEEI) and Luciano Caruso (GeSa). They are not only my tutors but also friends.

In conclusion, I have to cheer my whole family because it trusted completely me.

My last words go to Fernando, above all for staying by my side. Our lives crossed with each other in the everyday life as well as job and research as it was a coincidence, but I am sure that it was not. I dedicate the work to him.

Poca osservazione e molto ragionamento conducono all'errore; molta osservazione e poco ragionamento conducono alla Verità

Alexis Carrel

Index

1 Introduction

Precision Farming is generally defined as information and technology based farm management system to identify, analyse and manage variability within fields for optimum profitability, sustainability and protection of the land resource. In this mode of farming, new information technologies can be used to make better decisions about many aspects of crop production. The goal is not to obtain the same yield everywhere, but to manage and distribute inputs on a specific site in order to maximize long term cost/benefit. Applying the same inputs on the entire field may no be the best choice.

Precision Farming is helping many farmers worldwide to maximize the effectiveness of crop inputs. To be viable, both economic and environmental benefits must be considered, as well as the practical questions of field level management and technologies needed.

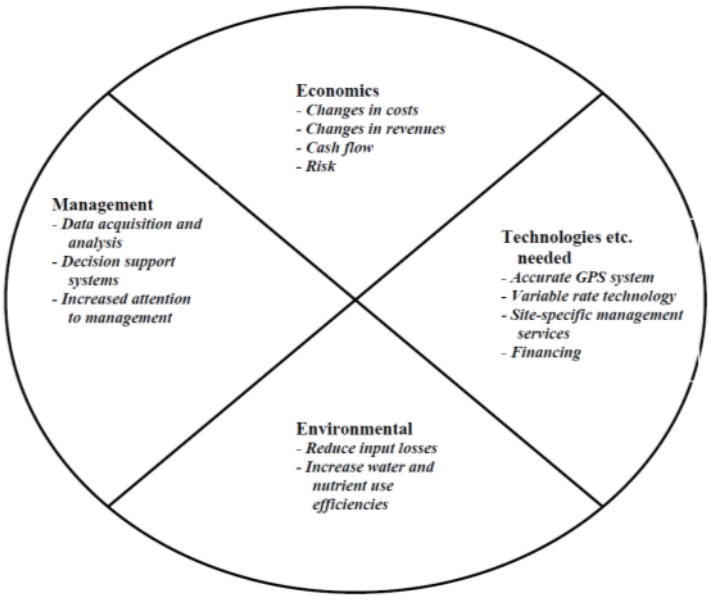

Figure 1 - Important issues related to precision farming (adapted from Davis, 2004)

The potential of precision farming for economical and environmental benefits could be visualized through reduced use of water, fertilizers, herbicides and pesticides besides the farm equipments. Instead of managing an entire field base upon some hypothetical average condition, which may not exist anywhere in the field, a

precision farming approach recognizes site specific differences within fields and adjusts management actions accordingly.

Precision agriculture offers the potential to automate and simplify the collection and analysis of information. It allows management decisions to be made and quickly implemented. In order to collect and utilize information effectively, it is important for everyone considering precision farming to be familiar with modern technological tools available. The vast array of tools includes hardware, software and the best management practices. These are briefly described in the following paragraphs.

For example, precision agriculture management practices can significantly reduce the amount of nutrient and other crop inputs used while boosting yields. Farmers thus obtain a return on their investment by saving on phytosanitary and fertilizer costs. The second, larger-scale benefit of targeting inputs—in spatial, temporal and quantitative terms—concerns environmental impacts. Applying the right amount of inputs in the right place and at the right time, it will benefit crops, soils and groundwater, and thus the entire crop cycle. Consequently, precision agriculture has become a cornerstone of sustainable agriculture, since it respects crops, soils and farmers. Sustainable agriculture seeks to assure a continued supply of food within the ecological, economic and social limits required to sustain production in the long term.

Precision agriculture therefore seeks to use high-tech systems in pursuit of this respectable and worthy goal.

The use of robotics is of great benefit for many industries, and for various reasons. For example, robotics can be used for tasks when there are concerns over human safety, or when the task is repetitive and can be done more productively by a robot working much longer hours than humans. And then there are times when robots simply offer a required level of precision that humans cannot provide.

The agricultural industry is no different in this regard. In order to remain competitive in what is now a global industry, farmers will have to be more productive, more efficient, and provide consistently good product. On top of this, the industry is also

suffering from a reduction in the available skilled and unskilled labour workforce. Robotic solutions leading to autonomous farming can be used to help.

The use of robots could significantly contribute to increase overall performances in intensive culture management and production efficiency, reducing costs, and, not least, to improve labour quality and safety. Particular attention should be focused on the last aspect: in agriculture, in fact, the development of autonomous systems able to remove or facilitate the operators in the workplace comes from the sensitivity to the issues of health and safety.

Robots can easily perform repetitive task, can undertake operations that are not possible with human operators since their cost in terms of time and/or required concentration is too high. Operations like precise fertilization and spraying of each single plant, precise mechanical weed control or harvesting specific procedures can be routinely performed with a robot that can also performs control tasks such as inspection and growth evaluation of each single plant. Meanwhile a robot can operate in a hazardous environment strongly reducing the exposition of human operators to dangerous chemicals.

1.1 The Goal

The project's main task is the integration of robotics technology and plant science; understanding and overcoming socio-economic barriers to technology adoption; and making the results available to growers and stakeholders.

Central to the present work is the development and use of an automated robotic platform that can autonomously navigate inside production greenhouses environments (especially tomatoes cultivation). The robot structure and architecture allows thinking on its reusability in other environment with similar structure of greenhouses, e.g. vineyards, for many and different applications.

Focusing always on all the safety aspects concerning the research, the described robot can carry sensors, instruments, farm implements, to automate or to augment production operations, including:

- Pesticide spraying
- Tomatoes detection for operation such as: crop load estimation, harvesting, precision spraying treatment.

After the robotic platform has been developed and sensors on board defined, most of the work has been focused on the second point and, with more detail, on the development of an intelligent computer vision system.

Computers have become ubiquitous in our daily lives. They perform repetitive, data intensive and computational tasks, more efficiently and more accurately than humans. It is natural to try to extend their capabilities to perform more intelligent tasks such as analysis of visual scenes – in brief the high-level tasks that we humans perform subconsciously hundreds of times every day with so much ease that we do not usually even realize that we are performing them.

In real human vision, the exact type, colour and viewpoint of a tomato is irrelevant to the decision that an object is a tomato. Similarly, we are able to detect people under widely varied conditions, irrespective of the colour or kind of clothing, pose, appearance, partial occlusions, illumination or background clutter. On the other side, computers are currently far behind humans in performing such analysis and inference.

Thus one goal of researchers working in computer vision and machine intelligence has been to grant computers the ability to see – visual analysis and interpretation of images or videos. One of the primary tasks is the detection of different classes of objects (in our case tomatoes) in images and videos. Such a capability would have many applications in the precision farming field.

This thesis introduces the problem of automate such repetitive tasks performed by humans and operators in harsh environment like greenhouses, discusses the challenges involved, and briefly presents the approach highlighting the contributions of the thesis.

1.2 Overview of the work

With the aim to facilitate workers in precision farming activities inside greenhouses environment, a versatile multifunction electrical vehicle with different automated subsystem has been developed.

The vehicle, named U-Go, has been built in order to be used as a multipurpose outdoor vehicle in different applications, so the control electronics have been designed to provide different choices for control modalities. The simplest one is the teleoperated modality. In this situation a remote user, using a joystick, can send direct commands to the robot in order to move forward, backward or turn left or right at different speeds. In order to allow the totally autonomous operations in the greenhouse, different sensors have been tested.

Among all, this work targets the problem of the implementation of a machine vision allowing tomato detection in images. For a more precise definition of our goal we can view an object detector as a combination of two key building blocks: a feature extraction algorithm that encodes image regions or parts of videos as feature vectors, and a detector that uses the computed features to provide object/non-object decisions (tomato or not tomato).

Differently from the approach currently used, (see §2.3) our contribution is focused on a learning phase. The system is data-based. It learns what a tomato is and detects it on a test image through a classification method (support vector machine is used). Consider the example of recognizing handwritten digits, illustrated in Figure 2. Each digit corresponds to a 28×28 pixel image and so can be represented by a vector x comprising 784 real numbers. The goal is to build a machine that will take such a vector x as input and that will produce the identity of the digit 0, . . . , 9 as the output. This is a nontrivial problem due to the wide variability of handwriting.

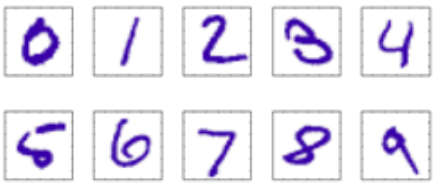

Figure 2 - Example of hand-written digits taken from US zip codes

It could be tackled using handcrafted rules or heuristics for distinguishing the digits based on the shapes of the strokes (as in the literature overview) but in practice such an approach leads to a proliferation of rules and of exceptions to the rules and so on, and invariably gives poor results.

This is fundamental to creating efficient object (tomato) detectors as unlike text documents where two words either match exactly or are different, matching or categorising objects in images is inherently ambiguous. Many factors contribute to this ambiguity, including the image formation processes, variations in illumination, partial occlusions, intra-class variation and context (Dalal 2006).

Due to the dataset sensitivity of the used algorithm, as much bigger is the dataset as the machine vision will provide precise detection.

The introduction of a simple and robust image descriptor simplifies the classification task allowing tomatoes to be discriminated easily with less training data and less sophisticated learning methods.

1.3 Challenges

In order to achieve our goal, the first challenge faced was to develop a robotic platform able to navigate autonomously along the environment path. In this context, an important role has been played by the choice of a robust algorithm for the navigation and the sensors mounted on board.

Related to the sensors, a cost-benefices analysis has been taken into account and in this perspective new low cost solutions have been designed and developed.

A second challenge has been the choice of an innovative computer vision approach allowing us to achieve the second goal: the development of a system vision for tomato detection. Finally, the choice of a detector applicable to our case.

The foremost difficulty in building a robust object detector is the amount of variation in images.

Figure 3 - Classical view of a tomatoes row

Several factors contribute to this:

- Firstly, the image formation process suppresses 3-D depth information and creates dependencies on viewpoint such that even a small change in the object's position or orientation respect to the camera centre may change its appearance considerably. A related issue is the large variation in scales under which an object can be viewed. An object detector must handle the issues of viewpoint and scale changes and provide invariance to them.
- Secondly, background clutter is common and varies from image to image. The detector must be capable of distinguishing object class from complex background regions.
- Thirdly, object colour and general illumination varies considerably, for example directs unlighted and shadows during the day to artificial or dim lighting at night. Thus a robust object detector must handle colour changes and provide invariance to a broad range of illumination and lighting changes.
- Finally, partial occlusions and overlapping create further difficulties because only part of the object is visible for processing.

Figure 4 (left and right) provides two classical view of tomato rows in a greenhouse (robot view) illustrating the difficulties listed above. Note the amount of variation in the images, in particular, the variations in appearance, colour and illumination.

Figure 4 - Tomatoes row view

1.4 Outline of Dissertation

This chapter introduces an overall system description, the key applications of our project and presents the overall goals of the thesis. It then continues with a brief background of robotics application in greenhouse, outlines our approach to the problem, and summarizes our main contributions. The remaining chapters are organized as follows:

- Chapter 2 describes the state of the art in precision farming greenhouses activities focusing particularly on those performed by using computer vision support.
- Chapter 3 presents a high-level overview of our project. It does not give implementation level details but it describes the overall system platform and provides an introduction to the vision system architecture.
- Chapter 4 deeply describes the robotic platform has been developed and on board sensors, showing the tools used and giving a motivation for the attempted choices.
- Chapter 5 presents the algorithm vision architecture and the developed vision algorithm.
- Chapter 6 shows and give an analysis of the obtained results.

2 State of Art

2.1 Robotics in Greenhouses: social impact

Nowadays a great awareness is spreading rapidly among the farmers concerning safety aspects.

It is clear from literature as the workplace and the activity performed inside a greenhouse are not always performed in safety conditions (Balloni *et al.* 2008a, Balloni *et al.* 2008b, Schillaci *et al.*, 2009). This due to the: "hard" environment (humidity, hot, dirty etc), the material and machines managed, wrong procedures and not adequate training of the personnel involved, etc..

Greenhouses, Figure 5, are characterised by small volumes of continuously recycled supporting resources, and have critical epidemic average ratios, enhanced by moisture and lighting continuance (for farming effectiveness) favouring the spread of biotic agents.

Greenhouses are translucent glass or plastic constructions for hastening the growth of plants. The distribution of plants inside greenhouses usually consists of an alternation of double rows of plants and narrow corridors for human operation and walkway. This kind of agricultural technique is massively used for intensive production of horticultural products in regions with adverse natural climatic conditions, since it allows a more effective use of water and daylight.

Figure 5 - Greenhouses environment

The favourable atmosphere created inside greenhouses for plant growth causes pests and undesirable organisms to thrive as well, making necessary the use of pesticides and other chemical products that must be sprayed directly on the plants (Acaccia *et al.* 2003, Pasinetti 1952)

Today solutions massively depend on heavy chemicals, plentifully distributed at given time intervals, making the greenhouse indoors highly toxic, with operator health shocks and forbidden re-entry long lasting delays.

The automation of spraying, as well as other greenhouse operations like monitoring and control of environmental conditions, harvest support, plant inspection, and artificial pollination, has a dramatic social and economical impact (Acaccia *et al.* 2003).

Recent studies reported confirmation that spraying operations have hazardous effects on the health of knapsack sprayer human operators, who are specially exposed when working inside greenhouses, in conditions of high temperature and poor ventilation (Acaccia *et al.* 2003).

Different studies were conducted about risk connected with preparing and spraying to high temperature and humidity. The problems related to a prolonged phytosanitary exposure are constantly studied and till now are considered indisputable (Aprea *et al.* 2011, Giambattistelli 2003, Jovita *et al.* 2011).

In Capri *et al.* 1999 the aim of the study was both to measure potential dermal and inhalation exposure in the greenhouse and to evaluate the risk to workers in such a scenario.

During pesticides spreading operations into greenhouses sited in Mediterranean area many workers don't use all the PPD prescribed by law because the high temperature and humidity. Use of vehicles and of a full helmet endowed with forced ventilation and air filtration devices for greenhouse agrochemical application appears to be related to greater comfort and better operator ventilation (Balloni *et al.* 2008d).

In Italy the agricultural sector is in second place as regards accidents, just after the building industry. There are many reasons for this but it can be said that human error

is often the cause. Carelessness, non-observance of safety regulations, progressive aging of the operators, poor technical training of employers and operators and lack of maintenance of safety devices are the main causes of accidents in agriculture. To these key elements, can be added the fact that many of the machines used daily in the fields were designed for completely different purposes or are now obsolete; often are also home-made by operators.

Due to these reasons in agriculture many accidents involve agricultural machinery. Of about one million, seven hundred and eighty thousand registered machines, at least five hundred thousand do not have the requisites to be in circulation and thus endanger the safety of the operators and others. This data is quite shocking in that it means that about one machine out of three of those used daily in the fields is obsolete and, no longer having the necessary requisites for use, should be taken out of circulation (Balloni *et al*. 2008a)

In the province of Ragusa, many of the examined machines in Balloni *et al*. 2008a are without bonnets and or chain guard, or they are not in good conditions. It appears those workers and often the employers and consequently the management are not greatly involved in safety aspects. To make adjustment to machineries and to form and to motivate workers are all activities that employers often consider as additional costs. A great benefice and support can come from the adoption of automated or robotic solutions. Actually the use of robot in every environment requires trained/skilled personnel, differently its usage can arise other issues. The aim should be to provide reliable and safe solutions that not require mechanical and electronic proficiency to be used.

In Balloni *et al.* 2008b, Nuyttens *et al.* 2005, Cerruto *et al*. 2007, Schillaci *et al.* 2009a, the results showed a large reduction in operator dermal exposure using the prototype of a self-propelled sprayer suitably designed to properly work in greenhouses with respect the normal spray lance.

In addition to the supplemental security issues, even a costs analysis may be advanced (Moltó *et al.* 2001). Pesticide treatments make up 30 and 42% of the production costs. One reason for the high cost of these treatments is due to the high

percentage loss of pesticide out of the canopy, due to the globular shape of their canopies and to the application of chemicals in the absence of trees.

The mechanized harvesting of specialty crops, such as tomatoes, is becoming necessary as labour costs continue to rise and as the availability of labour decreases.

In any production system there is a growing need to obtain higher quality products at a lower cost in order to be competitive. One solution to this challenge is the development of automatic systems that replace manpower in tasks when a person performs worse than an automatic device in terms of precision, repetitively and working cycle. Probably, harvesting is the process that has received the least amount of technological development for satisfactory automation (Jiménez *et al.* 2000).

The applications of instrumental robotics are spreading every day to cover further domains, as the opportunity of replacing human operators provides effective solutions with return on investment. This is especially important when the duties, that need be performed, are potentially harmful for the safety or the health of the workers, or when more conservative issues are granted by robotics.

2.2 Overview of Robotics platform for Greenhouse application

It is well known that the introduction of robotics in agriculture has not had the same success as it has in the manufacturing industry. This is mainly due to the fact that the agricultural environment is much less structured and consequently, it is more difficult to adopt robots in the automation of different agricultural processes (Muscato *et al.* 2005). The object, with which machines have to interact, have irregular size, location and shape; operating environment is quite hostile (humidity, dirty, etc) and drivable path are not well defined with possible unexpected obstacles

As previously mentioned, greenhouse activities often require hours of hard-work made by operators. Many of these works can be also very dangerous and uncomfortable because of chemicals, high temperature and humidity. With this environmental condition, even common agricultural operation can became heavy and stressful. Moreover, because of high temperature and humidity, operators often do not correctly wear safety clothes, increasing health risk (Sammons *et al.* 2005). In this condition an assist machine could be useful for operators in order to alleviate the work load.

Automation is a need for agriculture operations because of the reduction in the labour and increasing cost. There are numerous research investigations reported in literature in the area of robotic application in agriculture.

But frequently agricultural tasks are carried out in an outdoor environment and this involves:

- more expensive design and realization of the mechanical structures, because they have to be waterproof, dust-proof and resistant to anything that is present in a non-protected environment;
- the adoption of high-performance and outdoor-proof sensors and actuators;
- the development of elaborate, flexible control laws in order to cope with a great variety of conditions.

One of the biggest challenges is in developing and integrating robotic solutions into the farming landscape. The more unstructured and uncertain the environment is the

more machine intelligence is required to achieve any required precision in farming operations.

From the robotics viewpoint, for this reason, there are many advantages to use a typical greenhouse environment such as: controllable and/or partially known position of the plants, controllable shape of the plants (e.g. the growing direction and the height) and ground floor (at least in the driveable surface) more regular than in the open field etc. (Dario *et al.* 1994).

The robot locomotion mobility is performed on a plane and can be obtained through a narrow mobile platform or by track rails suitably joined to the greenhouse ceiling structure. Both the solutions have been examined and compared (Acaccia *et al.* 2003).

Due to their high cost and huge impact on the greenhouse, fixed or "invasive" solutions are not so common. Actually these systems are adopted in plant nursery greenhouses where it can be easy to spray chemicals from the ceiling. In any case environmental issues must be considered as chemicals pipe can have leakages, releasing dangerous substances in wrong places. Anymore, huge amount of toxic fluids are required to fill-in long pipes with non-trivial problem of post-spraying flushing with clean water. Instead, because of greenhouses environments are highly structured and regular with respect to the open field, they are well suited to be operated by automatic machines that do not implies much fixed cost for each greenhouse. Moreover, automatic machines can be re-used in different place and can solve tasks other than spraying.

Definitely, the high risk level for operators has driven different research group to find some solutions like autonomous or teleoperated robotics systems.

The available solutions are costly and insufficiently robust, the architecture or the software used is often old or heavy and they have limited working capabilities. So, in general, till this moment the potential of robotics has not been fully exploited in agricultural and there aren't robots, to be created and commercialized, as dexterous and skilful as trained workers (Gay *et al.* 2008).

The main design parameters for a robot that have to work in a greenhouse environment are analysed (Gay *et al.* 2008, Belforte *et al.* 2006, Blackmore *et al.* 2002). In Acaccia *et al* 2003 and Gay *et al.* 2008, the problem of the introduction and diffusion of robotic was faced. Analysing carefully the reasons that have prevented the employment of robotic systems in agriculture so far some important guide lines to design agricultural robots was introduced.

Among these, the capability to perform different tasks appears the most important feature for a robotic system conceived to operate in the agricultural context. Moreover, greenhouses are indeed the most suitable agricultural environment to the introduction of robotic systems for several technical, safety and economical reasons.

In Pedersen *et al.* 2008, economical impact for different automatic approach in agriculture for different parameters (fuel consumption, labour costs, autonomy, chemicals cost, maintenance) is analysed against manual approach.

Different autonomous machines have been developed for similar tasks in open field (Slaughter *et al.* 2008, Blasco *et al.* 2002).

In the last decade, different research group have been interested on these issues. The Aurora robot (Spain) is able to perform different tasks in an autonomous way with remote supervision (Mandow *et al.* 1996). In order to solve navigation problem, some visual feedback have been used (Dario *et al.* 1994).

At University of Genoa a project named “Mobile robots in greenhouse cultivation: inspection and treatment of plants” (Acaccia *et al.* 2003) has been developed. The paper presents a service robot for health monitoring and localized chemical, drugs and fertilizers dispensing to plants in greenhouses. The robot and its end-effectors have been conceived and designed specifically oriented to the specific environment and tasks. Only a virtual prototype has been realized and the authors suggest solution in order to solve different tasks: mobile or suspended solutions are compared, discussion about the use of an arm with an end-effector, a gripper equipped with a system vision to detect the fruits.

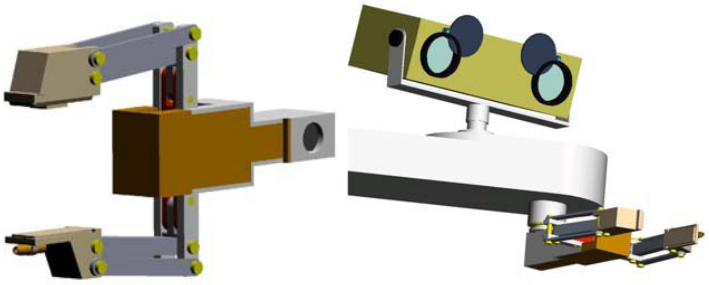

Figure 6 - The gripper and the eyes-hand system

Among the several sensor devices used for robot navigation, vision provides the richest source of information. In Palamas *et al.* 2006, a technique for autonomous navigation in a greenhouse pathway using pixel based image segmentation is presented.

In literature, different automatic guidance systems that allows reducing driver stress and a more relaxed working and an efficient use of machines and resources, were presented. Autoguidance, also called auto-steer, of tractors and agricultural machines that is based on a global navigation satellite system (GNSS) represents one currently available technology that can provide significant benefits for crop production in different growing environments. Today, numerous farmers have suspended the use of conventional markers from their operations and rely on cost-effective alternative methods to steer their farm equipment based on continuously measured geographic coordinates. There are several companies that sell such GPS systems for tractors or machines and other automated solutions (e.g. sprayers) (Deere, Trimble, Arvatec). Some papers highlight different approaches for guiding a vehicle using a Differential Global Positioning System (DGPS) based position sensor as the only external posture sensor (Buik *et al.* 2006, Heraud *et al.* 2009). Some machine adopts multiple guidance sensors (Holpp et al. 2006, Li M. *et al.* 2009, Wan et al. 2008, Murakami *et al.* 2008, Toru 2000).
In this thesis, an innovative solution, for robot autonomous navigation in tomatoes greenhouse cultivations, is described.

The next sections focus on the past and current solution related the development of greenhouse robotic platform. Particular attention has been taken in those equipped with system vision as support of precision farming activity.

Spraying robot

Different engineering solutions to the current human health hazards involved in spraying potentially toxic chemicals in the confined space of hot and steamy glasshouses are presented: from the two different multi-purpose robotic cells prototypes (Gay *et al.* 2008), through robots capable to realise the potential of steel pipes as a method of guidance in the greenhouse (Sammons *et al.* 2005), to the development of robots totally autonomous capable to work in every greenhouses (Mandow *et al.* 1996, Balloni *et al.* 2008c).

Figure 7 - Fixed point robotic cell operating in the greenhouse

A mobile platform for greenhouse chemicals spraying has been developed at University of Almeria (Sanchez *et al.* 2010). In this work, specifications for a greenhouses robot are first identified then the complete machine (named FitoRobot) has been built with ultrasonic sensors for the motion between plants rows. The machine is driven by an internal combustion engine. A commercial machine, named Fumimatic 400 (Fiuminatic), is available in Spain. It is not autonomous nor teleoperated but it is a complete spraying machine with a powerful Diesel engine and a 400 l tank for chemicals. In the area of Vittoria (RG), Italy, a local SME build and sold a small tracked vehicle with thermal engine. On board is mounted a complete spraying system; the machine is named “Vanco Spray” (Balloni *et al.* 2008c, Balloni *et al.* 2008d, Schillaci *et al.* 2009a, Schillaci *et al.* 2009b).

Cho and Ki (Cho *et al.* 1999) used fuzzy logic and machine vision technology for an autonomous speed sprayer for orchards. The input information to the fuzzy logic

controller is given by both machine vision and ultrasonic sensors. The vehicle heading was decided from the machine vision image while the range data, between the object and the sprayer, was determined from ultrasonic sensors.

Shin and Kim (Shin *et al.* 2001) developed an autonomous guidance system for a small orchard sprayer with ultrasonic sensors using artificial stainless steel pipe targets, due to its strong reflection of ultrasonic waves.

Misao (Misao *et al.* 2001) used image processing for an automatic steering system with red target boards to guide the power sprayer. Video camera was used to acquire the image of 3 the travel path including the red board targets. Image processing was used to determine the distance from the current vehicle position to the target, and then the actual position to the desired vehicle path was compared and corrected by automatic steering control.

The path in greenhouse is usually straightness to increase plant area, so high guidance precision is strictly demanded especially at the comer. The guidance control strategy in Fujuan *et al.* 2010 can deduce the heading speed of the robot when turning the comer and increase the guidance precision. Electromagnetism guidance method is adopted. In the spraying control system, the spraying pole can fold to avoid touching the strut and the greenhouse wall and stop spraying to save pesticide when the robot arriving at the end of the field and lay down a certain angle to spray when the robot arriving at the path in the field.

In Zhang *et al.* 2005, the electromagnetic induction mode is chosen as navigation methodology. The selection of spraying is relied on many factors, for example: the type of the plant, the kind and grade of the insect pest and the kind of the pesticide. The spraying mode could be chosen according to different situations, the robot is versatile. A signal switch is used for switching on or off induction wires so as to make the robot change the path according to task's requirements. A drawback of this work has to be identified in the operation of laying wires in the greenhouse underground; this causes big effort and cost every time the environment or greenhouse changes (not flexible solution).

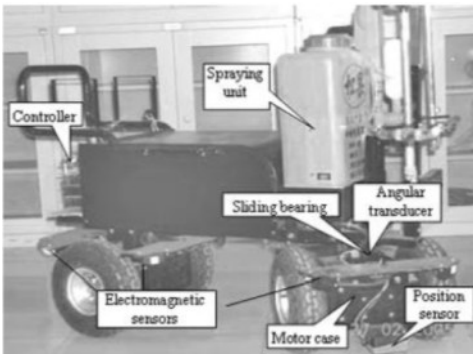

Figure 8 - Robotics platform

Harvesting robot

In large-scale greenhouse production, technological developments can reduce production costs; mechanization of crop maintenance and harvesting is one desirable way to accomplish this. In Hayashi *et al.* 2005, authors developed, about this context, three types of harvesting robot. The prototype strawberry-harvesting robot could judge maturity and make basic harvesting movements. The eggplant-harvesting robot achieved a harvesting rate of 29.1%, averaging 43.2 seconds per fruit.

Figure 9 - Strawberry and eggplant harvesting robot

In Van Henten *et al.* 2002, the author describes the concept of an autonomous robot for harvesting cucumbers in greenhouses. The paper focuses on the individual hardware and software components of the robot. These include, the autonomous vehicle, the manipulator, the end-effector, the two computer vision systems for detection and 3D imaging of the fruit and the environment and, finally, a control scheme that generates collision free motions for the manipulator during harvesting.

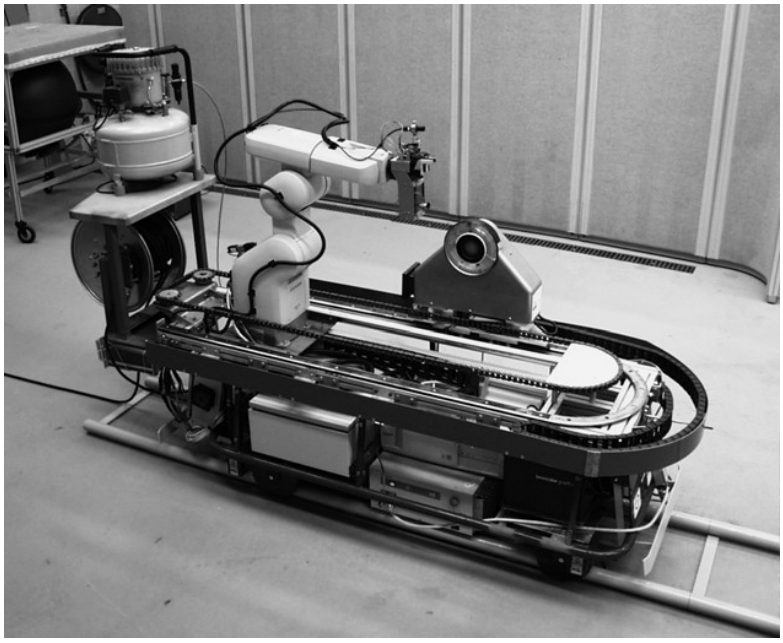

Figure 10 - A functional model of the harvesting robot.

The Kitamura *et al.* 2005 paper describes recognition and cutting system of sweet peppers for picking robots in greenhouse horticulture. This picking robot has an image processing system with a positioning system for two cameras and cutting a device to follow the sweet pepper by visual feedback control.

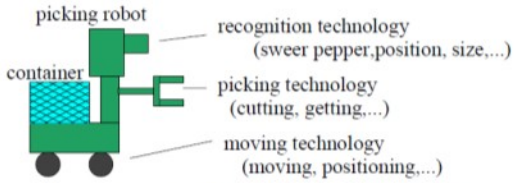

Fig. 3: Conceptual illustration of picking robot

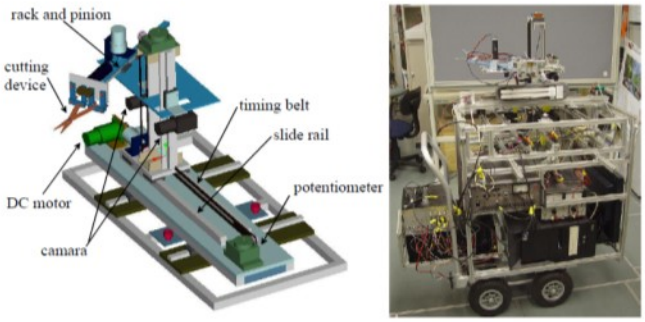

Fig. 4: Structure of picking robot

Figure 11 - Structure of picking robot

2.3 Precision farming based on computer vision

Agricultural automation may take advantage of computer vision resources, which can be applied to a number of different tasks, such as inspection (Brosnan *et al.* 2002), classification of plants (Tang *et al.* 2003, Neto *et al.* 2003, Steward *et al.* 2004), estimated production (Annamalai *et al.* 2004), automated collection (Plebe *et al.* 2001) and guidance of autonomous machines (Lulio *et al*. 2009).

In Sandini *et al.* 1990, the authors have classified three different main area of intervention related this field:

1. Navigation control;
2. Grading and quality control; (Blasco *et al.* 2003 and 2009)
3. Monitoring and localization of important objects;

Different approach of image processing can be applied depending on the assigned task. Related the first point, in this thesis, a new approach has been developed (see §4.7).

The homogeneity and appearance of the fruits have significant effect on consumer decision. For this reason, the presentation of agricultural produce is manipulated at various stages from the field to the final consumer and is generally oriented towards the cleaning of the product and sorting by homogeneous categories (Blasco *et al.* 2003b).

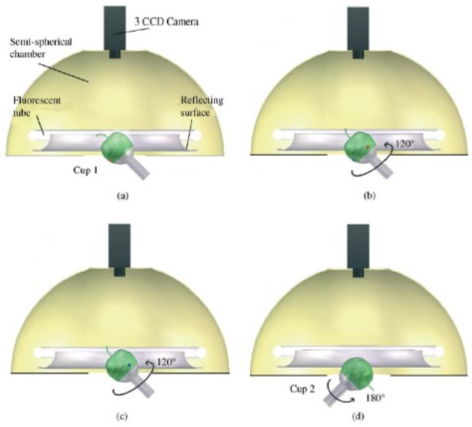

Figure 12 - (a) Acquisition of the first image; (b) acquisition of the second image - cup 1 rotates the fruit 120°; (c) acquisition of the third image - cup 1 rotates the fruit other 120° and (d) acquisition of the fourth image - cup 2captures the fruit and rotated 180°

The approached method (segmentation) is fast and appropriate for on-line processes, but depends much on the colour of the objects to be inspected. In fact, many factors can invalidate the results: illumination, colour, highlights, etc.

In Gomez-Sanchis *et al.* 2008, the study proposes a method for correcting the adverse effects produced by the curvature of spherical objects in acquiring images with a computer vision system. Its suitability has been illustrated in a specific case of citrus fruits. The images of this kind of fruit are darker in areas nearer the edge than in the centre, and this makes them more difficult to analyse.

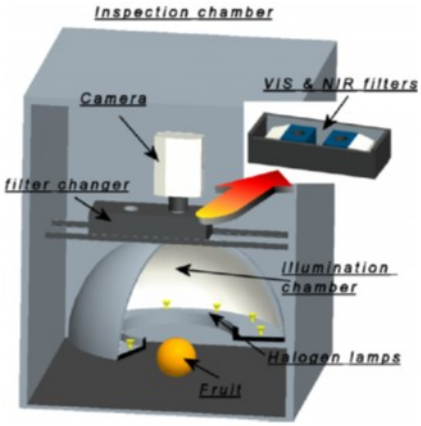

Figure 13 - Diagram of the image acquisition system

This methodology considers the fruit as being a Lambertian ellipsoidal surface and produces a 3D model of the fruit. By doing it becomes possible to calculate the part of the radiation that should really reach the camera and to make the intensity of the radiation uniform over the whole of the fruit surface captured by the camera, no matter what region is being sampled. Some tests have been carried out in order to prove that using the proposed correction methodology the reflectance in all the surface of the fruit is similar, minimizing the differences from the central area to the peripheral areas.

Relating to the second point, monitoring and localization of important objects inside a greenhouse, since there is a clear tendency of reducing the use of chemicals in agriculture, numerous technologies have been developed trying to obtain safer agricultural products and lower environmental impacts.

One of the precision farming activities that involve machine vision system is automatic weeding strategy (Blasco *et al.* 2002). The development of image processing methods to discriminate between weed, crop and soil is an important step (Xavier *et al.* 2008). Automatic weeding strategy could minimize the volume of herbicides that is sprayed to the fields. Image filtering technique (Ghazali et al. 2008) or segmentation (Jafari *et al.* 2006) has been used to process the images as well as a feature extraction method to classify the type of weed images.

In Quan *et al.* 2006, a semi-automatic technique for modelling plants directly from images is proposed.

Most of them focus on segmentation or filtering image processing technique based on the difference of the color existing between the fruit, the crop, the soil and so on (Haff *et al.* 2006, De Mezzo *et al.* 2007, Pajares *et al.* 2007, Xia *et al.* 2009).

In Svensson *et al.* 2002, the aim of the project has been to investigate the possibility of using image processing as a tool to facilitate grapevine management, in particular shoot counting and assessment of the grapevine canopy.

In Braun *et al.* 2010, a novel colour-based recording and evaluation technique is presented which allows determining the spatial distribution of foliage along approximately planar vine rows with high accuracy. The approach uses optical flow tracking to construct a composite from an image sequence and user-trainable bayesian colour classification to segment foliage from distinctively colored canvas carried along behind the row under evaluation.

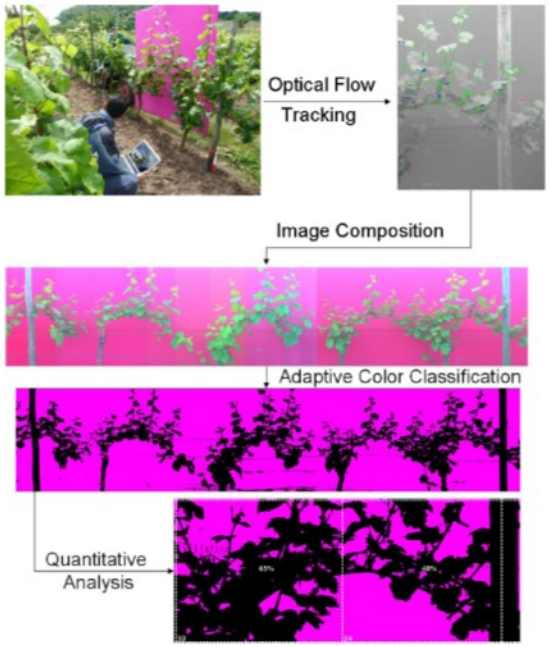

Figure 14 - Estimation process of the vineyard foliage

In De Mezzo *et al.* 2007, the author proposes a new method of pest detection and positioning based on binocular stereo to get the location information of pest, which is used for guiding the robot to spray the pesticides automatically.

In the context of monitoring and localization of important objects in the greenhouses environment, one of the most performed activities is the localization and detection of the fruit. It could guarantee a better phytosanitary product/pesticide distribution, harvesting procedure, detecting illness or pest diffusion, quality and growth control.

Different components make difficult such operation: occlusion of the leaf, brightness, huge variety of horticulture typology etc.

The work reported in Jiménez *et al.* 2000, is a review of previous studies that try to automate the fruit location finding process on trees using computer vision methods. The main features of these approaches are described, paying special attention to the sensors and accessories utilized for capturing tree images. The work focuses on different aspects: the image processing strategy used to detect the fruit, the results obtained in terms of the correct/false detection rates and the ability to detect fruit independent of its maturity stage.

The majority of these works use CCD cameras to capture the images and use local or shape-based analysis to detect the fruit.

Systems using local analysis, like intensity or colour pixel classification, allow for rapid detection and were able to detect fruit at specific maturity stages, i.e. fruit with a different colour from the background. However, systems based on shape analysis were more independent of hue changes and were not limited to detecting fruit with a colour different from the background one. Nevertheless these algorithms were more time consuming.

The approaches using range images and shape analysis were capable of detecting fruit of any colour, did not generate false positive and gave precise information about the fruit three-dimensional position. In spite of these promising results, the problem of total fruit occlusion limits the amount of fruit that can be harvested, depending on fruiting and viewing conditions.

2.3.1 Fruit detection in greenhouses

Aiming at realizing cucumber identification and location for cucumber harvesting robot in greenhouse, a segmentation algorithm for cucumber image was presented (Van Henten *et al.* 2002). Choosing an initial threshold to segment the images, it was dynamic revised based on shape characteristics of cucumber fruits after the initial segmentation threshold was judged. The pixel region which did not belong to fruit was removed

Figure 15 - (a) Original Images (b) Image after segmentation (c) Images after post-processing

Fruit detection is an intricate problem because the green cucumber fruit have to be found on a green background (Van Henten *et al.* 2002). From the two main

approaches, (a) recognition based on shape and (b) recognition based on spectral properties, the last seems to be the most promising.

The reasons for not detecting the fruit were mainly caused by illumination problems (reflection, flash intensity too high or too low), problems with separating two cucumbers, problems with separating fruit and stem, and by cucumbers completely hidden behind leaves.

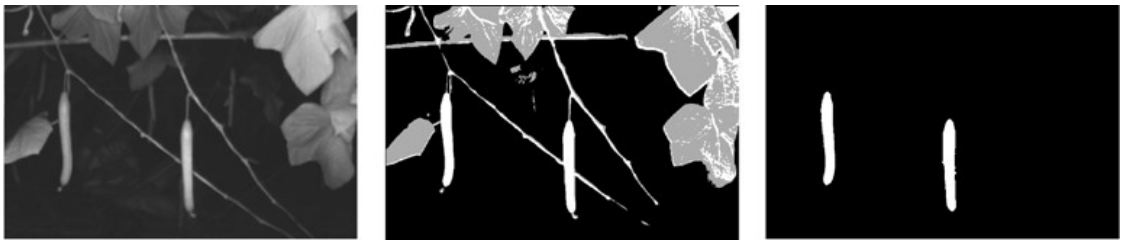

Figure 16 - Detection of cucumber fruit using computer vision. Original image taken by the camera mounted on the vehicle (left), segmentedimage (middle) and detected fruit (right).

In the image-processing algorithm of Kitamura *et al.* 2005, the fruit of the sweet pepper is recognized by binarization of HSI colour specification. HSI colour specification system is one of colour image expressions; it consists of three images that are hue, saturation, and intensity. However, this algorism cannot recognize fruits without enough lighting.

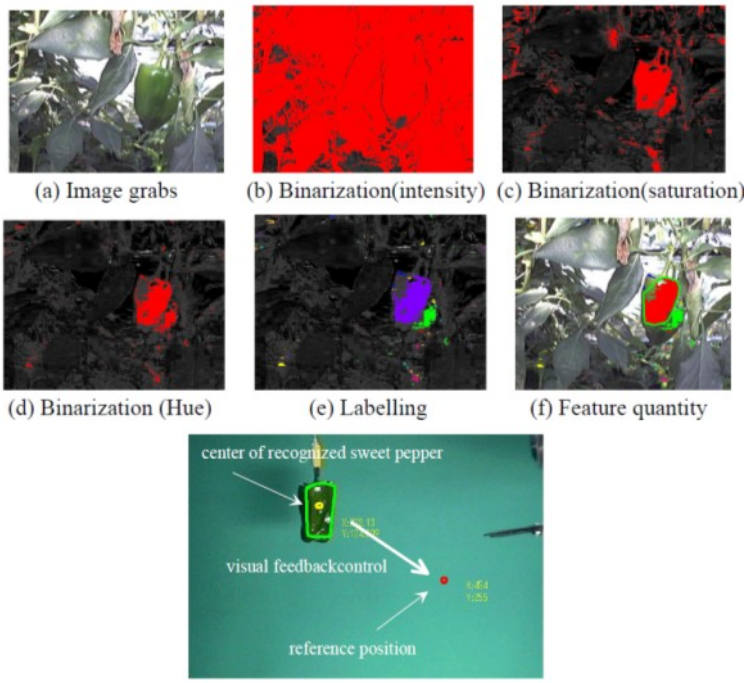

Figure 17 - Pepper detection flow

Most of the works, for developing their project, they have chosen the productive cycle of tomato plants in a greenhouse environment since it presents a sequence of cultural operations (as explained below) rather diversified and not banal. At the same time tomato plant is characterized by certain regularity in its structure that could partially simplify the development of cultural operations by means of a robotic system and so it is a valid test.

Tomato's fruit generally doesn't ripe simultaneously. On each tomato plant, green, yellow, orange, and red tomatoes can be found. Tomato can be detected basing on: colour, shape or highlights.

In Whittaker *et al.* 1987, digital image analysis with a modified circular Hough transform, proved to be able to locate tomatoes based on shape and not colour, even when the scenes contained substantial background noise and the fruit were partially hidden from view.

The circular Hough transform is indeed an acceptable method for locating and identifying tomatoes in an image that has been acquired under natural field conditions. The circular Hough transform method does not depend upon a colour difference between the fruit and the foliage. It has the ability to operate in the presence of natural "noise" in an image such as bright spots caused by specular reflectance and shadows cast by plant leaves onto the tomato. Finally, the circular Hough transform is valid for situations in which the perimeter of the tomato is partially occluded. The algorithms presented in the research are computationally intensive on a serial processor and cannot at the present time be performed in real time.

In Arefi *et al.* 2011, the harvesting robot was designed to have the ability to detect all the types of tomato colours and pick up only the ripen ones. Because colour range of UT and RT (Ripe Tomato and Unripe Tomato) is close together and colour of RT is not uniform (there is yellow- red colour pixels in a RT), no appropriate algorithm has been reported yet to detect the RT on the plant. Hence, the object of this study was to introduce and develop a new algorithm for recognition and localization of RT from

the background (green, branches, leaves and greenhouse space) and UT (yellow and orange tomato) based on the colour quantification and shape of fruit.

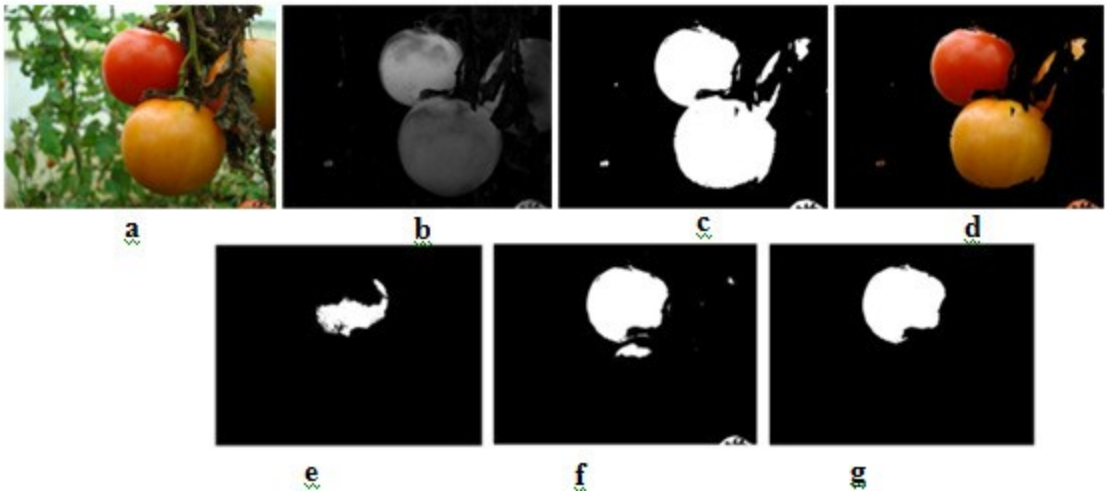

Figure 18 - Typical RT recognition. (a) Original color image (b) Gray image (c) Binary image (d) Image after removing the background (e-f) Extraction of red pixels, and g. RT recognition.

In Hayashi *et al.* 2005, authors have developed a stereoscopic vision system that relies on highlights (Figure 19). The system consists of a halogen light, two cameras, a travelling carriage, and a computer. Figure 19 shows the method of detecting ripe tomatoes in a cluster. First, the stereoscopic vision system detects the red area from the R-G images. Second, the highlights of each fruit inside the red area are separately detected. Finally the system calculates the fruit position in three dimensions from the stereoscopic images.

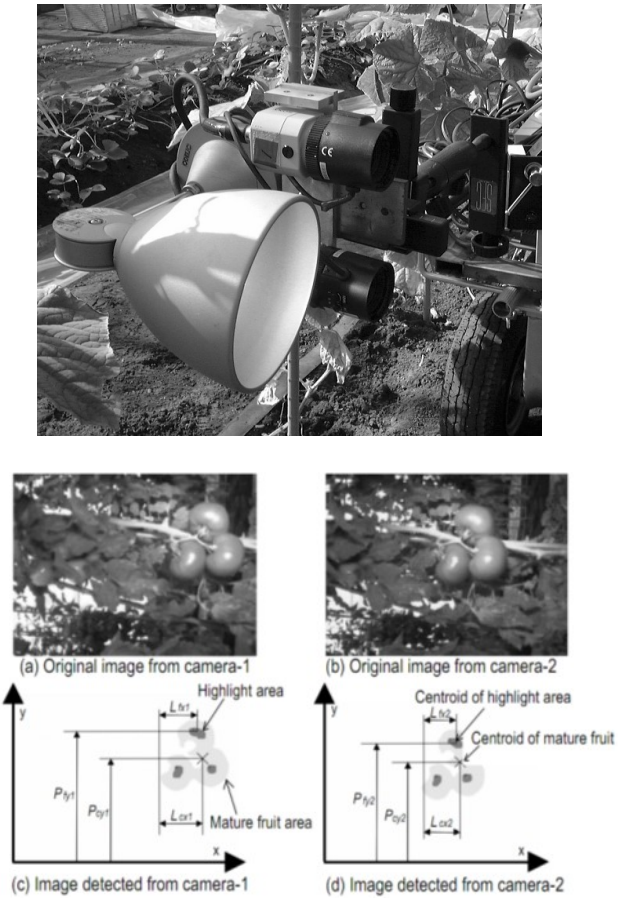

Figure 19 - The stereoscopic vision system and the method of detecting red tomatoes in a cluster

In Yin H. *et al.* 2009 a robotic system for harvesting tomatoes in greenhouses is designed. Effective recognition of ripe tomatoes from complex background is the key technology of the harvesting robotic system. In this work, the colour feature of ripe tomatoes is employed. The ripe tomato is segmented by K-means clustering using the L*a*b* color space. To extract a single integrity ripen tomato, mathematical morphology method is used to de-noise and handle the situations of tomato overlapping and shelter.

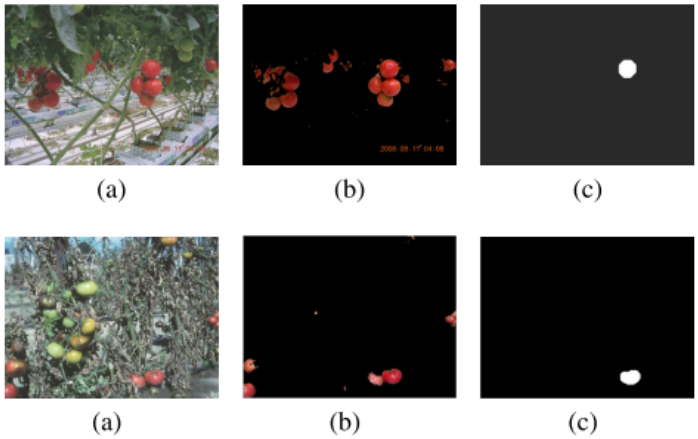

Figure 20 - The tomato with complex background. (a) The input image; (b) The segmented tomato; (c) The extracted tomato

In Kondo *et al.* 2008, a research on a tomato fruit cluster harvesting robot was started to automate the whole process. It consists of a system for artificial vision, a manipulator and an end-effector.

This machine vision system consisted of two identical colour TV cameras (VGA class), four lighting devices with PL (polarizing) filters, and two image capture boards. Two images were acquired at a time and RGB colour component images were converted into HSI images. Using colours on the HSI images, main stems, peduncles, and fruits were discriminate and an end-effector grasping point on the main stem was recognized based on physical properties of the tomato plant.

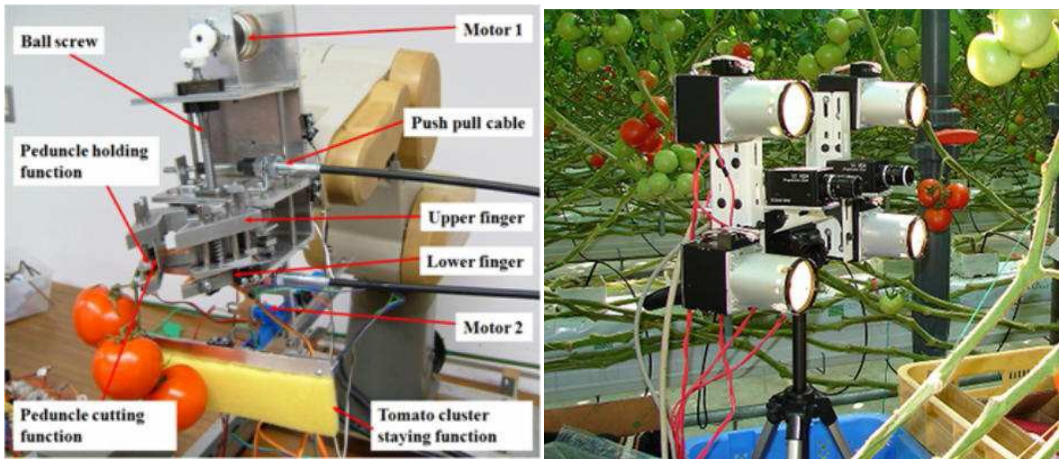

Figure 21 - End-effecter and machine vision

Fruit image segmentation issue on colour difference between mature fruits and backgrounds under natural illumination condition is an important and difficult

content of fruit-harvesting robot vision. Some studies concerning fruit image segmentation have been presented in the last few years. However, these studies are focused on particular fruit and different from segmentation results. In Yin J. *et al.* 2008, four kinds of segmentation methods are presented and applied into fruit image segmentation. The tests show that these methods can segment successful several kinds of fruits image, even tomatoes.

Figure 22 - Tomato image and its segmented image (image size: 640x480)

The Yang *et al.* 2007 paper presents a method to detect and recognize mature tomato fruit clusters on a complex structured tomato plant containing clutter and occlusion in a tomato greenhouse for automatic harvesting purpose. A colour stereo vision camera (PGR BumbleBee2) is applied as the vision sensor. The proposed method performs a 3D reconstruction with the data collected by the stereo camera to create a 3D environment for further processing. The Colour Layer Growing (CLG) method is introduced to segment the mature fruits from the leaves, stalks, background and noise. Target fruit clusters can then be located by depth segmentation. The experimental data was collected from a tomato greenhouse and the method is justified by the experimental results. The experiments included severe self and stereo occlusion.

Figure 23 - Mature fruits at 32~35cm distance with stereo occlusion (a) raw left image (b) raw right image (c) CLG colour segmentation (d) fruit cluster in the image with depth filter

In these approaches many factors can infect the quality results and the same detection.

One of the drawbacks of the segmentations method based on the colour information relies on the fact that it can detect only fruit at the same maturity or colour distribution: how detect tomato when different color appearance occurs? As shown in Figure 24:

Figure 24 - Classical view of a tomatoes raw

Occlusion, highlights presence, illumination variability, etc are all parameter that have to be taken into account and till the moment it doesn't exist a robust method to solve the detection issues. Our approach starts directly from the object/tomato and proposes a new algorithm method based on detecting classes of objects, in our case tomato (Blasco *et al.* 2003a).

Our focus is on developing robust feature extraction algorithms that encode image regions as high dimensional feature vectors that support high accuracy object/non-object decisions. To test our feature sets we adopt a relatively simple learning framework that uses linear Support Vector Machines to classify each possible image region as a "tomato" or as a "non-tomato". The approach is *data-driven* and purely bottom-up using low-level appearance and motion vectors to detect objects.

3 Materials and Methods

3.1 The Greenhouses

The Mediterranean landscape is characterized from a huge number of greenhouses for crop cultivation as shown in figure below.

Figure 25 - Overview of the greenhouse industry at Pachino (Province of Siracusa, Sicily, Italy)

Figure 26 - Overview of the greenhouse industry at Licata (Province of Agrigento, Sicily, Italy)

The present work is focused on activity performed on tomato cultivation.

The experimental trials were carried out in a tomato greenhouse located in the province of Ragusa (Sicily, Italy) (Figure 27). The plants were arranged in twin rows with distance between plants of about 0.6 m, distance between twin rows of 0.85 m, and row spacing of 1.80 m (Figure 28):

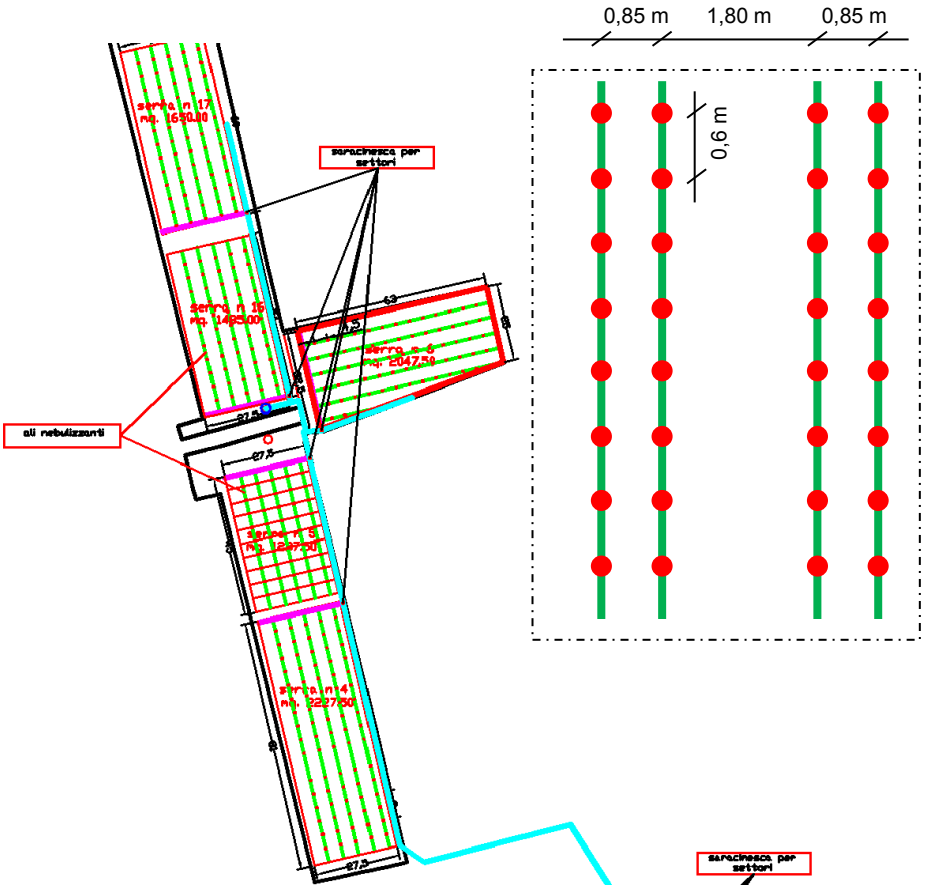

Figure 27 - Greenhouses plant where the tests has been performed (Ragusa, Sicily, Italy)

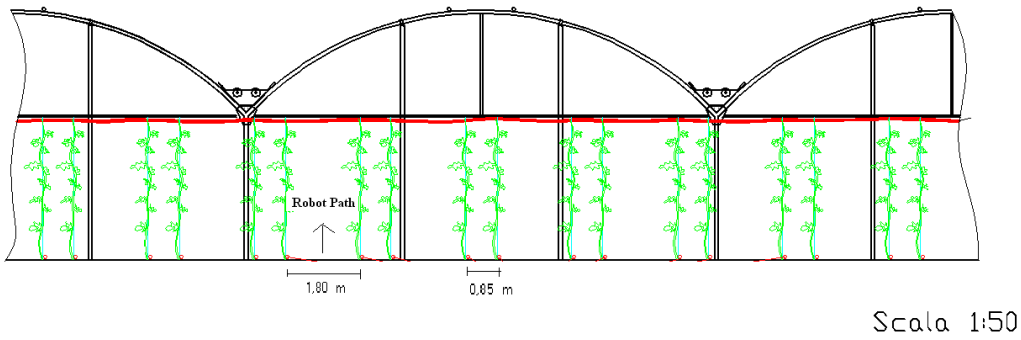

Figure 28 - Section of a Greenhouse sample

The greenhouse had a metallic structure, covered with plastic film. The minimum height was 2.70 m, the maximum 4.50 m. It had 15 spans, each 29 m long and 8 m wide, so the total surface was some 3600 m^2. A lateral aisle 1.10 m wide is provided for internal movements of operators during crop activities.

There are around 7500 tomato varieties grown for various purposes. Tomato varieties are roughly divided into several categories, based mostly on shape and size.

- "Slicing" or "globe" tomatoes are the usual tomatoes of commerce, used for a wide variety of processing and fresh eating.
- Beefsteak tomatoes are large tomatoes often used for sandwiches and similar applications. Their kidney-bean shape, thinner skin, and shorter shelf life make commercial use impractical.
- Oxheart tomatoes can range in size up to beefsteaks, and are shaped like large strawberries.
- Plum tomatoes, or paste tomatoes (including pear tomatoes), are bred with a higher solids content for use in tomato sauce and paste, and are usually oblong.
- Pear tomatoes are obviously pear-shaped, and are based upon the San Marzano types for a richer gourmet paste.
- Cherry tomatoes are small and round, often sweet tomatoes generally eaten whole in salads.
- Grape tomatoes, a more recent introduction, are smaller and oblong, a variation on plum tomatoes, and used in salads.
- Campari tomatoes are also sweet and noted for their juiciness and low acidity. They are bigger than cherry tomatoes, but are smaller than plum tomatoes.

The work is mainly focused on the "cherry" tomatoes variety; a typical crop wall is shown in the figures below:

Figure 29 - Typical Cherry tomatoes wall

The robot structure, the sensors and the vision system developed has been designed in order to guarantee the achievement of the goal: development of a vehicle capable to autonomously navigate in the greenhouses previously described, equipped with sensors (of different cost solutions) for the navigation and a vision system for the detection of tomatoes. The robot will be equipped with a remote operable sprayer that can use information coming from the artificial vision subsystem to optimize the spraying process. The next paragraphs are an overview of the work, leaving to the next chapters the task of a deeper explanation of the architecture and algorithm used.

3.2 U-Go Robot

The mechanical structure of the robot has been designed in order to be compliant to different requirements (see §4.1). Different sensors have been tested and new solutions developed in order to guarantee mainly the autonomous navigation along the greenhouses path (Figure 30) and the detection of the tomatoes fruit.

Figure 30 - Typical greenhouse path

Related to the navigation sensors, two different categories can be identified. Those based on:

- DGPS and Laser scanners (high cost solution)
- Webcams and ultrasounds (new low cost solutions) (Longo *et al.* 2010a)

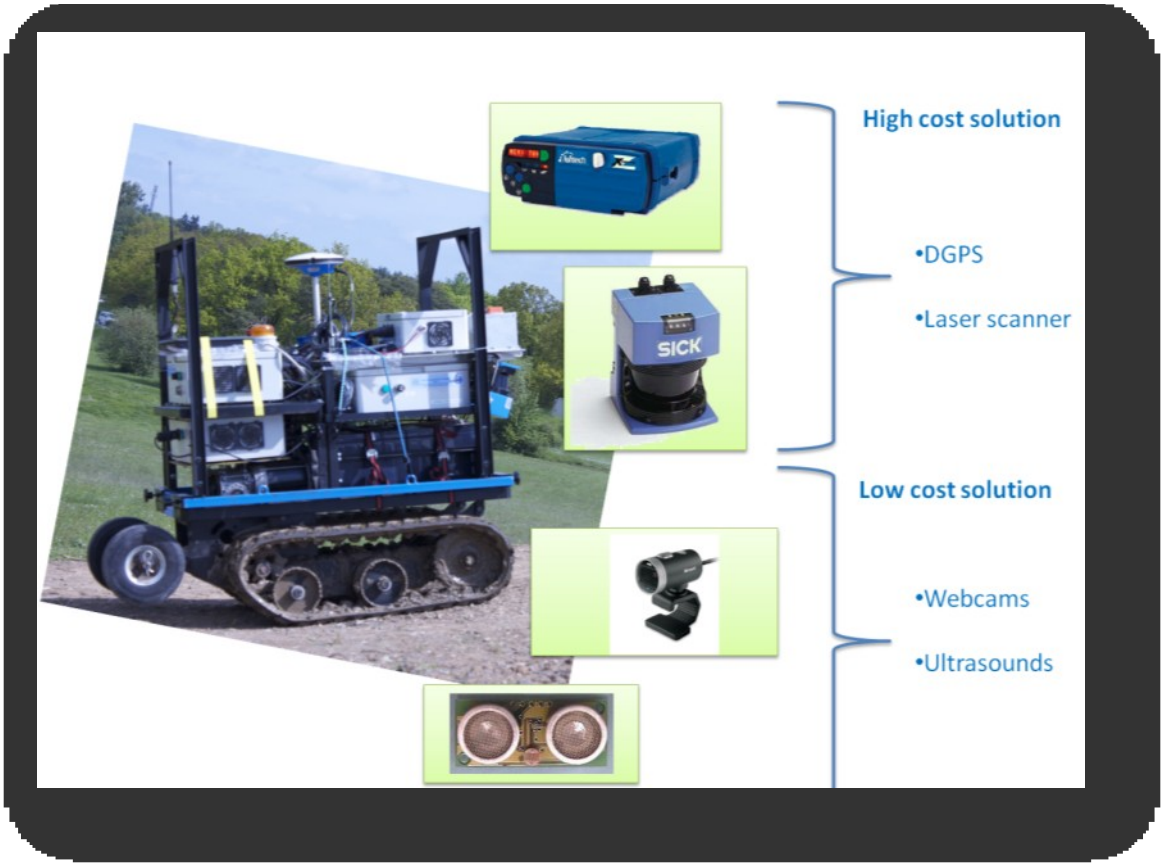

Figure 31 - system modalities overview

Actually a third modality is in development. The aim of the work presented (Bonaccorso *et al.* 2011) is the investigation of the potentialities of a very low cost Global Navigation Satellite System (GNSS) based localization architecture.

The needed accuracy in the position solution is obtained through the adoption of the "Sicili@Net" infrastructure, a free of charge Ground Based Augmentation System (GBAS) offered by the National Institute of Geophysics and Volcanology – INGV. Several test campaigns have been conducted to test both the augmentation and the guidance systems in dynamic conditions and performances have been evaluated in different environments (open spaces, urban environments).

In the next chapters a description of the architecture, of the sensors and of algorithm used will be carried out taking into account only the first two modalities previously mentioned. A computer vision approach for the detection of the tomato based on a commercial web cameras will be introduces in the next paragraph and more detailed in §5.

3.3 The Vision System and Tomato Detection

Computer vision (or machine vision) is the science and technology of machines that see. Here see means the machine is able to extract information from an image, to solve some task, or perhaps "understand" the scene in either a broad or limited sense.

Applications range from (relatively) simple tasks, such as industrial machine vision systems which, say, count bottles speeding by on a production line, to research into artificial intelligence and computers or robots that can comprehend the world around them.

Computer vision seeks to apply its theories and models to the construction of computer vision systems. Examples of applications of computer vision include systems for:

- Controlling processes (e.g., an industrial robot or an autonomous vehicle).
- Detecting events (e.g., for visual surveillance or people counting).
- Organizing information (e.g., for indexing databases of images and image sequences).
- Modelling objects or environments (e.g., industrial inspection, medical image analysis or topographical modelling).

- Interaction.

The classical problem in computer vision, image processing, and machine vision is that of determining whether or not the image data contains some specific object, feature, or activity. This task can normally be solved robustly and without effort by a human, but is still not satisfactorily solved in computer vision for the general case: arbitrary objects in arbitrary situations. The existing methods for dealing with this problem can at best solve it only for specific objects, such as simple geometric objects (e.g., polyhedral), human faces, printed or hand-written characters, or vehicles, and in specific situations, typically described in terms of well-defined illumination, background, and pose of the object relative to the camera.

Tomato detection is a computer vision system that determines the locations and sizes of objects in arbitrary (digital) images. It detects objects features we are interested and ignores anything else, such as buildings, trees and every kind of object in the background.

Our algorithm can be regarded as a specific case of object-class detection. In object-class detection, the task is to find the locations and sizes of all objects in an image that belong to a given class.

Often, a window-sliding technique is employed. That is, the classifier is used to classify the (usually square or rectangular) portions of an image, at all locations and scales, as either tomato or non-tomato (background pattern).

This approach has been followed in our research and it can be summarized in two main steps (Figure 32).

The first, the learning phase, is in off-line mode and it can be performed once you decide your objective and dataset. The dataset creation has been a main and determinant step in order to achieve the results expected. Finally, it's up to the classifier to choose what the object is and what not.

The second phase, the detection ones, can be called on-line phase. It starts considering a test image and applying to that one a dense image scanning. Once a

classifier has run and performed the detections, a fusion of multiple data (non maxima suppression technique) is often required.

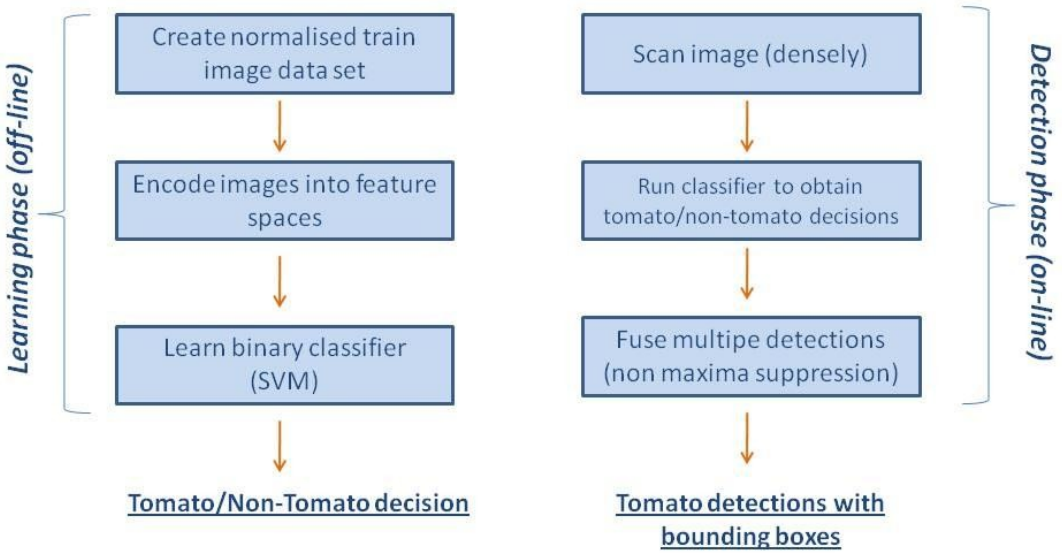

Figure 32 - Tomato Detections Overview

The power of this method consists in the reusability of its architecture for other applications. The achievement of the results whished depends strongly on the dataset (e.g. tomatoes, apples, strawberry etc.) and in the descriptor choice.

For more and deeper details on the vision algorithm please refer to §5.

4 U-Go Robot

4.1 Introduction to the robot

This section will present the U-Go Robot, a rugged outdoor vehicle under developed at DIEEI Robotic Laboratories. The main target for this new machine is to be a multifunction system able to comply with different field of applications. The DIEEI Robotic Group currently has different active projects in the field of autonomous navigation, new GNSS technologies, agriculture and precision farming applications, artificial vision, data fusion and so on. The work has been a challenging opportunity to test different subsystems working together.

U-Go Robot is an acronym that means "Unmanned Ground Outdoor Robot". This robot has been developed at DIEEI Robotic Laboratories mainly to solve problems like transportation, navigation and inspection in very harsh outdoor environments, to perform inspections into volcanic environments and as a test bed for new GNSS (Global Navigation Satellite System) localization technologies.

Moreover, in this thesis, the robot has been used as a multifunctional vehicle able to mainly operate inside greenhouses for precision farming applications.

The whole hardware and software kit has been developed in order to perform different kind of activity; in the next paragraphs the hardware configuration of the sensors, the computation units and the used framework for the development of the control algorithm, will be shown.

4.2 The robot structure

The mechanical structure has been designed in order to be compliant to different requirements. First of all, the robot must be able to move inside greenhouse corridors; moreover it must be able to move on different uneven terrains and must not generate too high pressure on the terrain (in order to meet agricultural requirements). The robot must be able to carry at least 200 kg payload over a flat

road (in order to be able to carry an agricultural spraying machine or other tools) and climb on sloping roads with some reduced payload.

According to these specifications, the robot two main dimensions are 0.6 m wide and 1.2 m long. Moreover it uses rubber tracks instead of wheels for locomotion and its weight is about 150 kg. Figure 33 shows the U-Go Robot. Two 12 V – 180 Ah sealed lead-acid batteries are mounted on the mechanical structure on the rear side of the robot. Each rubber track is actuated by means of a 24 V – 650W brushed DC motor and suitable gearboxes are mounted in the front side of the robot. Two roller chains connect gearboxes output shaft with tracks input shaft; in case of emergency they can be easily opened to allow free tracks motion. Above the two DC motors it is possible to see the box that contains the power electronics.

Figure 33 - The tracked U-Go Robot in the lab and during a teleoperated outdoor test.

Finally, on the top of the robot there is another box that contains computers, all the necessary electronic circuits needed for autonomous navigation, the emergency stop button and the safety flash. The computers box can be disconnected if only teleoperation is needed.

4.3 The on-board electronics

The control electronics have been designed to provide different choices for control modalities.

The simplest one is the *teleoperated modality*. In this situation a remote user, using a joystick or simply a computer keyboard, can send simple direct commands to the

robot in order to move forward, backward or turn left or right at different speeds. The user can also read some system parameters like battery voltage, power electronics temperatures or motors currents. The only components required for this operating mode is the power electronics box and a radio link between the robot and the remote station. The radio link sends digital data at the speed of 9600 bit/s over a UHF radio signal. The remote station is composed by the joystick, the radio and a standard PC or laptop used to read joystick movement by means of suitable software. The PC could be replaced by apposite suitable electronic circuit capable to read movements of an analogical joystick and to send digital commands to the radio. These commands will be received by the radio mounted on-board the robot and then sent to the power electronics. A block diagram of this system modality is shown on Figure 34.

The other two possible modalities for controlling the robot are semi-autonomous mode and autonomous mode. Both these two modalities rely on an on-board computer and on several sensors mounted on the robot.

In the first mode, a remote base station is also required. In the second mode, the remote base station could be avoided, however for safety and logging reasons it is always better to keep it. In semi-autonomous mode, the remote user send high level commands to the robot (like “go forward”); then the robot, using on-board sensors can compensate the whole trajectory for unexpected disturbances and to reach waypoints and targets.

In autonomous mode, the robot will be able to find its way through corridors in the greenhouse, to find optimal path on a not well defined road, to reach a target by GPS waypoints and to avoid obstacles. For safety reason a remote or local user can always issue an emergency “stop” command to the robot. Next chapters will briefly describe the different sensors and algorithms used.

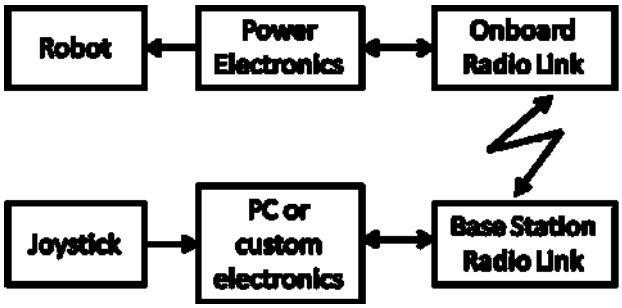

Figure 34 - Block diagram of the system in teleoperated mode

4.4 Interfaces and sensors

The on- board sensors suite is composed by:

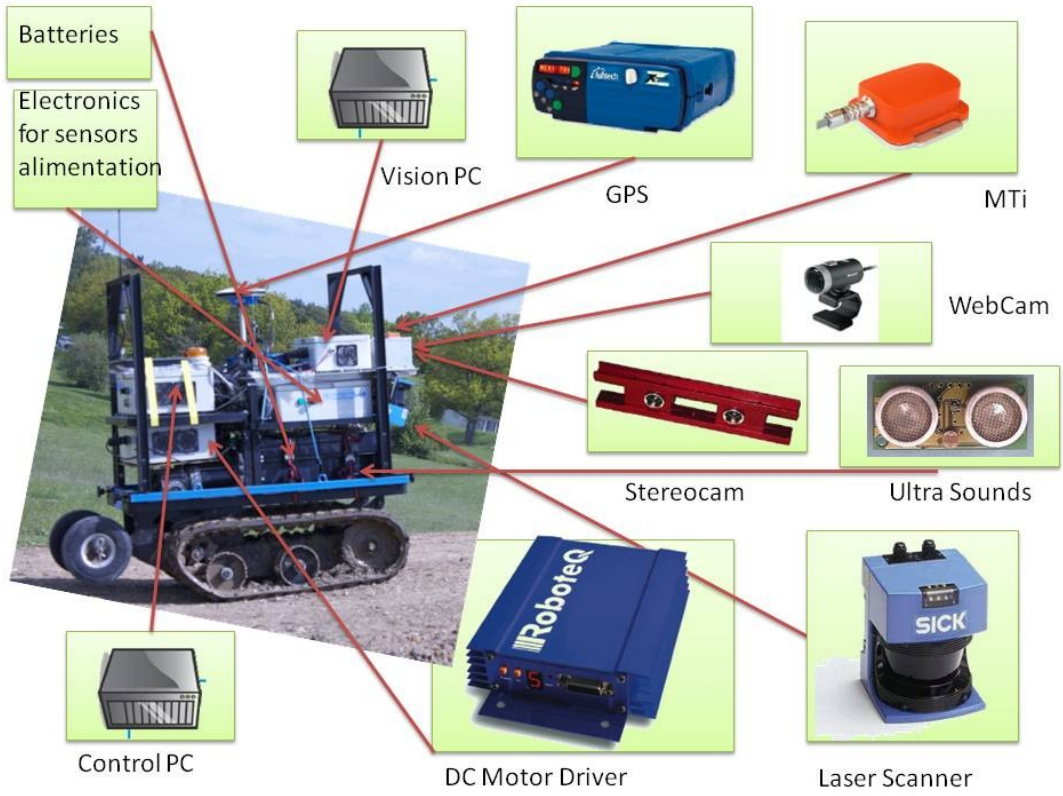

Figure 35 - Hardware configuration and disposition for U-Go Robot

- Global Navigation Satellite System (GNSS) receiver
- X-Sens MTi Attitude/Heading sensor
- Laser range finder (LRF)
- Stereocam for Eye-Bird vision

- Control and Vision PCs
- DC Motor Driver
- Vision System
- Webcam and US-Sonars

Information coming from these devices are fused together in order to implement a self-localization and obstacle avoidance algorithm. In Figure 35 a block diagram of the sensors system architecture is shown.

Different cost solutions, depending on the sensors used, will be explained in §4.5.

4.4.1 Global Navigation Satellites Systems (GNSSs)

The sensor provides information on the absolute position of the robot in geographic coordinates. Due to the functioning with the module for differential correction, it provides a very accurate measurement (with an error in the order of centimetres) of the position of the robot. The Differential GPS (DGPS) is a method to improve the accuracy of a standard GPS signal by using one or more GPS reference stations with exactly known positions, each equipped with a GPS receiver. It is obtained by sending signals to correct the error between the different stations and the mobile station with unknown position.

DGPS requires the use of at least two GPS receivers:

- **Base:** permanently positioned in a fixed location with known coordinates with great accuracy.
- **Mobile:** used in a generic position (e.g. equipped on board the robot) with unknown coordinates.

If the two receivers are not too far one from each other, they will be more or less at the same weather conditions and subjected to the same error in the calculation of the position. The base station knows its location (latitude, longitude, altitude) and then accurately can calculate the error between the estimated position and unknown location. If this error is communicated to the receiver mobile, for instance through radio modems, it can correct its estimated position, greatly reducing the localization error. The information transmitted from the receiver to the mobile base is called *differential correction*. The precision offered by this tracking system can also reach a few centimetres.

4.4.1.1 LLA to Local coordinates conversion

WGS84 is a mathematical model of the Earth from a geometric, geodetic and gravitational point of view, built on the basis of the scientific and technological knowledge available in 1984. From the geometrical point of view, WGS84 is a particularly conventional Earth system, or more precisely a Cartesian reference system used to describe the earth, whose characteristics are:

- *center*: on the mass center of the earth
- *Z-Axis*: passing through the North Pole.
- *X axis*: passing through the Greenwich meridian.
- *Y axis*: chosen to complete a right-handed triad, i.e. such that an observer placed along the Z axis sees the X-axis overlaps to the Y-axis with counter-clockwise motion.

To this reference system is associated an ellipsoid obtained by rotating an ellipse around its minor axis. A point P is uniquely determined by the following triplet of geographic coordinates LLA:

- **Latitude μ:** is the angular distance of the point P measured from the equator along the meridian passing through that point.
- **Longitude l:** is the angle between the meridian passing through P and the prime meridian of Greenwich. It is positive in the west front and negative by the east of Greenwich.
- **Altitude h:** distance along the normal to the ellipsoid between P and the same ellipsoid.

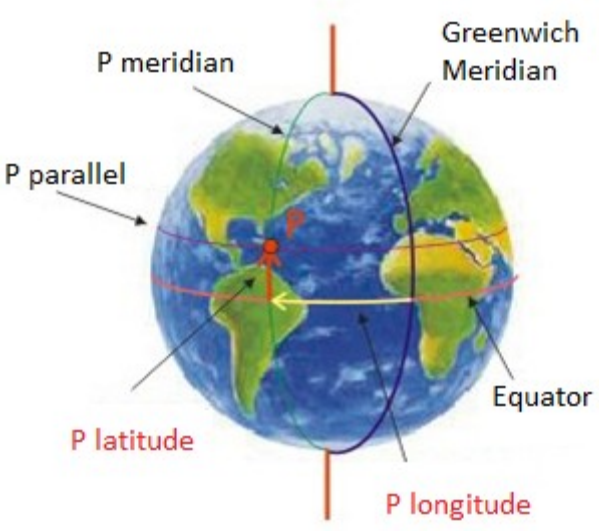

Figure 36 - Latitude and longitude coordinates

The ultimate goal is the conversion of the LLA coordinates in ENH (East, North and Height) local coordinates but, first, they must be converted into an intermediate system of reference: ECEF (Earth-Centered Earth-Fixed). It's also a Cartesian coordinate system similar to the WGS84 system. It's the system of reference rotating with the Earth; the X, Y, Z are represented in meters. The XY plane is coincident with the equatorial plane with the respective unit vectors pointing directions of longitude 0° and 90°, while the Z axis orthogonal to this plane pointing in the direction of the north pole. The point (0, 0, 0) denotes the centre of the earth, hence the name "Earth-Centered." In this system of coordinates, a generic point P is defined by the triple of coordinates(x, y, z).

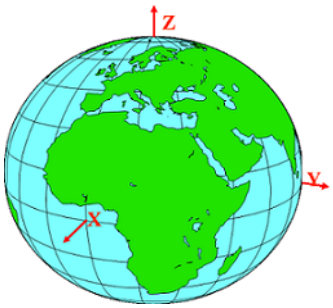

Figure 37 - ECEF coordinate system

The conversion from LLA to ECEF can be done through the use of these relationships:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r\cos\lambda\cos l + h\cos\mu\cos l \\ r\cos\lambda\sin l + h\cos\mu\sin l \\ r\sin\lambda + h\sin\mu \end{bmatrix}$$

where:

$$\lambda = \tan^{-1}\left((1-f)^2\tan\mu\right)$$

$$r = \sqrt{\frac{R^2}{1+\left(\frac{1}{(1-f)^2}-1\right)\sin^2\lambda}}$$

and $f = 298.257223563$ is the flattening coefficient and $R = 6378137$ is the equatorial radius.

Now you can make the conversion from geocentric Cartesian coordinate system (ECEF) to local Cartesian coordinate system (ENH).

P0 is a known geocentric coordinates (X0, Y0, Z0), the origin of the Eulerian triad, while P be a generic point, with geocentric coordinates (X, Y, Z).

We calculate first the coordinate differences between P and P0.

$$\Delta X = X - X_0$$

$$\Delta Y = Y - Y_0$$

$$\Delta Z = Z - Z_0$$

Then the coordinates (e, n, h) of the generic point P are obtained by using the rotation formulas between the two systems:

$$\begin{bmatrix} e \\ n \\ h \end{bmatrix} = R(\mu, l) \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix}$$

Where R is the Rotation Matrix:

$$R(\mu, l) = \begin{bmatrix} -\sin l & \cos l & 0 \\ -\sin \mu \cos l & -\sin \mu \sin l & \cos \mu \\ \cos \mu \cos l & \cos \mu \sin l & \sin \mu \end{bmatrix}$$

4.4.1.2 The Standard: NMEA 0183

The purpose of the standard NMEA0183 is to send to the computer processed data from the GPS receiver in a series of strings, with a certain frequency (typically 1 Hz). Each string is included between a $ terminator and a carriage return '\r\n'; they identify exactly the beginning and end of the string. After the $, each string contains a prefix identifying the device is working properly sending the string. For example,

for GPS receivers the prefix is 'GP'. The fields of each string have variable length and they are separated by ','.

The GPS string we are interested is the GGA, which contains essential data for 3D localization. An example of the GGA string is the following:

$GPGGA , 122730.00 , 3731.5177 , N , 01504.4368 , E , 1 , 06 , 1.2 , 154.3 , M , 39.4 , M , , 0000*5A\r\n

where:

- **GPGGA:** String identifier
- **122730:** UTC system Time; corresponding to 14:27:30.
- **3731.5177:** latitude in DM format (Degrees, Minutes); corresponding to 37°31.5177'.
- **1504.4368:** longitude in DM format (Degrees, Minutes), corresponding to 15° 4.4368'.
- **154.3:** altitude in meters above the sea level.
- **39.4:** geode height (sea level) on the ellipsoid WGS84.

Thanks to this information, the geodetic coordinates LLA can be extracted, then converted in geocentric Cartesian coordinates ECEF and finally in the local coordinates ENH.

4.4.1.3 The DGPS System used

The DGPS system, used for robot satellite tracking, is based on the model produced by the Ashtech Z-Xtreme.

Figure 38 - Ashtech Z-Xtreme module

It consists of two modules, a base and a mobile. Each module is powered up by a battery even if the base module, being fixed, can be powered by a generator. The module provides six ports:

- **Power:** used to feed the module, if the battery is not provided.
- **Radio:** the port used for connecting the module to the radio modem, necessary for the communication between base and mobile. The base module is connected to a transmitter to communicate the differential correction; the module will be connected to a mobile antenna to receive it.
- **Serial Port A, B, C:** serial ports used for the output of differential correction and navigation data, as well as the connection between base and mobile in alternative way to the radio modem.
- **GPS:** connecting port of the DGPS module to the GPS antenna.

A PC can be connected to any of the three serial ports and Mobile Base. This is necessary to set the two modules and to obtain a proper working of the DGPS system. The Base module signal will be set in order to:

- provide the differential correction to the output port B.
- provide navigation data to the output port C.

In addition, it is necessary to save on the Base module, the geodetic coordinates of the antenna mounted on a tripod. In this way, the Base module, knowing with precision the position of the antenna, is able to estimate the error and notify the differential correction to the mobile module. This output on serial port B is transmitted via radio modem.

The Mobile module is set in order to:

- receive the differential correction in the input port where is connected a radio antenna.
- provide navigation data to the output port C.

At this point, the mobile module is able to correct its position and provide it to the serial port connected to a PC. The parameters of the connection between mobile and PC module are:

- **Baud rate:** 9600 bps
- **Data bit:** 8
- **Parity:** No
- **Stop bit:** 1
- **Flow Control:** Hardware

4.4.2 Laser Scanner

The laser scanner used is a Sick, model LMS200 (Figure 39). This type of sensors detects the distance between points detected by scanning the environment, calculated from the average of a laser beam deflected through a rotating mirror mechanism, and the instrument itself.

Figure 39 - Laser Scanner Sick LMS 200

These devices allow measuring an angular range of 180°. In terms of resolution, the smallest detectable angular variation is typically 0.5° but it is also possible to obtain

a resolution of 0.25°. The power requirement is 24 VDC with a power consumption of 20W. The optimal temperature for its operating range is between 0°C to 50°C.

The interface used is a serial RS-232 or RS-422. The weight is about 4.5 Kg and dimensions are156x155x210 mm (LxWxH). Laser scanning technology allows the digital acquisition of tridimensional objects as point clouds. The digital geometric description of the object is discrete, the resolution set for the acquisition defines the density of the point cloud and so the detail of the representation.

Every point is described by a spatial position in x y z coordinates respect to the origin represented by the position of the scanner.

4.4.2.1 Sensor communication

The Sick receives commands as a stream of bytes through the serial port. In response, it sends distance measures through packet data.

A typical data stream could be as follows:

2 128 214 2 176 105 65 79 23 76 23 77 23.....

To get the data from the sensor, you need to send a string of initialization is as follows:

- **Hexadecimal form:** 02 00 02 00 20 24 34 08
- **Decimal form:** 2 0 2 0 32 36 52 8

Successfully received the string, the sensor begins to forward the measures. This information is organized in packets whose format is:

STX	ADR	LenL Low byte	LenH High byte	CMD	Data LenL	DataL enH	Data 0° Low byte	Data 0° High byte	Data 1° Low byte	Data 1° High byte	..... (more data)	Status	CRC Low byte	CRC High byte

Figure 40 - Data Packet of the distance measures

The first five fields of the packet are header. They contain information about the address of the recipient (typically a PC), total size (in bytes) of data stream (excluding the CRC field) and size (in bytes) of data section. The overall size of the

header is 7 bytes. In the package, there's a part of the packet data (16 bits) reserved to distance measures (Data Low and Data High fields).

Totally there are n x 16 bits for data, where n is the number of measurements performed by the sensor. If the resolution is 0.5 ° and the scan range of 180 ° there is a total of 361 measures (722 data bytes). For each measurement, the device sends the least significant byte first, followed by the most significant.

The measurement unit is the millimetre. For example, if the sensor detects a distance of 5000 mm it will send two bytes in decimal form: 136 and 19 (88 and 13 in hexadecimal), in fact, 136 + 19 x 256 = 5000.

The last three fields in the packet (3 bytes) are part of trailers and contain information about the state of the system and a checksum (typically a 16 bit cyclic redundancy code).

To stop forwarding data from the sensor, an additional string has to be sent to the device. The string is as follows:

- **Hexadecimal form:** 02 00 02 00 20 25 35 08
- **Decimal Form:** 2 0 2 0 32 37 53 8

4.4.3 DC Motor Driver

It allows the robot moving through commands sent via RS-232. The independent actuation of the robot tracks allow a differential drive configuration (§4.7.1.4).

4.4.4 Stereocam: Bird-Eye Vision

Mainly for outdoor use, e.g. vineyard, on the robot there are also two high quality stereoscopic cameras; each one has a resolution of 1.3 Megapixel; they are equipped with fixed focus lens of 4.0 mm. The CCD sensors of these cameras have a good noise immunity and sensibility; moreover, it is possible to adjust all the image parameter, e.g. exposure gain, frame rate, resolution. The cameras are mounted on a rigid support; it permits to adjust in a simply way the camera distance in a range 5-20

cm (Figure 41).The images come from the two camera are synchronized with an 8 KHz clock, generated by using IEEE1394 interface.

Figure 41 - Stereocamera

The sensor is composed from two video cameras, based on fixed position, in order to provide a 3D vision to the user.

Through this device, it is available a 3D vision for the implementation of artificial vision algorithm (e.g. the "Bird Eye Vision"). The terrain morphology is used both to detect the presence of obstacles in the mission path and to recognize and define a drivable surface in the proximity of the robot. An approach similar to the one described in Dahlkamp *et al.* 2006 is adopted: a supervised algorithm permits to create a database of the scenarios, allowing the on-line identification of possible safe patterns. However, while in Dahlkamp *et al.* 2006, a simple camera is adopted, in our application, a stereo camera pair will be used. Image processing is based on the OpenCV open source library that provides useful, quick and reliable basic functions to build up computer vision algorithms. An auto-calibration tool for stereo camera has been developed, in order to easily and quickly compute cameras intrinsic, extrinsic and distortion parameters. The computed parameters are used in the real-time image processing procedures so as to obtain a top down "bird's eye" view (Figure 42), from which information about the path in front of the robot can be extracted. The adoption of stereo cameras also allows detecting obstacles and getting out elevation data from the images, in order to find drivable corridors.

Figure 42 - Original image (on the right); reconstructed top down "bird's eye" view (on the left)

4.4.5 PCs for vision and control

The two calculation units are separated for a greater efficiency and consist of two PCs (UNO - 2182) Intel CoreDuo. The Vision PC provides sensory information from the images coming from the stereocameras processed by graphics library (OpenCV).

An additional PC is dedicated to the final control algorithm with the collection of data coming from sensors, to the processing and actuation of the DC motor driver. All sensors are connected to the PC for control via RS-232, while the connection between the two calculation units is via UDP protocol over Ethernet.

The two on board PC have been chosen on the basis of the characteristics required by the application: the structure is robust and suitable for industrial use, they do not use cooling fans for use in dusty environments and solid-state hard disk has been used to improve robustness to vibration.

4.4.6 Low cost solution for localization

Another main target of this research activity has been to develop and test different low cost sensors that can enable the multifunctional tracked electrical vehicle to move autonomously inside a greenhouse with tomatoes cultivation, regardless of use of the DGPS system.

Big effort has been put on finding solutions that require minimum or no apparatus installation in the greenhouses while performing centimetre-level accuracy at a fraction of the cost of a DGPS commercial system.

Two main kinds of sensors systems have been developed using low cost ultrasound sensors. The first system uses a set of eight ultrasound transmitter/receiver couple. These sensors can compute time by time the distance and the rotation angle of the robot with respect to the plants rows. This information could be used to correct the robot trajectory, allowing the system to move along the centre line between rows.

The second system that has been tested is a Local Positioning System (LPS). It uses a set of low cost ultrasound receiver sensors mounted on the greenhouse structure in a regular grid while an ultrasound transmitter is mounted on-board the robot that is moving between rows. The system allows knowing the relative coordinate of the robot with respect to a reference inside the greenhouse. At this stage, different trials on the sensors have been performed inside a greenhouse in different condition; these trials allowed evaluating systems accuracy and performance.

As previously mentioned, sensors like DGPS and 2D Laser scanner are widely used; they have very good performance but the high cost and their use cannot be addressed to most SME in the agriculture field. The two methodologies proposed here are instead based on very low cost ultrasound sensors and suitable measurement algorithms. They allow obtaining the position of a moving machine along rows with respect to corridors boundary or with respect to a reference system defined in the greenhouse. The two systems can be used at the same time on the same machine for a better accuracy (smart data fusion algorithm could be developed) or it is possible to use only one of the two at time.

Different tests have been done in a real greenhouse in order to evaluate performance and capabilities of the two systems and results have been reported; the obtained accuracy is in any case in the order of few centimetres.

In next section, the two different low cost solutions, both based on ultrasound sensors, for robot autonomous navigation in tomatoes greenhouse cultivations, are described.

Finally, another low cost solution will be shown. The algorithm uses Visual Odometry to calculate position, speed and directions of the movement for the mobile robot moving through narrow paths, by using only the images coming from a single CCD camera. This methodology has different advantages and could be very cheap compared to other solutions; moreover it can be applied to a wide range of narrow path similar to the greenhouse corridors that consists of a floor, a ceiling and two limits walls.

4.4.6.1 Self-centering system

The first sensor system is composed by eight SFR08 sensors from Devantech and exploits the particular environment of tomatoes cultivation in greenhouse. Generally speaking, the system can operate in all those places where a wide, quite regular, vertical leaves surface is available.

Each sensor is a module that comprises both transmitter and receiver transducers and all the related electronics in order to perform distance measurement in the range 0.03 m – 6 m. The used ultrasound wave is in the 40 kHz band and the radiation diagram of the transducers is quite large. The module uses the acoustic wave generated by the transmitter and reflected by an obstacle in front of the module itself, to measure the distance by means of the wave time of flight (about 300 m/s in free air). The sensor module is shown in Figure 43.

Figure 43 - The SFR08 sensor module

Exploiting these features it was possible to use this sensor to measure distance against leaves walls. Due to the wide area covered by the sensor, the single

measurement is not biased by the particular leaf or by local leaves structures but it takes a kind of mean value of the distance. In more detail, in the particular area covered by the sensor, if a single small leaf is on a different plane with respect to the most leaves vertical plane, its position will be neglected by the sensor because of the very small acoustic energy it can reflect; in fact the module has a threshold below which it cannot 'see' targets in front of it.

These sensors and related control electronics, can compute time by time the distance and the rotation angle of the robot with respect to the plants rows by using a triangulation algorithm. This information could be used to correct the robot trajectory, allowing the system to move along the centre line between rows.

4.4.6.2 The LPS system

The second system that has been developed is a Local Position System. Like the most common GPS system, the LPS give back the position of the system, for example the robot, while moving or standing still. The main difference is that the GPS uses absolute coordinates while the LPS uses relative coordinates with respect to a reference system locally defined, for example one of the vertex of the greenhouse. The LPS system that has been used was developed at DIEEI during past research activities for different applications (Andò *et al.* 2006, 2008, 2009). The system does not rely on some special structure of the greenhouse, but it can be used in all kind of environment indoor or outdoor. Moreover different tests performed in real greenhouse has shown that plants, cable, pipe and other infrastructures that normally can be found in every greenhouse, does not interfere with normal system operations. The LPS system uses a set of low cost ultrasound receiver mounted on the greenhouse structure in a regular grid (for example one sensor on each pillar). An ultrasound transmitter is then mounted on the robot that is moving between rows; every time the fixed receivers hear the signal sent by the robot, the system computes the robot position with respect to the reference system using a trilateration algorithm. Using the robot position and as the greenhouse map is known, it is possible to use standard navigation algorithms used for outdoor DGPS navigation, exploiting the

same centimetre-level precision at a fraction of the cost of a DGPS commercial system.

4.4.6.3 Artificial vision system for odometer estimation

The aim of this research is to implement and test an artificial vision algorithm that will allow autonomous navigation of the robot through greenhouses corridors. This target can be achieved exploiting some special features and light condition that are normally found in a typical greenhouse with tomatoes cultivation (Figure 44). For example, lateral vegetation, a very high illuminated ceiling, some pattern differences between lateral vegetation and drivable path on the floor and so on. With this methodology in mind, some others high cost solutions (for example DGPS or 2D Laser Scanner) are automatically excluded.

Figure 44 - Common robot path in a greenhouse

Data coming from the artificial vision are then used to obtain useful information that will be used to drive the robot through the aisle. The algorithm uses Visual Odometry to calculate position, speed and directions of the movement for the mobile robot moving through narrow paths, by using a single CCD sensor (a common commercial low cost webcam – Figure 45).

Figure 45 - Webcam used for the trials.

Odometry is a technique used to estimate the position of a moving machine using sensors like encoders, accelerometers and so on. Visual Odometry is a process that tries to find the same information of traditional odometry, but using only the images from CCD cameras. The algorithm does not rely on previous information about surrounding environment. It uses perspective or central projection into a new plan.

In few words, the original frame caught from the webcam, after a stabilization phase, is transformed using a bird-eye like algorithm (Figure 46). This perspective transformation allows pointing out the pattern differences between lateral vegetation and the driveable path. With simple image manipulation algorithms, the vanishing point of the image is determined.

Figure 46 - Corridor view: a) Original frame b) Bird-eye view.

Using this information and the actual position of the robot (determined in the previous algorithm step) the robot orientation and offset is then calculated. The application solves different problems as the stabilization and calibration of image stream, the extrapolation of information from the image stream and the speed of execution of the algorithm. In Figure 47 a block diagram of the implemented software is represented. The FluxLab object is the core of the application. It calculates three parameters: the alpha angle, the offset and the speed of the robot.

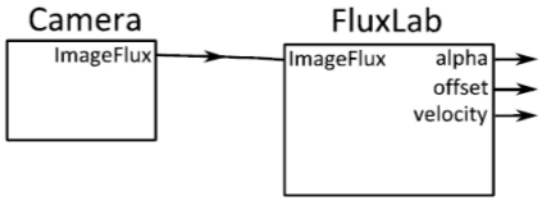

Figure 47 - Block diagram of the implemented software.

The alpha angle is the angle of the robot with respect to the aisle axis while the offset is the distance of the robot from one side wall. The speed parameter is calculated with respect to the floor using the optical flow methodology.

4.4.7 MTi

The sensor provides an absolute orientation of the robot. The MTi is a unit that use MEMS devices as inertial measurement sensors, capable of calculating real-time roll, pitch and yaw, as well as acceleration, angular velocity, data related to the magnetic field and to the temperature.

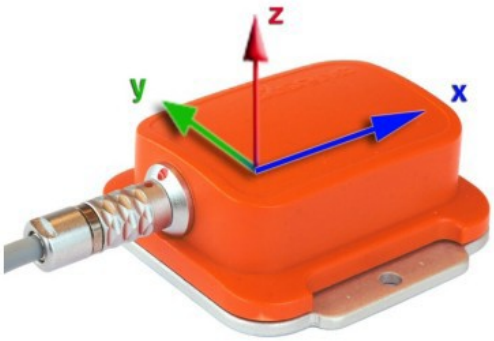

Figure 48 - MTi sensor

Fields of use of the device are numerous:

- Robotics
- Aerospace Applications
- Autonomous Vehicles
- Marine Industry
- Motion Capture

The orientation of the MTi is calculated using a Extended Kalman Filter (EKF), and data related the magnetic field are exploited to stabilize direction, in particular, the measure of gravity and magnetic north offsetting the unlimited growth of errors associated with the integration of the gyroscopic measurements.

A diagram of the inertial device is shown in Figure 49, where there are represented the various sensors, 3D accelerometer, 3D gyroscope, 3D magnetometer, and temperature sensor. All sensor data will be sent to a digital signal processor (DSP) and then will be processed in a PC via serial communication.

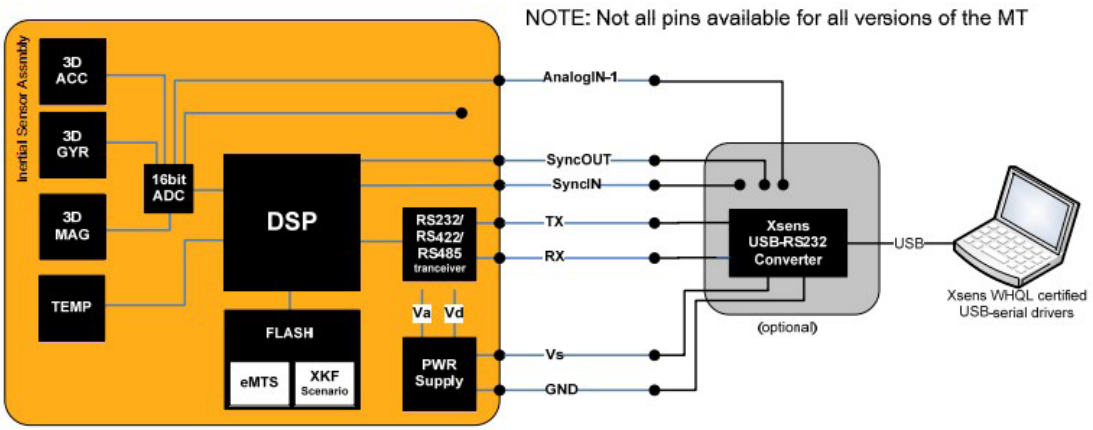

Figure 49 - Functioning scheme of MTi sensor

The specifications of the inertial device are the following:

Dinamic Range	All angles in 3D
Angular Resolution	0.05°
Repeatability	0.2°
Static Accuracy (roll/pitch)	0.5°
Static Accuracy (Yaw)	1.0°
Dinamic Accuracy	2° RMS
Refresh Rate	From 1 Hz to 120 Hz.

Table 1 - MTi specifications

The output values can be presented in different parameters:

- Quaternions
- Euler angles: roll, pitch, yaw
- Rotational Matrix

The sensor is able to provide the orientation of the reference system S, within the frame of the sensor, compared to the reference system G, that is, in other words, the inertial system of the earth.

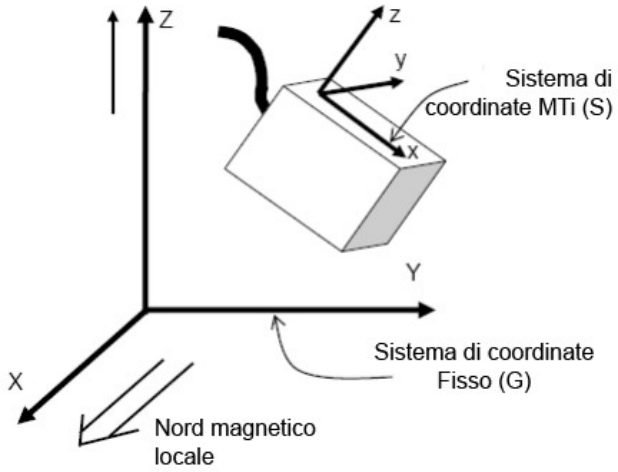

Figure 50 - Orientation of the reference system S, within the frame of the sensor, compared to the reference system G, within the frame of the earth

The axes will be oriented respect to the "earth reference system" according to these conventions:

- *X* is positive when it corresponds to magnetic north
- *Y* is positive when it corresponds to magnetic West
- *Z* is positive when it points upward

Obviously they are defined with respect to magnetic north and not to true north, due to the magnetic declination.

4.4.8 The Vision System for Tomato Detection

Two simple web cams have been mounted on board of the robot. They provide the data image flow necessary for the tomato detection. More details will be added in the next chapter (Chapter 5).

4.5 System Architecture Overview

As the large sensor kit available, the modularity of the architecture of the robotics platform used (see 4.6) will vary depending on the choices, application environment (greenhouses, vineyards, etc.) and the budget available.

The system control architecture is shown in Figure 51. It mainly consists of one computer, in which reside software modules used to receive information from the various sensors used (sonar, laser, webcam etc) and data from the computer where is implemented the artificial vision system. The communication between the different sensors and computers is through serial port or Ethernet and UDP protocol.

Navigations algorithms have been developed using the Microsoft Robotics Developer Studio (MRDS) tool. This programming platform is developed by Microsoft just to interact with different and customizable robotic devices and integrate also a Visual Simulation Environment (VSE). This permits easily to simulate control algorithms and architectures before real robot testing. This development environment designed for robotics allows to:

- create a simulated custom scene, simulating the sensors and the traction of the vehicle
- implementing communication services with the real hardware
- create control loops using block diagrams
- implementing control algorithms in a powerful object-oriented language like C#
- easily simulate robotics algorithms using physically realistic 3D simulated models
- testing the same algorithms in a real context
- exploit the service-oriented paradigm for a rapid and intuitive development

MRDS is particularly useful for moving a project from simulation toward the real robot implementation. In fact, it is possible to replace each simulation entity with a corresponding tool of the real world. For all these reasons, MRDS is proposed in the robotics world as a powerful tool for the development of robotics applications.

The control module, therefore, resides in a PC that receives and processes sensors information provided by the GPS device Ashtech Z-Extreme, the laser scanner SICK LMS200, the Bird Eye Vision system, XSens MTi, the vision system for odometry, the bumpers, the ultrasound sensors etc.

In figure, the whole architecture is shown:

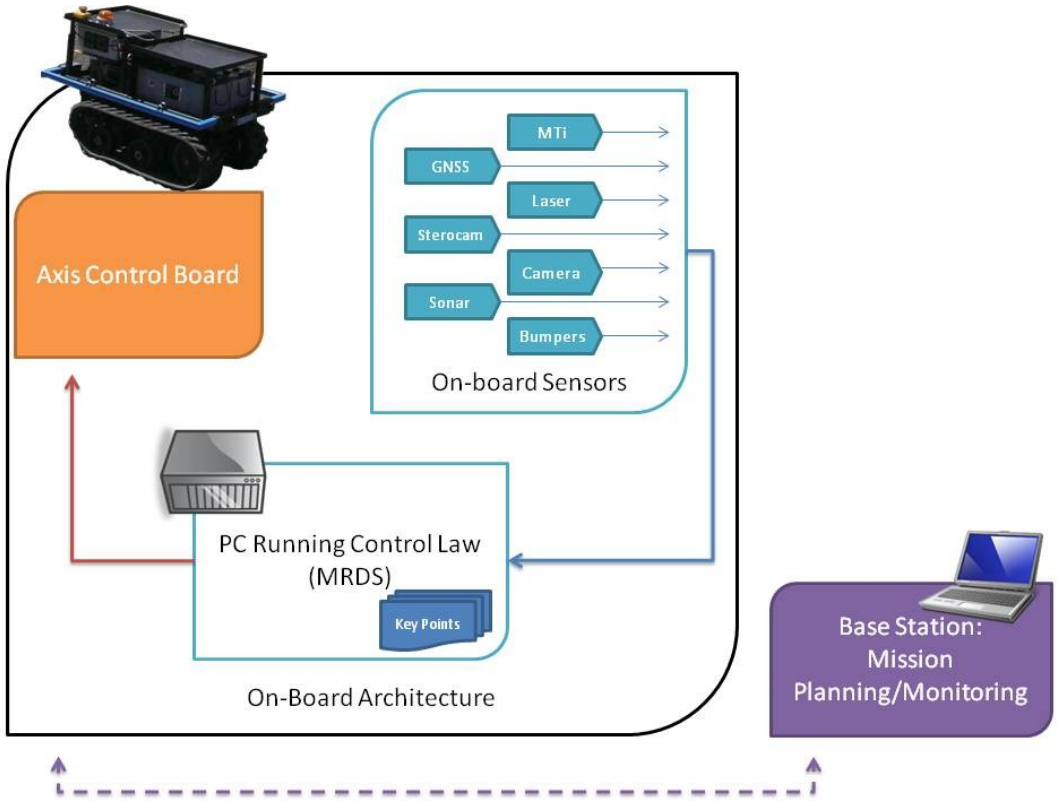

Figure 51 - System control architecture

As previously mentioned, two different solutions for the localization of the robot can be shown:

- Low cost: based on ultrasound system (Self-centering or LPS system)
- Higher cost (GPS and Laser Scanner)

Thanks to the use of MRDS, the control architecture has been developed in order to guarantee the usage of the entire kit of sensors, choosing each time the most suitable for the requested application and, sequentially, for the development of the control law.

In the previous chapter the hardware toolkit has been explained. In the next sections the used framework will be shown, and an overview of the control algorithms and the final architecture will be offered.

4.6 Microsoft Developer Robotics Studio

Microsoft Developer Robotics Studio (MRDS) has been chosen as control software for the robotics platform. It's a framework, Windows based, representing a significant tool for the development of robotics applications.

In particular MRDS was created by developers want to give innovative solutions to problems encountered in the robotics field. In fact, MRDS born after a large analysis dedicated to discovering the world of robotics and the problems most commonly encountered. Microsoft, after studying the needs not only of professionals but also of hobbyists, students and researchers, has drafted a user's "wish list" that can be translated as:

- Simple configuration of sensors and actuators and ability to use them asynchronously
- Ability to start and stop software components dynamically
- Ability to monitor the robot during normal operation
- Ability to access, in a distributed architecture, to multiple robots simultaneously from a single user node or from multiple users to single robot
- Ability to reuse the same software components for different robots

The points listed reflect most of the daily problems encountered in software development for a new robotic platform: they are related to the monitoring and remote control of the robot as well as the need of re-writing control software every time a change on the sensors used occurs (which is a widespread issue in the field).

Encouraging the reuse of code already written and interoperability across the network, Microsoft aims to open the doors of robotics to more people and provide a new development platform, which solves the most trivial problems, leaving to the researchers more time for more complex problems and for design of new algorithms.

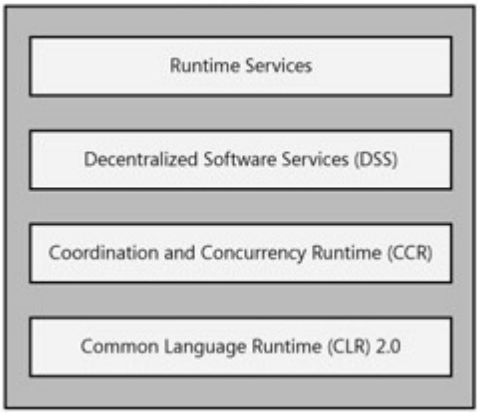

Figure 52 - MRDS Runtime

MRDS is a development environment that offers a service-oriented programming, graphical tools, tutorials, and documentation. The applications created with MRDS are divided into services linked one to each other through exchanging messages.

A MRDS application, as shown in Figure 52, uses three levels: CCR, DSS and .NET Common Language Runtime (CLR) allowing the interaction between different services in a message oriented architecture.

4.6.1 CCR, DSS components and Asynchronous Programming

4.6.1.1 Concurrency and Coordination Runtime

A substantial part of the development environment is represented by the component "Concurrency and Coordination Runtime" (CCR) of the framework .NET that makes asynchronous programming much easier than using traditional tools.

The CCR component is a part of .NET library and can be used by any developer wants to avoid the typical problems of thread programming. It provides asynchronous programming where the concurrency is managed by the component CCR.

In practice, a robot is equipped with sensors and actuators that operate simultaneously, assigning to each one a service. The component CCR manages the proper functioning in a multithreaded architecture.

Without the CCR the developer is forced to use events and semaphores to coordinate the different services and complex algorithms and to prevent threading deadlock of asynchronous operations. With the integration of CCR component in MRDS, it's possible to avoid all the mentioned issues having to handle each type of asynchronous operation, leaving to the user the ability to split the code into services, and taking care only of the exchanging messages between services.

4.6.1.2 Decentralized Software Services

The DSS is an application model allowing the developer to monitor services in real time. It follows a set of rules used to define some of the Web technologies. By definition, the web is a stateless environment, however, through the DSS, to each service is associated a unique identifier within the subnet URI (Unified Resource Identifier) which becomes the access key to the service in a MRDS applications and allows to display its status through the net. Peculiar characteristic of the DSS is the separation between state and behaviour of the service.

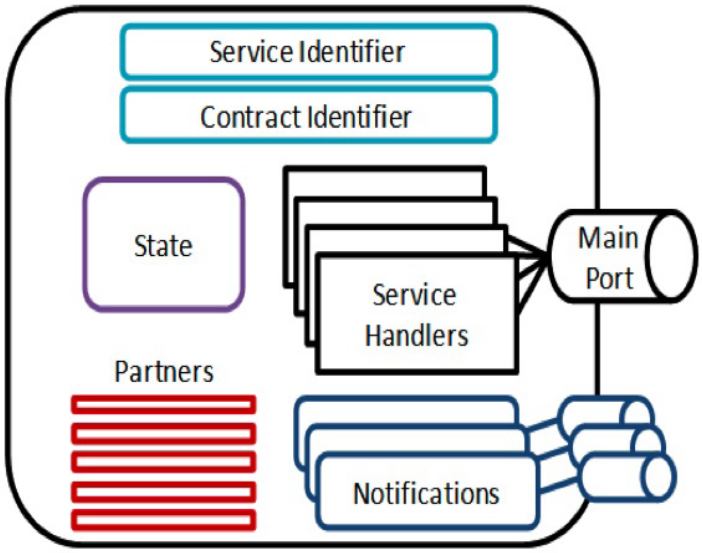

Figure 53 - DSS component

The DSS, therefore, defines the components of each service in MRDS as shown in Figure 53.

Service Partners. Service Partners are special kind of services with which the main service exchanges messages.

Service Identifier. In MRDS, a unique identifier, Universal Resource Identifier (URI), is assigned to each service.

Contract Identifier. The contract defines the operation of the service and is identified by a URI; you must know the URI of the contract to use the service.

Service State. The status of the service is a set of variables representative of the service. It can be monitored at any time through a web interface of the MRDS tools or other services that know the identifier.

Main Port. It is the main port of the service where the messages, to be processed to generate a response output, occur. On the port, there are defined one or more actions that the service is ready to provide.

Service Handlers. Each action will work differently as described in the Service Handler where it is detailed the processing of the incoming message in order to provide the output message.

Notification. It allows to link more services: they will answer in case of asynchronous changes of status of the service (paragraph 3.3).

To implement subscriptions, the Extensible Markup Language (XML), which provides rules for handling data structures, event notifications, and multiple accesses on the web, has been used in MRDS.

MRDS uses standard protocols such as TCP combined with XML and HTTP.

4.6.2 Visual Programming Language

A solution for beginners or for a rapid prototyping of complex applications is the use of the MRDS Visual Programming Language (VPL). VPL is a graphical tool intuitive and easy to use whose main objects are blocks and connections between them: it simplifies the creation and editing of even complex algorithms.

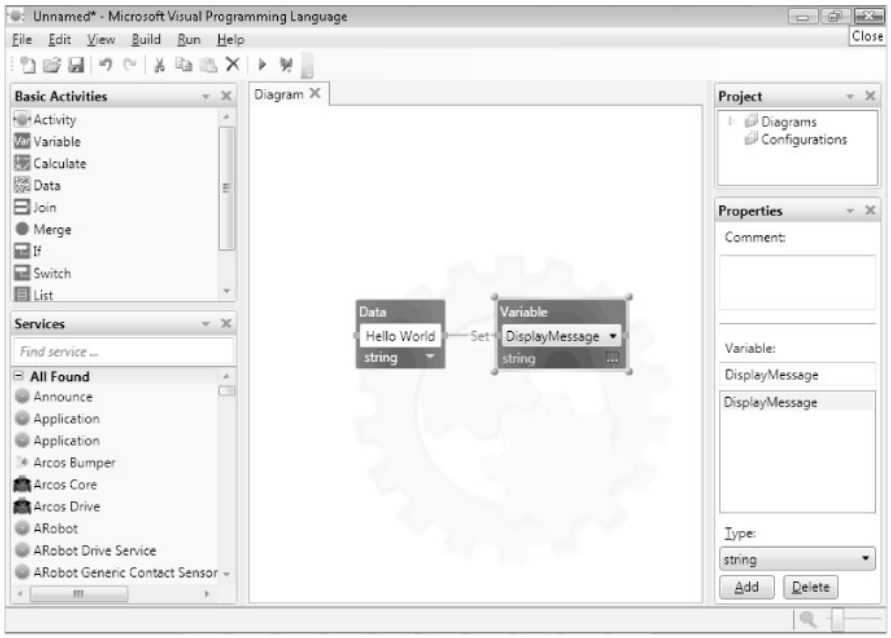

Figure 54 - VPL context

As shown in Figure 54, a VPL application consists of blocks for actions/services and connections. In addition, the activities in VPL are blocks performing standard actions as the data manipulation, the use of global variables for a project, or calculations.

VPL, although a graphical environment, is based on messages flows. The different blocks exchange messages through the main port or by creating flows through notifications.

The different work flows, running in parallel, follow simple rules for a competitive access to data in a consistent manner.

In the following sections, practical applications for the completion of the final application through the use of components of Microsoft Robotics Developer Studio are illustrated.

4.6.3 The VSE component

The Microsoft VSE (Visual Simulation Environment) provides an integrated environment that allows you to perform simulations, even complex ones, in order to test different applications. It gives the ability to create a simulated scenario, not only robots but also environments and other entities, such as obstacles, using a realistic

3D rendering. The VSE uses the AGEIA PhysX engine for the physical simulation of the environment. Without it, the simulation would be useless, since it could in no way represent the world that you are trying to simulate. For example, entities would be not affected by gravity or friction. Finally, the physics engine allows entities to interact with the environment in a realistic way.

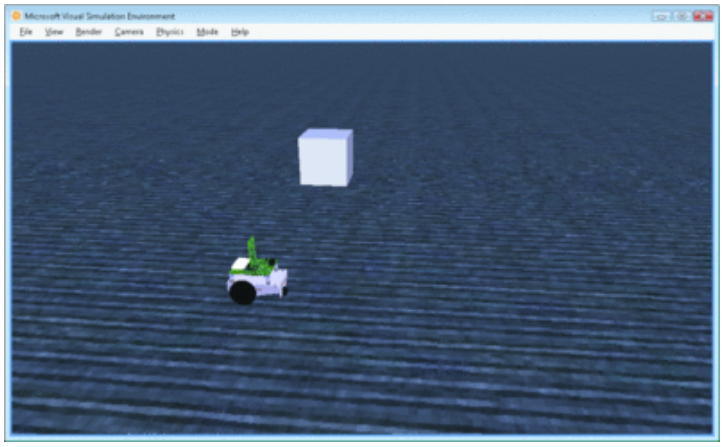

Figure 55 - Simulation environment

The VSE gives the possibility to create custom simulation scenes essential to test, in security, the applications and the control algorithms developed. The simulation phase, has also given the opportunity to simulate the noise and "bottlenecks" in the existing context. The presence of a tool such as VPL has permitted to build control loops in a graphical and intuitive manner.

MRDS allows "painless" transfer by the simulated environment to the real one, thus giving the user the ability to focus on simulation to better understand the role of parameters in the control laws and to identify the optimal values.

4.6.4 U-Go robot applications

U-Go Robot applications are different, thanks to the simplicity in the management and reuse of code provided by MRDS. This last one was a useful tool for carrying out the various applications and control algorithms.

The sensors management provides the opportunity of changing sensor in relation of the control algorithm.

The code, so written, is modular respecting the interfaces between the services; moreover, the code writing is simple, neat and easy to understand.

MRDS makes possible different applications of U-Go Robot, with different sensor configurations, due to the simplicity in the sensors management through services. To add a new feature of a new sensor is sufficient adding a new sensor service and a new control algorithm service for the new application.

4.6.5 MRDS Interface Sensors

As previously mentioned, in MRDS each sensor corresponds to a service. This section is a MRDS service overview of the sensors on board of U-Go robot. For each service the interfaces provided to the navigation service will be detailed.

4.6.5.1 GPS Service

A GPS service provides an interface between the hardware module and control service. The service implements the serial interface, manages the string received through the serial port, unpacks and decodes the string attributing a value to the different variables.

The handler of the output values is related to the action “GetLLA” of the main port. The service architecture consists of a thread.

The service architecture consists of a hardware thread handles RS-232, the processing of the received string and updating of the service variable. The handler attached to the main port allows forwarding updated values of the VPL flow.

The input to the VPL action GetLLA of the MyGPS service is null; the service output variables are the following:

- ***ucttime:*** The time provided to the GPS module in UCT format.
- ***Gpsqi:*** A value related to the GPS signal quality.
- ***numSats:*** Satellites number received by GPS module.
- ***latitude longitude e altitude:*** current GPS position.

- ***nuoviDati:*** boolean value, indicating that the values are generated from current data than the previous.
- ***ggaString***: complete string received in input, forwarded to the output for the log file or other functions.

All information is contained in a single output string properly processed and validated. The string is sent with a frequency of 1 Hz or 5 Hz. The field *nuoviDati* is set to a true value when a new string has been validated and unpacked.

4.6.5.2 MTi Service

The management service of the inertial sensor (MTi) is similar to the GPS service. It uses a library provided by the manufacturer of the sensor available for the most common programming languages. It is necessary to properly import this library from the installation folder of the sensor at the following path:

C:nProgramminXsensnSoftware DevelopmentnCMTDllComnXsensCMT.dll

To start the service, as the GPS service previously mentioned, a null message shall be sent as VPL input. Inside the service, a thread manages the processing of the input data sensor. The outputs are the data provided by the sensors:

- ***Roll Pitch Yaw:*** Orientation values in the system reference of the sensor within the frame of the inertial system.
- ***AccX AccY AccZ:*** acceleration values in all three spatial directions.
- ***gyrX gyrY gyrZ:*** components of the acceleration detected by the gyroscopic sensor.
- ***magX magY magZ:*** components of the magnetic field detected by the sensor.
- ***Temperature:*** temperature value measured by the MTi internal sensor.

4.6.5.3 Laser Scanner Service

To provide the management of the Laser Scanner has been used a service developed and tested by Ben Axelrod (Axelrod): it allows the management of the sensor and its initialization.

The sensor requires an initialization string when the application starts and the output from the service is an array of measures along the different radial directions.

The proper functioning of the sensor is secured by a further flow VPL, as shown in Figure 56: the first flow is executed only once when the application starts and ensures the proper initialization of the sensor. Secondly it queries sensors and updates to the new values the variables in the diagram.

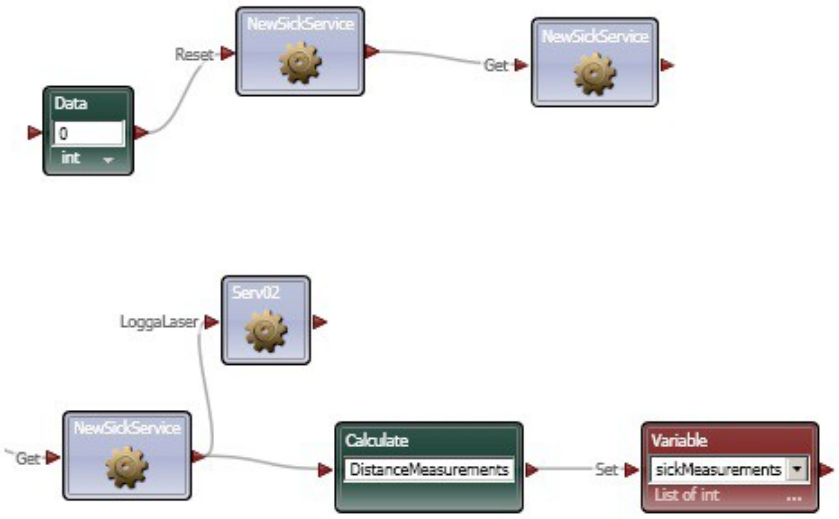

Figure 56 - VPL initialization and management flows of the Scanner Laser. The second operation may be linked to a timer for the sensor polling

4.6.5.4 Bird-Eye Vision

An important sensory information for the final application is the result of the algorithm based on machine vision library OpenCV. The vision algorithm, called “Bird Eye Vision" (Aranzulla *et al.* 2010), through the processing of images from stereocamera, provides information on a viable path and any obstacles located at an increased distance from the laser scanner range.

The service and its implementation are more useful in the case of outdoor inspection, e.g. vineyard context, and till now, it hasn't been used for greenhouse environments.

The result of the vision algorithm is a further sensory information for the control service and consists of two arrays, one attractive and one repulsive, each containing five values. The values represent the forces of attraction and repulsion toward the path travelled in an opposite direction to the obstacles identified.

The exchange of data between two computers is guarantee d by the communication service via UDP protocol, as described in section 4.6.5.6.

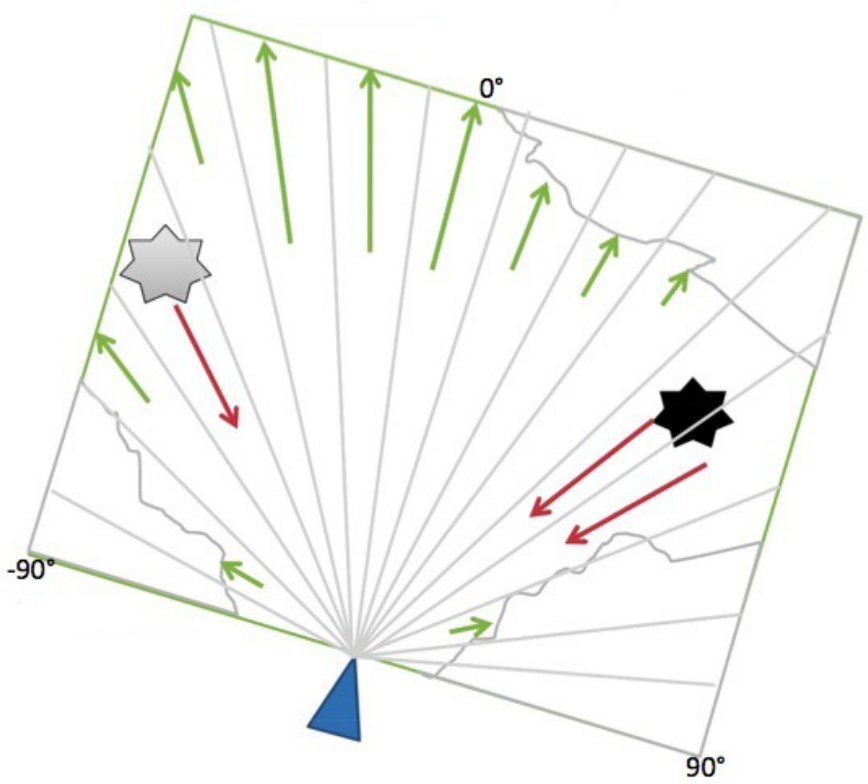

Figure 57 - Representation of the output vectors of the vision algorithm: in green the attractive forces, reds are repulsive.

The directions of the five values of the two arrays are radial directions along five fixed angles: -23°, -14°, 0°, 14°, 23° in the reference system shown in Figure 4.4.

4.6.5.5 Ultrasound sensor (US) Service

On-board the robot there are 8 US sensors:

- E0, E2, E4 on the left side
- F0, F2, F4 on the right side
- E6 on the front position

- E6 on the back

In Figure 58 is shown the exactly pose on the robot

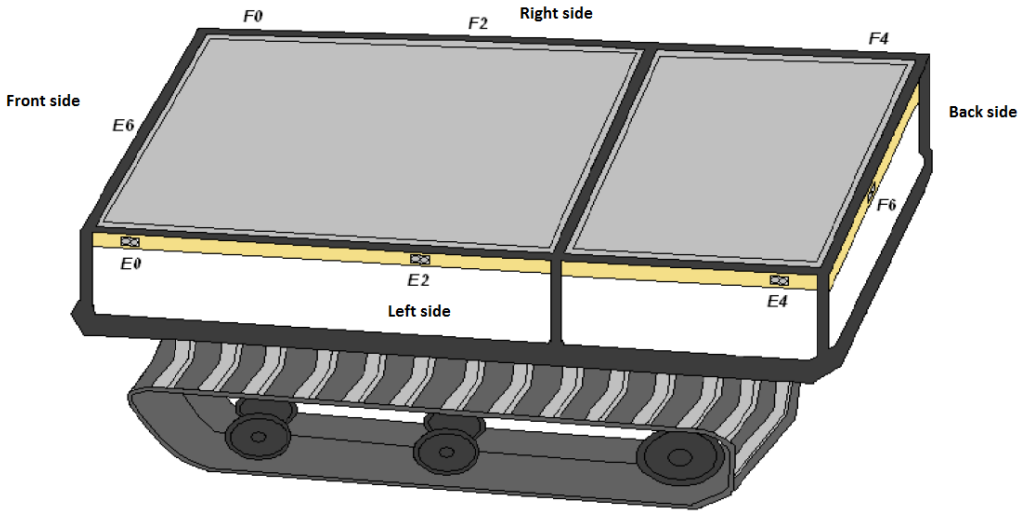

Figure 58 - Position of US sensors on the robot

Reading the measure of each sensor, the aim is to obtain parameters useful for the robot driving. In more detail, the algorithm allows to extract two parameters:

- *delta*: translation of the robot axis respect to the corridor's axis (negative if the robot is translated towards the left or positive in the opposite case)
- *theta*: rotational angle between the robot axis respect to the corridor's axis (negative if anticlockwise, positive viceversa)

To better understand the meaning of Delta and Theta refers to the following explanation and figure.

Delta estimation

The robot translation can be measured through the difference between the sensors measures on the left side and those on the right side.

In other words as:

difference[1] = misure_E0 – misure_F0

difference[2] = misure_E2 – misure_F2

difference[3] = misure_E4 – misure_F4

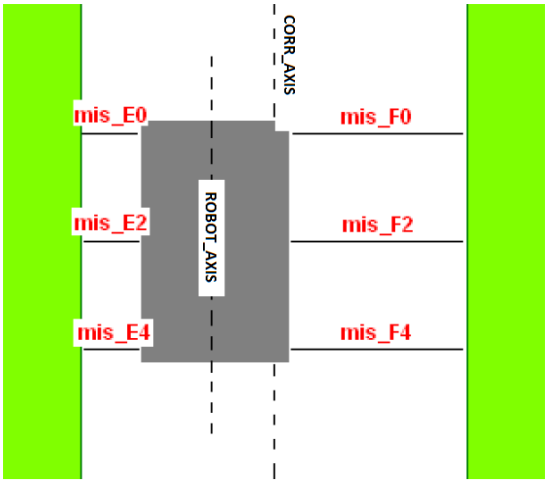

Figure 59 - robot translation respect to the corridor axis

Theta estimation

To know the orientation of the robot, it's necessary to take sensor measurements in subsequent couple on the same side, and note the distance between the sensors, we can calculate the angle, as seen in Figure 60.

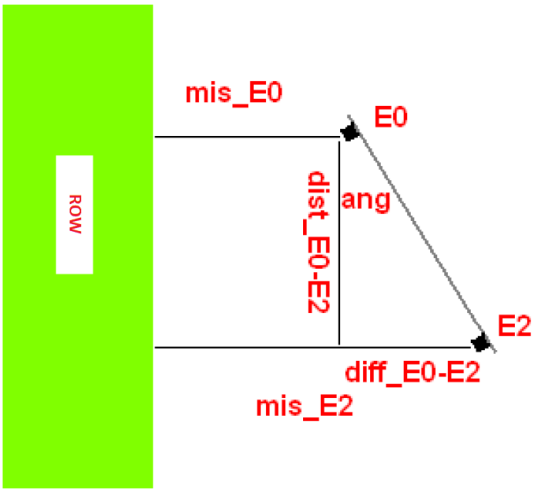

Figure 60 - robot orientation respect to the row.

For example, supposed sensors calculate the distance from a perfectly flat wall, known sensor measurements of E0 and E2 and the distance between the sensors, which is set at 0.50 m, it is possible to know the angle between the two sensors by the following formula:

$$angle\_1 = \arcsin[(misure\_E0 - misure\_E2) / distance]$$

angle_1 is the first angle calculated for the left side of the robot.

The second angle is calculated through the difference between E2 and E4, and in the same way for the right side.
Obviously it should be noted that the row is not perfectly flat, but still contains the points where the foliage is sparser. Consequently, the measures do not have high accuracy, therefore, the orientation will be known within a certain margin of error which will be determined on the basis of experimental tests.
Finally the robot orientation is estimated considering the average of the calculated four angles.

4.6.5.6 UDP Service

The information provided by the algorithm of computer vision (paragraph 4.2.4) is sent to the navigation algorithm via UDP protocol. In fact, a PC has been dedicated to the execution of image processing and an Ethernet connection allows the data exchange with the PC dedicated to the navigation control.

So a MRDS service to allow these operations has been created for the final application.

In more detail, a thread allows the network communication via UDP and the data processing. The service has been customized in order to receive the data from the vision system. The outputs of the action "GetVision" are two arrays updated each time new data come from the Vision PC:

- ***attractiveArray*** is an array of five values: along five directions are shown attractive forces.
- ***repulsiveArray*** is a further array of five values representing the obstacle forces for the obstacle detection in addition to the data performed by the Scanner Laser.

These values are the result of the algorithm performing the detection of the drivable path and the detection of the obstacles located at a greater distance from the Scanner Laser.

4.7 Navigation

The navigation algorithm, running on the MRDS platform, takes care of generating the control reference for trajectory of the used robot; in our case the code has been customized for the U-Go Robot.

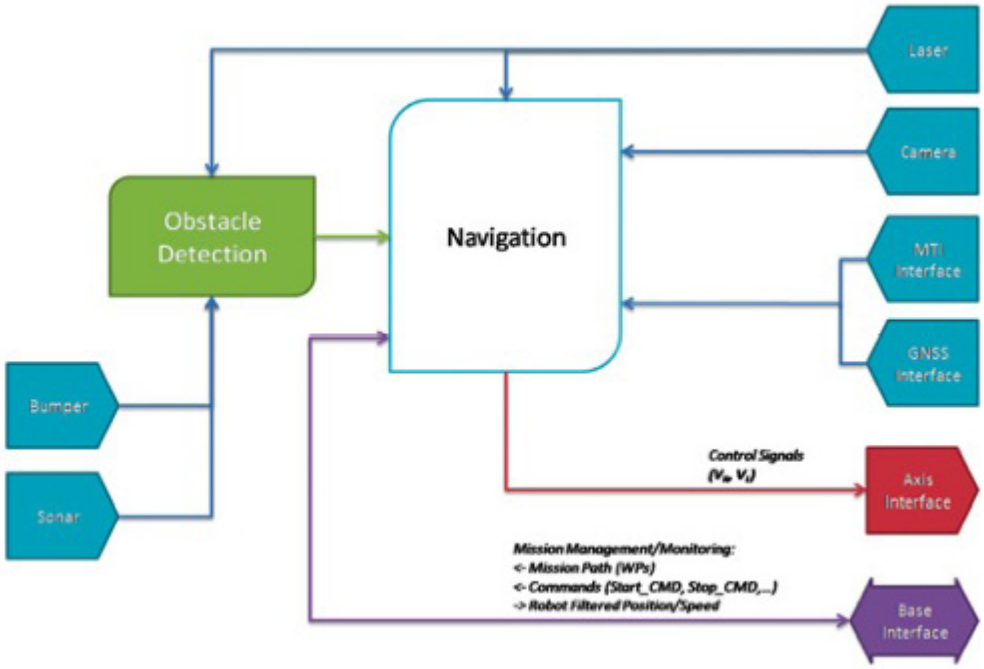

Figure 61 - System control architecture

First of all, different algorithms have been tested using data coming from the laser scanner and the DGPS. Relying on these data, an Obstacle Avoidance (OA) algorithm was developed. One of the techniques used for OA was the Potential Field Method (PFM) which allows the motion control to avoid collisions with the obstacles detected by sensors during the motion itself, without losing the main task; for example to achieve a target configuration identified by a GPS waypoint. The result of this technique is a sequence of movements that allows to safely drive the vehicle towards the target without collisions.

4.7.1 Autonomous Navigation Algorithm

This section is dedicated to the presentation of algorithms applicable to the mobile robot, subject of the related work.

4.7.1.1 Target Following

Among the different algorithms for autonomous navigation, one of the algorithms used in this work is the tracking of targets, shown in Figure 62. The operation of the algorithm is based on the compensation of the angle between the robot and the target, while the robot is going towards the target point.

Varying the rotation speed of the robot, the algorithm compensates the error between the orientation vector of the robot and the vector that identifies the target.

Considering the coordinates of the position of the robot (Xr and Yr) and Xg and Yg the coordinates of the goal, the angle that the robot detects is given by:

$$\theta goal = \tan^{-1}\left(\frac{Yg - Yr}{Xg - Xr}\right)$$

a rotational speed proportional to the error angle will be applied:

$$\omega = K_\omega \theta_{error} = K_\omega(\theta_{goal} - \theta_{robot})$$

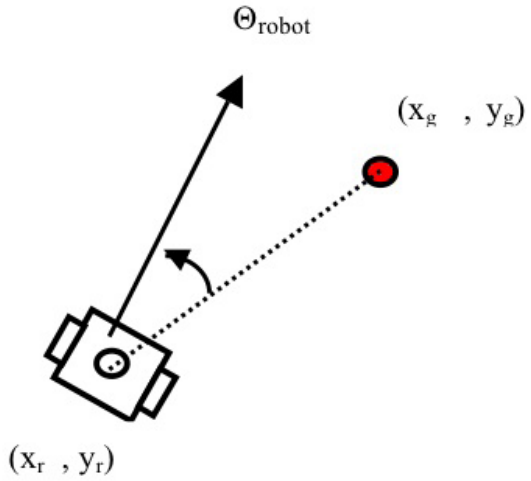

Figure 62 - Representation of the algorithm for tracking targets

4.7.1.2 Trajectory Tracking

In some applications it is preferred that the robot follows a precise trajectory rather than just the target. The navigation of the robot through trajectory tracking (in Figure 63) takes care of the position (Xr and Yr) and orientation of the robot (θrobot).

The goal is the trajectory track given by joining two contiguous points (X1, Y1) and (X2, Y2) of the path the robot must follow. A straight line (trajectory to be followed) in that tracking step, has the equation:

$$\frac{y - y1}{y2 - y1} = \frac{x - x1}{x2 - x1}$$

- (X1; Y1) and (X2; Y2) are the coordinates of the two extremes of the trajectory
- x and y are the variables for the straight line equation in the plane

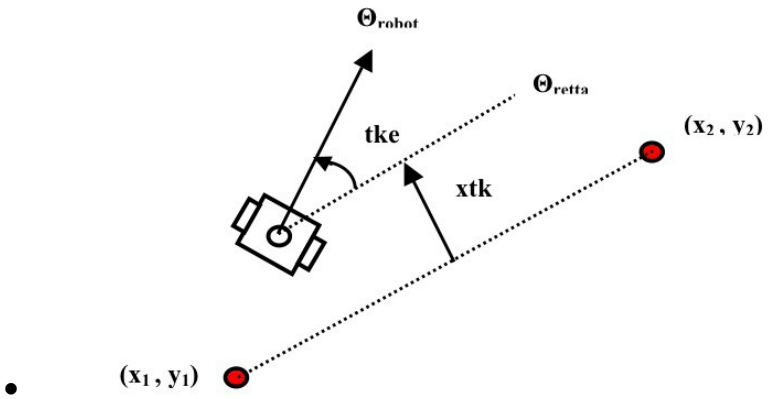

Figure 63 - Representation of the algorithm for trajectory tracking

In this case the coefficients of error to be compensated are two. We need a combined action of two weights given by:

$$e_{tot} = K_1 e_{orientation} + K_2 e_{distance}$$

with:

$$e_{orientation} = \theta_{robot} - \theta_{straight\ line}$$

the orientation error is:

$$e_{\underline{distance}} = \frac{Y_r - Mx_r - q}{\sqrt{1+m^2}}$$

The error distance between the robot and the straight line that identifies the trajectory is given by the intersection of the normal to the straight line through the robot.

M is the angular coefficient and q is the ordinate of the origin of the straight line trajectory.

4.7.1.3 Estimation of the Robot orientation

The calculation of the orientation of the robot can be derived directly from measurements made by an inertial sensor (§4.4.7), or approximated by the difference between two consecutive positions (Figure 64).

Let (X1, Y1) and (X2, Y2) the two consecutive positions of the robot, the angle of orientation for the robot will be:

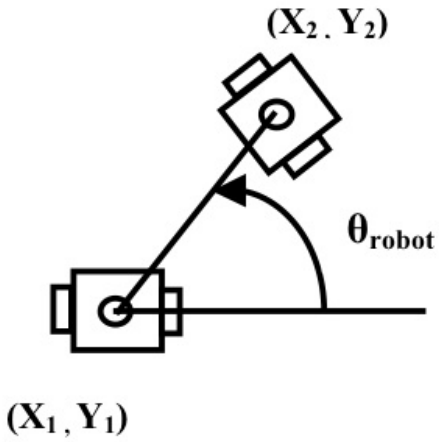

Figure 64 - Angle representing the orientation of the robot

$$\theta robot = \tan^{-1}\left(\frac{Y_2 - Y_1}{X_2 - X_1}\right)$$

In both solutions, estimated using inertial sensor or calculated between two successive positions, the estimated orientation of the robot is used in navigation algorithms discussed earlier.

4.7.1.4 Differential Drive

The robot has a differential traction (Differential Drive) which consists of two independent actuated wheels or tracks.

As shown in Figure 65, the traction is a combination of differential rotation speeds of two wheels. The robot can rotate on itself with two equal speeds (but in opposite direction for the two wheels).

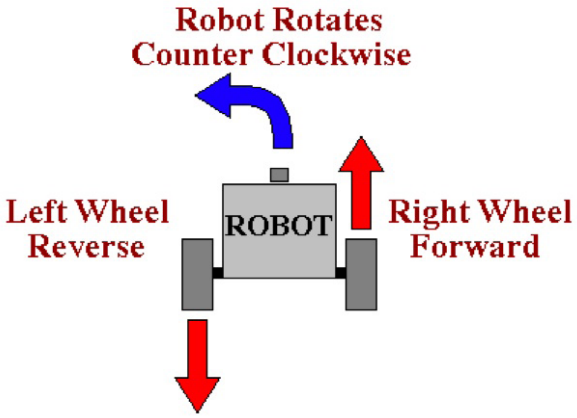

Figure 65 - Differential drive principle

The angular speed (ω) and the linear speed (v) of the traction differential vehicle are:

$$\omega = \frac{Vr - Vl}{b}; \; v = \frac{Vr + Vl}{2}$$

Where b is the distance between the two wheels (wheelbase) and Vr e Vl are the rotation speeds

4.7.2 Potential Field Method

The Potential Field Method (PFM) is a technique of obstacle avoidance that translates the problem of autonomous navigation in the presence of obstacles in a vector problem as shown in Figure 66.

Sensory information and information about the target to be reached are vectors for a mathematical treatment of the problem. The method is to represent the robot as a particle moves in configuration space, under the influence of a force field.

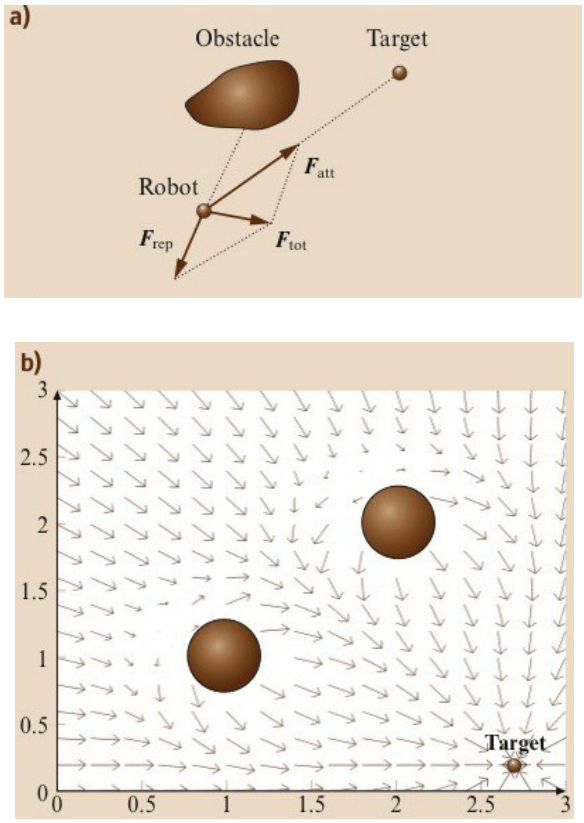

Figure 66 - Potential Field Representation: (a) Forces vector system applied to the robot during the navigation (b) Vectorial field configuration in robot workplace

The target location applies on the particle a force of attraction F_{att}, while the obstacles apply repulsive forces F_{rep}. At any instant, the motion is processed in such a way that the vehicle follows the direction of the artificial induced force. This force is derived from the sum of the attraction and repulsion vectors:

$$F_{tot}(t_i) = F_{att}(t_i) + F_{rep}(t_i)$$

The force of attraction is a vector having the direction joining the center of the robot and the target location. The module is equal to the attraction constant:

$$F_{att} = K_{att}n_{goal}$$

The repulsion force, F_{rep} is the result of the repulsive forces applied by obstacles. Let q be the position vector of the robot and pj the j-th obstacle position vector. The single repulsion force applied by the j-th obstacle is inversely proportional to the distance of the robot from the obstacle, multiplied for a repulsion constant K_{rep}. The

single force of repulsion is a vector having the direction of the joining line between the centre of the robot and the obstacle j-th.

The final algorithm implementation is given by a weighted vector sum composed by several vectors each attractive or repulsive. Each sensor has been translated into a vector for the final algorithm. The following sections describe the different forces used and their combination.

4.7.3 PFM for U-Go Robot

The autonomous modality of the U-Go robot, consists on reaching the coordinates of the waypoints along the path that the robot has to accomplish, they represent the forces of attraction for the robot.

The waypoints can be set in manual mode, choosing which points you want the robot marks or entering previously the coordinates of the waypoint, expressed in the related reference system respect to the workplace.

The vectors of force used by the algorithm of navigation are given by a weighted sum of the individual vectors according to the formula:

$$F_T = \_{WP}F_{WP} + \_{laser}F_{laser} + \_{CV_a}F_{CV_a} + \_{CV_r}F_{CV_r}$$

The attractive forces (F_{WP} and F_{CVr}), are given by the vector that identifies the waypoint and by the data coming from the vision algorithm (it suggests a preferred path); while the forces of repulsion are given by measurements of the laser scanner and vision algorithm that provides more information on obstacles at distances greater than the range of the laser scanner.

The different vectors of forces are centred in the robot with suitable coordinate transformations for the coherence between reference systems. Figure 67 is a representation of the PFM applied to U-Go Robot:

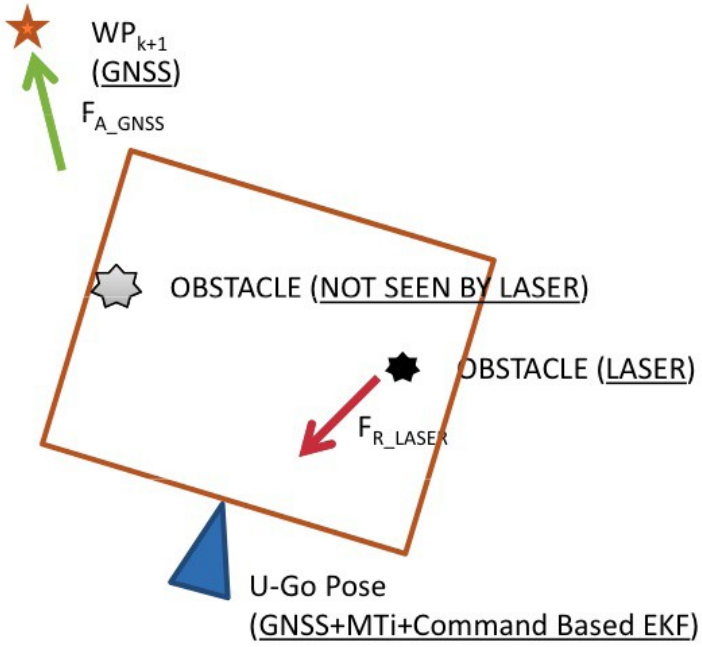

Figure 67 - PFM method

The position and orientation of the robot in the configurations space are the values coming from GNSS and inertial sensor (MTi). An Extended Kalman Filter (EKF) has been also implemented to make up for absences of GPS.

4.7.4 PFM Implementation in MRDS

The control implementation in the MRDS is contained within a service. The final version, including all the sensory information, is implemented by the handler action "BirdNavigation" named by the computer vision algorithm and described in paragraph 4.4.4.

All sensory information related to the determination of the position and orientation of the robot, the obstacles detection by using laser scanner and the information from stereo cameras for machine vision are forwarded to the service:

- XR and YR: are the position coordinates coming from GNSS expressed in the reference system of the workplace
- Distance Array: they are all measurement coming from scanner laser, each one for every 0.5° (for a total of 180° on front of the robot)

- New Data: is a boolean value indicating the occurrence of a new data coming from sensors
- Yaw: is the unique value of the inertial sensor used for the estimation of the robot orientation respect to the North (Reference System of the Earth)

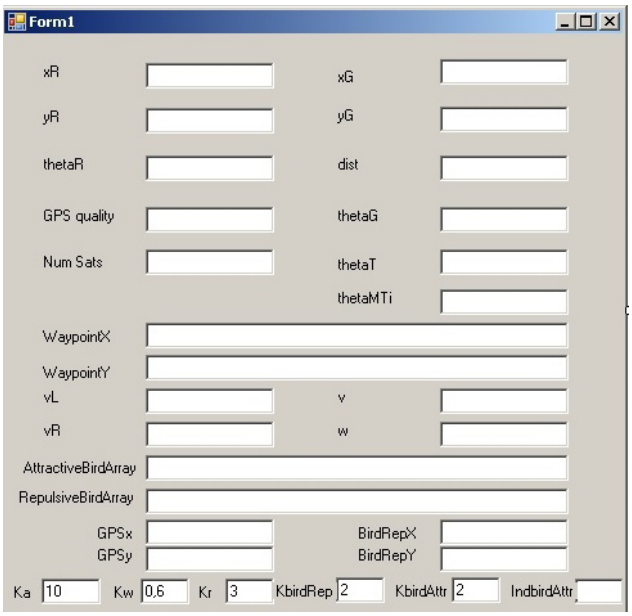

Figure 68 - Feedback form of the PFM in MRDS

- BirdAttractiveArray and BirdRepulsiveArray are two arrays results of the image processing coming from stereocam: only the highest value of the attractive force has been chosen, which is responsible for the choice of the best path. Instead, all values of the repulsive forces are used for 'obstacle avoidance
- GPS quality and Number of Satellites values are information about the GNSS for estimating the quality of the signal and the information provided

The outputs are: the linear and rotation speed references (V_{lin} and V_{turn}) for the DC Motor Driver.

4.8 Control Architecture in MRDS

This section contains an explanation of the robotics application: more details will be added on the final architecture used.

The VPL global architecture is divided into streams of messages exchanged between different services and cooperating one with each other. The variables of the VPL diagram represent the state of the robot determined from the sensors outputs.

The final architecture presented in this section refers to the case where the position of the robot is provided by using DGPS. Similar considerations must be made in cases where such information comes from "low cost" solutions previously discussed.

4.8.1 Sensor Polling in VPL

For a proper working of the final application and the management of the robot in autonomous and manual mode, synchronization of timers and sensors has been used.

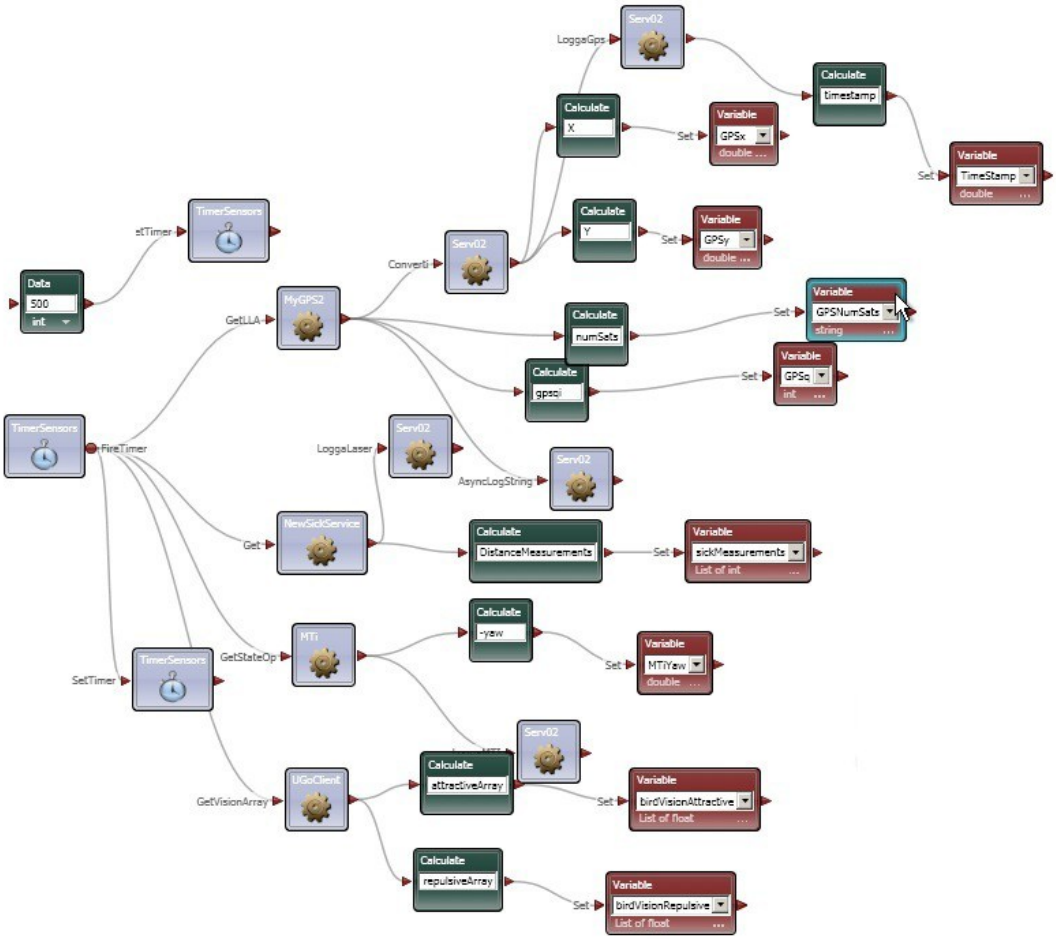

Figure 69 - Sensor polling in VPL in the final application.

The management sensors timer allows asynchronous interrogation of all the sensors services interface and the update of the VPL diagram global variables. As shown in Figure 69, the timer is initialized for operation at a frequency of 2 Hz and sends messages to each sensors service in order to update all variables. In the diagram in Figure 69 additional services (e.g. Serv02) for the log of the data file coming from different sensors are shown.

4.8.2 VPL Data Flow

The flow diagram architecture has been developed taking into account the different navigation modalities, autonomous and manual, and taking into account the necessary sensors.

In Figure 70 the commands flow allowing the managing of the joypad, is developed by VPL. It guarantees the use of a wide range of joypad. In succession a service specifically created (GamePadManager) manages the state of the robot according to the modalities settings and to the timer variables for the final control diagram.

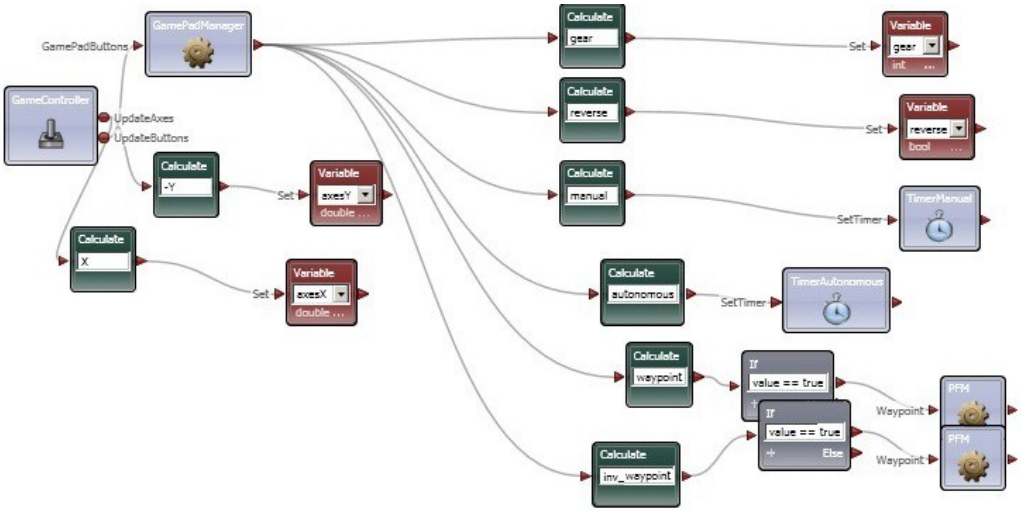

Figure 70 - VPL diagram for the joypad commands management with different modalities and functionalities.

- ***"gear"* and *"reverse"* variables:** are the driving speed (1, 2 or 3 with higher speed at level '3') and the flags for the reverse mode (for robot backward motion).

- **"*TimerAutonomus*" e "*TimerManual*":** are two timers that set with a positive or void value allowing the modalities choice: autonomous or manual.
- **Waypoint management service:** used for waypoints setting (add, remove or reverse) in the autonomous navigation modality.
- **"*X*" e "*Y*" Axis:** are the joypad values, translating in moving commands in the manual navigation modality.

In Figure 71 the data flow needed for the DC Motor Driver action in the manual navigation modality is shown. The "*TimerManual*" is set in a value different from zero if the manual modality is chosen (figure 5.8). A message is sent to the DDCingo service containing the axis values of the joypad that are translated in V_{lin} e V_{turn} commands for the differential drive of the vehicle.

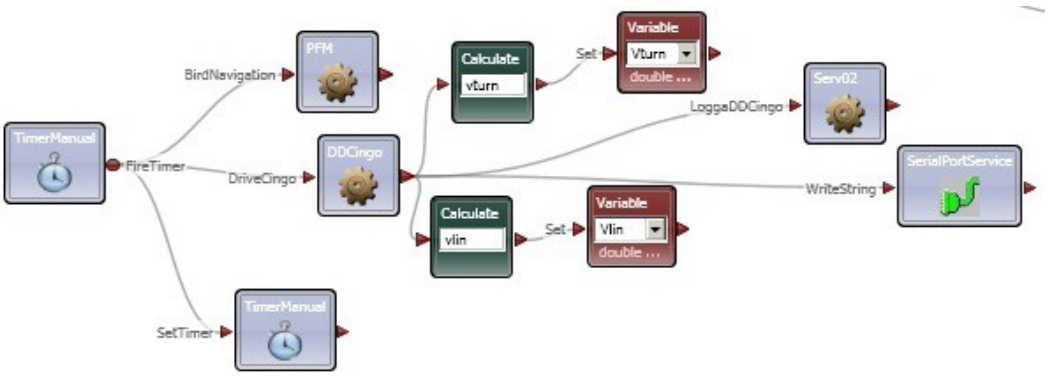

Figure 71 - Manual Navigation management

The movement string is created and sent through VPL message by using a RS-232 protocol ("*SerialPortService"*) to the engine driver.

An additional message is sent to the PFM service for displaying the feedback form in manual mode.

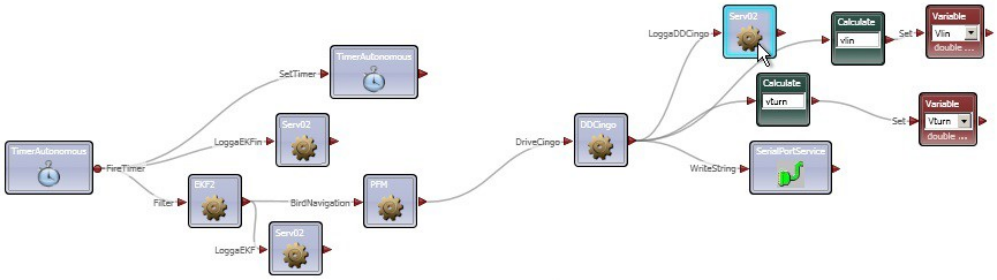

Figure 72 - Autonomous Navigation management.

In Figure 72 the data flow for the autonomous navigation composed of a succession of the following service:

"EKF2": service implementing the Kalman filter for the interpolation of the data in the case of GNSS lacks: the output is GPS measurement if the quality is acceptable, otherwise it is a measure interpolated by the filter.

***"PFM"* service:** with action "*BirdNavigation*" having in input all the data coming from the several sensors and in output the speed references for the "*DDCingo*" service.

***"DDCingo"* service:** with the action "*DriveCingo*". It translates the speed references coming from PFM in commands to be sent through RS-232 protocol to the DC motor driver.

***"serv02"* service:** dedicated to the log file of the EKF output values or commands sent to the tracks.

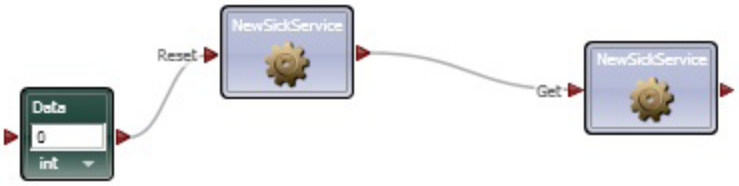

Figure 73 - Laser Scanner initialization

In Figure 73 the flow for the initialization of the laser sensor: it is performed only once at application startup to initialize, by sending a string of start to the laser scanner. In Figure 73, after initializing the sensor, measurements are in an array of 360 elements (180° with a resolution of 0.5°).

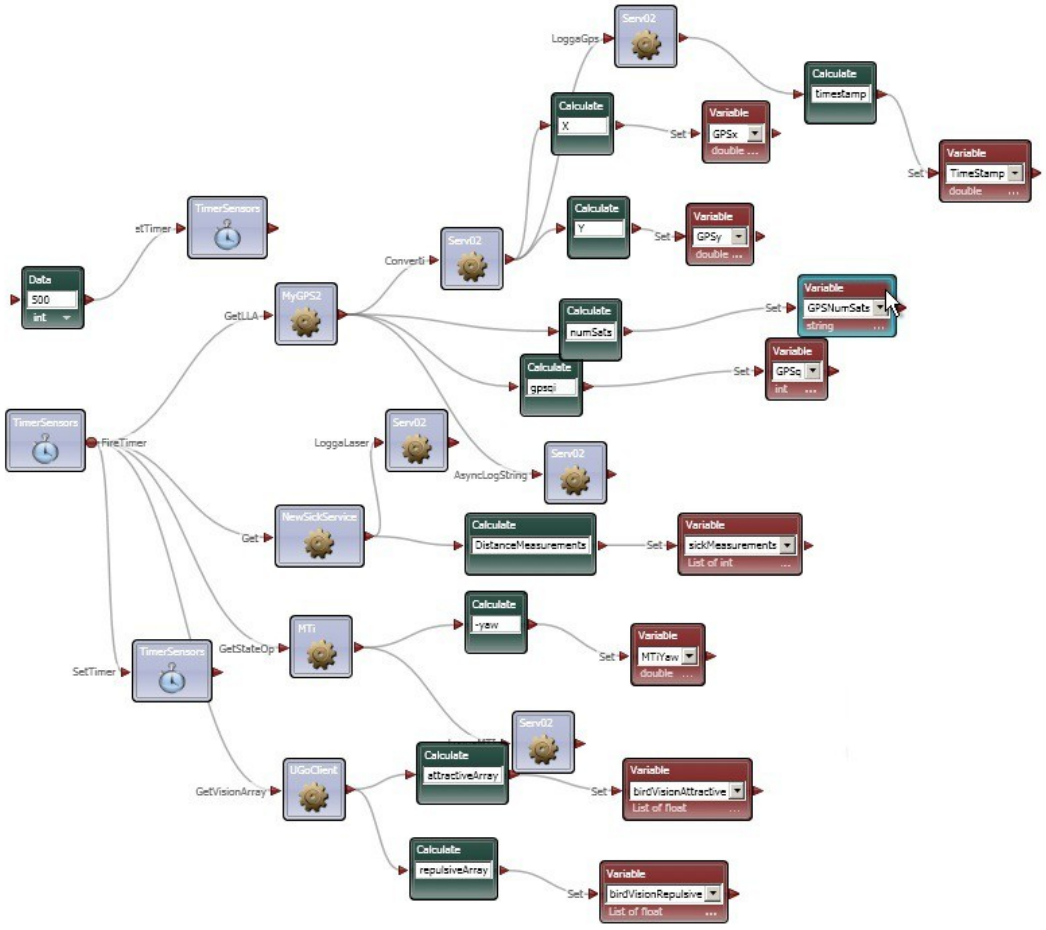

Figure 74 - Sensors timer initialization and polling

In Figure 74 (the same of Figure 69, repeated here in order to make it easy to analyse) the flow for the interrogation of multiple sensors (sensor polling) is represented. The timer is initialized at the starting of the VPL application in the first flow of Figure 74, while the second flow deals with interviewing various sensors through the interface services, logging services with "*Serv02*" and updating global variables (red blocks).

These variables represent the state of the system and they are used in navigation services as inputs for the services.

4.8.3 Final Architecture

Figure 75 shows the entire and complete architecture for the final application. It consists by the flows for the two navigation modalities and sensors polling.

The cooperation between the different flows, even if disconnected one from each other, is managed through the sharing of system variables that are represented by global variables in VPL.

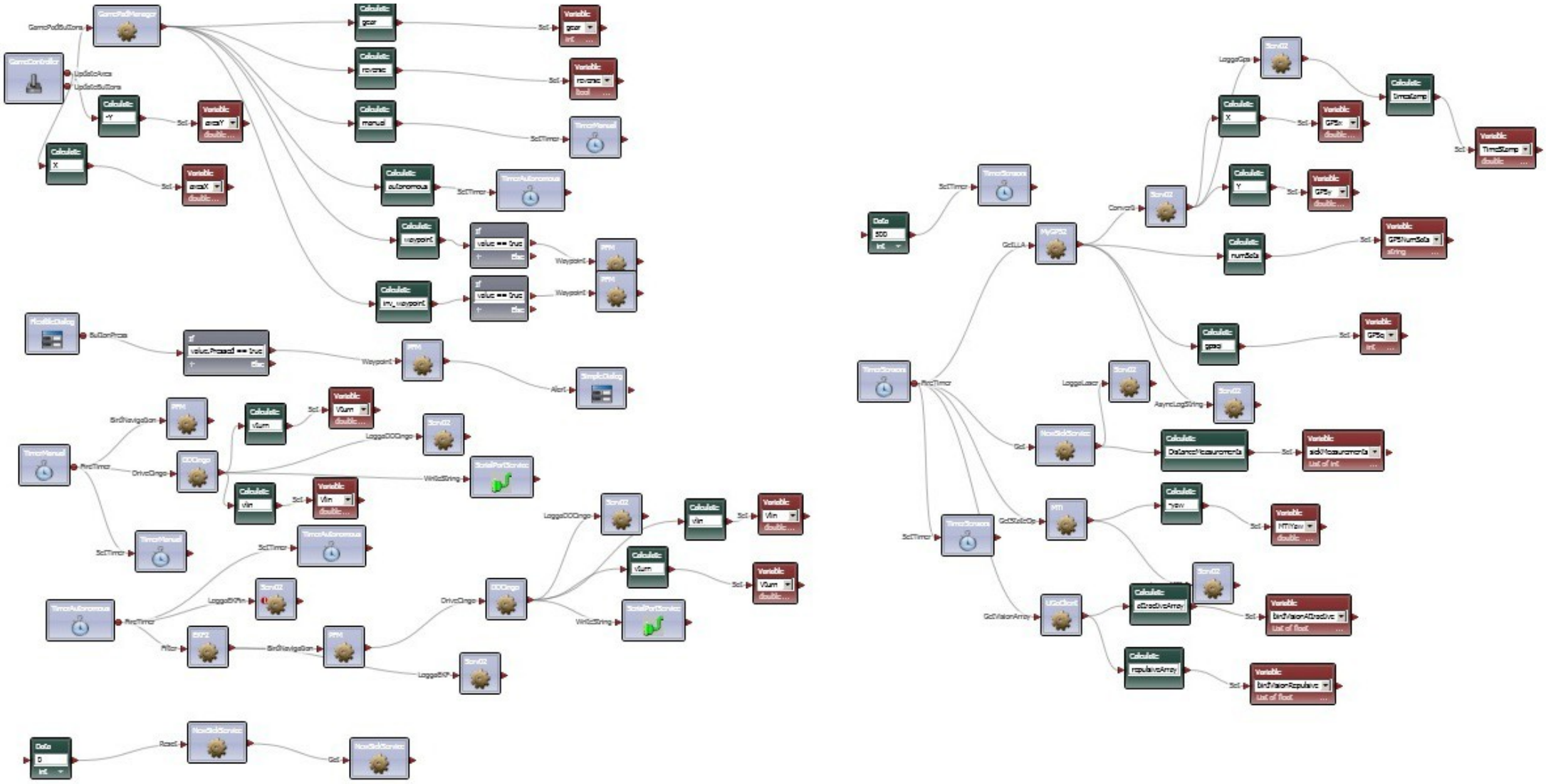

Figure 75 - Complete diagram of the final architecture

5 The vision system for tomato detection

5.1 Preliminary concept on object recognition

Pattern recognition develops and applies algorithms that recognize patterns in data. These techniques have important applications in character recognition, speech analysis, image analysis, clinical diagnostics, person identification, machine diagnostics, and industrial process supervision. Many chemistry problems can be solved with pattern recognition techniques, such as: recognizing the provenance of agricultural products (olive oil, wine, potatoes, honey, etc.) based on composition or spectra; structural elucidation from spectra; identifying mutagens or carcinogens from molecular structure; classification of aqueous pollutants based on their mechanism of action; discriminating chemical compounds based on their odour; classification of chemicals in inhibitors and non-inhibitors for a certain drug target.

To better understand details of the present work, some basic notions of pattern recognition will be shown as follows. A pattern (object) is any item (chemical compound, material, spectrum, physical object, chemical reaction, industrial process) whose important characteristics form a set of descriptors. A descriptor is a variable (usually numerical) that characterizes an object. A descriptor can be any experimentally measured or theoretically computed quantity that describes the structure of a pattern: spectra and composition for chemicals, agricultural products, materials, biological samples; graph descriptors and topological indices; indices derived from the molecular geometry and quantum calculations; industrial process parameters; chemical reaction variables; microarray gene expression data; mass spectrometry data for proteomics.

The major hypothesis is that the descriptors capture some important characteristics of the pattern, and then a mathematical function (machine learning algorithm) can generate a mapping (relationship) between the descriptor space and the property. Another hypothesis is that similar objects (objects that are close in the descriptor space) have similar properties. A wide range of pattern recognition algorithms are

currently used to solve chemical problems: linear discriminating analysis, principal component analysis, partial least squares (PLS), artificial neural networks, multiple linear regression (MLR), principal component regression, k-nearest neighbours (k-NN), evolutionary algorithms embedded into machine learning procedures, support vector machines.

5.1.1 Visual Feature

In computer vision and image processing the concept of feature is used to denote a piece of information which is relevant for solving the computational task related to a certain application. More specifically, features can refer to:

- the result of a general neighbourhood operation (feature extractor or feature detector) applied to the image
- specific structures in the image itself, ranging from simple structures such as points or edges to more complex structures such as objects.

When features are defined in terms of local neighbourhood operations applied to an image, a procedure commonly referred to as feature extraction, one can distinguish between feature detection (Feature Detection, Wikipedia) approaches that produce local decisions whether there is a feature of a given type at a given image point or not, and those who produce non-binary data as result.

When feature extraction is done without local decision making, the result is often referred to as a *feature image*. Consequently, a feature image can be seen as an image in the sense that it is a function of the same spatial (or temporal) variables as the original image, but where the pixel values hold information about image features instead of intensity or colour. This means that a feature image can be processed in a similar way as an ordinary image generated by an image sensor. Feature images are also often computed as integrated step in algorithms for feature detection.

A specific image feature, defined in terms of a specific structure in the image data, can often be represented in different ways.

A common practice is to organize the information provided by all these descriptors as the elements of one single vector, commonly referred to as a *feature vector*. The set of all possible feature vectors constitutes a *feature space*.

In computer vision, visual descriptors or image descriptors are descriptions of the visual features of the contents in images, videos, algorithms, or applications that produce such descriptions. They describe elementary characteristics such as the shape, the colour, the texture or the motion, among others.

Feature extraction involves sparse or dense representation of image regions as feature vectors while the detector architecture specifies exactly how the spatial occurrences of these feature vectors related to each other are exploited to obtain detection decisions. Feature extraction typically captures intensity patterns, texture details, and/or shape and contour information. There are two contrasting views in computer vision on how to compute feature vectors:

- One approach is based on sparse features extracted from a set of salient image regions. The motivation is that not all image regions contain useful information: many are uniform, texture-less, or too cluttered to use.
- The alternative approach is to densely compute feature vectors on image regions.

Note that the differences between the sparse and dense approaches are not as great as they may seem as the detection of salient regions requires a dense scan of the input image.

5.1.2 Image Descriptors

Descriptors (Visual Descriptors, Wikipedia) are the first step to find out the connection between pixels contained in a digital image and what humans recall after having observed an image or a group of images after some minutes.

Visual descriptors are divided in two main groups:

1. **General information descriptors**: they contain low level descriptors which give a description about colour, shape, regions, textures and motion.

2. **Specific domain information descriptors**: they give information about objects and events in the scene.

General information descriptors consist of a set of descriptors that covers different basic and elementary features like: colour, texture, shape, motion, location and others. This description is automatically generated by means of signal processing.

- **COLOUR**: the most basic quality of visual content.
- **TEXTURE**: also, an important quality in order to describe an image. The texture descriptors characterize image textures or regions. They observe the region homogeneity and the histograms of these region borders
- **SHAPE**: contains important semantic information due to human's ability to recognize objects through their shape.
- **MOTION**: defined by four different descriptors which describe motion in video sequence. Motion is related to the objects motion in the sequence and to the camera motion. This last information is provided by the capture device, whereas the rest is implemented by means of image processing.
- **LOCATION**: elements location in the image is used to describe elements in the spatial domain.

5.1.3 Feature Detection and Matching

Regarding the detector, several different models and techniques have been studied in the literature. We broadly divide them into two categories:

- One common approach is to learn to recognize classes of similar image regions that commonly occur in the given object class. Broadly this can be termed a parts based approach. This approach is not applicable in the case of tomato detection.
- Another, perhaps simpler, approach is to implicitly encode spatial information in the form of rigid templates of feature vectors. This scheme is usually based on densely computed image representations, but sparse

representations can also be used. The overall detection is then provided by an explicit template matching or by use of state-of-the-art machine learning algorithms such as kernel-based Support Vector Machines (SVM).

In more detail:

Template matching (Template matching, Wikipedia) is a technique in digital image processing for finding small parts of an image which match a template image.

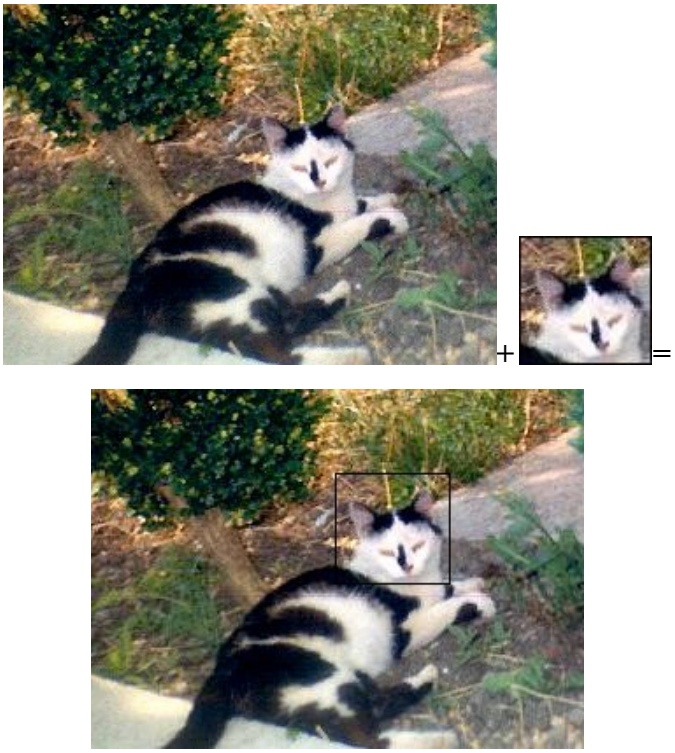

Figure 76 - Template matching concept

Support Vector Machines (SVM): is a concept in computer science for a set of related supervised learning methods that analyse data and recognize patterns, used for classification and regression analysis. The standard SVM takes a set of input data and predicts, for each given input, which of two possible classes the input is a member of, which makes the SVM 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 other words, the goal of SVM modelling is to find the optimal hyperplane that separates clusters of vector in such a way that cases with one category of the target variable are on one side of the plane and cases with the other category are on the other size of the plane. The vectors near the hyperplane are the support vectors. The figure below presents an overview of the SVM process.

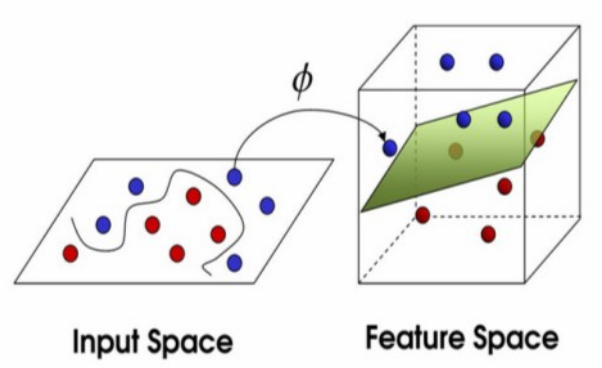

Figure 77 - Principle of Support Vector Machine

5.2 Tomato Detection: the method

As mentioned earlier, an object detector can be viewed as a combination of an image feature set and a detection algorithm.

This thesis focuses on various aspects of the image encoding. It proposes and provides a study of different image encoding schemes, but on the other hand deliberately adopts a standard and a relatively simple learning and testing framework for object detection.

The overall object detection architecture is built around a method for classifying individual image regions. This is divided into two phases. The learning phase creates a binary classifier that provides object/non-object decisions for fixed sized image regions ("windows"); while the detection phase uses the classifier to perform a dense multi-scale scan reporting preliminary object decisions at each location of the test image. These preliminary decisions are then fused to obtain the final object

detections. Both the learning phase and the detection phase contain three stages. Figure 78 depicts these.

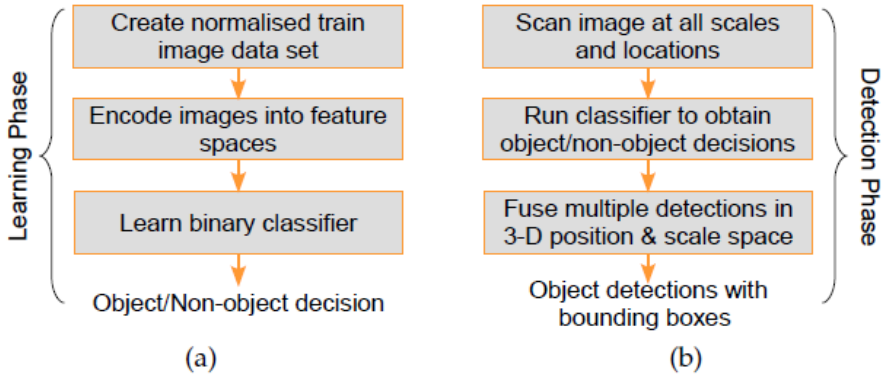

Figure 78 - Overall object detection architecture. (a) The learning phase extracts robust visual features from fixed size training windows, and trains a binary object/non-object classifier over them. (b) The detection phase uses the learned binary classifier to scan the test image at all locations and scales for object/non-object decisions.

The first stage of learning is the creation of the training data. The positive training examples are fixed resolution image windows containing the centered object, and the negative examples are similar windows that are usually randomly sub-sampled and cropped from set of images not containing any instances of the object.

The binary classifier is trained using these examples. Ideally, each positive window contains only one instance of the object, at a size that is approximately fixed with regard to the window size. Details of the data sets used and how we annotate the images are given in §5.4.

This simple window architecture has various advantages. It allows a conventional classifier to be used for detection and relieves the classifier of the responsibility to be invariant to changes in position and scale (although invariance to other types of transformations, changes in pose and viewpoint, and illumination still has to be assured). It also means that the classifier works in relative coordinates (feature position relative to the centre of the current window) which allows relatively rigid template-like feature sets to be used. On the other hand it means that the classifier is run on a large number of windows, which can be computationally expensive and

which makes the overall results very sensitive to the false positive rate of the classifier.

Any classifier can be used for the purpose, but SVM or AdaBoost are common. In this thesis we have chosen to focus mainly on the issue of robust feature sets for object recognition, so we have selected a simple, reliable classification framework as a baseline classifier for most of the experiments.

We use linear SVM as our baseline binary classifier. During detection, the input test image can be scanned at all scales and locations. For each scale and location, the feature vector is computed over the detection window, just as in the learning phase, and the binary classifier is run to produce object/non-object decision for the window. Image regions that contain objects typically produce multiple firings and it is necessary to fuse these overlapping detections into a single coherent one. The overall detection score depends both on the dataset creation and on how finely the test image is scanned and the detections fused.

5.3 Overview of the Encoding Process

Our proposed image feature sets are based on dense and overlapping encoding of image regions using descriptors. Different kind of descriptors are faced in literature; in the next paragraphs a really sub category, the ones analyzed in our approach, will be detailed.

Figure 79 gives an overview of overall encoding process.

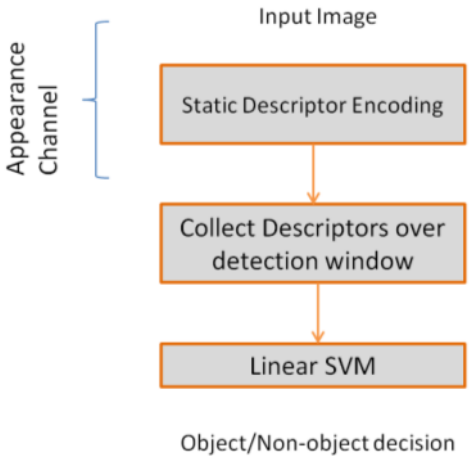

Figure 79 - The appearance channel uses the current image and computes static descriptors over it. Descriptors are collected over the detection window and the combined vectors are fed to a classifier for object/non-object classification.

5.3.1 HOG Descriptors

This section gives an overview of the static HOG feature extraction chain. The method is based on evaluating a dense grid of well-normalized local histograms of image gradient orientations over the image windows. The hypothesis is that local object appearance and shape can often be characterized rather well by the distribution of local intensity gradient or edge directions, even without precise knowledge of the corresponding gradient or edge positions.

We now sketch and motivate each step of this process (Figure 80). The method has been applied in literature in the case of pedestrian detection but we strongly believe in its reusability in other and different context such as ours.

- The first stage applies an optional global image normalization equalization that is designed to reduce the influence of illumination effects. Image texture strength is typically proportional to the local surface illumination so this compression helps to reduce the effects of local shadowing and illumination variations.
- The second stage computes first order image gradients. This step capture contour, silhouette and some texture information, while providing further resistance to illumination variations. The locally dominant colour channel is

used, which provides colour invariance to a large extent. Variant methods may also include second order image derivatives, which act as primitive bar detectors – a useful feature for capturing, e.g. bar like structures in bicycles and limbs in humans.

- The third stage aims to produce an encoding that is sensitive to local image content while remaining resistant to small changes in pose or appearance. The adopted method pools gradient orientation information locally in the same way as the SIFT (Lowe 2004) feature. The image window is divided into small spatial regions, called "cells". For each cell we accumulate a local 1-D histogram of gradient or edge orientations over all the pixels in the cell. This combined cell-level 1-D histogram forms the basic "orientation histogram" representation. Each orientation histogram divides the gradient angle range into a fixed number of predetermined bins. The gradient magnitudes of the pixels in the cell are used to vote into the orientation histogram. Figure 80 illustrates the notion of a cell and orientation histogram within it.
- The fourth stage computes normalization, which takes local groups of cells and contrast normalizes their overall responses before passing to next stage. Normalization introduces better invariance to illumination, shadowing, and edge contrast. It is performed by accumulating a measure of local histogram "energy" over local groups of cells that we call "blocks". The result is used to normalize each cell in the block.
- The final step collects the HOG descriptors from all blocks of a dense overlapping grid of blocks covering the detection window into a combined feature vector for use in the window classifier.

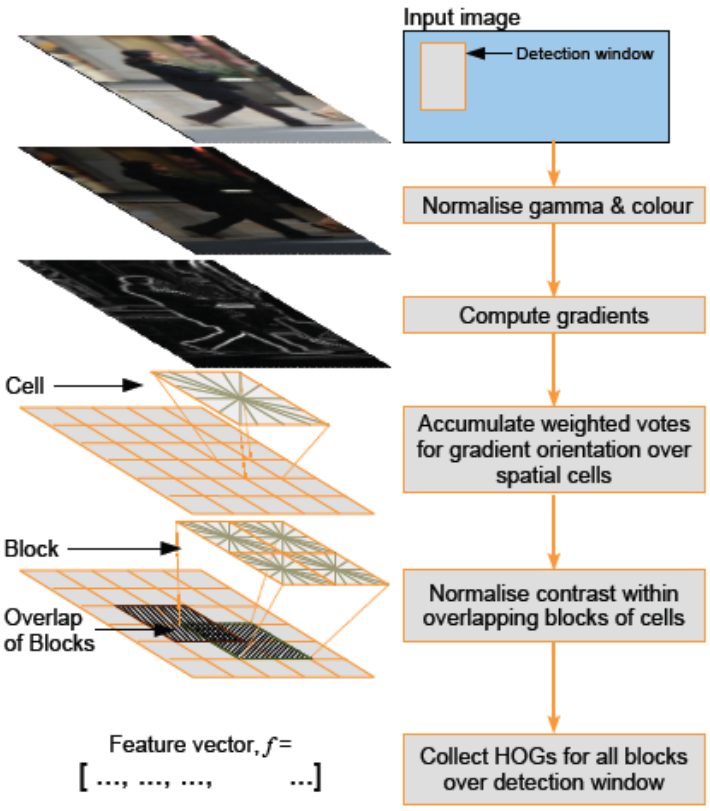

Figure 80 - An overview of static HOG feature extraction. The detector window is tiled with a grid of overlapping blocks. Each block contains a grid of spatial cells. For each cell, the weighted vote of image gradients in orientation histograms is performed. These are locally normalized and collected in one big feature vector.

The HOG representation has several advantages. The use of orientation histograms over image gradients allows HOGs to capture local contour information, i.e. the edge or gradient structure that is very characteristic of local shape. In conjunction with the spatial quantization into cells, it allows them to capture the most relevant information with controllable precision and invariance (e.g. by changing the number of bins in orientation histograms and the cell size).

Gamma normalization and local contrast normalization contribute another key component: illumination invariance. The use of overlapping of blocks provides alternative normalizations so that the classifier can choose the most relevant one.

In the present work, the results does not show the use of HOG descriptor in the architecture, but considering the future developments, in the effort of obtain better

results and performance, this kind of descriptor will be integrated and fused with the current configuration.

5.3.2 Grayscale Pixels Descriptors

A grayscale (or gray level) image is simply one in which the only colours are shades of gray. The reason for differentiating such images from any other sort of colour image is that less information needs to be provided for each pixel. In fact a `gray' colour is one in which the red, green and blue components all have equal intensity in RGB space, and so it is only necessary to specify a single intensity value for each pixel, as opposed to the three intensities needed to specify each pixel in a full colour image.

Often, the grayscale intensity is stored as an 8-bit integer giving 256 possible different shades of gray from black to white. If the levels are evenly spaced then the difference between successive graylevels is significantly better than the graylevel resolving power of the human eye.

Grayscale images are very common, in part because much of today's display and image capture hardware can only support 8-bit images. In addition, grayscale images are entirely sufficient for many tasks and so there is no need to use more complicated and harder-to-process colour images.

Starting to the definition of a grayscale image, given an image 24x24 pixels, the descriptor of a feature sample will be a vector of 576 values representing each a grayscale value (from 0 to 255).

Given a tomato sample of 24x24 pixel as that in Figure 81,

Figure 81 - Tomato sample

the descriptor of this tomato will be a vector o 576 values. Repeating the computation for the 500 samples and collecting all the data in one element, the result will be a matrix of 575x500 pixels.

Averaging each column of the matrix, the result will be a new vector of 575 values that can be shown in Figure 82

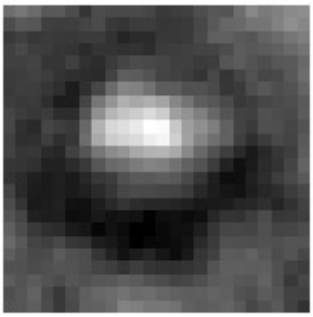

Figure 82 - Representation of average pixel descriptors

All results in §6 consider the use of this kind of descriptors. They have several advantages, first of all the simplicity and the low calculation load.

5.3.3 Colour Pixel Descriptors

Image category recognition is important to access visual information on the level of objects (buildings, cars, etc.) and scene types (outdoor, vegetation, etc.). In general (Koen *et al.* 2010), systems for category recognition on images and video, use machine learning based on image descriptions to distinguish object and scene categories. However, there can have large variations in viewing and lighting conditions for real-world scenes, complicating the description of images and consequently the image category recognition task.

A change in viewpoint will yield shape variations such as the orientation and scale of the object.

In addition, changes in the illumination of a scene can greatly affect the performance of object and scene type recognition if the descriptors used are not robust to these changes. To increase photometric invariance and discriminative power, colour descriptors have been proposed which are robust against certain photometric changes

(Koen *et al.* 2010). There are many different methods to obtain colour descriptors, however, the approach analysed in the work is really simple. We call it improperly as colour descriptor but it can't be considered as a real one. In other word it is an extension of the grayscale descriptor adding the colour component.

A feature/image can be obtained by the combination of the three primary channels: red, green and blue. Starting from the same concept explained in §5.3.2, given a colour image 24x24x3 pixels, the descriptor of a feature sample will be simply a vector of 576x3 (one for each RGB channel) values.

5.4 Annotation Methodology

All annotations shown in the present work have been made on the original images (figures below).

Figure 83 - Samples of original images

Each annotation consisted of a centre point and a bounding box surrounding the object. Typically the bounding boxes contain the object, in this case the entire or partial tomato (occlusion issue).

500 tomatoes samples have been extracted and the same number of negative (not tomato) samples.

Figure 84 - Positive samples

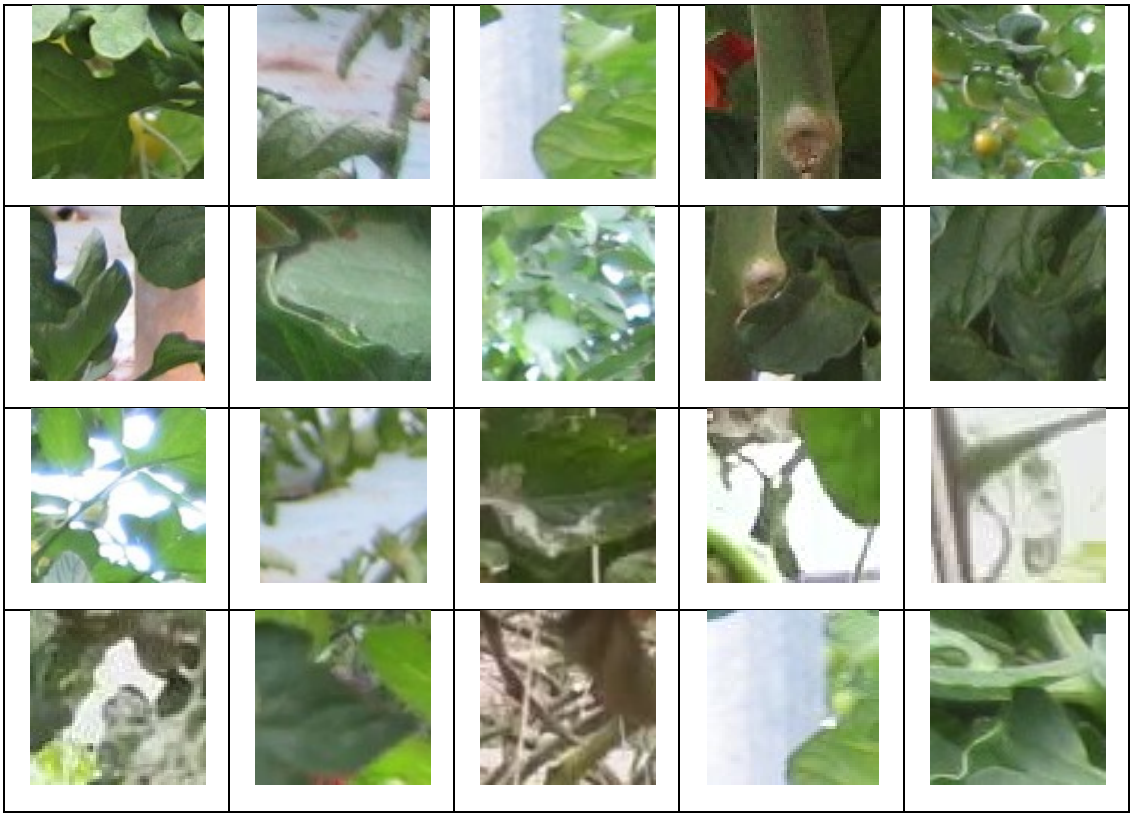

Figure 85 - Negative samples

The image regions belonging to the annotations were cropped and rescaled to 24x24 pixel image windows. On average the tomatoes height is 24 pixels in these normalized windows to allow for an approximately 6 pixel margin on each side. In practice, we leave a further 6 pixel margin around each side of the image window to ensure that flow and gradients can be computed without boundary effects. The margins were added by appropriately expanding the annotations on each side before cropping the image regions.

5.5 The Learning Process

We use linear Support Vector Machines as our benchmark classifiers. More in detail, the Matlab implementation of the SVM algorithm proposed by Chang *et al*. 2011 has been used.

First, we train a preliminary detector on the positive training windows and an initial set of negative windows. The (centered and normalized) positive windows are supplied by the user, and the initial set of negatives is created once and for all by randomly sampling negative images.

A preliminary classifier is thus trained using these inputs. Second, the preliminary detector is used to exhaustively scan the negative training images for hard examples (false positives). The classifier is then re-trained using this augmented training set (user supplied positives, initial negatives and hard examples) to produce the final detector. The number of hard examples varies from detector to detector, and in particular depends on the initial detector's performance. If too many large numbers of hard examples are generated, the set of hard examples is sub-sampled in order to optimize the RAM memory usage. This retraining process significantly and consistently improves the performance of all the tested detectors (approximately reducing false positive rates by an order of magnitude in most cases). Additional rounds of retraining make little or no difference to the overall results and so we do not use them.

The output of the learning process to be provided for the detection phase is a *model*: a predict tool allowing the classifier, during the detection phase, to discriminate between tomato and not-tomato selection in a test image.

5.6 The Detection Process

Once the learning (or retraining) phase has been performed, the test image can be scanned densely (moving each time of one pixel).

Figure 86 - Scanning Windows in a test image

The model before calculated can be applied to each window scanned, in order to provide a decision (classifier is again the owner): tomato or not tomato.

More in detail: the classifier runs to each window scanned and associates a confidence value (an index representing the tomato property). The most high confidence values should be associated to tomato, the lowest to the background.

Two issues arise from this process: multiple and hard detection.

Multiple issues occur every time an object is detected in more bounding box: a tomato should be associated to just one bounding box.

Hard detection occurs in the case of false detection: the classifier indicates as tomato a feature that is not.

In the following paragraphs a description of the approaches used to solve the two issues will be detailed.

5.7 Non Maxima Suppression: single or multiple scales

During the detection phase, the binary window classifier could be scanned across the image at multiple scales. Actually in the case of the robot moving along the path, the distance between the webcam and the raw will be approximately the same. The size of tomato (depending by the growth - about the same on every raw) in the images acquired from the camera on the same raw doesn't have big changes so, if you are interested to the detection of tomato on the same raw and not different, multi scale feature can be avoided.

In this case the software will leave to the user the possibility to set up the environment depending on the greenhouse condition. A calibration phase (offline) is required: the operator, looking on the image acquired from the camera, sets the size (in pixel) of the tomato he wants to detect, drawing a square around the fruit as shown in the image below:

Figure 87 - Tomato Calibration

The algorithm produces a multiple overlapping detections for each tomato so a non-maximum suppression technique is required to fuse the information of the same object.

Non-maximum suppression is often used along with detection algorithms. The image is scanned in windows and if the pixels are not part of the local maxima they are set to zero. This has the effect of suppressing all image information that is not part of local maxima.

If you are interested to a multi scale algorithm, this typically produces multiple overlapping detections for each object instance. Also in this case, these detections need to be fused together.

We propose somewhat based on representing detections in a position scale pyramid: each detection step provides a weighted point in this 3-D space and the weights were the detection's confidence score. A non-parametric density estimator is run to estimate the corresponding density function and the resulting modes (peaks) of the density function constitute the final detections, with positions, scales and detection scores given by value of the peaks. Figure 88 illustrates the steps

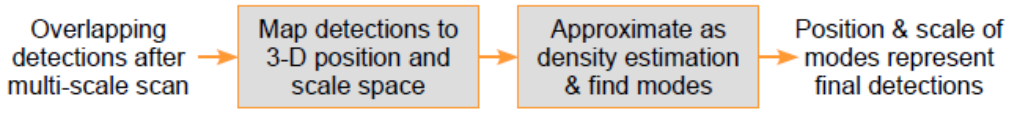

Figure 88 - An overview of the steps of the non-maximum suppression algorithm. The detection window is scanned across the image at all positions and scales, and non-maximum suppression is run on the output pyramid to detect object instances.

5.8 Results evaluation

We know that most features in one image are likely to match the other image, although some may not match because they are occluded or their appearance has changed too much.

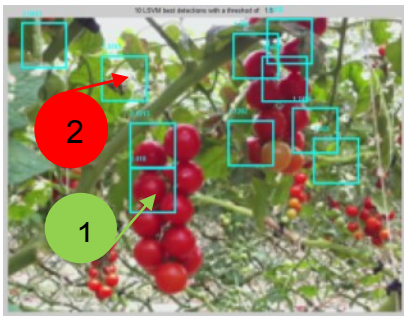

Figure 89 - (1) Correct and (2) uncorrected matching sample

On the other hand, if we are trying to recognize how many known objects (tomatoes) appear in a cluttered scene (Figure 89), most of the features may not match.

Furthermore, a large number of potentially matching objects must be searched, which requires more efficient strategies, as described below.

Given a Euclidean distance metric, the simplest matching strategy is to set a threshold (maximum distance) and to return all matches from other images within this threshold. Setting the threshold too high will result in too many false positives, i.e., incorrect matches being returned. Setting the threshold too low will result in too many false negatives, i.e., too many correct matches being missed.

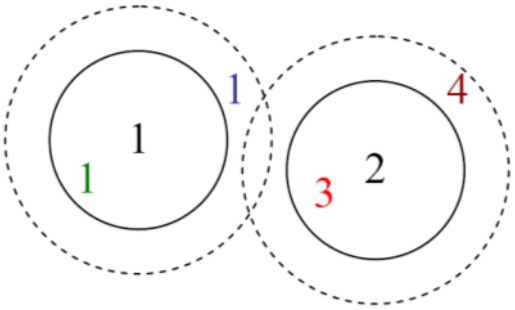

Figure 90 - An example of false positives and negatives. The black digits 1 and 2 are the features being matched against a database of features in other images. At the current threshold setting (black circles), the green 1 is a true positive (good match), the blue 1 is a false negative (failure to match), and the red 3 is a false positive (incorrect match). If we set the threshold higher (dashed circle), the blue 1 becomes a true positive, but the brown 4 becomes an additional false positive.

We can quantify the performance of a matching algorithm at a particular threshold by first counting the number of true and false matches and match failures, using the following definitions:

- TP: true positives, i.e., number of correct matches;
- FN: false negatives, matches that were not correctly detected;
- FP: false positives, proposed matches that are incorrect;
- TN: true negatives, non-matches that were correctly rejected.

Figure 91 shows a sample confusion matrix (contingency table) containing such numbers.

We can convert these numbers into unit rates by defining the following quantities

	True matches	True non-match.		
Pred. matches	TP = 18	FP = 4	P' = 22	PPV = 0.82
Pred. non-match.	FN = 2	TN = 76	N' = 78	
	P = 20	N = 80	Total = 100	
	TPR = 0.90	FPR = 0.05		ACC = 0.94

Figure 91 - Sample table showing the number of matches correctly and incorrectly estimated by a feature matching algorithm. The table shows the number true positives (TP), false negatives (FN), false positives (FP), true negatives (TN). The columns sum up to the actual number of positives (P) and negatives (N), while the rows sum up to the predicted number of positives (P') and negatives (N'). The formulas for the true positive rate (TPR), the false positive rate (FPR), the positive predictive value (PPV), and the accuracy (ACC) are given in the text.

- true positive rate (TPR),

$$TPR = \frac{TP}{TP+FN} = \frac{TP}{P};$$

- false positive rate (FPR),

$$FPR = \frac{FP}{FP+TN} = \frac{FP}{N};$$

- positive predictive value (PPV),

$$PPV = \frac{TP}{TP+TN} = \frac{TP}{P'};$$

- accuracy (ACC),

$$ACC = \frac{TP+TN}{P+N}.$$

In other words, the indexes have been used are:

PPV = *(True Positive) / (All Predicted Tomatoes)*

TPR = *(True Positive) / (Total Number of Tomato to be predicted);*

In the information retrieval the terms ***precision*** is used instead of PPV (how many returned feature are relevant) and ***recall*** is used instead of TPR (what fraction of relevant feature was found).

Ideally the precision and recall value should be as much possible close to 1 for any different value of threshold. In ideal condition the precision-recall value has the trend shown in Figure 92.

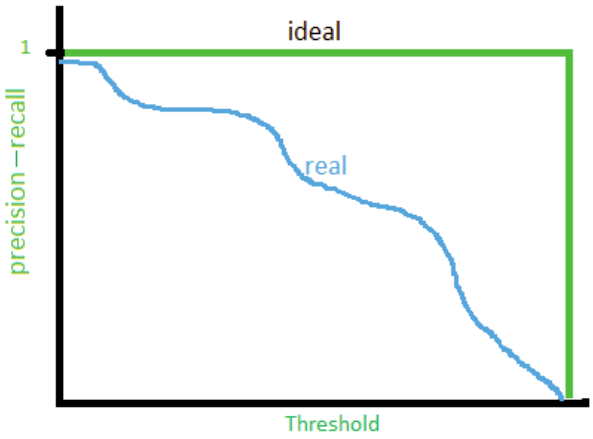

Figure 92: Ideal and real trend of the precision-recall curve

The algorithm performs better as much bigger is the area under the precision-recall curve.

6 The results

In this chapter some experimental results will be shown. The first part of the tests have been focused in order to verify the functioning and performance of the control algorithms for the autonomous navigation- Later, each developed sub-system has been tested and as consequence improved to obtain better results. A deep analysis has been faced in the last part of the results chapter relating one of these subsystems: the vision algorithm for the tomato detection.

6.1 Robot navigation test

To evaluate the performance of the robotics platform, different tests have been performed. The U-Go Robot has been built in order to be used as a multipurpose outdoor vehicle in different application. Its technical specification meets requirements both for teleoperated as for autonomous motion.

In this paragraph a brief review of the all the tests performed will be shown.
Teleoperated mode has been used on volcanic environment (Muscato *et al.* 2008) in order to test system reliability and payload capabilities. About 180 kg of materials and instrumentation were carried on the top of the Mt. Etna volcano (about 3300 m asl) on behalf of INGV (Istituto Nazionale di Geofisica e Vulcanologia) in order to allow their technicians to build some gases monitoring stations. In Figure 93 the U-Go Robot during this test is shown.

Figure 93 - Teleoperated robot test on the Etna volcano

Preliminary tests in autonomous modality were performed first in the simulated environment and only after on the real context to detect and reduce possible bugs. The algorithm in the real context was extended later with the use of the artificial vision.

These algorithms have been developed to allow autonomous navigation in case of GPS signal failure.

Even if the data coming from the GPS were not always continuous, the robot reached the target following again the path. If the GPS data are missing for a longer time, the navigation algorithm will take advantage of the information coming from the developed artificial vision system.

6.1.1 PFM in simulated mode

In this paragraph, there will be shown the results obtained in simulation (Figure 94) by using the obstacle avoidance technique: the control algorithm of the Potential Field.

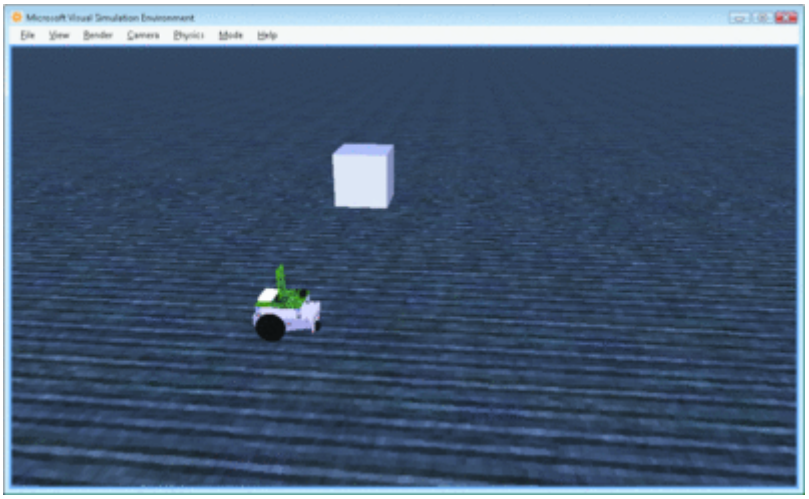

Figure 94 - U-Go Robot in Simulated Environment

This algorithm exploits the sensors information coming from the laser scanner device that allowing investigating the environment and avoiding the obstacles along the path. The choice of the parameters is crucial for the algorithm performance.

As already detailed, the PFM considers the robot as a particle moving in the configuration space, under the influence of forces field. The target location applies

on the particle an attraction force F_{att}, while the obstacles a repulsive F_{rep}. Once calculated, it's possible to define a resultant force F_{tot}: its direction shall be followed by the vehicle in order to reach the target avoiding the collisions.

In absence of obstacles, the robot follows perfectly the targets, being the resultant coincident with the attraction ones. To verify it, a test on a rectangle has been performed. Four targets have been planned with coordinates (x, y): (-12.6, 8.22), (-21.0, 12.65), (-18.0, 18.7), (-9.23, 14.3). In the real context, these points correspond to a square behind the DIEEI laboratory. They are local cartesian coordinates *enh* respect to an eulerian tern *neh* (Stoppini *et al*., 2011) with origin in a specific point P0 of the square..

Considering:

- **v and K_w** :the velocity reference for the differential drive.
- **K_w and K_a**: the attractive and repulsive obstacle constant
- **sleep**: the delay for the localization measures (in the real context they correspond to the GPS measurement)
- **threshold**: the distance the robot have to keep from the obstacles

Setting the following variables results are shown:

- **v = 0.5 m/s ; K_w = 0.3 ; K_a = 10 ; K_{rep} = 0 ; sleep = 0 ms goal threshold = 1 m**

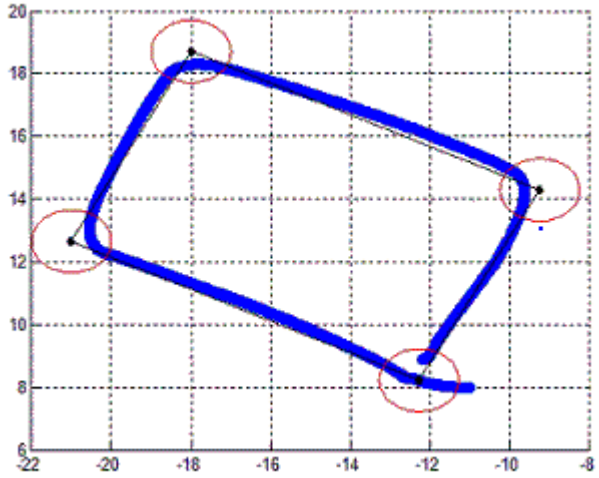

Figure 95 - Square path test; dark blue line represents robot trajectory, black dot represent target waypoint while red circles represent the target threshold area

The robot follows the target in a right manner, driving a rectangle where the vertexes are the planned goals. The red circles in the graph, in the following tests, represent a threshold value: over it, the algorithm considers the target as reached and the movement towards the next goal will be planned.

The second test, performed in ideal condition (in the ideal case GPS sends continuous measurements; simulating that case no delay has been introduced: sleep=0) consists on placing the targets in a "*serpentine*" configuration. Because of the delays absence, the test has been performed with a high translation speed (half of the maximum speed allowed).

- **v = 0.5 m/s ; K_w = 0.3 ; K_a = 10 ; K_{rep} = 0 ; sleep = 0 ms; goal threshold = 1 m**

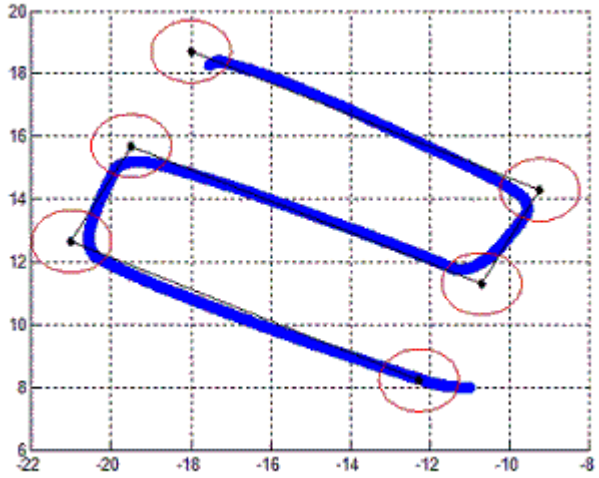

Figure 96 – Robot drives a serpentine path in order to simulate the rows configuration inside a greenhouse

Also in this case, the repulsive constant has been maintained null, because of the total absence of obstacles in the scene. The trajectory of the robot, as pointed out in the graph, is very similar to the ideal ones (traced in black). Further tests will be planned in order to evaluate algorithms performances in terms of maximum error between target trajectory and the real one.

The third test has been performed adding obstacles along the path, located in the center of each side of the square. The test has been repeated different times, with increasing speed, always in ideal condition, namely without introducing delays on the control ring.

Obviously, the obstacles presence determines a repulsive force that causes freaky changes of the robot direction in proximity of them. Four cubes of 0.5 m^3 have been added with the objective of simulating the obstacles to be avoided during the simulation. The block are located to the coordinates: (-16.6 , 10.3), (-19.5 , 15.7) , (-13.6 , 16.5) , (-10.7 , 11.2), in the center of each side of the rectangle

- **v = 0.3 m/s ; K_w = 0.2 ; K_a = 10 ; K_{rep} = 1 ; sleep = 0 ms; goal threshold = 3 m**

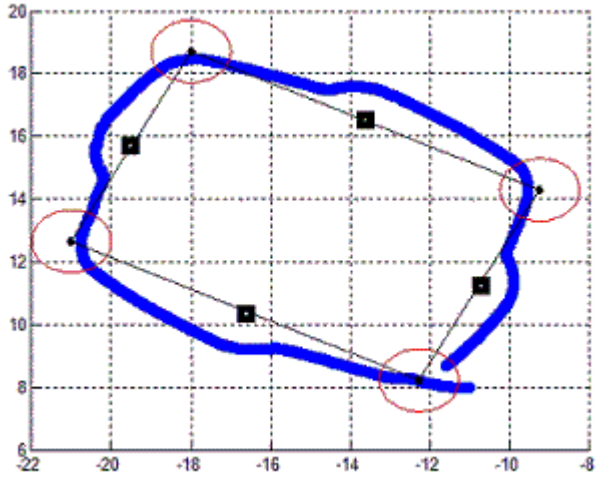

Figure 97 –Robot avoids obstacles (black square) during the performing of the square path

The obstacles presence produces repulsive forces that cause a safe deviation of the robot. When the obstacles are located outside the area detected from the laser scanner, the repulsive component is null and the movement is dominated by the attraction forces toward the goal (same behave of the previous test).

Reducing the threshold from the obstacle, the reaction of the robot respect to the presence of the obstacle is delayed. If the speeds are high or the pose measures are provided with a certain delay, it can cause robot collisions. A threshold increase, instead, gives to the robot a higher sensibility towards the obstacles with a consequent increasing of the direction changes.

- **v = 0.5 m/s ; K_w = 0.3 ; K_a = 10 ; K_{rep} = 1 ; sleep = 0 ms; goal threshold = 3 m**

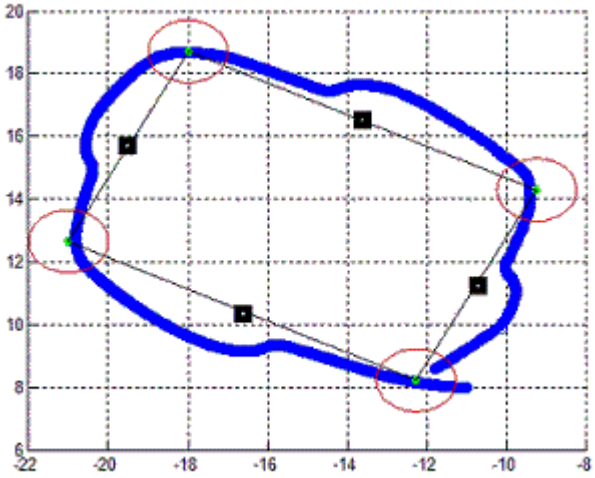

Figure 98 – Robot path with an increase of a speed to 0,5 m/s that needs an adjustment of the Kw variable

The vehicle behavior, even if with higher speed, is similar to the previous case. To quickly compensate the orientation error respect to the target, the K_w parameter has been increased, returning a robot trajectory very similar to the ideal case (black rectangle).

- **v = 0.8 m/s ; K_w = 0.4 ; K_a = 10 ; K_{rep} = 1 ; sleep = 0 ms; goal threshold = 3 m**

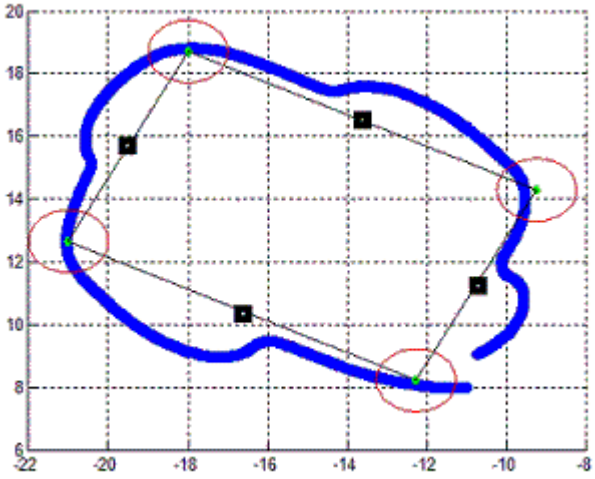

Figure 99 – Further test increasing speed parameters to 0,8 m/s

The reaction of the robot respect to the obstacles presence is similar to the two previous cases. Considering the maximum translation speed allowed for the robot entity in simulation, about 1 m/s and the speed in the real test is close to this, we can

deduce that Potential Field Algorithm implemented has an optimal behave in absence of delay.

The last fourth test, initially without adding any delay, forecasts two couples of obstacles, located on the first and third side of the rectangle. The linear speed is low, because of the complexity of the scene proposed. It is necessary to reduce the obstacle threshold parameter because the robot could feel too strong the proximity between the two blocks, traducing it in too frequent direction changes of the vehicle.

- **v = 0.3 m/s ; K_w = 0.2 ; K_a = 10 ; K_{rep} = 1 ; sleep = 0 ms; goal threshold = 2 m**

Figure 100 – Configuration with double obstacle and narrow space between them

By using moderate speed values, the robot avoids correctly all the obstacles, as shown in the graph above.

Finally all the tests are repeated, introducing a time delay of 1 second that simulates the low update frequency of the GPS module used in the real context.

v = 0.5 m/s ; K_w = 0.3 ; K_a = 10 ; K_{rep} = 0 ; sleep = 1 s; goal threshold = 3 m

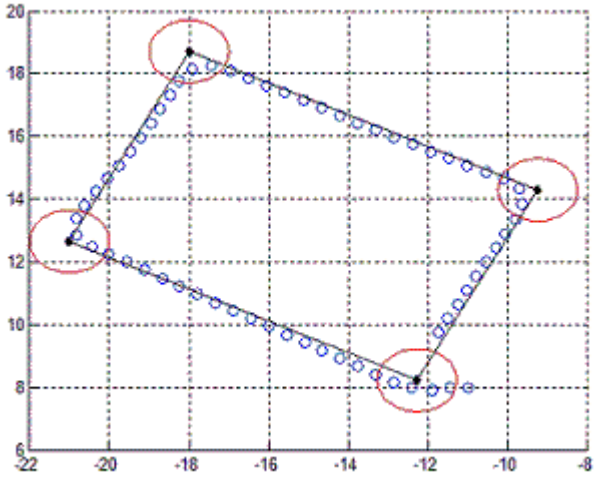

Figure 101 –Square path performed by robot adding a delay component to simulate real GPS performance

Despite the delay introduced, the robot follows correctly the targets. The trajectory is near to the ideal ones. The reduced number of blue circles underlines as the sample frequency of the pose is reduced considerably. The GPS provides information related to the robot pose once in a second (frequency of 1 Hz).

- **v = 0.5 m/s ; K_w = 0.3 ; K_a = 10 ; K_{rep} = 0 ; sleep = 1 s; goal threshold = 3 m**

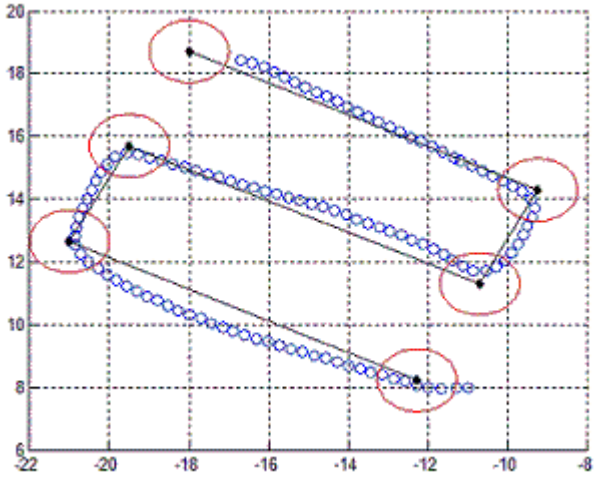

Figure 102 – Robot performs a serpentine path with a sleep parameter set to 1s

Also in this case, the robot reaches the planned way-points. The trajectory followed by the robot in some points differs from the ideal ones. However, it would be enough to increase the K_w parameter to obtain similar trajectories.

- **v = 0.3 m/s ; K_w = 0.2 ; K_a = 10 ; K_{rep} = 1 ; sleep = 1 s; goal threshold = 3 m**

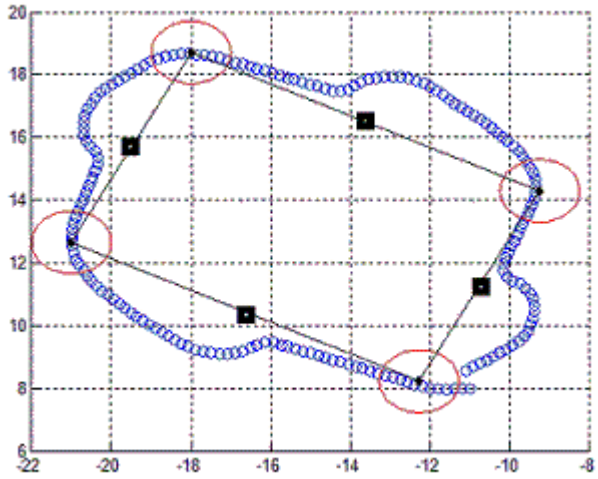

Figure 103 – Test on obstacle avoidance introducing a delay on the robot pose estimation of 1 s

Thanks to the low speed, the robot can avoid the obstacles and reach the targets (it drives inside the threshold area highlighted by red circles).

An alternative solution is to increase the parameter related to the obstacle threshold avoiding suddenly robot direction changes. The decrease of the parameter K_w, causes larger trajectories because the robot has a lower compensation of the orientation errors respect to the target.

As it will be deduced from next tests, it is not always possible to increase the parameter related to the obstacle threshold in case of multiple obstacles. The robot would suffer excessively of the proximity between the obstacles, causing not predictable trajectories.

- **v = 0.2 m/s ; K_w = 0.1 ; K_a = 10 ; K_{rep} = 1 ; sleep = 1 s; goal threshold = 4 m**

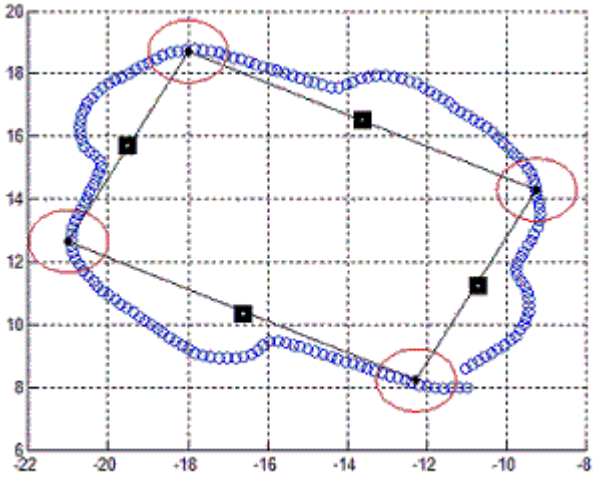

Figure 104 – Robot performing square path: increasing speed value, the obstacle parameter should be increased too in order to reach all way-points

Again the result is satisfactory. The parameters are not independent one from each other and their choice can require amount of time in the real application. It will be the user, through his experience, to determine the optimal parameters for the application observing the environment proposed.

Moving the frequency rate update to 1 Hz introduces strong constraints on the speed performed by the robot. Increasing the speed, the robot can drive longer in a shorter time.

Having a reduction of the information number respect to the ideal case causes a minor control and a slower planning of the speed references for the robot.

- **v = 0.5 m/s ; K_w = 0.3 ; K_a = 10 ; K_{rep} = 1 ; sleep = 1 s; goal threshold = 3 m**

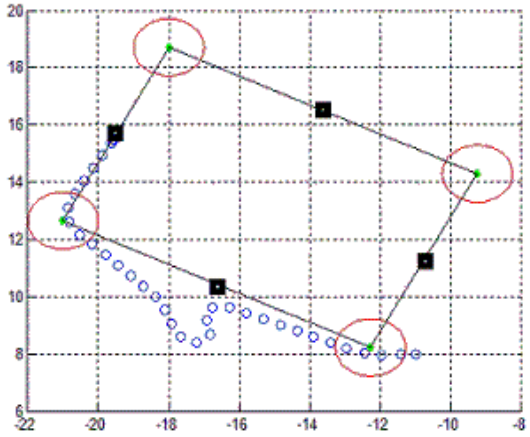

Figure 105 – Robot colliding an obstacle when a sleep parameters of 1s and a speed of 0,5m/s are set

As shown in the figure above, the algorithm was not be able to avoid collision. It has been set a different parameters tuning studied for this kind of environment. In any case, the best choice is to work with low speed.

- **v = 0.3 m/s ; K_w = 0.2 ; K_a = 10 ; K_{rep} = 1 ; sleep = 1 s; goal threshold = 2 m**

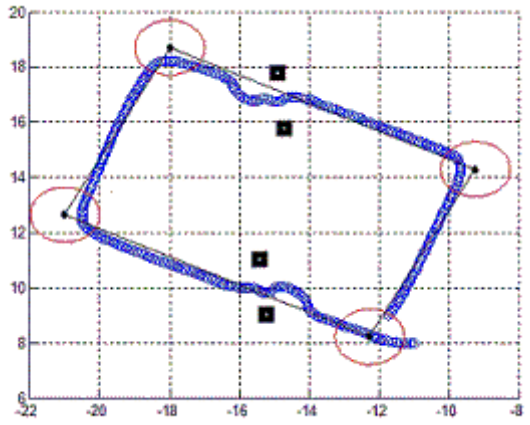

Figure 106 – Robot path with a reduction of the speed (sleep =1) in a configuration with double obstacle for each side

If the speed setting is low, the control algorithm is able to perform correctly and in safety the task avoiding all the obstacles. The robot drive, when obstacles occur, can be more linear decreasing the "obstacle" parameter: the robot will be less sensible to obstacle influence.

6.1.2 PFM in the real context

The tests in the real context have been performed in the square place behind the DIEEI laboratory. The tests are the same shown in the simulated context, in order to demonstrate a similarity between them. The figures below show all the trajectories of the robot (the blue small circles are the coordinates of the robot while the black lines are reference trajectories). That has been possible thanks to the AGEIA physical engine: without it, there would not be any correspondence between simulated and real context.

- **v = 0.5 m/s ; K_w = 0.3 ; K_a = 10 ; K_{rep} = 0 ; freq. GPS=1Hz; goal threshold = 3**

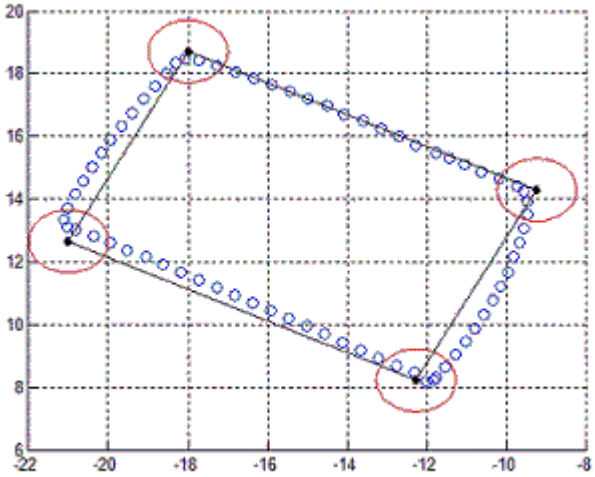

Figure 107 – Robot in real context driving a square path

The robot follows correctly the targets: in the second and fourth side the trajectories are larger than the simulated model probably because the low number of satellites detected by GPS.

- **v = 0.5 m/s ; K_w = 0.3 ; K_a = 10 ; K_{rep} = 0 ; freq. GPS=1Hz ; goal threshold = 3**

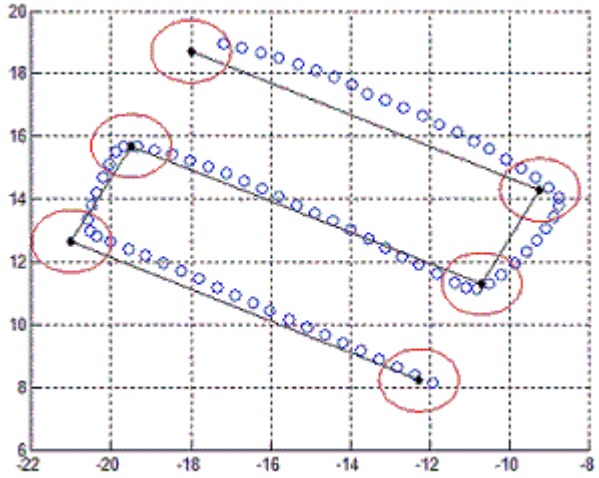

Figure 108 – Robot in real context driving a serpentine path

The serpentine trajectory has been performed satisfactory by the robot: the error computed is of centimeter order. This precision has been reached thanks to the use of the differential GPS device.

- **v = 0.3 m/s ; K_w = 0.2 ; K_a = 10 ; K_{rep} = 3 ; freq. GPS=1Hz ; goal threshold = 3**

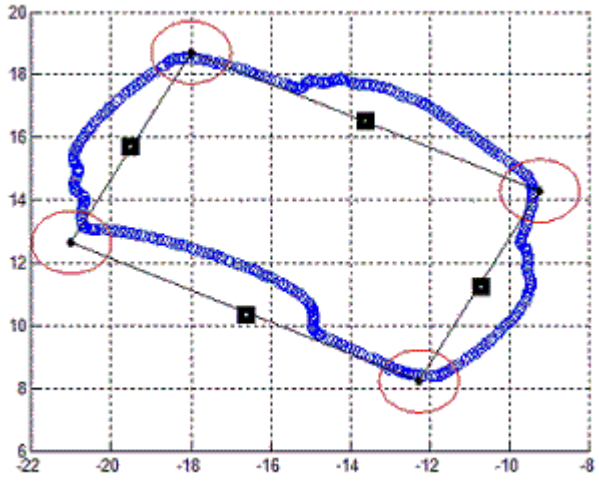

Figure 109 – Obstacle avoidance algorithm test on real context

In this test the repulsive constant has been increased to have a better robot response in proximity of the obstacles. The robot avoids all obstacles and reaches the four target points. Decreasing the parameter K_w allows obtaining less large trajectories but it can cause faster direction changes (not safety condition).

- **v = 0.3 m/s ; K_w = 0.2 ; K_a = 10 ; K_{rep} = 3 ; freq. GPS=1Hz ; goal threshold = 2**

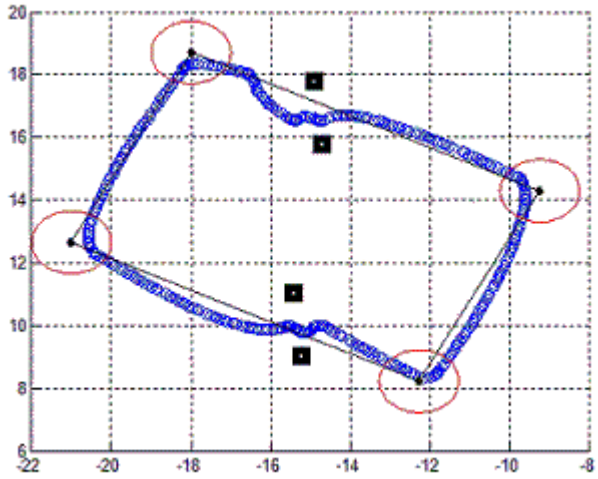

Figure 110 –Obstacle avoidance algorithm test in presence of double boxes

The overall result is very similar to the simulated context and the robot trajectory very similar to the ideal ones reaching correctly all the way-points.

6.1.3 M-Elrob

Different sensors have been tested to allow for autonomous operations not only in greenhouses but also in open field.

A deeper evaluation of the robot performance has been performed during the M-Elrob competition (military purpose), performed in Hammelburg (Germany) on May 2010; in Figure 111 the robot performing a task during the opening ceremony.

Figure 111 - U-Go Robot in autonomous modality during M-ELROB 2010 (b)

In Figure 112the robot path extracted from Google Earth (yellow line). The very long straight line is the path the robot traveled between the camp area and the opening ceremony area.

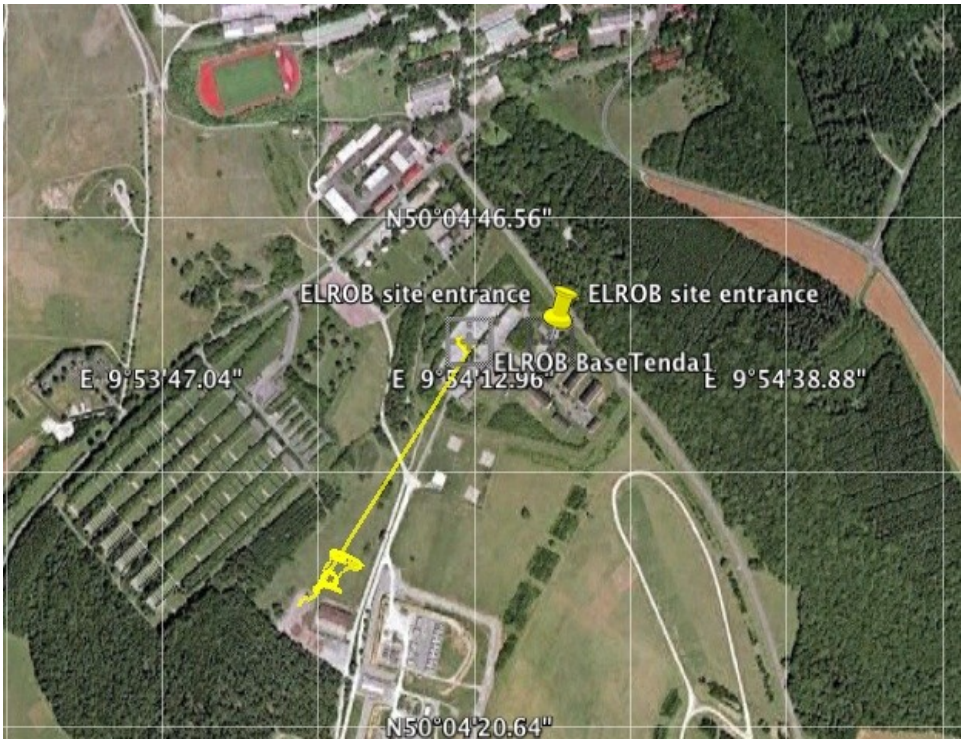

Figure 112 – The yellow line represents the U-Go Robot moving inside the military base in Hammelburg (Germany)

An example of the obstacle avoided has shown in Figure 113 and Figure 114:

Figure 113 – Examples of obstacles present in the environment

Figure 114 - The u-go robot during an autonomous task in a rocky area

while in Figure 115 again the robot during the official ceremony.

Figure 115 – U-Go robot during the official ceremony

Using the same configuration of the tests performed in the University context (see §6.1.2) the robot's task, during the trial, was to reach some waypoints (as shown in Figure 116 from Google Earth) avoiding all the (rocky) obstacles. Here, in red, it has been highlighted the ideal path, matching simply two consecutive waypoints; in yellow the path obtained from the GPS data during the trial execution. U-Go Robot reached all waypoints with good results completing the set path.

Figure 116 – Robot path avoiding obstacles

The totally new scenario and the achievement of the tasks can be considered a great result for the application requested and for the finality of the project.

6.1.4 Tests in Vineyards

Other tests have been performed in the vineyards located near Viagrande (CT) (Longo et *al.* 2010b, Schillaci *et al.* 2010).
The vineyard environment is very similar to the greenhouse's one, so it simplifies its reuse.

In Figure 117 U-Go Robot drives along the vineyards rows during the tests.

One of the possible applications of this electrical tracked robot is as a semi-automatic barrow to carry bins with grapes out of the rows. To perform these tasks, the operator can just drive the robot between two different rows, while the robot can autonomously move along the row using GPS and Laser range finder. In this way operators can concentrate their self in the harvesting task, while the robot perform the transport task; it also have special algorithms in order to not be dangerous for operators and vineyards. Moreover, because the robot is electrically powered, there is no danger for operators of toxic exhausts. When the robot uses a D-GPS, it can be fully autonomous and can perform fully autonomous operations (Schillaci *et al.* 2010). Also in vineyards spraying application can be possible increasing, also there, workers safety.

Figure 117 - U-Go Robot during tests in the vineyards (Viagrande, Sicily)

6.2 Low cost sensors: results

6.2.1 Self-Centering System

Using the considerations previously faced, a set of eight sensors and a small electronic board that collect data from each sensor, were mounted around a wooden-made robot mock-up (used only for testing purpose). In Figure 118 the experimental test-bed is shown. The front and rear sensors, during these experiments, are not used but they can be useful for safety reasons to avoid collision of the robot with obstacles and operators.

Figure 118 - The experimental setup with eight SFR08 sensor modules

The other six lateral sensors are used in order to compute the position of the robot with respect to the plants rows. Each sensor measures its distance to the plants and all these measures, using some filtering algorithm and some trigonometric consideration, give back the robot position (offset with respect plant rows) and orientation. Actually only two lateral sensors could be necessary for this algorithm;

the third sensor (the central one) is used for validating measurement of the other two and to compensate for wrong measurement that can happen. Information about robot offset and orientation are then used to correct the robot trajectory, allowing the system to move between rows. In Figure 119 and Figure 120 some data acquired during the real experiment are shown. During these tests, the eight sensors were put between the two rows with different orientation and offset. At the same time, the real position was measured by using the ultrasonic sensors and by using a rule and a goniometer. Using this methodology, it was possible to evaluate system performances and accuracy.

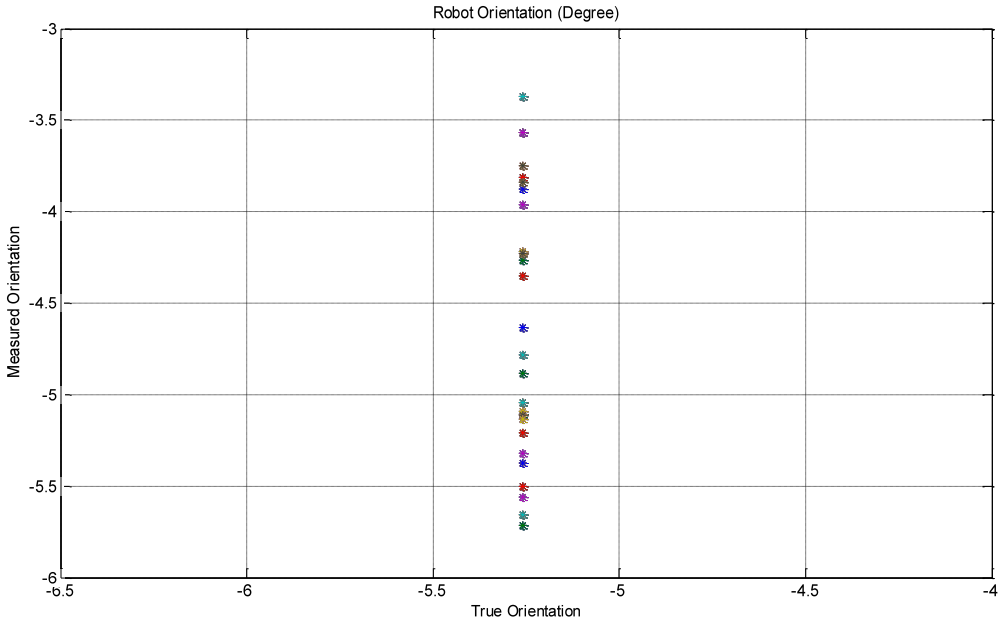

Figure 119 - Orientation test: on the X axis there is the true orientation of the robot while along the Y axis there are orientation measures by using the SFR08 sensors

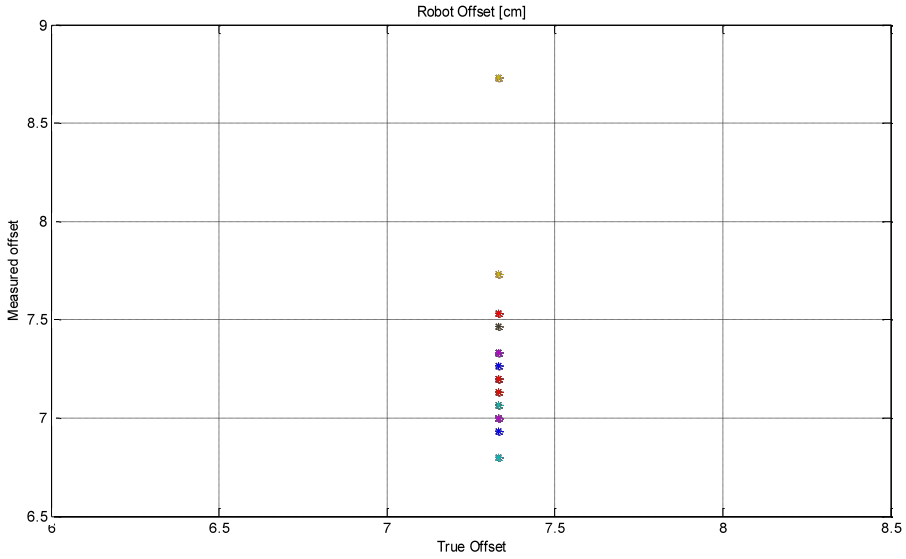

Figure 120 - Offset test: on the X axis there is the true offset of the robot while along the Y axis there are offset measures by using the SFR08 sensors

6.2.2 LPI

As detailed in §4.4.6.2 in this section the results of the experimental test are listed showing as the system can exploits the same centimetre-level precision at a fraction of the cost of a DGPS commercial system.

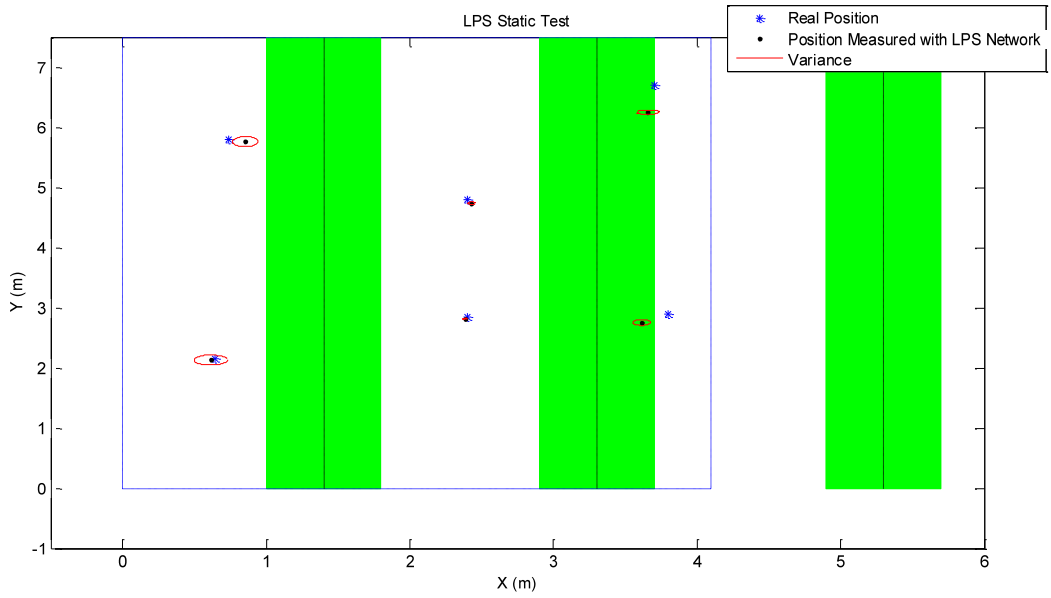

Figure 121 - LPS system: static test.

In Figure 121:

- *: is the real position of the robot
- ˙: is the position measured with the LPS algorithm and the ellipse (in red line) around it represents the variance of the cloud points acquired. More in details, the ellipse axis are the measures of the variance of all the acquired point; among these points, only the average value has been represented with a back black dot.

Different tests have been performed in a greenhouse in order to validate the system capabilities. Eight fixed receivers have been mounted inside the greenhouse covering an area of about 14 m^2. Static tests have been performed in order to evaluate the system accuracy. The mobile transmitter has been placed on different position inside the area covered by the fixed sensors.

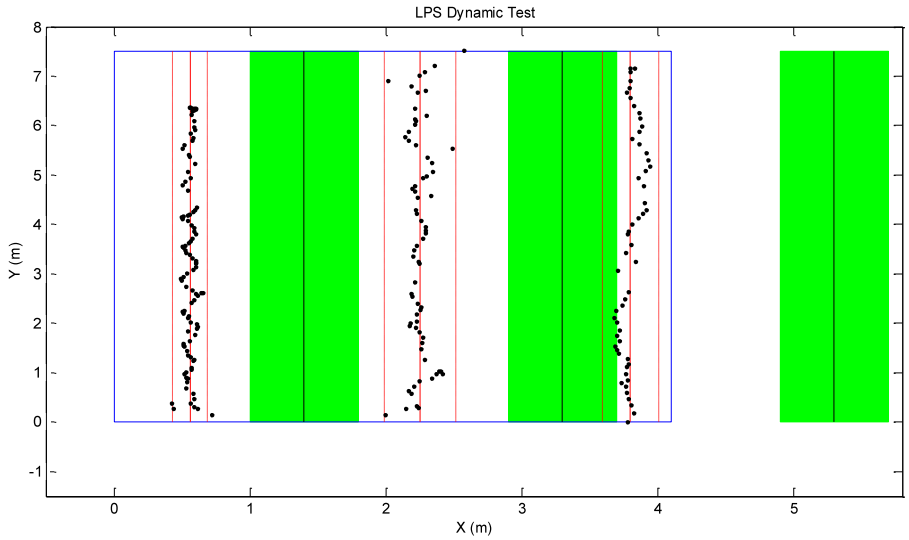

Figure 122 - LPS system: dynamic test.

The real position has been measured with a rule and compared with the position estimated by the LPS system. In Figure 121 some results are shown. In Figure 122 the results of a dynamic test are shown. In this case, the mobile transmitter has been moved along rows and subsequent positions have been recorded. Dark dots represent the system measurement, while the solid line represents the trajectory mean value. Dash-dot lines represent the data variance. In both figures, green boxes represent the plants.

6.2.3 Artificial Vision

Different tests have been done in a real greenhouse environment in order to evaluate overall performance and capabilities of the low cost guidance visual system.

In Figure 123 an experimental results is shown. The cart with laptop and webcam mounted on-board is moving in a greenhouse corridor at a speed of about 2 km/h that is comparable with normal speed used by operating machine during spraying operations in greenhouses. At the same time the on-board webcam is acquiring the optical flow and the software is elaborating in real time the data.

The system has been calibrated before starting the experiment using the chessboard technique.

On the image (extracted from the video stream) different information are then superimposed. The red central cross is the vanishing point while the red line on the top is the movement direction. In the black box on the top left, the velocity vector is represented. In the black box on bottom left the different parameters are shown in a numerical format (velocity vector and alpha angle).

Figure 123 - Some algorithm results: the robot is moving in the corridor.

6.3 Tomato detection: results

As described in §5, the architecture of the detection system can be summarized in the following two steps: the learning and the detection phase.

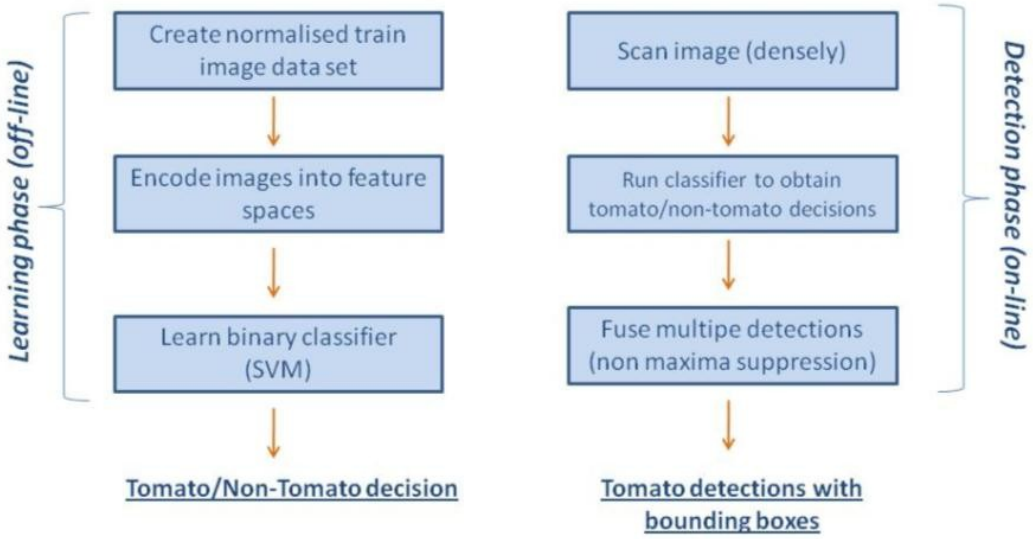

Figure 124 - Tomato Detections Overview

Figure 124 has been reported again here in order to make more readable the present paragraph and distinguish better the two phases.

The learning phase (in *off-line* modality) is based entirely on the dataset creation as well as the classifier choice. As previously mentioned, it has been used linear support vector machine as binary classifier.

Increasing the number of positive and negative samples, guarantees better performance and results in the detection phase. In this thesis 500 samples of positive and negative tomatoes have been selected, but dataset can be further improved and expanded.

The classifier output for the detection phase is a *model* or in other words a predicting tool allowing once again to the classifier, during the detection phase, to discriminate between tomato and not-tomato selection in a test image.

The test images used for the results are shown in the figures below:

Figure 125 - Test_Image_1

Figure 126 - Test_Image_2

Figure 127 - Test_Image_3

Figure 128 - Test_Image_4

Figure 129 - Test_Image_5

The images have been extracted from a video and different conditions (illumination, distance from the raw and the tomato detection) have been tested in order to demonstrate the robustness of the proposed approach.

The video were acquired in tomato greenhouses located in Vittoria (Ragusa - Sicily) during daylight in the month of June 2011.

Different typology of tomato has been analysed; the method can be exploited for any kind (it will be needed just to change the dataset) but, in this thesis, the results are focused on cherry variety.

In the Detection phase, the test image has been scanned densely, but in order to reduce the time and the calculation load, a different scanning can be adopted (e.g.

moving the scanning windows every time of 3 or more pixels and not 1). In this last case (a "not densely" scanning) the results are perfectly comparable with that one's shown in the present work.

In the Detection phase, the classifier returns a vector where for every window scanned a confidence value is associated to that window (represents how much the selected window is or not a tomato).

The size of the test image is 320*240 pixels and the scanning window is 24*24 pixels. The vector in output to the classifier will have a size of:

confidence value number of scanned windows* (22912 in our case).

A threshold value can be used in order to select which of the confidence value can be associated to a tomato or not.

6.3.1 Multiple Detection Issue

Performing the detection, the highest values of the confidence parameter are related to tomatoes, but the most of the times a tomato can be detected different times.

In order to solve this problem, a non maxima suppression method has been used (§5.7)

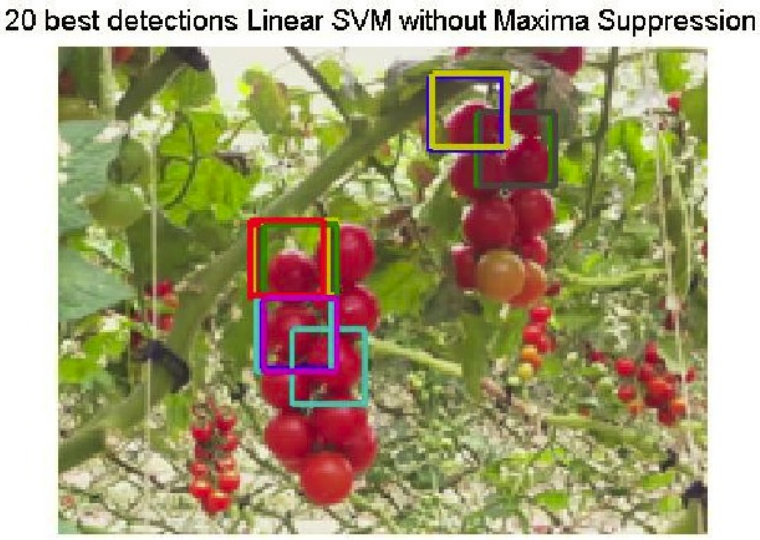

Figure 130 - 20 best detection without Non Maxima Suppression

As previously mentioned, this has the effect of suppressing all image information that is not part of local maxima, leaving only one bounding box around the best one.

6.3.2 Hard Detections Issue

Another important aspect we have faced is related to the hard detection removal (false positive or, in other words, not-tomato that is considered tomato).

To better understand, let's a look of some results obtained (non maxima suppression has already performed) varying the threshold parameter. In a first analysis, it has been considered threshold values in the range from 0 to 2.

Let's consider the image related Test_Image_1 for a threshold value of 0.1 (Figure 131):

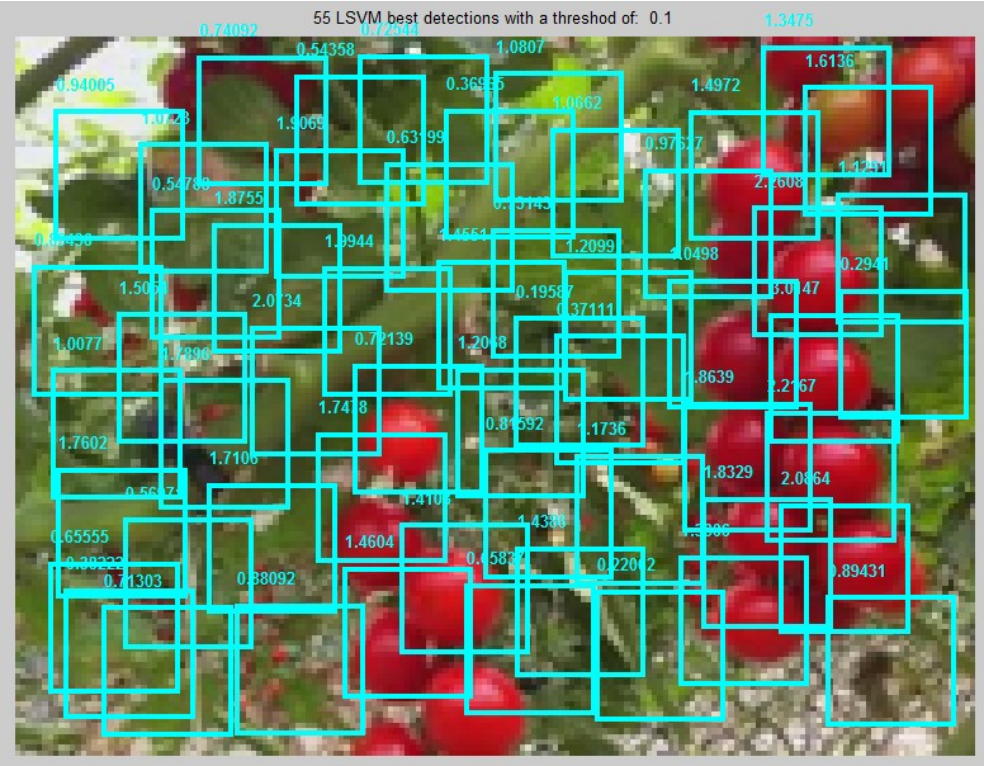

Figure 131 - 55 detections over a threshold of 0.1 (Test_Image_1)

As can be seen, 55 detections have been found over a threshold of 0.1. In the figure it is possible to see the correspondence between the bounding box (square) and the confidence value coming from classifier in the detection phase. As it is possible to see some of this detections are related to tomatoes, some others not.

If the threshold value increases, the number of the detections will decrease, as shown in Figure 132:

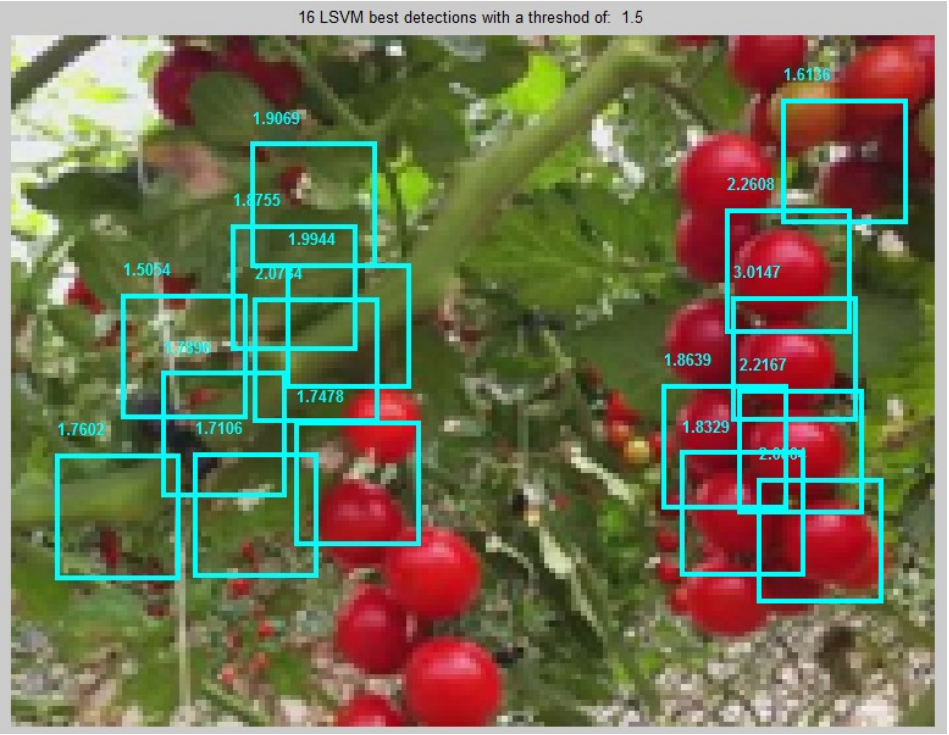

Figure 132 - 16 detections over a threshold of 1.5

Now, 16 detections will be found with a threshold of 1.5. As it can be seen, some of these detections are related to false detections (or "hard detections", not tomato that the method recognizes like tomato).

Increasing again the threshold value (2.0), the precision (number of true detections) increases but some tomatoes are missed.

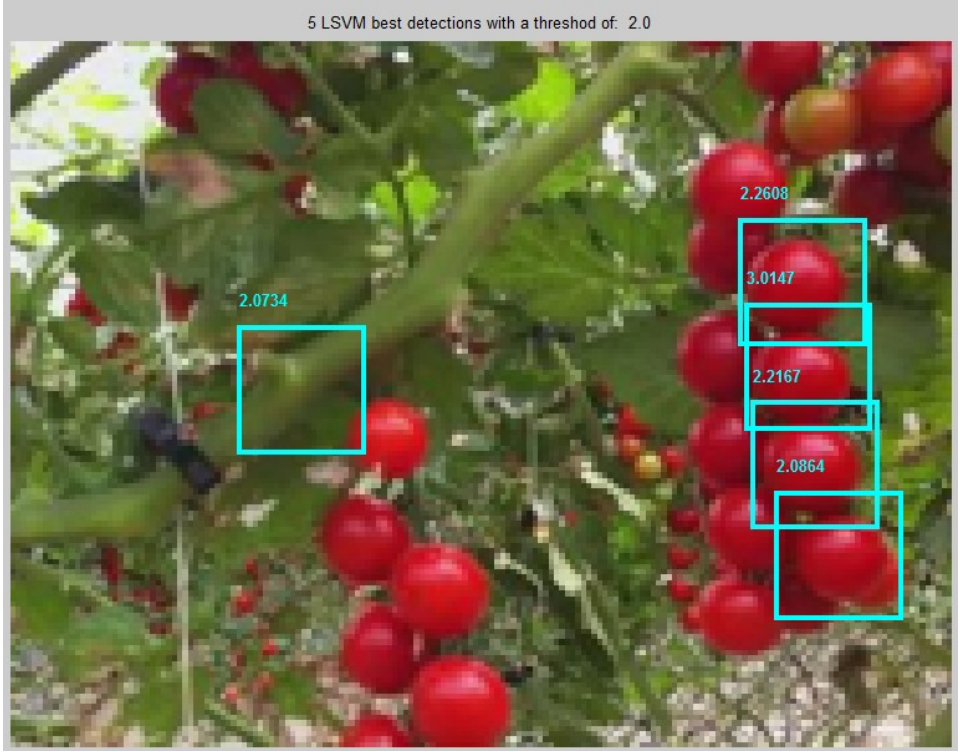

Figure 133 - 5 detections over a threshold of 2.0

How is it possible to avoid, remove or decrease the number of *hard detections* (above all for high values of threshold)?

A retraining method can be adopted in order to improve the results evaluation. It consists on detect the false detection and, coming back to the learning phase, give them again to the classifier.

Once this action will be performed, the hard detections will be removed (see Figure 138 and Figure 139 where hard detection over the threshold of 1.5 has been removed). Obviously, the case of hard detections will arise again for other test images, but it is a demonstration that increasing the dataset is a key point in order to achieve better results.

The following images represent the detections found in pre and post retraining for a threshold value of 1.5:

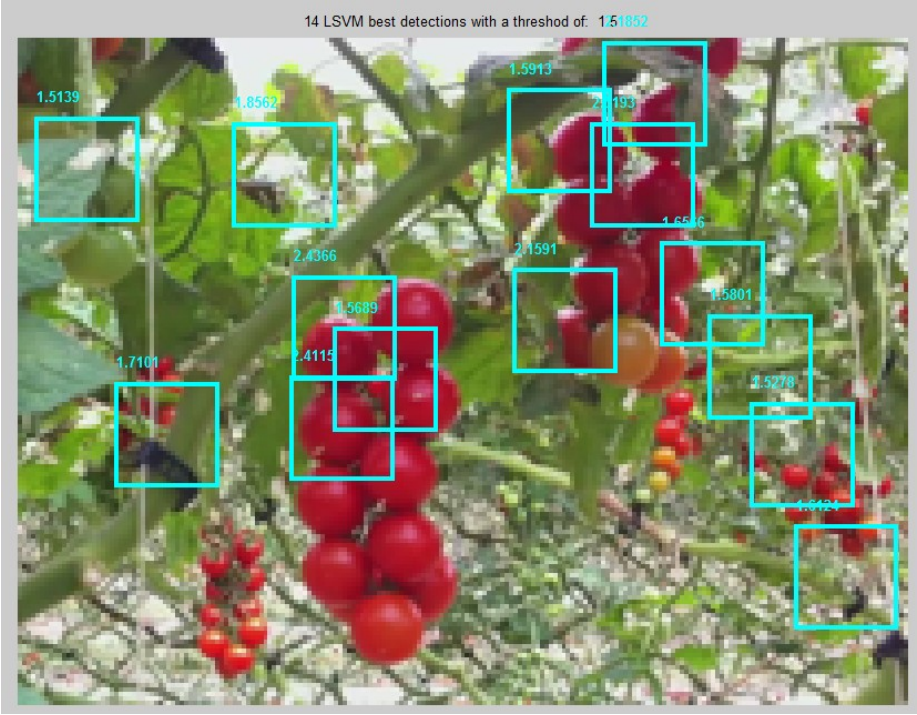

Figure 134 - pre-retraining - 14 detections over a threshold of 1.5

14 detections where the true positive are:

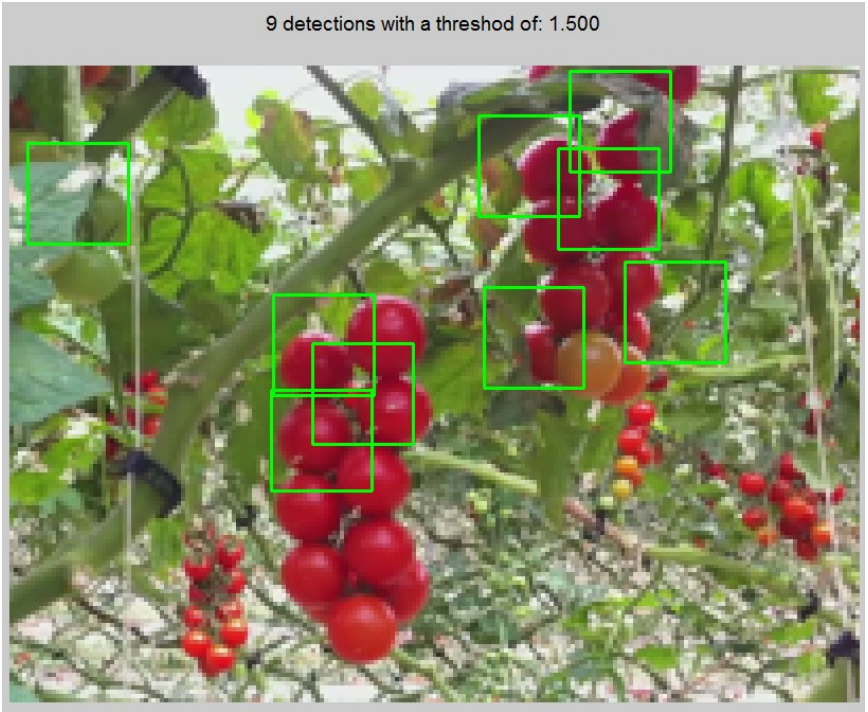

Figure 135 - 9 pre-retraining - true detections over a threshold of 1.5

and the false detection are the following:

Figure 136 - pre-retraining - false detections over a threshold of 1.5

The windows in Figure 137 are hard detections have been added to the dataset in order to give them to the classifier for the retraining.

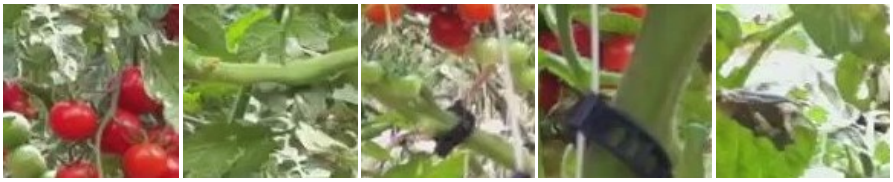

Figure 137 - Hard detections samples

Once this action has been performed, let's run again the detection on the same image test and let's a look of the result obtained for the same threshold value:

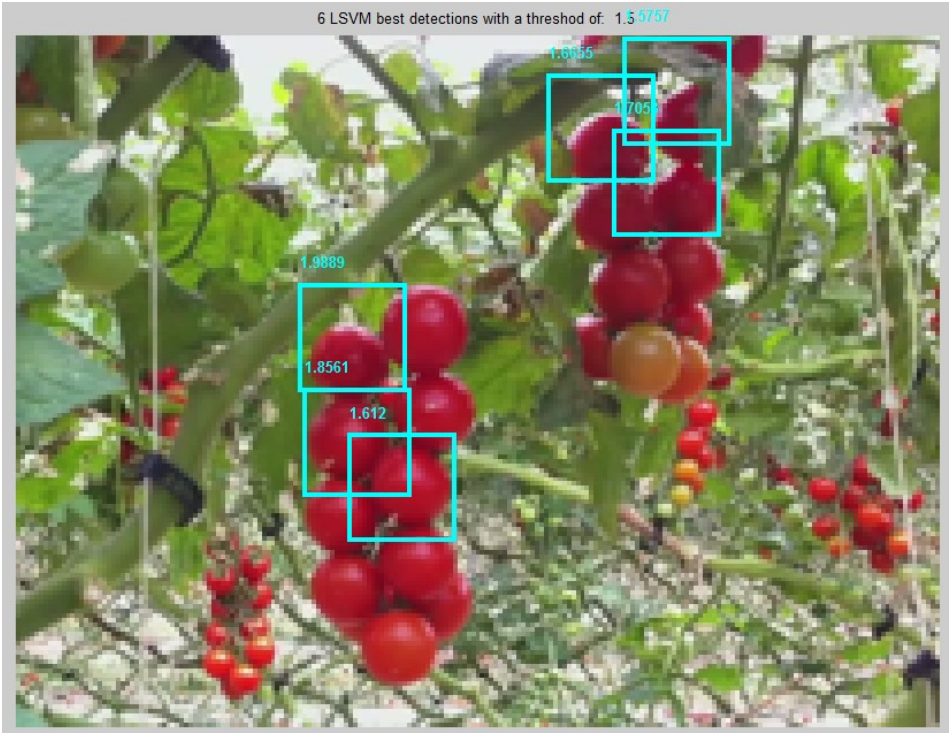

Figure 138 - Post-retraining - 6 detections over a threshold of 1.5

No more hard detections have been found.

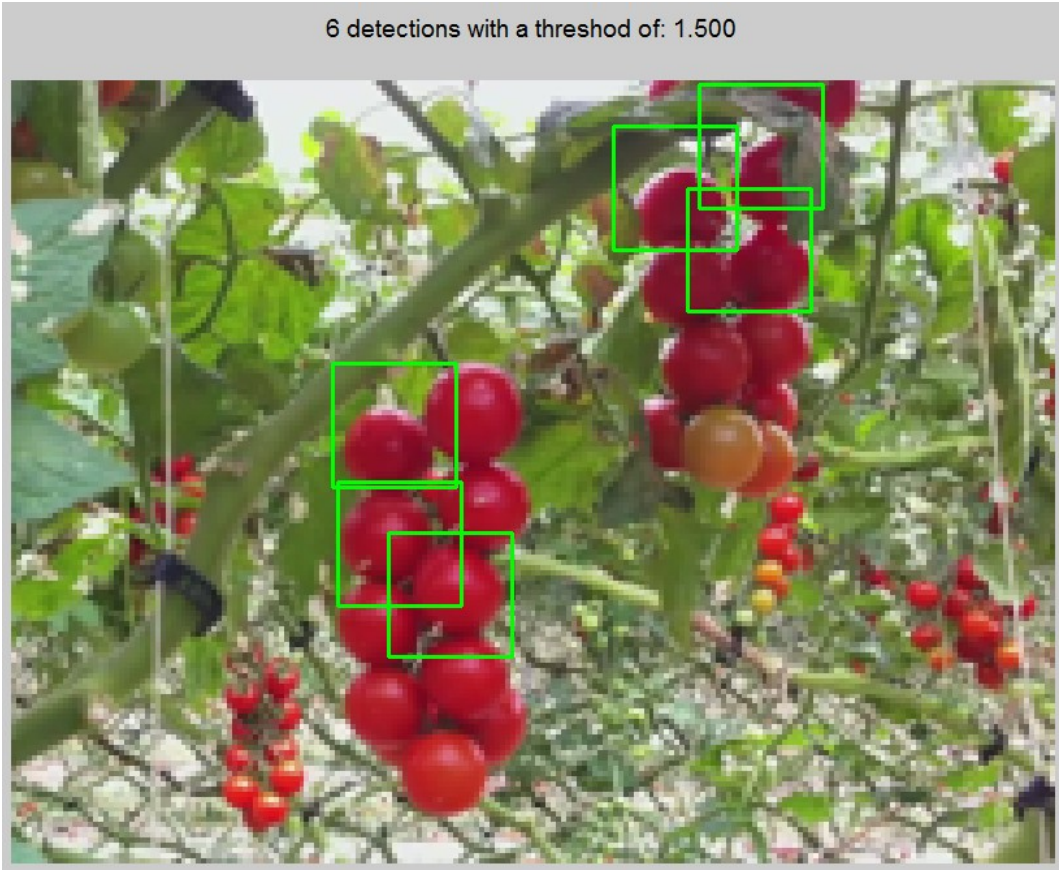

Figure 139 - Post-retraining - 6 true detections over a threshold of 1.5

These results give evidence on the dataset relevance and allow us to identify sample (the hard one's) we have previously "forgotten" or "undervalued" in the creation of the not-tomatoes dataset.

There are some features in the background that incorrectly, because of themselves appearance can be confused with true samples.

Figure 140 - Hard detection sample

The retraining approach can help to detect these features and remove them. Certainly not all the hard detection can be removed but in any case we can try to reduce their number.

6.3.3 Precision Recall evaluation

The following graphs show the recall, the precision and the precision-recall trend for different test images (shown in §6.3). Each curve is obtained varying the threshold used to filter the best confidence values in the range 0-2. This range rate comes from a heuristic evaluation.

A deeper description of the precision and recall meaning can be found on §5.8. They can be expressed by the following formulas:

Precision = *(True Positive) / (All Predicted Tomatoes)*
Recall = *(True Positive) / (Total Number of Tomato to be predicted);*

If higher would be the threshold (high severity with hard selection of the best detections), higher is the precision, but lower the recall. This trend has been confirmed in our results shown below for different test images.

The ideal threshold setting is the highest possible recall and precision rate. This goal is not always achievable, because the higher the recall rate, the lower the precision rate, and vice versa. Setting the most appropriate threshold for a category is a trade-off between these two rates and strongly depending on the application purpose.

In other words, depending on the application context, if we are interested to have a high precision, the threshold should be set higher, viceversa, if we prefer obtaining a higher recall the threshold can be set lower.

Same results are represented for different test images showing results absolutely comparable in order to demonstrate the repeatability and robustness of the method.

The analysis can be repeated for all graphs relates different test images:

- Lower threshold -> major recall and viceversa
- Lower threshold -> minor precision and viceversa
- Precision – recall trend is better as it is near to 1

Test Image 1:

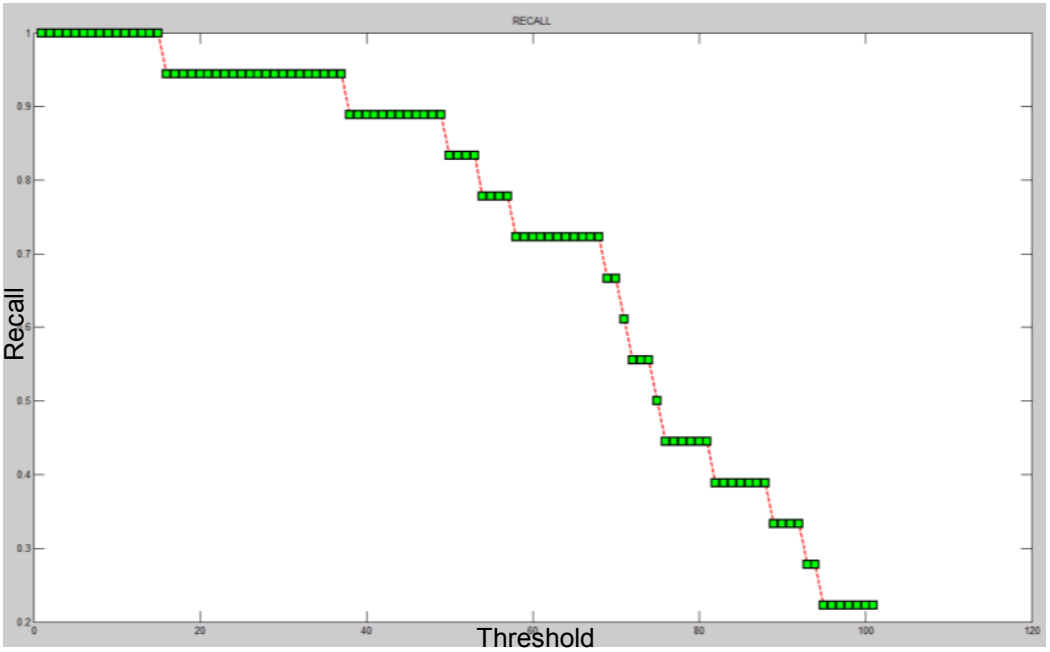

Figure 141 - Recall Test_Image_1

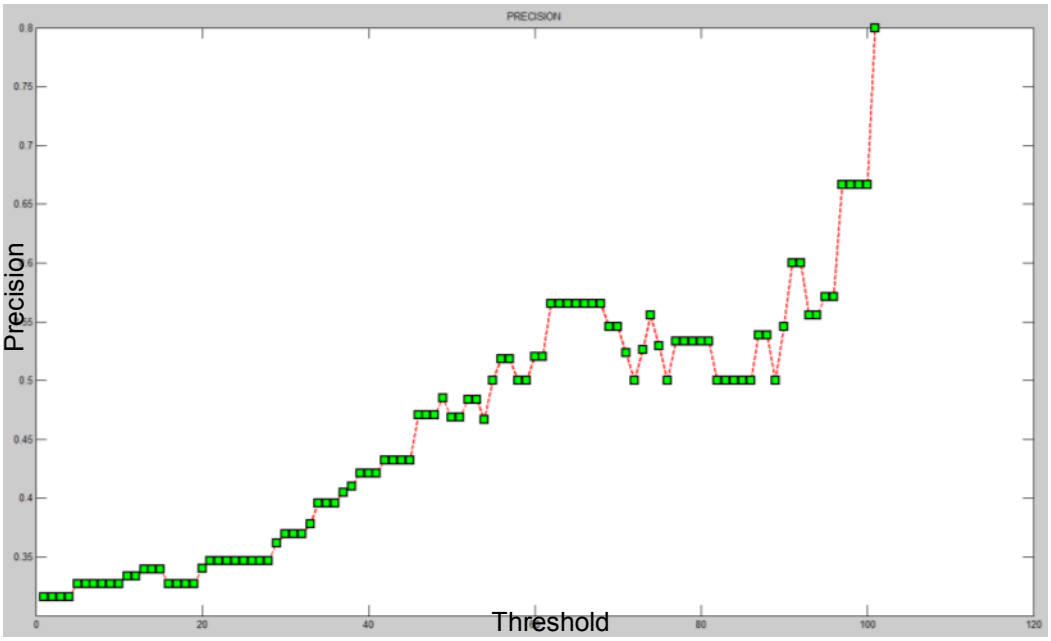

Figure 142 - Precision Test_Image_1

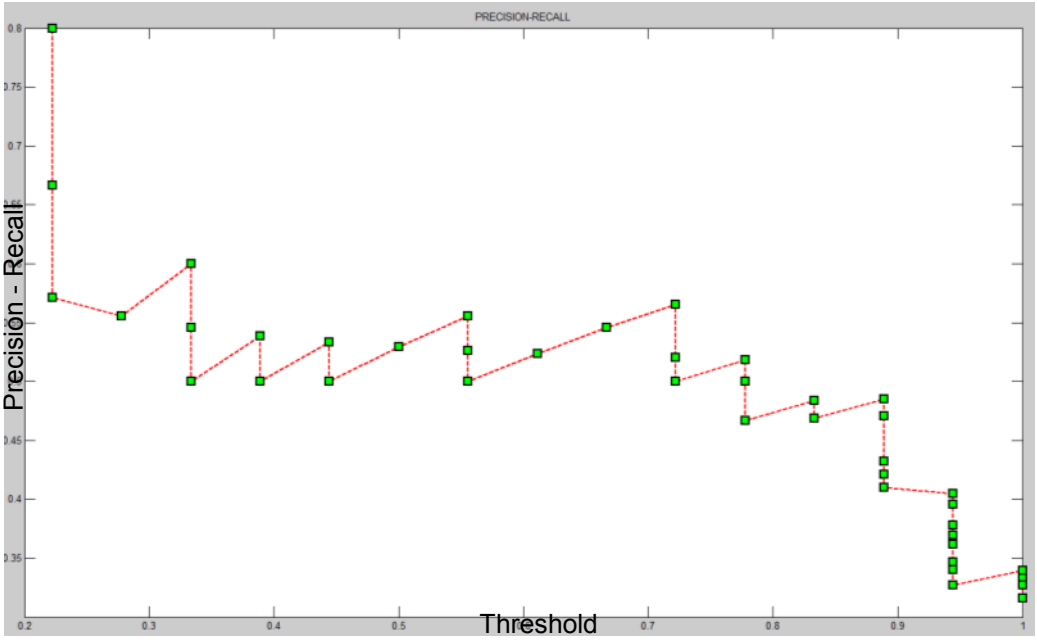

Figure 143 - Precision-Recall Test_Image_1

The Figure 144 shows the tomato detections for different threshold values. It shows as decreasing this values the number of true detections respect all the predicted value increase with a consequently increase of the precision.

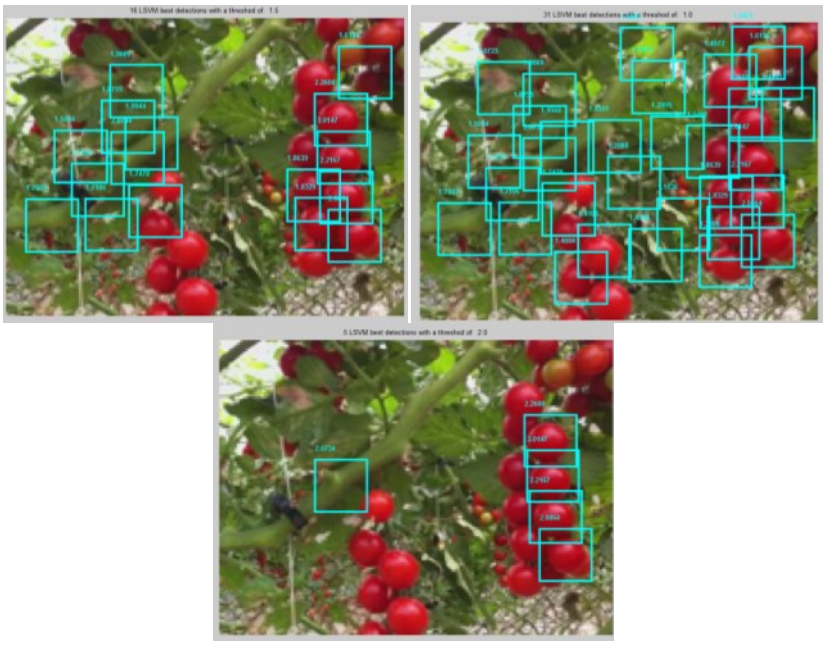

Figure 144 - Detection varying the threshold: 1, 1.5 and 2 (Test_Image_1)

Test Image 3:

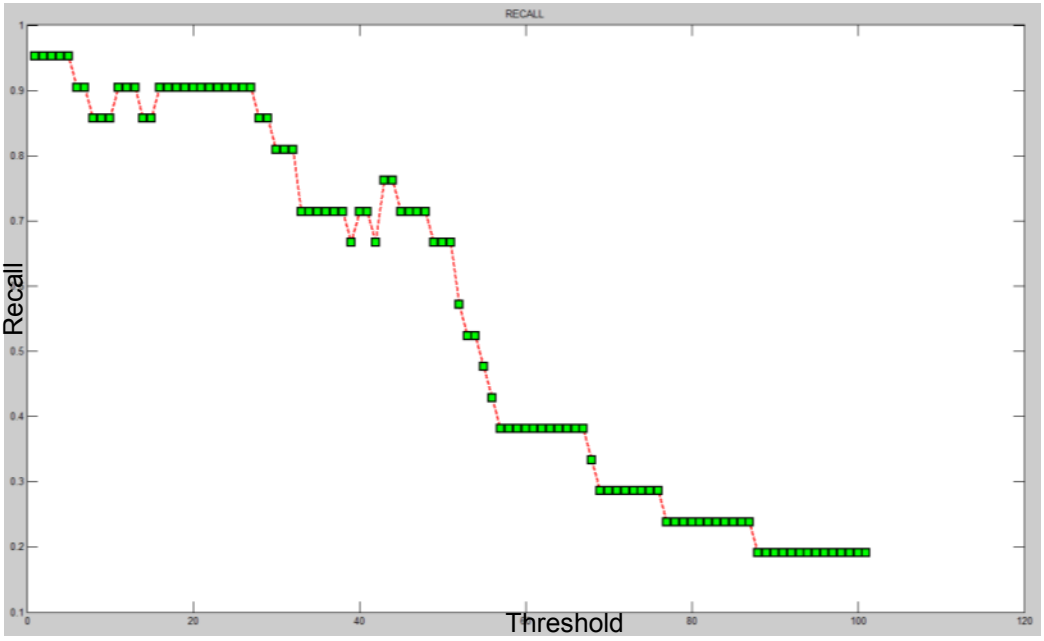

Figure 145 - Recall Test_Image_3

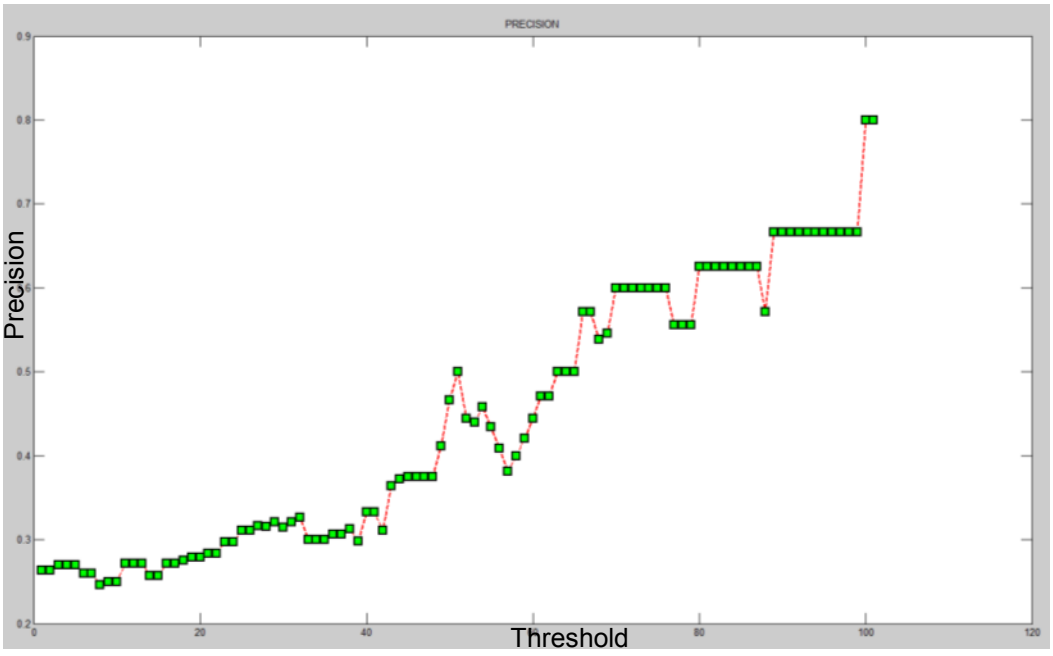

Figure 146 - Precision Test_Image_3

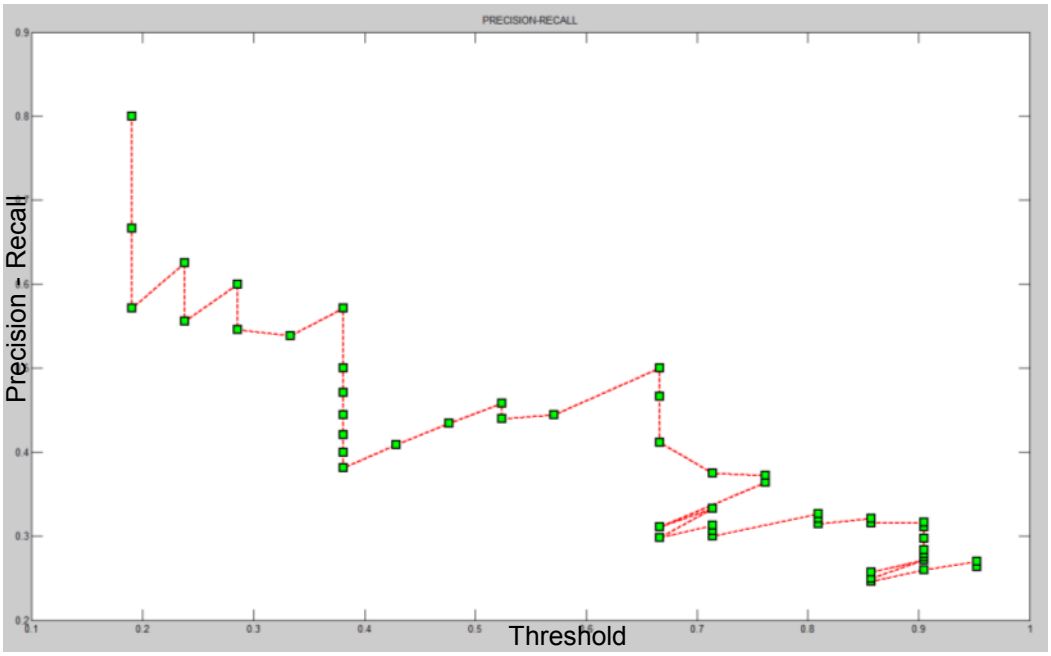

Figure 147 - Precision-Recall Test_Image_3

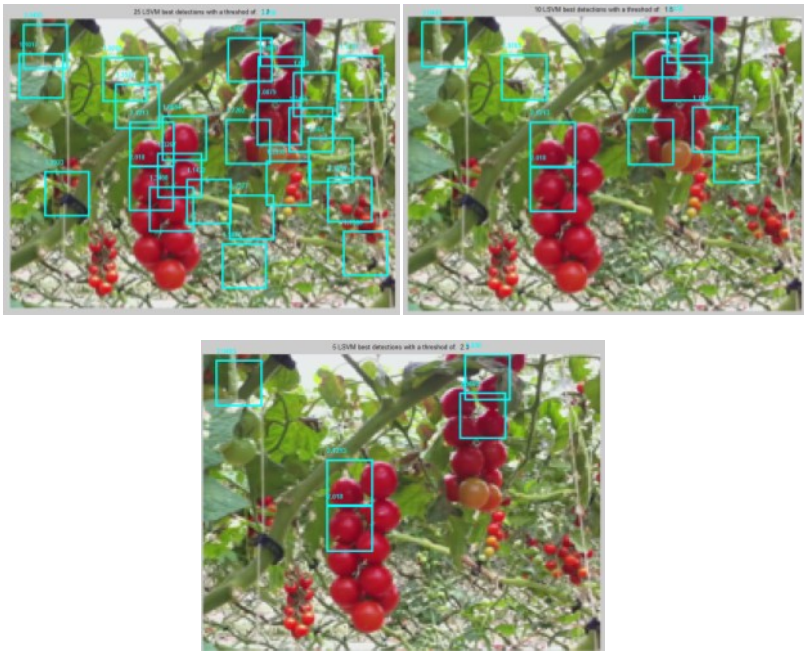

Figure 148 - Detection varying the threshold: 1, 1.5 and 2 (Test_Image_3)

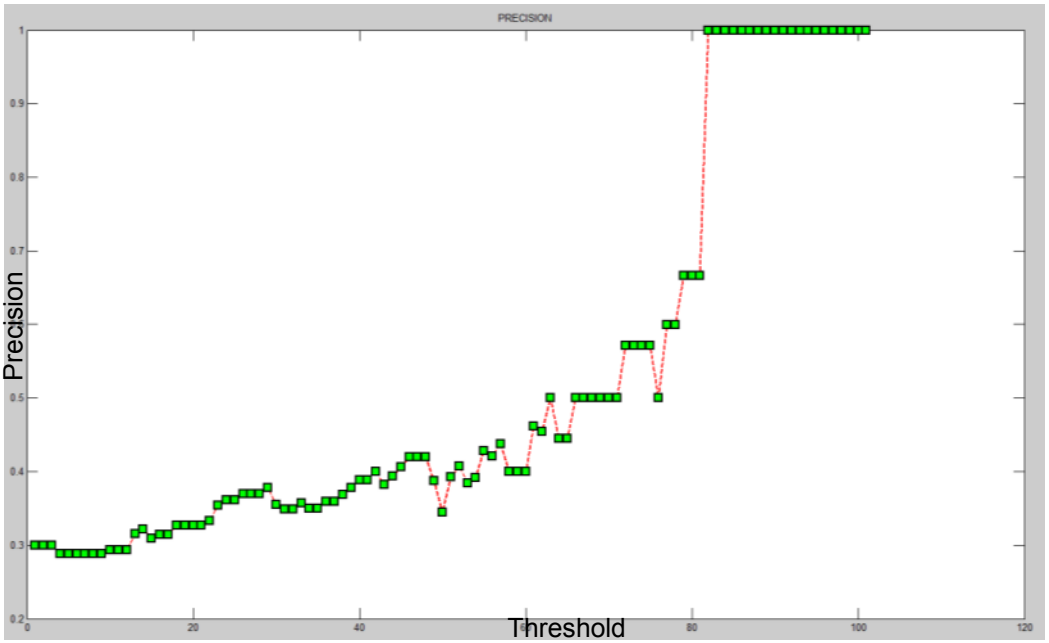

Figure 149 - Precision Test_Image_2

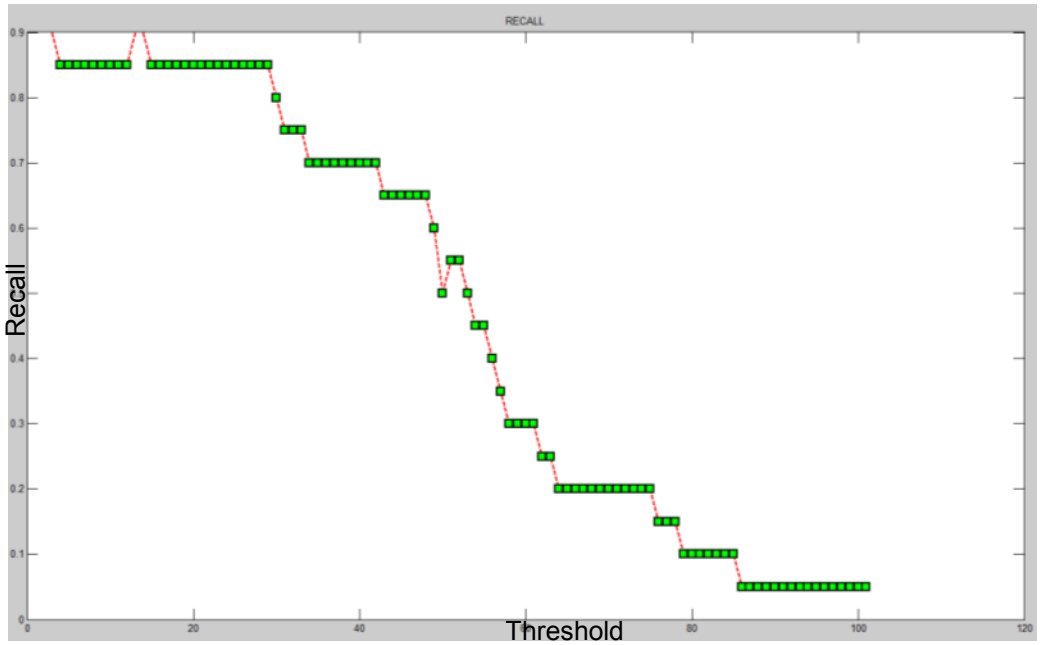

Figure 150 - Recall Test_Image_2

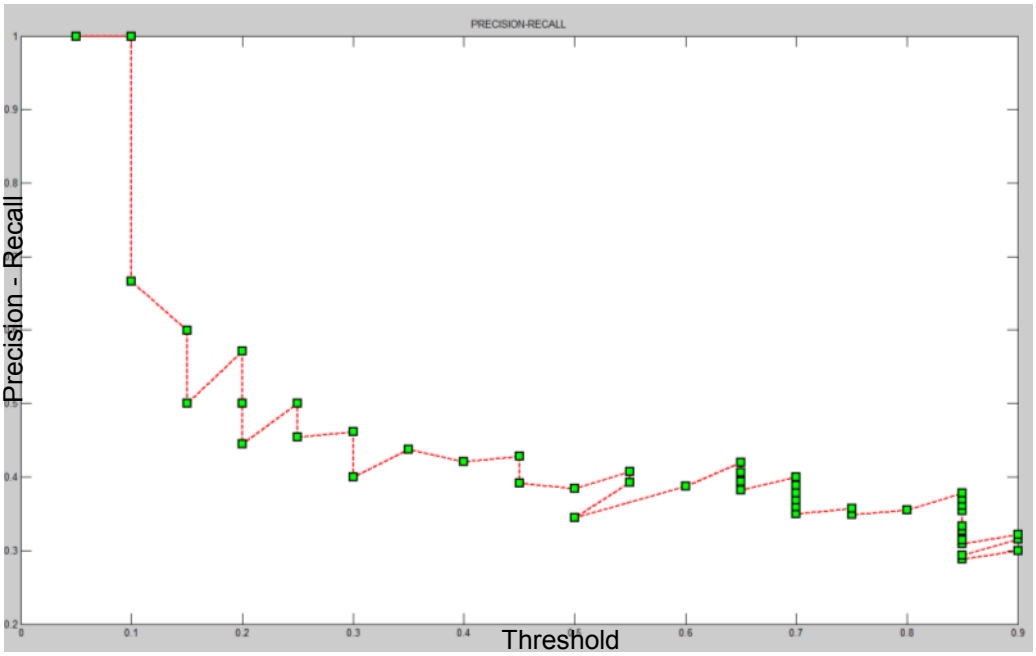

Figure 151 - Precision-Recall Test_Image_2

For better understand the importance of the retraining process, let's compare the results pre and post phase:

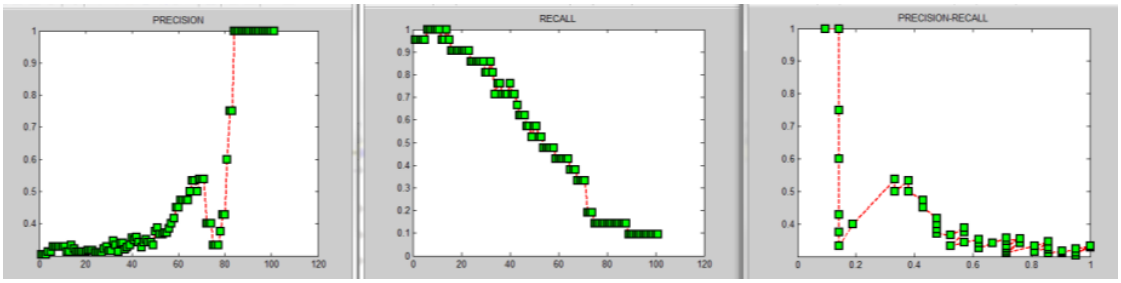

Figure 152 - Precision, Recall and Precision-Recall curves in the pre-retraining phase (Test_Image_3)

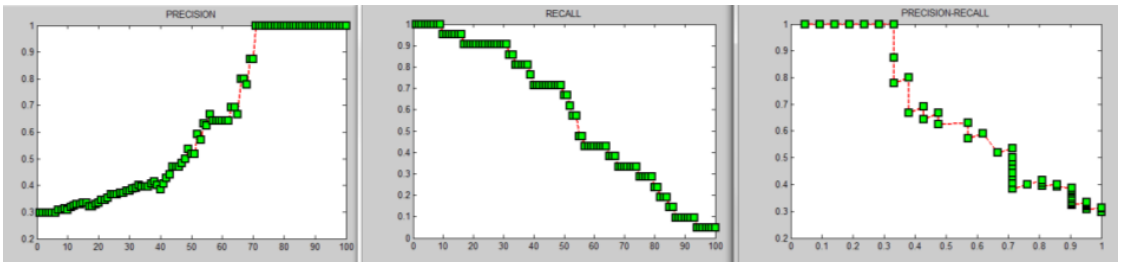

Figure 153 - Precision, Recall and Precision-Recall curves in the post-retraining phase (Test_Image_3)

A better trend, for all the three curves examined, has been obtained after the retraining process: a higher precision and lower recall for lower values of threshold and a bigger area under the curve of precision-recall.

Summarizing, if the scope of the research is to detect the grape of tomato, leaving a high threshold, the precision of the detection method is high and allow detecting the tomato grapes presence. Decreasing the threshold, more tomatoes will be detect but with a worse precision (because of the false detections). In this last case, if a high precision is requested the false positives number can decrease by using the retraining process or increasing the dataset).

Figure 154 - 20 True Positives against 26 False Positive with threshold of 0.2

Figure 155 - 13 True Positive and 9 False Positive with a threshold of 1

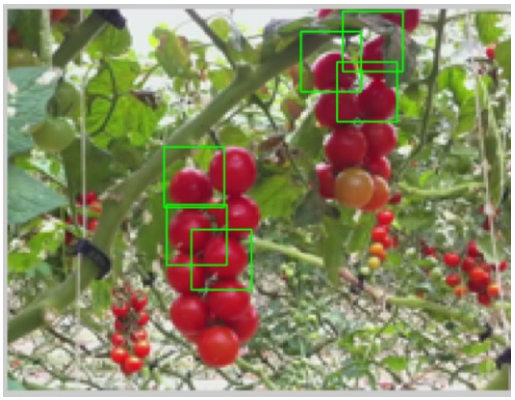

Figure 156 - 6 True Positive and no False Positive with a threshold of 1.5

The method allows, till the moment, to detect the tomato grapes presence with a really high precision (setting a high threshold value and so increasing the precision). If we are interested to count the number of tomatoes, a better precision for low threshold value should be obtained and this can be performed increasing the dataset or using a more performing descriptor to describe tomatoes.

To better understand the algorithm performance and to verify the compatibility with a future real time implementation (in particular for the detection phase), the average processing time of each algorithm step is reported in Table 2.

Step	***Average Time***
Training phase (for 500 samples of 24x24 pixels)	0.481962 s
Detection phase (images of 320x240 pixels)	1.3919 s (12 pixels steps for the sliding windows)

Table 2 – Average Processing Time for Training and Detection phase

In the detection phase the average time estimation is strongly dependent on how many pixels the sliding windows move on the test image (minors are the moving pixels, major the number of sliding windows and so higher the average time). From experimental analysis it has been chosen moving the sliding windows of 12 pixels because it doesn't affect the obtained results and algorithm performance (the results

are comparable to the case of a dense detection) finally improving algorithm execution time.

These data has been performed using a standard personal computer with an Intel(R)Xeon(R) CPU with 2.67GHz and 8Gb RAM.

Since the overall algorithm implementation has been performed using the Matlab tool, in order to obtain real-time performance, a C language implementation (using OpenCV and suitable SVM code as support libraries) is required.

7 Conclusions and Perspectives

In this work, some limits and benefits of introducing robotics in agricultural context have been faced and deepened. It seems to be clear how the adoption of intelligent system in such a context, can increase operations safety level for operators and quality of the chemicals distribution. A cost reduction of the phytosanitaries and fertilizers can be reached as well as a lower environmental pollution and better greenhouse product quality.

A suitable robotic platform has been designed and the different subsystems on board have been developed and tested reaching the objective prefixed in terms of integration and functionality.

The performed study, in the first phase of the work related to the Microsoft Robotics Studio framework, represents an important step towards the creation of a versatile robotic platform and a services library capable to achieve the different algorithms for the autonomous navigation. The possibility of simply managing the sensors and the navigation algorithms through the Visual Programming Language has allowed a fast algorithm development and left more time for the testing of the hardware/software combination on the robotics platform.

Sensors like DGPS and 2D Laser scanner are widely cited in literature and different tests have been performed in the present work. They have very good performance but due to the high cost, their use cannot be addressed to most SME in the agriculture field. The low cost methodologies proposed here are instead based on very low cost ultrasound sensors and suitable artificial vision algorithms (one for the robot navigation and one for detecting fruit/tomatoes along the greenhouse rows).

All the developed sensors and algorithms are used also for guidance and safety purpose. They allow obtaining the position of a moving machine along rows with respect to corridors boundary or with respect to a reference system defined in the greenhouse. At the same time they allow a safety level increase on performing such

activities in a hot, dirty and hard greenhouse environment, reducing the exposition and the employment of the workers during harmful operation.

Different tests have been done in a real greenhouse in order to evaluate performance and capabilities of the systems and results have been reported.

Finally, the used architecture has versatile features and can adapt to different applications: from the usage as service robot in greenhouse context to new and open field applications, such as in vineyards.

Despite the results obtained, future improvements are related to:

- Implementation of new and more complex navigation algorithms with different characteristics using the sensors and the communication interfaces already existing. Such solutions can improve the robot safety level and capabilities and consequently can raise workers safety during performing dangerous operations.
- Testing of the final algorithm on more and different scenarios.
- Creation of a MRDS services library in order to allow an efficient reuse of the existing code.
- In collaboration with DIEEI and Istituto Nazionale di Geofisica e Vulcanologia (INGV), further tests will be performed on the top of Etna volcano. Actually INGV manages more than 80 GPS station in Sicily and in Calabria. This network makes possible the planning of new applications (i.e. volcanic surveillance) in proximity of the Etna main craters.
- Integration with the tomato detection system in order to perform autonomously precision farming activities.

Related to the last point the implemented algorithm has shown good performance during experimental trials and its usage allows the detection of the grape of tomatoes with such acceptable precision.

Despite all, taking into account the state of art available nowadays, the results and

performances related the object/fruit detection offer great improvement cues. Till now there is no a mathematical law or algorithm allowing to represent or recognize nature: no general rules.

So many components and factors should be taken into account that, currently, a robust and reliable method allowing object detection doesn't exist. The developed algorithm has not such accuracy to allow harvesting operation or counting precisely tomatoes in a crop. Despite all opposition, improvements in the detection method can be provided and are related to:

- Using more performing descriptors (e.g. HOG or colour descriptors) to describe the fruit, extending and improving the dataset to obtain better precision and to limit the number of false detections.
- Once a higher precision and accuracy level is reached, the algorithm could be used for multiple application such as:
    - fruit classification;
    - illness fruit detection and local treatment application;
    - finally, 3D pose localization for harvesting purpose (Del Bimbo *et al.* 2010);
- Re-testing the algorithm in other and various agricultural environment, such as in vineyards for the grape detection (Nuske *et al.* 2011). In this last case, because of the grape feature and structure, maybe it should be preferred to implement the whole grape detection and not of the single fruit.

The present work represents a mixture and a fusion of different scientific disciplines and knowledge: robotics, computer vision, system integration, safety regulations and agricultural concepts are fused together.

The achievement of such a wide and various results have been possible leveraging the DIEEI (Dipartimento di Ingegneria Elettrica Elettronica ed Informatica of the University of Catania) skills of years of experience and involvements in different robotics projects. On the other hand, the Dipartimento GeSA (Dipartimento di Gestione dei Sistemi Agroalimentari e Ambientali of the University of Catania) has contributed with its experience in the field of agricultural machinery, with an

analysis of the agronomic and environmental context and providing many contacts with local SME. Finally the cooperation with the MICC (Media Integration and Communication Center of the University of Florence) related to the computer vision aspects has allowed the development of the tomato detection methodology.

The collaboration between DIEEI, GeSA and MICC, has allowed reaching of more exciting results in a context, those agricultural, still in investigation and in growth upon the technology profile. Moreover most of this work cover the area of interest of the project "Veicolo mobile a guida autonoma per la distribuzione di agrofarmaci in serra", co-funded and supported by MIPAAF (Ministero delle Politiche Agricole Alimentari e Forestali), that is currently in progress with the cooperation of Dept. GeSA, DIEEI and local SMEs.

Always keeping in mind all the aspect related the safety and the health of the workers, new and always safer solution will be provided in order to *facilitate* man tasks; *never* a machine could *replace* him.

8 Bibliography

ACACCIA G.M., MICHELINI R.C., MOLFINO R.M., RAZZOLI R.P. (2003). *Mobile robots in greenhouse cultivation: inspection and treatment of plants*. in Proc. of ASER 2003, 1st International Workshop on Advances in Service Robotics, 13-15 March, Bardolino, Italy.

ANDÒ B., *Sensors that provide security for people with depressed receptors,* IEEE Magazine on Instrumentation and Measurements, Vol.9, N.2, pp. 58-63, April 2006.

ANDÒ B., N. SAVALLI, *CANBUS Networked Sensors Use in Orientation Tools for the Visually Impaired Wired versus Wireless Technology*, IEEE Magazine on Instrumentation and Measurements, Vol. 11 N.1, pp.49-52, February 2008.

ANDÒ B., S. BAGLIO, S. LA MALFA, V. MARLETTA, *A multisensor guide system to assist visually impaired in unfamiliar environments*, Proceedings of the ASME 2009 International Design Engineering Technical Conferences & Computers and Information in engineering Conference, IDETC/CIE 2009, pp.1-5

ANNAMALAI P., LEE W. S., BURKS T. F., *Color vision system for estimating citrus yield in real-time.* In: CSAE ANNUAL INTERNATIONAL MEETING, 2004.

APREA C., SCIARRA G., LUNGHINI L., BOZZI N. (2001). *Il monitoraggio biologico dell'esposizione professionale e non ad antiparassitari.* Ann. Ist. Super. Sanità, vol. 37, n. 2, pp. 159-174

ARANZULLA P., BONACCORSO F., BRUNO C., CANTELLI L., LANTERI G., LONGO D., MELITA D., MUSCATO G., PENNISI A., PRESTIFILIPPO M.. *An innovative autonomous outdoor vehicle based on microsoft robotic studio*; Proceedings of CLAWAR'2010: 13th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, Nagoya, Japan, 31 August -03 September 2010; p. 97, 104; World Scientific Publishing Co Pte Ltd (24 Aug 2010)

AREFI A., MOTLAGH A. M., MOLLAZADE K., TEIMOURLOUC R. F., *Recognition and localization of ripen tomato based on machine vision*, Australian Journal of Crop Science, AJCS 5(10), pp 1144-1149, 2011

ARVATEC, http://www.arvatec.it/index_eng.php?view=categoria&categoryId=5&Agriculture

Axelrod B., http://www.benaxelrod.com/

BALLONI S., BONSIGNORE R., CAMILLIERI D., CARUSO L., CONTI A., SCHILLACI G. (2008). A *Survey of Safety Aspects Concerning Horticultural Farm Machineries*. Atti su CD-rom del Congresso Internazionale "Innovation Technology to Empower Safety, Health and Welfare in Agriculture and Agrofood Systems", Ragusa, Italy, 15-17 September.

BALLONI S., CARUSO L., CERRUTO E., EMMA G., SCHILLACI G.(2008). *A prototype of self-propelled sprayer to reduce operator exposure in greenhouse treatment*, Atti su CD-ROM dell'International Conference on "Innovation Technology to Empower Safety, Health and Welfare in Agriculture and Agro-food Systems", Ragusa, Italy 15–17 September.

BALLONI S., CARUSO L., CONTI A., SCHILLACI G., LONGO D., MUSCATO G. (2008). *Preliminary study for the development of an electrical autonomous vehicle for safe agricultural chemicals distribution inside greenhouses*. Atti su CD-ROM dell'International Conference on "Innovation Technology to Empower Safety, Health and Welfare in Agriculture and Agrofood Systems", Ragusa, Italy 15–17 September.

BALLONI S., CARUSO L., CONTI A., SCHILLACI G., LONGO D.,MUSCATO G. (2009*). Development of an electrical multifunctional autonomous vehicle able to cultivate covered crops and to safe distribute agricultural chemicals inside greenhouses.* Atti del XXXIII CIOSTA - CIGR V Conference 2009 "Technology and management to ensure sustainable agriculture, agro-systems, forestry and safety", Reggio Calabria, Italy, 17-19 june, Vol.1, pp. 355-359.

BALLONI S., CARUSO L., CONTI A., SCHILLACI G., VALENTINO M., LORETO C., FENGA C., RAPISARDA V (2008). *Use of a helmet endowed with forced ventilation and air filtration devices in greenhouse application of agrochemical treatments using an innovative prototype of self-propelled sprayer vehicle*. Atti su CD-ROM dell'International Conference on "Innovation Technology to Empower Safety, Health and Welfare in Agriculture and Agrofood Systems", Ragusa, Italy 15–17 September.

BELFORTE G., DEBOLI R., GAY P., PICCAROLO P. E RICAUDA AIMONINO D., (2006). *Robot design and testing for greenhouse applications. Biosystems Engineering*, 95, 3, pp. 309–321.

BJUGSTAD N., TORGRIMSEN T. (1996). *Operator safety and plant deposits when using pesticides in greenhouses*. J. Agric. Engng Res., 65, pp. 205-212.

BLACKMORE S., HAVE H., FOUNTAS S., *A specification of behavioural requirements for an autonomous tractor*. Automation Technology for Off-Road Equipment, Proceedings of the July 26-27, 2002 Conference, pp 33-42, Chicago, Illinois, USA, July 26, 2002.

BLASCO J., ALEIXOS N., ROGER J.M., RABATEL G., MOLTÓ E., *Robotic Weed Control using Machine Vision*, Biosystems Engineering, Vol. 83, Issue 2, October 2002, Pages 149-157

BLASCO J., ALEIXOS N., MOLTÓ E., (2003). *Machine vision system for automatic quality grading of fruit.* Biosystems Engineering, 85(4), 415–423.

BLASCO J., ALEIXOS N., GÓMEZ-SANCHÍS J., MOLTÓ E., *Recognition and classification of external skin damage in citrus fruits using multispectral data and morphological features,* Biosystems Engineering 103 (2009), pp 137–145

BLASCO J; ALEIXOS N; MOLTÓ E, *Machine Vision System for Automatic Quality Grading of Fruit*, Biosystems Engineering, Vol. 85, Issue 4, August 2003, Pages 415-423

BONACCORSO F., CANTELLI L., LONGO D., MELITA C.D., MUSCATO G.; *On-field test of a low cost, high precision augmented global navigation satellite system for agriculture applications;* Proceedings of CLAWAR 2011: the 14th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, Paris, France, 6 – 8 september 2011, pp 317 - 324, ISBN-10: 1439831610, ISBN-13: 978-1439831618

BRAUN T., KOCH H., STRUB O., ZOLYNSKI G., BERNST K., *Improving* pesticide spray application in vineyards by automated analysis of the foliage distribution pattern in the leaf wall, Vehiche Technology Symposium (CVT), Kaiserslautern, Germany, 16-18 March 2010

BRAUN T., ZOLYNSKI G., KOCH H., STRUB O., BERNS K., Visual Analysis of Vineyard foliage distribution, Proceedings of the International Conference of Agricultural Engineering, 2010

BROSNAN T., SUN D., *Inspection and grading of agricultural and food products by computer vision systems – a review*. Computer and Electronics in Agriculture, v. 36, p. 192-213, 2002.

BUIK R. 2006. Gps guidance and automated steering renew interest in precision farming technique. Trimble Navigation Limited. July, pp. 1-10.

CAPRI E., ALBERICI R., GLASS C. R., MINUTO G., TREVISAN M. (1999). Potential Operator Exposure to Procymidone in Greenhouses. J. Agric. Food Chem., 47, pp. 4443-4449

CERRUTO E., BALSARI P., OGGERO G., FRISO D., GUARELLA A., RAFFAELLI M. (2007). Operator safety during pesticide application in *greenhouses: a survey on Italian situation*. GreenSys 2007, Naples.

CHANG C.-C., LIN. C.-J.; *LIBSVM : a library for support vector machines*. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011

CHO, S. I., N. H. KI. 1999. *Autonomous speed sprayer using machine vision and fuzzy logic.* Transaction of the American Society of Agricultural Engineers, Volume: 42 (40); Page(s): 1137-1143

DALAL N., *Finding People in Images and Videos*, PhD Thesis. Institut National Polytechnique de Grenoble / INRIA Grenoble , Grenoble, July 2006.

DAHLKAMP H., KAEHLER A., STAVENS D., THRUN S., BRADSKI G., *Self-supervised monocular road detection in desert terrain*. In G. Sukhatme, S. Schaal, W. Burgard, and D. Fox, editors, Proceedings of the Robotics Science and Systems Conference, Philadelphia, PA, 2006.

DARIO P., SANDINI G., ALLOTTA B., BUCCI A., BUEMI F., MASSA M.,FERRARI F., MAGRASSI M., BOSIO L., VALLEGGI R., GALLO E.,BOLOGNA A., CANTATORE F., TORRIELLI G. E MANNUCCI A., 1994.*The agrobot project for greenhouse automation*. Acta Hort. (ISHS) 361, pp. 85-92.

DEL BIMBO A., FRANCO F., PERNICI F., *Local shape estimation from a single keypoint*, Proc. of Computer Vision and Pattern Recognition, International Workshop on Non-Rigid Shape and Deformable Image Alignment (NORDIA),2010, pp. 23–28.

DE MEZZO B.; RABATEL G. ; FIORIO C., *Weed leaf recognition in complex natural scenes by model-guided edge pairing*, 4th ECPA European Conference on Precision Agriculture, pp 141-147, Berlin, DEU, 15-19 June 2003

DE MEZZO B., RABATEL G., FIORIO C., *Weed leaf recognition in complex natural scenes by model-guided edge pairing*, 4th European Conference on Precision Agriculture, Berlin , Germany, 26 November 2007

DEERE, http://www.deere.com/servlet/com.deere.u90785.productcatalog.view.servlets.ProdCatProduct?tM=FR&pNbr=0500_PC

DE VREEDE J.A.F., BROUWER D.H., STEVENSON H.,VAN HEMMEN J.J. (1997). *Exposure and Risk Estimation for Pesticides in High-Volume Spraying*, Ann. Occup. Hyg., Vol. 42, No. 3, pp. 151–157.

Feature detection (computer vision), http://en.wikipedia.org/w/index.php?title=Feature_detection_(computer_vision)&oldid=455241471 (last visited Dec. 3, 2011)

FRISO D., BALDOIN C., BONDESAN D., (2008). *Prospettive di miglioramento nella distribuzione meccanica di fitofarmaci al pomodoro in coltura protetta per la salvaguardia dell'ambiente e della salute*. Ragusa 4 aprile, pp. 33-43.

FUJUAN.V., *Control system design of spraying robot*. Computer and Communication Technologies in Agriculture Engineering (CCTAE), 2010 International Conference, pp 8 - 11, Chengdu, June 2010

FUMIMATIC, http://www.fumimatic.com/primefumi.htm.

GAY P., PICCAROLO P., RICAUDA AIMONINO D., E DEBOLI R. (2008). *Robotics for work and environment safety in greenhouse*. Atti su CD-ROM dell'International Conference on "Innovation Technology to Empower Safety, Health and Welfare in Agriculture and Agro-food Systems", Ragusa, Italy 15–17 September.

GHAZALI K.H., RAZALI S.; MUSTAFA M.M., HUSSAIN A., *Vision System for Automatic Weeding Strategy in Oil Palm Plantation using Image Filtering Technique*, Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008. 3rd International Conference, pp 1 - 5, Damascus, 7-11 April 2008

GIAMBATTISTELLI S. (2003). *La valutazione del rischio nei processi chimici*. Ancona, Italy, 16 May

GOMEZ-SANCHIS J., MOLTÓ E., CAMPS-VALLS G., GÓMEZ-CHOVA L., ALEIXOS N., BLASCO J., *Automatic correction of the effects of the light source on spherical objects*. An application to the analysis of hyperspectral images of citrus fruits, Journal of Food Engineering 85 (2008), pp 191–200

HAFF R. P., PEARSON T., *Spectral band selection and testing of edge-subtraction leaf segmentation*, Transactions of the ASABE, 2006, Vol. 49(4), pp 1105−1113

HAYASHI S., OTA T., KUBOTA K., GANNO K., KOND N., *Robotic Harvesting Technology for Fruit Vegetables in Protected Horticultural Production*, Information and Technology for Sustainable Fruit and Vegetable Production, FRUTIC 05, 12-16 September, Montpellier, France, 2005

HERAUD J.A.,LANGE A.F. 2009. *Agricultural Automatic Vehicle Guidance from Horses to GPS: How we got here, and where we are going*. Presentation Agricultural Equipment Technology Conference,9-12 february 2009. American Society of Agricult And Biological Engineers. 33. February 2009. pp. 1-67

HOLPP, M. DURR, L., *A Automatic guidance system for tractors in fruit farming*, VDI BERICHTE, 2006, VOL 1958, pages 207, Germany

JAFARI A., MOHTASEBI S. S., JAHROMI H. E., OMID M., *Weed Detection in Sugar Beet Fields Using Machine Vision*, International Journal of Agriculture & Biology, Vol. 8, No. 5, 2006

JIMÉNEZ A.R., CERES R., PONS J.L., *A Survey of Computer Vision Methods for Locating Fruit on Trees*, Transactions of the ASAE, vol. 43(6), pp 1911-192, 2000

JOVITA O., OLUROMINIYI O. (2011). *Pesticide use and related health problems among greenhouse workers in Batinah Coastal Region of Oman*. Journal of Forensic and Legal Medicine 18, pp. 198-203

KITAMURA S., OKA K., *Recognition and cutting system of sweet pepper for picking robot in greenhouse horticulture*, Mechatronics and Automation, 2005 IEEE International Conference, pp 1807 - 1812, Vol. 4, 29 July-1 Aug. 2005

KOEN E. A. VAN DE SANDE, THEO GEVERS AND CEES G. M. SNOEK, *Evaluating Color Descriptors for Object and Scene Recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 32 (9), pages 1582-1596, 2010

KONDO N., YAMAMOTO K., YATA K., KURITA M., *A Machine Vision for Tomato Cluster Harvesting Robot*, 2008 ASABE Annual International Meeting ASABE, Rhode Island Convention Center, Providence, Rhode Island, June 29 – July 2, 2008

LI M., IMOU K., WAKABAYASHI K., YOKOYAMA S., *Review of research on agricultural vehicle autonomous guidance*, Int J Agric & Biol Eng, Vol. 2 No.3, pag 1 - 16, 2009, DOI: 10.3965/j.issn.1934-6344.2009.03.001-016

LI Y., XIA C., LEE J.,*Vision-based Pest Detection and Automatic Spray of Greenhouse Plant,* Industrial Electronics, 2009. ISIE 2009. IEEE International Symposium, pp 920 - 925, Seoul, 5-8 July 2009

LONGO D., PENNISI A., CARUSO L., MUSCATO G., SCHILLACI G.; *An autonomous electrical vehicle based on low-cost ultrasound sensors for safer operations inside greenhouses*. International Conference Ragusa SHWA2010 - September 16-18, 2010 Ragusa Ibla Campus-Italy "Work safety and risk prevention in agro-food and forest systems"

LONGO D., PENNISI A., BONSIGNORE R., MUSCATO G., SCHILLACI G.; *A multifunctional tracked vehicle able to operate in vineyards using GPS and laser range-finder technology*. International Conference Ragusa SH-WA2010 -September 16-18, 2010 Ragusa Ibla Campus-Italy "Work safety and risk prevention in agro-food and forest systems"

Lowe, D. G., *Distinctive image features from scale-invariant keypoints*. International Journal of Computer Vision (IJCV), 60(2):91–110, 2004.

LULIO L.C., TRONCO M.L., PORTO A.J.V..*JSEG-based Image Segmentation in Computer Vision for Agricultural Mobile Robot Navigation*. Computational Intelligence in Robotics and Automation (CIRA), 2009 IEEE International Symposium, pp 240 - 245, Daejeon, 15-18 Dec. 2009

MANDOW A., GOMEZ-DE-GABRIEL J.M., MARTINEZ J.L., MUNOZ .F., OLLERO E A. GARCIA-CEREZO (1996). *The autonomous mobile robot AURORA for greenhouse operation*. IEEE robotics and automation magazine. Vol. 3 issue 4, pp. 18-28.

MISAO, Y. 2001. *An image processing based automatic steering power system*. American Society of Agricultural Engineers annual international meeting, Paper number: 013106

MOLTÓ E., MARTIN B., GUTIÉRREZ A., (2001*). Pesticide Loss Reduction by Automatic Adaptation of Spraying on Globular Trees*. J. agric. Engng Res., 78 (1), pp. 35-41

MRDS (Microsoft Robotics Developer Studio) http://msdn.microsoft.com/en-us/robotics/aa731517

MURAKAMI N., ITO A., JEFFREY D. WILL, MICHAEL STEFFEN, K. INOUE, K. KITA, S. MIYAURA, *Development of a teleoperation system for agricultural vehicles*, Computers and Electronics in Agriculture, Volume 63, Issue 1, August 2008, Pages 81-88

MUSCATO G., CALTABIANO D., GUCCIONE S., LONGO D., COLTELLI M., CRISTALDI A., PECORA E., SACCO V., SIM P., VIRK G.S., BRIOLE P.,

SEMERANO A., White T.; *ROBOVOLC: A Robot for volcano exploration -Result of first test campaign*, Industial Robot: An International Journal, Vol. 30, N.3, pp. 231-242, 2003.

MUSCATO G., PRESTIFILIPPO M., ABBATE N., RIZZUTO I., *A Prototype of an Orange Picking Robot,* Past History, the New Robot and Experimental Results", Industrial Robot: An International Journal Vol.32, • N.2, 2005.

NETO J. C., MEYER G. E., JONES D. D., SURKAN A. J., *Adaptive image segmentation using a fuzzy neural network and genetic algorithm for weed detection*. In: Asae Annual International Meeting, 2003.

NUSKE S., ACHAR S., BATES T., NARASIMHAN S., SINGH S., *Yield Estimation in Vineyards by Visual Grape Detection,* International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, September 25-30, 2011.

NUYTTENS S., WINDEY B., SONCK B. (2005). *Comparison of exposure for five different greenhouse spraying applications*. Atti del Convegno XXXI CIOSTA-CIGR V Congress "Increasing Work Efficiency in Agricolture, Horticolture and Forestry", September 19-21, University of Hohenheim, Stuttgart, Germany, ISBN 3-00-016346-8, pp. 98-105.

PAJARES G., TELLAECHE A., BURGOSARTIZZU X.-P., RIBEIRO A., *Design of a computer vision system for a differential spraying operation in precision agriculture using Hebbian learning*, IET computer vision 2007, vol. 1, no3-4, pp. 93-99

PALAMAS, G.HOUZARD, J.-F.; KAVOUSSANOS, M.; *Relative Position Estimation of a Mobile Robot in a Greenhouse Pathway*, IEEE International Conference on Industrial Technology (ICIT2006), Mumbai, December 15-17, 2006, pp. 2298-2302; ISBN: 1-4244-0726-5

PARISEK Z., RUZSA Z., GORDOS G., *Mathematical algorithms of an indoor ultrasonic localisation system*, Infocommunications Journal 2009/IV VOLUME LXIV, 2009

PASINETTI L.. *Malattie delle piante*, Hoepli, Milano, 1952.

PEDERSEN S. M., FOUNTAS S. AND BLACKMORE S. *„Agricultural Robots applications and Economic Perspectives*, Service Robot Applications, Yoshihiko Takahashi (Ed.), ISBN: 978-953-7619-00-8, InTech, Austria, 2008

PLEBE A., GRASSO G., *Localization of spherical fruits for robotic harvesting. Machine Vision and Applications*, v. 13, p. 70-79, 2001.

QUAN L., TAN P., ZENG G., YUAN L., WANG J., S. BING KANG, *Image-based plant modeling*. ACM Trans. Graph. 25(3): 599-604 (2006)

ROWE D.E., MALONE S., YALES Q.L. (2000). *Automated greenhouse spray system for increased safety and flexibility Crops* Science 40, pp.1176-1179

SAMMONS P. J., FURUKAWA T. AND BULGIN A., *Autonomous Pesticide Spraying Robot for Use in a Greenhouse*, 2005 Australian Conference on Robotics and Automation (CD-ROM), December 5-7, 2005, Sydney, pp. 1-8, 2005

SANCHEZ-HERMOSILLA J., RODRIGUEZ F., GONZALEZ R., GUZMAN J. L., BERENGUEL M. (2010). *A mechatronic description of an autonomous mobile robot for agricultural tasks in greenhouses*, Mobile Robots Navigation, Alejandra Barrera (Ed.), ISBN: 978-953-307-076-6, INTECH,

SANDINI G., BUEMI F., MASSA M.; ZUCCHINI M. *Visually guided operations in green-houses*. Intelligent Robots and Systems '90. 'Towards a New Frontier of Applications', Proceedings. IROS 1990. IEEE International Workshop, vol.1, pages 279 - 285, Ibaraki, Japan, 3-6 Jul 1990

SCHILLACI G., BALLONI S., CARUSO L., CONTI A., RAPISARDA V. (2009). *Health and Safety Aspects Connected with the Use of a Self Propelled Sprayer in Greenhouses*. Atti del XXXIII CIOSTA - CIGR V Conference 2009 "Technology and management to ensure sustainable agriculture, agro-systems, forestry and safety", Reggio Calabria, Italy, 17-19 june.

SCHILLACI G., BALLONI S., CARUSO L., CONTI A., PENNISI A., LONGO D., MUSCATO G., *Prove di funzionamento telecomandato e autonomo di un veicolo elettrico multifunzionale destinato alle colture in serra*, Proceedings of the IX Convegno Nazionale dell'Associazione Italiana di Ingegneria Agraria AIIA 2009, Memoria n. 7-37, 12-16 september 2009, Ischia Porto, Italy

SCHILLACI G., BONSIGNORE R., PENNISI A., LONGO D., MUSCATO G., *A multifunctional remote controlled and/or autonomous electrical vehicle able to operate in slope vineyard*, Proceedings of the Third International Congress on Mountain Steep Slope Viticulture, Castiglione di Sicilia (CT), Italy, 12-14 May 2010.

SHIN B.,KIM S.. 2001. A*utonomous guidance system for small orchard sprayer with Ultrasonic Sensors*. American Society of Agricultural Engineers Meeting Presentation, Paper number: 01-1193

SLAUGHTER D.C., GILES D.K., DOWNEY D. (2008). *Autonomous robotic weed control systems: a review*; Computers and electronics in agriculture, Vol. 61, pp 63-78, Elsevier

STEWARD B. L., TIAN L. F., NETTLETON D., TANG L., *Reduced-dimension clustering for vegetation segmentation*. Transactions of the ASAE, v. 47, p. 609-616, 2004.

STOMBAUGH, T. S., SCOTT A. SHEARER .2001. *DGPS-based guidance of high-speed application equipment.* American Society of Agricultural Engineers Annual Meeting, Paper number: 011190

STOPPINI A., RADICIONI F., *Materiale didattico aggiornato per il corso di topografia* - A.A 2010/2011"

SVENSSON H. B. J., *Assessment of Grapevine Vigour Using Image Processing*, Master's Thesis in Image Processing, 2002

TANG L., TIAN L., STEWARD B. L., *Classification of broadleaf and grass weeds using gabor wavelets and an artificial neural network*. Transactions of the ASAE, v. 46, p. 1247-1254, 2003.

TELLAECHE A., BURGOSARTIZZU X.P., PAJARES G., RIBEIRO A., *A Vision-based Classifier in Precision Agriculture Combining Bayes and Support Vector Machines, Intelligent Signal Processing,* 2007. WISP 2007. IEEE International Symposium, pp 1 - 6, Alcala de Henares, 3-5 Oct. 2007

Template matching, http://en.wikipedia.org/w/index.php?title=Template_matching&oldid=461904134 (last visited Dec. 3, 2011).

TORU T., *Research in autonomous agriculture vehicles in Japan*, Computers and Electronics in Agriculture, Elsevier, Vol. 25 (2000) pag 133–153

TRIMBLE, http://www.trimble.com/agriculture/guidance.aspx

VAN HENTEN E.J., HEMMING J., VAN TUIJL B.A.J., KORNET J.G., MEULEMAN J., BONTSEMA J., VAN OS E.A., *An Autonomous Robot for Harvesting Cucumbers in Greenhouses*, Autonomous Robots, Springer, Volume 13, Number 3, November 2002 , pp. 241-258

Visual descriptors, http://en.wikipedia.org/w/index.php?title=Visual_descriptors&oldid=425976808 (last visited Dec. 3, 2011).

WAN, X. AND LIU, G., *Automatic Navigation System with Multiple Sensors* 2008, in IFIP International Federation for Information Processing, Volume 259;Computer and Computing Technologies in Agriculture, Vol. 2; Daoliang Li; (Boston: Springer),pp. 769–776.

WHITTAKER D., MILES G. E., MITCHELL O. R., GAULTNEY L. D., *Fruit Location in a Partially Occluded Image*, Transactions of the ASABE. 30 (3), pp 0591-0596, 1987

XAVIER P. BURGOS-ARTIZZU, A. RIBEIRO, A. TELLAECHE, G. PAJARES, *Optimisation of natural images processing using different evolutionary algorithms.* IEEE Congress on Evolutionary Computation 2008: 1268-1275

XIA C., LI Y., CHON T-S, LEE J-M, *A stereo vision based method for autonomous spray of pesticides to plant leaves*. Industrial Electronics, 2009. ISIE 2009. IEEE International Symposium, pp 909 - 914, Seoul, 5-8 July 2009

YANG L., DICKINSON J., WU Q.M.J., LANG S., *A Fruit Recognition Method for Automatic Harvesting*, Mechatronics and Machine Vision in Practice, 2007. M2VIP 2007. 14th International Conference,pp 152 - 157, Xiamen, 4-6 Dec. 2007

YIN H., CHAI Y.,YANG, S.X.; MITTAL, G.S., *Ripe Tomato Extraction For A Harvesting Robotic System*, Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference, pp 2984 - 2989, San Antonio, TX, 11-14 Oct. 2009

YIN J., MAO H., XIE Y., *Segmentation Methods of Fruit Image and Comparative Experiments*, Computer Science and Software Engineering, 2008 International Conference, pp 1098 - 1102, Wuhan, Hubei, 12-14 Dec. 2008

ZHANG B., YU S., XU H., YANG. S., *A Mobile Robot System Based on Electromagnetic Navigation*. Electrical Machines and Systems, 2005. ICEMS 2005, Proceedings of the Eighth International Conference, pp 1559 - 1562, Nanjing, 29-29 Sept. 2005