Autonomous mobile robotics in open environments

Jorge Eduardo Aviles Mejia

To cite this version:

Jorge Eduardo Aviles Mejia. Autonomous mobile robotics in open environments. Automatic. Université de Limoges, 2023. English. NNT : 2023LIMO0017 . tel-04094226

HAL Id: tel-04094226
https://theses.hal.science/tel-04094226v1

Submitted on 10 May 2023

Thèse de Doctorat

Université de Limoges
ED 653 : Sciences et Ingénierie (SI)
Faculté des Sciences et Techniques – Institut de Recherche XLIM

Thèse pour obtenir le grade de
Docteur de l'Université de Limoges
Image signal et automatique

Présentée et soutenue par
Jorge Eduardo AVILÉS MEJIA
Le 7 avril 2023

ROBOTIQUE MOBILE AUTONOME EN MILIEU OUVERT

Thèse dirigée par Ouiddad LABBANI-IGBIDA et Joanny STÉPHANT

JURY :

Président du jury
M. Patrick DANÈS, Professeur – LAAS Toulouse

Rapporteurs
M. Olivier LY, Professeur – LaBRI – Université Bordeaux
M. Roland LENAIN, DR – INRAE

Examinateurs
M. Eric LUCET, Professeur – CEA
M. Anthony GELIBERT, Directeur Scientifique – Carbon Bee

This thesis is dedicated to my beloved parents and brothers.

Dreams without goals, are just dreams

Denzel Washington

Acknowledgements

I would like to express my thankfulness and gratitude towards my advisors, Dr. Ouiddad Labbani and Dr. Joanny Stéphant for giving me the opportunity to work on this fascinating project and for their invaluable support and guidance throughout this entire doctoral degree program.

I am deeply grateful to Dr. Olivier Ly and Dr. Roland Lenain for acting as reviewers for this thesis. I also want to thank Dr. Patrick Danès, Dr. Eric Lucet and Dr. Anthony Gelibert for accepting to be part of the thesis jury.

Special thanks to Dr. Daniel Soto and Dr. Jesús Franco for having contributed to this project. In the same way, I would like to thank my colleagues for the shared moments over the past years.

Finally, I would also like to extend my deepest gratitude to my family. Thank you for joining me on this grand adventure and for your unconditional support.

Jorge Avilés

Droits d’auteurs

Table of contents

List of Figures

List of Tables

Notations

$\begin{pmatrix} \vec{i}_W \\ \vec{j}_W \\ \vec{k}_W \end{pmatrix}$	World coordinate system
$\begin{pmatrix} \vec{i}_R \\ \vec{j}_R \\ \vec{k}_R \end{pmatrix}$	Platform coordinate system
$\begin{pmatrix} \vec{i}_{Xeye} \\ \vec{j}_{Xeye} \\ \vec{k}_{Xeye} \end{pmatrix}$	Frontal perception sensor coordinate system
$\begin{pmatrix} \vec{i}_{Xtool} \\ \vec{j}_{Xtool} \\ \vec{k}_{Xtool} \end{pmatrix}$	Proximal perception sensor coordinate system
$\boldsymbol{x}$	State vector
$\boldsymbol{u}$	Control vector
$(\overline{x}, \overline{y})$	Coordinates of a detected crop in pixels
υ_c	Linear velocity of the platform $[m/s]$
ω_c	Angular velocity of the platform $[rad/s]$
$\mathcal{I}_{Y_1}$	Crops instance segmentation generated by YOLACT++
$\mathcal{I}_{Y_2}$	Weed instance segmentation generated by YOLACT++
$\mathcal{I}_{Y_s}$	Crop segmentation based on feature extraction
L_{r_i}	Reference line
$(\widehat{x}_{vp}, \widehat{y}_{vp})$	Vanishing point coordinates in pixels

L_s	Interaction matrix
$\boldsymbol{s}$	Visual features
$\boldsymbol{s}^*$	Desired values of the visual features
J	Model predictive control cost function

1

Robotics in agricultural environments

Chapter summary

Robots have been employed to transform many global industries in the last decades, they can effectuate tasks that human hands can not, allowing to save money and execution time. Just like humans, robots can execute tasks completely autonomous based on the perception of their environment. The present chapter discusses a general review of agricultural robotics. Unlike other sectors, agricultural robots operate in natural environments imposing great challenges. Application of these robots in digital farming has changed traditional agriculture, one of the main interests in digital farming is replacing pesticides by agricultural robots in order to eliminate weeds. To deal with this, two international competitions are described where this work took place. Besides, the contributions of this manuscript are presented.

Table of contents

1.1 Introduction

Agriculture provides the basis of subsistence for the population through the production of food and raw materials. From 1900 to 2022, the global population increased from 1.6 billion to 8 billion [1] and, according to the Department of Economics and Social Affairs of the United Nations (UN), the world's population could reach its peak at nearly 11 billion around 2100, demanding innovations in food production and technological advancements to provide farmers with tools and resources to make farming more sustainable. As a result of this demand, robots have been employed in the last years in agricultural tasks reducing cost-production, execution time and manpower.

Despite these robots can interact with humans to perform different tasks, the trend for the future is to provide them full autonomy. However, farm work is mostly done in outdoor environments where conditions are constantly changing, which implies robots must have a robust perception system to extract information of their environment. The perception system allows robots the ability to make decisions, as illustrated in Figure 1.1, so that, the better the perception, the better the robot's performance.

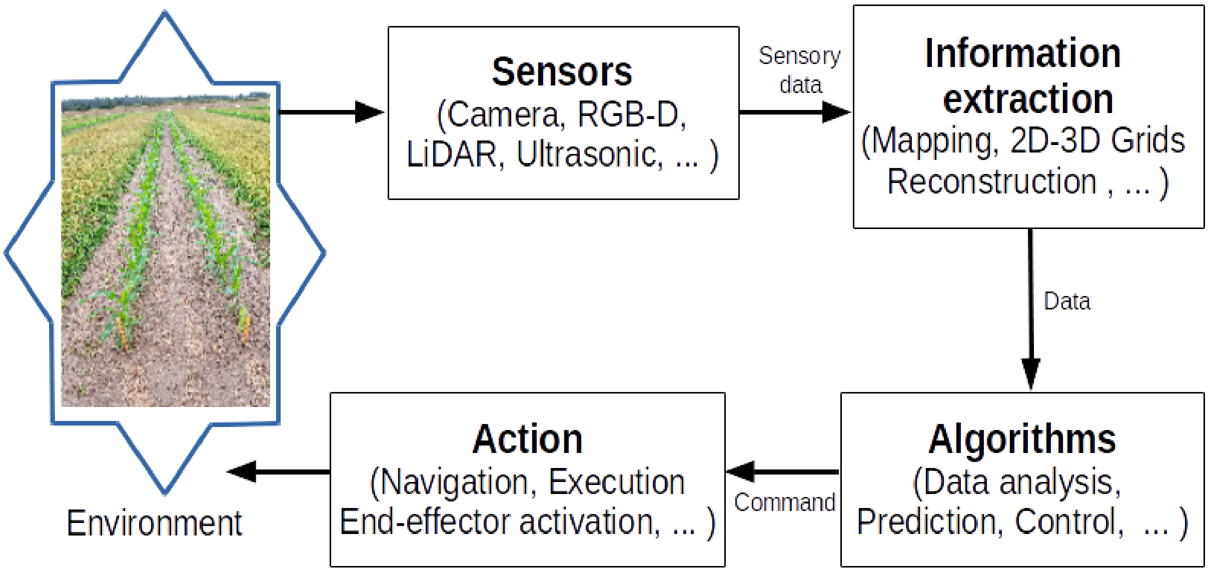

Figure 1.1: General scheme of the perception/decision/action loop of an autonomous robot.

The use of information technology has generated the so-called "Precision agriculture", which aims to optimize the resources needed to improve crop quality. To mention one example: in the pest control procedure, farmers apply the same rate of fertilizers in the field. However, some parts of the crops are less infected than others (which can be originated by the type of soil, weather conditions, etc.), generating waste of crops and pesticides. With the advent of precision agriculture, constant field monitoring can be done to provide data, indicating the most affected zones. Then, robots can effectuate optimal pesticide spraying or implement another mechanism to eliminate weeds.

1.2 Agricultural robotics: A general review

The following review presents the use of robots in agriculture in a general way without detailing the methods implemented for each application.

1.2.1 Soil preparation and planting

Planting is the process of putting plants into the ground, the classical way to execute this task requires human labor to manually insert each seed into the soil. Generally, this process is restricted by the distance between each plant to optimize its growth, which requires a high-precision level. Before performing planting, a previous soil preparation is carried out allowing to eliminate undesirable plants and to incorporate soil fertilizers to replace the nutrients necessary for the development of crops.

The commercial robot Cäsar [2] can fertilize the soil autonomously or remotely controlled. The vehicle is shown in Figure 1.2 - a), it is able to drive autonomously using stored route plans with an accuracy of 2 to 3 cm. Although the robot possesses an obstacle detector, the prerequisites for autonomous operation are: an unimpeded route within an enclosed fruit plantation and trained staff in the workspace, in case of an obstacle detection the robot makes an emergency stop, after the obstacle has been removed it is necessary to restart the vehicle.

Another alternative to execute arable farming is the Greenbot [3] (Figure 1.2 - b)), which is equipped with lifting devices, at the front of the robot there is a lifting device that can lift up to 750 kg and at the rear can lift up to 1500 kg. One of its main advantages is the four steered wheels increasing manoeuvrability, besides its user-friendly software allows to plan its own route and soil management independently.

In the same way, a variable-rate seeding is presented in [4] for corn planters. After performing laboratory analyses of soil samples and generating a map of the site, a 4-row pneumatic planter and a GPS receiver are attached to the tractor (Figure 1.2 - c)). The receiver reports the seeding position and the tractor's pose to the Android tablet, which matches with the previous map generated. The planter-unit controller adjusts the motor speed based on the calculations carried out by the tablet.

The development of autonomous seeding systems has been studied for several crops [5], [6], [7]. The soil digging process plays a crucial role to ensure successful seed planting because seeds require specific digging depth and the seeding systems must consider the soil moisture level to effectuate optimal planting.

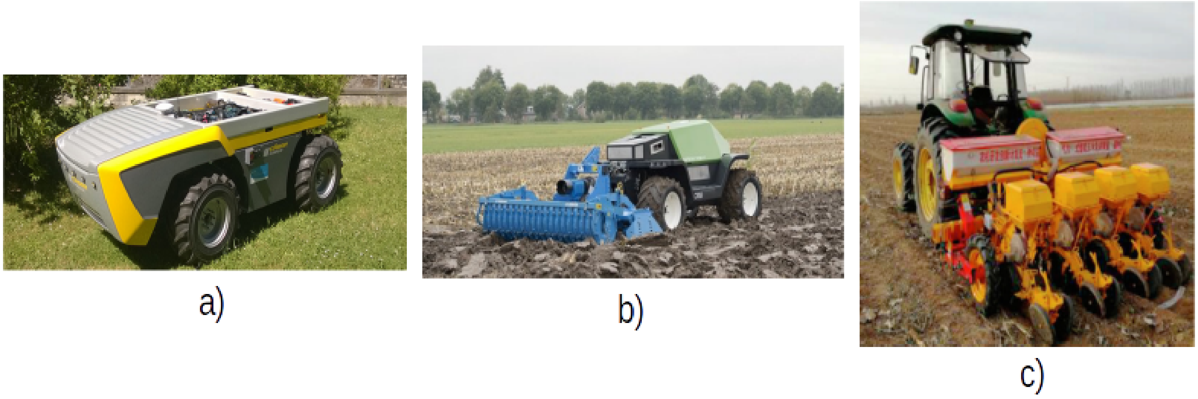

Figure 1.2: Examples of platforms employed in soil preparation and planting. a) Cäsar [2]. b) Greenbot [3]. c) John Deere tractor with pneumatic planter [4].

1.2.2 Plant monitoring and yield estimation

Plant diseases can kill or stunt plants having a direct impact in the world economy. For this reason, a continuous inspection process have to be implemented in order to identify the infection for implementing pesticide application to eradicate the source infection. Traditionally, farmers observe any abnormality by themselves and apply chemical products making this task highly time-consuming.

The plant growing stage is a factor that has to be considered during plant monitoring; crops in the early-middle stage present low density and small dimensions. To achieve adequate detection, a proximal visual sensing was proposed in [8]. An opaque box is used to act as a dark room (Figure 1.3 - a)), inside it, a rotating table allows to vary the leaf viewing angle. The acquisition system is composed of a multi-spectral camera, operating in the three spectral channels: green (540 nm), red (660 nm) and near infrared (800 nm). The sensing measurements carried out from different angles, rather than perpendicularly to the leaf's surface, improve the sensitivity from 9% at 0° up to 73% at 60°.

The Robot One (Figure 1.3 - b)) [9] is a multi-task agricultural robot able to detect and classify plants using a computer vision system, the stereoscopic depth cameras provide multiple data to the algorithms programmed in the robot allowing to produce predictive plant growth models considering the current growing stage and disease.

The use of multispectral sensors on board of unmanned aerial vehicles has been implemented recently [10], [11] as Figure 1.3 - c) shows, the purpose of these projects is to obtain quantitative data on a larger scale. The data collected by the sensors can be recorded for a future analysis [12], [13], [14] or computed in real time, sending the information to a base station which can estimate diverse parameters like plant disease, yield and biomass.

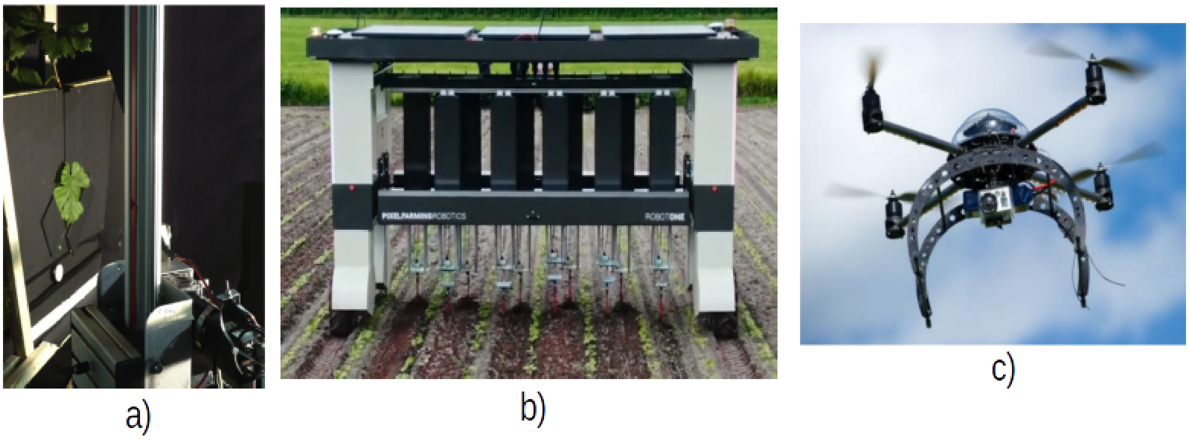
a) b) c)

Figure 1.3: Examples of platforms employed in plant monitoring and yield estimation. a) Proximal sensing platform [8]. b) Robot One [9]. c) Multispectral sensor on board an Unmanned Aerial Vehicle [10].

1.2.3 Spraying

Chemical substances can be sprayed on plants for pest-control, growth improvement or fertilizers. Generally, the spraying process in farms is applied uniformly, which generates waste of chemical products spread due to the crop infestation level is not the same. The design and construction of an autonomous robot for spraying pesticides is exposed in [15]. The spray system consists of a tank where the pesticide is stored, a pump and four lateral valves controlled by an onboard microprocessor as shown in Figure 1.4 - a). The robot's environment has to be known because its navigation is restricted to a steel pipes guidance system, reflecting markers placed on the ground are used to turn on and off the pump.

Auto Spray Systems commercialize the R150-ATJ electric platform [16] to help small-scale farmers to carry out spraying and pellet spreading. The twin rotating high-power nozzles can reach up to 12 meters spray and the container can store until 100 liters of chemical substances (Figure 1.4 - b)). The navigation system offers multiple autonomous modes: an integrated RTK with previous routes recorded and the "follow me" option, to make the robot follows an operator around the field.

With the increasing implementation of aerial vehicles nowadays, DJI designed an octocopter (AGRAS MG-1) for precision application of liquid pesticides and fertilizers (Figure 1.4 - c)). Like any other aerial vehicle, the main restriction is the operating payload, which is directly associated with the hovering time. A tank with a volume of 10 liters is fixed to the drone, the hovering time is around 12 minutes and the 4 nozzles allow a maximum spray speed of 0.5 L/min (per nozzle) [17].

A more extended review on advancements of spray technologies in agriculture is presented in [18].

Figure 1.4: Examples of platforms to perform spraying task. a) Spraying robot designed by [15]. b) Spraying system R150-ATJ [16]. c) DJI octocopter AGRAS MG-1 [17].

1.2.4 Harvesting

Harvesting is the process of picking and collecting crops. This activity can be divided in three steps: target localization, robotic arm manipulation and activation of the cutting mechanism. The first step implies a system recognition to identify the target; computer vision systems predominate over other approaches to solve this problem. Then, the robot arm has to move through crops without damaging them. Finally, the plant stem will be cut to collect the crop.

One of the most refined harvesting robots is the Agrobot robot [19], which is shown in Figure 1.5 - a). It is an autonomous strawberry harvestor. The flexible platform can be adapted to fit into any farming configuration, its 24 independent arms pick the fruit in a dedicated container. To isolate strawberries from the rest of the plant, a fusion algorithm between color and infrared depth sensors is implemented. Grasping without damaging the fruits has become the key difficulty. In the strawberry's case, its firmness can be an obstacle due to an erroneous end-effector pressure will damage the fruit. To deal with this, the Agrobot's harvest does not touch the fruit, the robotics arms grip and cut the stem.

Similarly, the design and control of an apple harvesting robot was proposed by *Zhao De-An et al.* [20]. The recognition vision system performs in real-time to detect and locate apples automatically, recognizing Fuji apples with a radius of 40 mm (which is the average radius of Fuji apples). A joint structure with multiple degrees of freedom was used as Figure 1.5 - b) shows. The end-effector includes a gripper and an electric cutting device, the average harvesting time is around 15 seconds per apple. The results report 39 apples recognized by the vision system, of which 30 apples were picked successfully, six apples failed to be picked due to the occlusion phenomena generated by the branches and the three remaining fell down to the ground as a result of apples were of a smaller radius.

The major adversity to carry out harvest task is occlusion because it can disturb two of the main stages: target localization and end-effector manipulation. *Boaz Arad et al.* developed a sweet pepper harvesting robot [21], illustrated in Figure 1.5 - c). To show the occlusion impact, the robot was tested in two scenarios: an idealized scenario with low occlusion removing leafs and no clustering interfering, and a commercial scenario without modifications. The success rates for fruit detection and fruit picked in the container were 97% and 71% respectively for the idealized scenario, whereas 79% and 49% respectively for the commercial scenario.

Some other propositions have been published to collect cucumbers [22], cherries [23], asparagus [24], tomato [25], just to mention some relevant works.

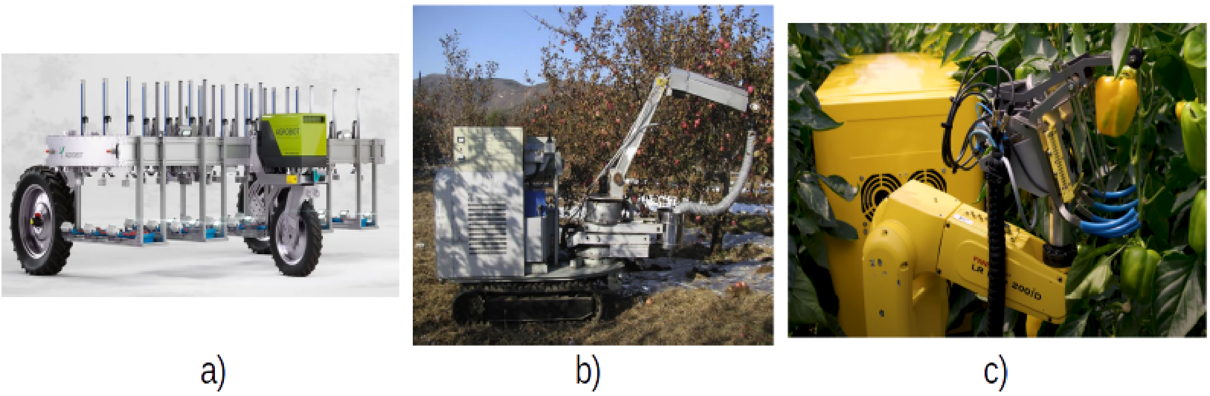

Figure 1.5: Harvesting platforms. a) Agrobot robot [19]. b) Apple harvesting robot [20]. c) Pepper harvesting robot [21].

1.2.5 Weeding

Weeding is the removal of weeds (any plant that grows in an unwanted place) from the field. For many years, herbicides have been used as weed control. However, extensive use of pesticides produces vapors that can become air pollutants, and also contaminates soil and water. Among the weed control methods, two can be highlighted: mechanical and chemical. The former refers to any technique that involves farm equipment, while the latter refers to any technique that involves the application of chemical substances.

Naïo Technologies develops and commercializes agricultural robots to perform autonomous mechanical weeding [26]. The electric Dino robot [27] uses a guidance system that combines the RTK-GPS data and visual information to achieve a navigation precision of 2 cm. The robot possesses multiple weeding tools like: kult parallelograms, V-shape hoe blades, L-knife hoe blades and Finger hoes (Figure 1.6 - a)).

In a similar manner, Farming Revolution [28] rents an electrical farming robot shown in Figure 1.6 - b). The robot's length can be adjusted to operate in many fields. The plant recognition system performs on up to 80 different plants (lettuce varieties, onions, maize, sugar, ...) and can distinguish small plants (from 1 cm). Its mechanical weeding tool executes intra-row (distance between two consecutive plants) and inter-row (distance between two rows of plants) weeding.

One of the initial works to develop an autonomous agricultural robot for mechanical weed control is presented in [29]. The robot has two vision systems: one to control the weeding tool and the other to recognize rows. To implement the plant classification in real-time, just few features were considered. The row recognition guided the robot with an accuracy of 2 cm.

In the same way, *Xiaolong Wu et al.* [30] proposed a multi-camera system for weed control, the weeds are detected and tracked across the camera system to estimate the timing to activate the end-effector (Figure 1.6 - c)). The system combines mechanical and chemical tools. For large weeds, a spray task is effectuated while small weeds are stamped mechanically.

1.3 Problem statement

Although weeds are part of the ecosystem and help to protect and restore soils, they affect crop growth reducing the amount of water, sunlight and nutrients absorbed by crops. Apart from damaging crops, weeds increase the production cost by employing more human labor and external resources. Therefore, weed control plays a

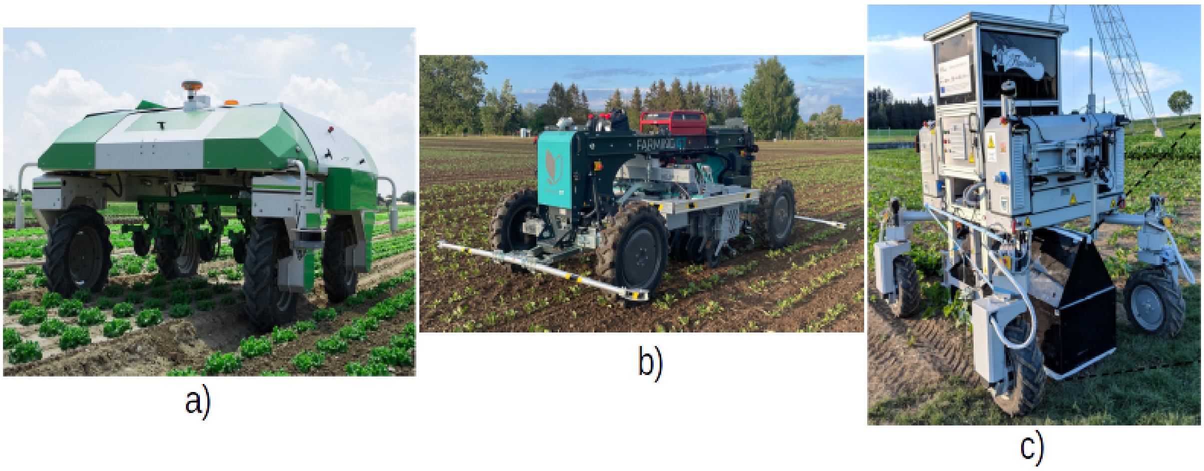

Figure 1.6: Weeding platforms. a) Dino robot (Naïo Technologies) [27]. b) Electrical farming robot of Farming Revolution [28]. c) Weeding platform proposed by [30].

crucial role in agriculture.

Approximately, 3.5 million tonnes of pesticides are utilized annually worldwide [31]; it is the weed control method most widely used in agriculture in recent years. Figure 1.7 is a map of the pesticide consumption by country. The top three pesticide consuming countries in the world are China, with an annual consumption of 1.76 million tonnes, followed by the USA and Brazil, with a consumption of 408,000 and 380,000 tonnes per year, respectively.

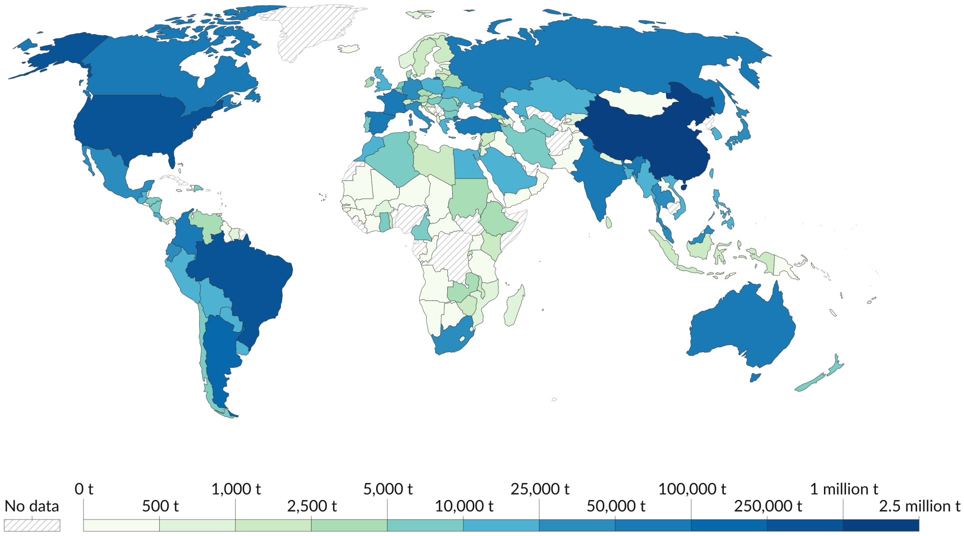

Figure 1.7: World map shows pesticide consumption by country [32].

Another important indicator is the normalization of pesticide application per hectare of cropland shown in Figure 1.8, which is more representative in quantitative terms. Due to China is the world's largest grain producer, it leads the list with an application of 13.07 kg per hectare. The USA are replaced in the second position by Argentina, with an average application of 6.07 kg/ha and Brazil completes the top three by applying 5.94 kg/ha.

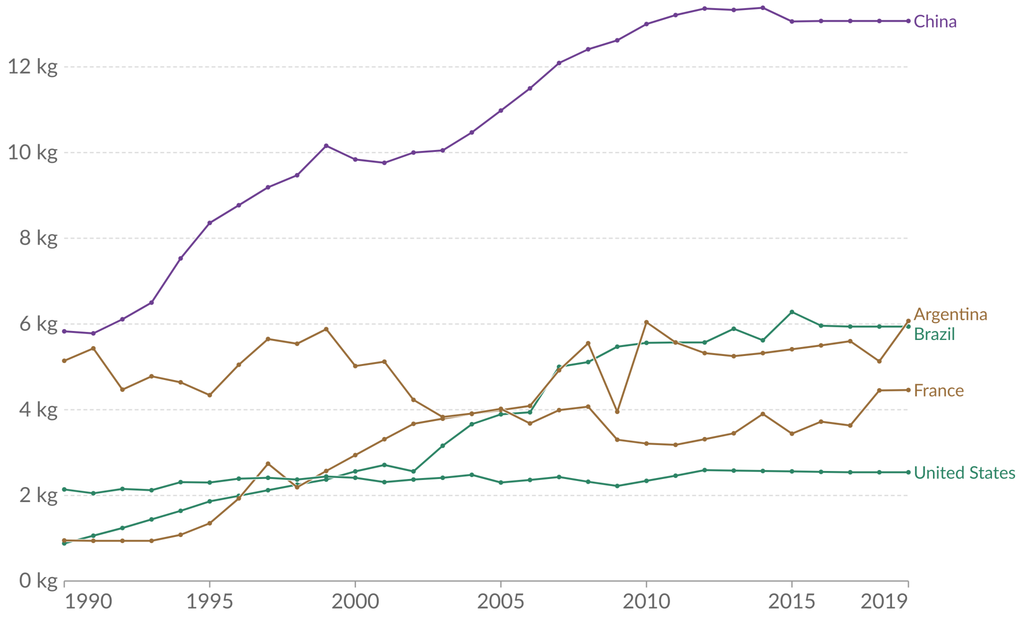

Figure 1.8: Average pesticide application measured in kilograms per hectare [32].

However, the application of these substances generates negative side effects on the ecosystem and human health [33]. Exposure to pesticides can be through inhalation, ingestion or contact with the skin, resulting in more than 26 million cases of non-fatal pesticide poisonings, about 3 million cases need hospitalization and approximately 220,000 deaths are reported [34]. Among the negative health effects associated with pesticides are: gastrointestinal, dermatological, respiratory, neurological and carcinogenic [35], [36].

On the other hand, the use of pesticides contaminates soil, water and they remain in the crops. Residues of pesticides can be found in everyday consumer products like fruit, wine, animal feeds. Moreover, washing and peeling food cannot completely remove the residues [37]. The spraying of these substances produces vapors that can become air pollutant and are dispersed to other sides.

The reduction of pesticide application is a pending subject in agriculture, and its eradication is the goal in the long term. Alternatives to conventional pesticides have been introduced recently like natural predators, biological control and hand weeding. These alternatives achieved good results on a lesser scale, but on a larger scale in more realistic agriculture conditions the problem remains open.

1.4 The ROSE Challenge

In 2017, the French Ministries of Agriculture and Ecological Transition in partnership with the French National Research Agency (ANR), launched the Robotics and sensors at the Service of Ecophyto (ROSE) challenge [38] to solve the technological problems identified by the agricultural community. The ROSE challenge aims to promote the development of scientific advances in the management of weeds and its organization is in charge of the National Metrology and Testing Laboratory (LNE) and the National Research Institute for Agriculture, Food and the Environment (INRAE).

The ROSE challenge focuses on intra-row weed control, which is the space between plants in the same crop line (see Figure 1.9). Multiple evaluation campaigns were organized in an open-field of the AgroTechnoPôle at Montoldre, France in order to measure the progress of each team.

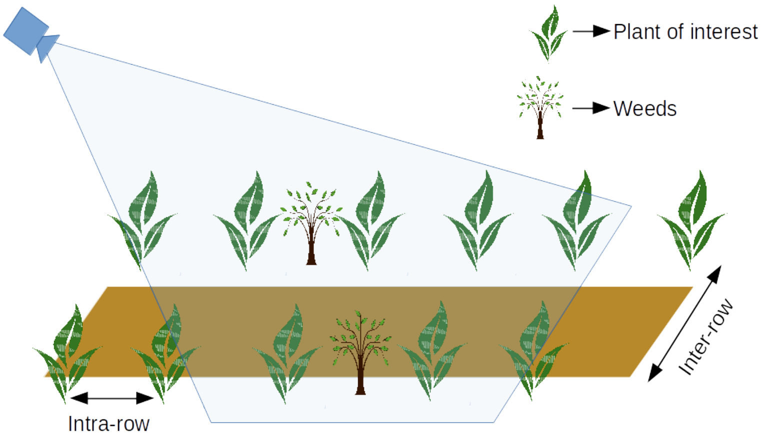

Figure 1.9: A typical parcel with intra- and inter-row distances.

The challenge is based on three main tasks:

- Detection of weeds and plants of interest (crops)
- The weeding action
- The overall intervention of the system

Regarding the detection task, the systems are evaluated on their ability to determine the position of weeds and plants of interest during a transition on the same parcel. For the weeding action, each team intervenes on the parcel to achieve the weeding task (weeds are located at indicated positions and easily visible by markers), the number and percentage of weeds destroyed is counted to measure the effectiveness of each solution. Finally, the overall intervention involves the entire chain: detection, decision and actuation. A reference parcel is utilized to compare the results of the global evaluation, comparing the results between the proposed systems with the results obtained in the reference parcel using conventional weed treatment.

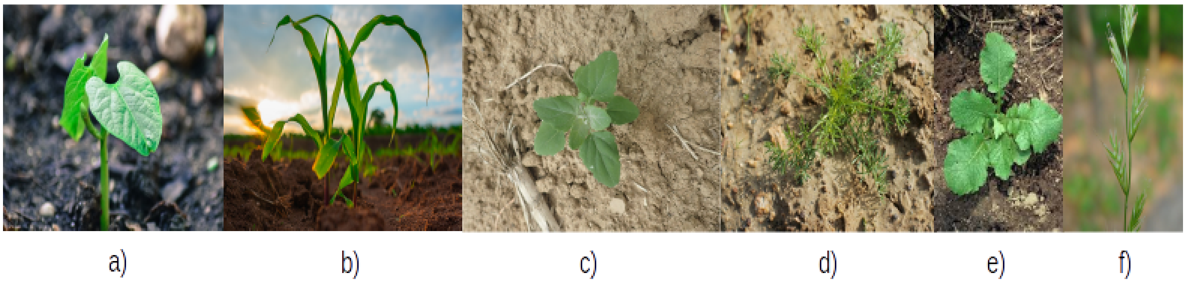

Figure 1.10: Plants in the ROSE challenge. Plants of interest: a) Bean and b) Maize. Weeds: c) Chenopodium, d) Matricaria, e) Moutarde, and f) Ryegrass.

The plants of interest are:

- Maize: presenting an inter-row distance of 75 cm and an intra-row distance of 15 cm. Two rows of crops are planted on the parcel with a length of 46.5 m. (see Figure 1.11).
- Bean: presenting an inter-row distance of 37.5 cm and an intra-row distance of 7-8 cm. Three rows of crops are planted on the parcel with a length of 46.5 m. (see Figure 1.11).

The weeds utilized during the ROSE challenge Matricaria, Chenopodium, Moutarde and Ryegrass. Crops and weeds are illustrated in Figure 1.10

1.5 The ACRE competition

ACRE (Agri-food Competition for Robot Evaluation) is one of the autonomous robot competitions organized by the METRICS (Metrological Evaluation and Testing of Robots in International Competitions) project [39]. The robot performance is tested in real-world agricultural scenarios to provide objective and repeatable results, promoting the transition from traditional agricultural practices to Agriculture 4.0.

Like the ROSE challenge, the ACRE competition was conducted on INRAE AgroTechnoPôle fields at Montoldre, France. An experimental field of 4 hectares is divided in numerous parcels and put at the disposal of participants. Similarly, the plants

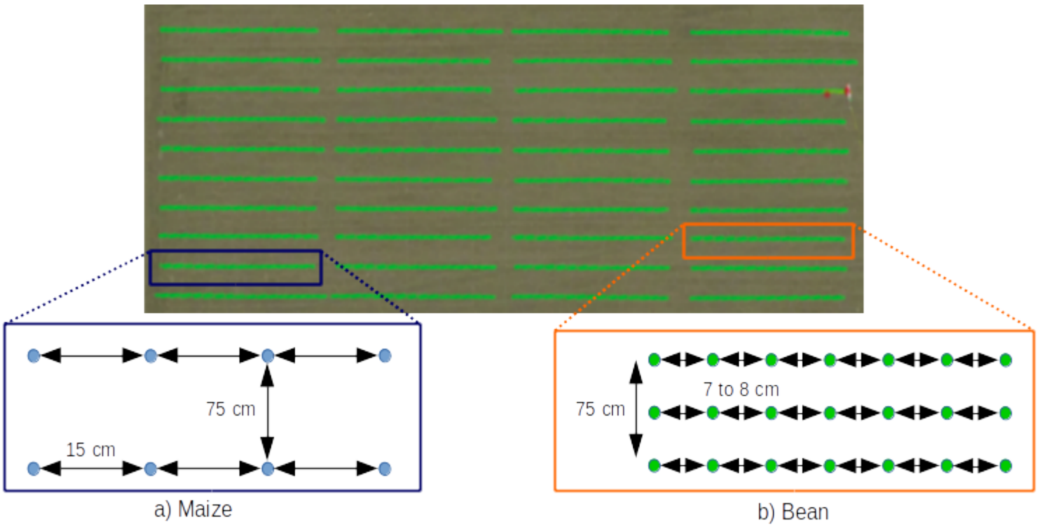

Figure 1.11: Experimental field AgroTechnoPôle and crop parcel examples (Maize and Bean).

of interest and weeds are the same to those used in the ROSE challenge. The ACRE competition involves a set of benchmarks to be performed listed below:

- Plant discrimination: To differentiate crops from weeds in the parcel. The evaluation compares plant classification performance by the robot with ground truth (human expert).
- Field navigation: To cross and move between parcels without damaging crops (see Figure 1.12). The quantity of crop damaged and the time to complete the task are evaluated.
- Leaf area estimation: To estimate the leaf area of the plants. The results obtained by the robot are compared with the ground truth, in the same way as crop differentiation.
- Weed destruction: For intra-row weeding without damaging crops. A parcel with visual markers on crops and weeds is analyzed before and after the robot executes weeding, the platform can be driven remotely.
- Biomass estimation: To estimate the above-ground biomass. The estimation given by the robot is compared to the crop's weight, to know the precision of each solution.
- Intra-row weeding: Which is similar to weed destruction but for this benchmark, the robot has to carry out the mission autonomously (weeding and navigation) and the visual markers are removed from the parcel.
- Crop mapping: To generate a map of the crops. A mapping error is computed as the discrepancy between the map generated by the robot and the ground truth map.

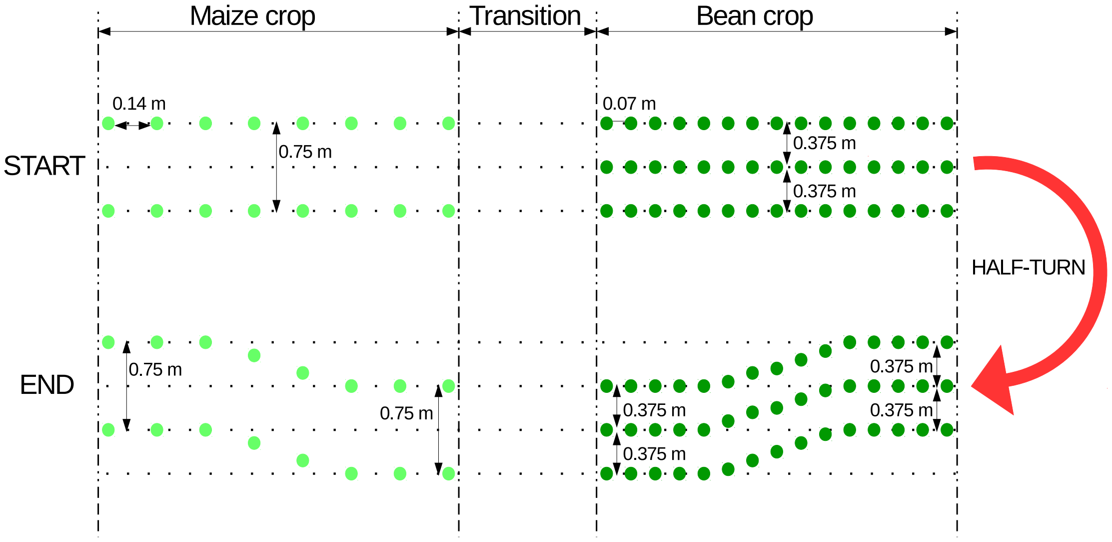

Figure 1.12: Parcel specifications for the field navigation benchmark.

1.6 Contributions

This thesis aims to propose an autonomous robotic system to carry out mechanical weeding. The focus on intra-row weeding represents a great challenge due to the compact distance between consecutive plants. The inter-row weeding is solved using a mechanical mechanism attached in front of the platform. A general architecture is presented in Figure 1.13. Although this architecture will be detailed in the following chapters, in a very general manner the proposed solution can be described as a robotic platform equipped with mechanical tools to carry out the weeding task and two multi-modal sensors developed for this project. The first multimodal sensor was fixed in front of the platform to execute autonomous navigation and the second multimodal sensor was attached under the platform to manipulate the weeding tools.

Global task:

* The main contribution of this work is an autonomous mechanical solution to perform inter and intra-row weeding. To achieve this goal, multiple tasks have been accomplished. We propose and develop perception systems and strategies dedicated to different purposes.

Detection task:

* Two methods to distinguish crops and weeds are proposed. The first is based on image processing techniques using crop feature extraction and the second is learning-based classification. The use of synthetic images in this work, allowed to reduce the number of images needed to train the learning-based method. To create the synthetic

images, a sample of the ground truth collected images was utilized.

Both methods perform well for crop/weed detection. However, each has some drawbacks. Using the first method, some weeds remain after the segmentation and the classification given by the second method generates some inconsistent masks.

* To leverage both approaches, a third detection method is proposed that consists of the fusion of the first two methods. This new method uses the strengths of each algorithm to improve crop/weed segmentation, solving the leakage mask issue of semantic segmentation of plants and weeds under natural dynamic conditions.

Each one of the previous methods is able to be executed in real time on an onboard system.

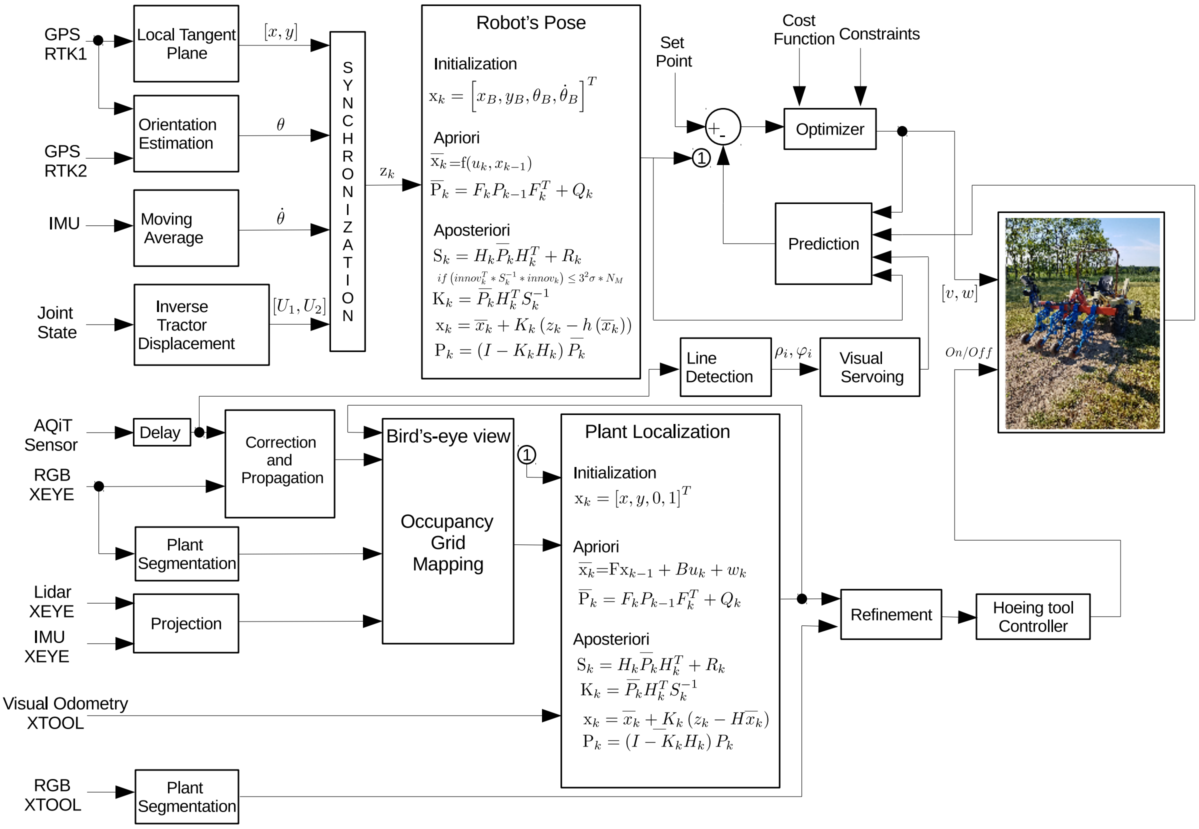

Figure 1.13: Autonomous robotic system proposed in this thesis.

Navigation task:

Having solved the crop/weed segmentation, this information is utilized for autonomous navigation and autonomous mechanical weeding.

* A robust algorithm to navigate using visual perception is presented. The proposed approach extracts the crop lines from the previous crop segmentation to estimate the vanishing point. The vanishing point can be computed for any number of crop lines.

However, the vanishing point approach can be used when crop detection is available. In the absence of crops, other alternatives are presented.

* A navigation strategy based on a bird's-eye view is proposed to guide the platform in the absence of crop detection. This approach segments vegetation to navigate in the transition zones. Similarly, a model predictive control is presented to navigate without visual perception.

Weeding task:

* Finally, to perform the mechanical weeding, two methods are proposed using the crop segmentation previously defined. The first one utilizes the proximal detection to activate the weeding tool and the second one propagates the detected crops given by the frontal detection. To perform crop propagation, a Kalman filter merging multiple sensors operating at different frequencies to improve the pose estimation of the robot is presented.

Validation task:

* To validate each one of the previous contributions, multiple realistic scenarios in the robotics simulation environment Gazebo were created using digital twins of the real platforms. The conditions of the experimental field were recreated in these scenarios, besides other challenging situations.

1.7 Thesis outline

To carry out an autonomous weeding, several tasks have to be done, among them: crop and weed detection, safe navigation and effective weeding tool actuation as shown Figure 1.14.

The thesis is organized as follows:

Chapter 2 - Visual plant detection and segmentation: this chapter is devoted to crop and weed detection. The proposed algorithms employ image processing techniques and artificial intelligence. Their low computation cost allows to implement them in real-time onboard an embedded system, improving some shortcomings reported in the literature.

Chapter 3 - Autonomous navigation based on visual perception: the crop detection method introduced in the Chapter 2 is used in this chapter to navigate through

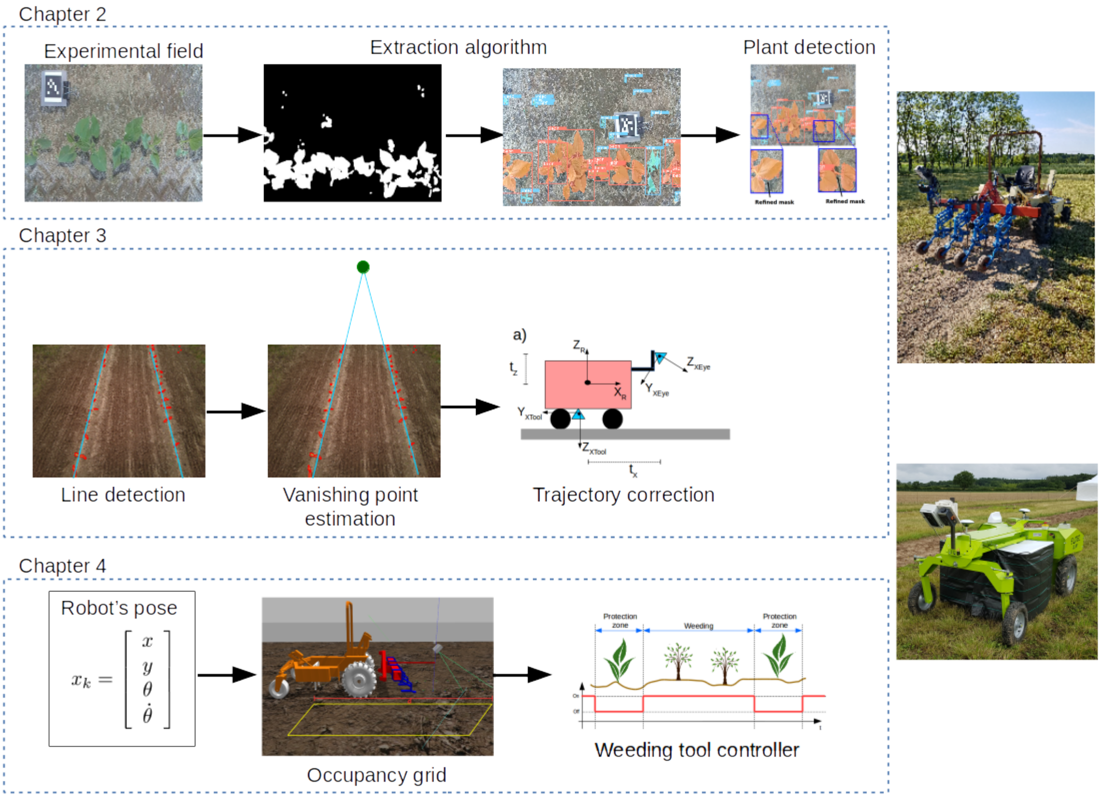

Figure 1.14: Thesis outline

the experimental field without relying on GPS. The autonomous navigation based on vanishing point estimation allows to cross the parcel, transit between parcels (consecutive parcels separated by a transition zone) and change of parcel (half-turn to change non-consecutive parcels).

Chapter 4 - Weeding actuation and robot localization: weeding action requires the highest possible accuracy to eliminate weeds without damaging crops. In this chapter, the weeding strategies employed in this work are exposed. They are aimed at an intra-row weeding task, which is more complicated than inter-row weeding.

Multiple simulation scenarios are utilized to validate the proposed approaches. In the same way, the obtained experimental results are presented in each one of the previous chapters.

Chapter 5 - Conclusions and outlook: the conclusions and the outlook of the present work are exposed.

1.8 Conclusion

The review presented in this chapter shows the multiple sectors where agricultural robots have been employed. Although there are some commercial and academic propositions, it is evident that a lot of work still has to be done. Each robot possesses different systems to execute the tasks for which it was conceived. However, this chapter concludes that there are two in particular that are linked and play a crucial role in agricultural robotics: perception and navigation systems.

The conditions in agricultural environments are always changing and the implementation of robust algorithms to extract information is imperative. On the other hand, most navigation systems rely on GPS technology or they are limited to a rail system which imposes restrictions on the robot's workspace. A poor performance of one of these systems is going to impact the manipulation of the robot's end-effector (for whatever purpose it may serve).

The presentation of the ROSE challenge and the ACRE competition allows to encourage the robotics community to substitute the use of pesticides by autonomous weeding robots. Especially, the intra-row weeding requires a high precision performance to eliminate weeds and protect crops. Generally, it is carried out mechanically, but there are other alternatives like laser-weeding. The smaller the intra-row distance, the faster and more robust the algorithms for plant detection and the weeding tool have to be.

The next chapter will address crop and weed detection, which is the first stage in the autonomous weeding task.

2

Visual plant detection and segmentation

Chapter summary

In this chapter, an algorithm to detect and improve plant segmentation is exposed. This detection will serve to navigate and to activate the weeding tool. The proposed algorithm is based on the fusion of a fully connected network (YOLACT++) and an image processing strategy that relies on plant features, it compensates for the shortcomings of each method performing independently. That is, the image processing strategy fulfils plant segmentation by generating finely refined masks; however, some weeds are segmented incorrectly. On the other hand, YOLACT++ performs well on weeds and plants, but when it estimates inaccurate bounding boxes, imperfect masks are generated. The algorithm can be executed in real-time on an onboard computer generating crop and weed segmentation masks and can be extended to multiple types of crops and weeds. To validate the performance of the algorithms, multiple experiments were conducted in real agricultural conditions described in the introduction chapter.

Table of contents

2.1 Introduction

Agricultural robots aim to optimize efficiency in diverse tasks and can be automatized to execute autonomous missions. A key requirement in agricultural robotics is the ability to detect plants for any subsequent treatment, like harvesting and picking, thinning and pruning, weed control, chemical application, among others.

The use of visual sensing technology in agriculture for crop and weed detection is one of the tasks that has rapidly gained popularity in companies as well as research projects in recent years. The knowledge of weed infestation is a critical process to prevent crop damage, impact on human health and soil pollution. Moreover, its impact on biodiversity is incalculable. On the other hand, crop detection is part of a new concept called 'precision agriculture', that aims to improve crop production, reduce working time, and control irrigation management employing data from multiple sources.

The fact that agriculture is carried out in outdoor environments imposes a substantial number of factors to be considered. The first and most important for a visual perception system is the unpredictable and constant illumination changes, which are linked to the weather conditions. Then, the kind of crop/weed to be detected. Each plant has different physical properties that can help to recognize them easily. Another factor is the type of ground which can contrast or can be homogeneous complicating the crop/weed detection.

To overcome some of the factors previously described, an algorithm to carry out plant/weed detection and segmentation based on the fusion between artificial intelligence and image processing techniques is presented. In the following chapters, this detection will be used to navigate in the parcel and to control the weeding tool to eliminate weeds without damaging crops. A review of different crop/weed detection methods is exposed at the beginning of the chapter, which helps to identify the problems to be solved. Then, the proposed algorithms are explained and finally, the results obtained are shown.

2.2 A review of different methods for crop and weed detection

Crop segmentation plays a major role in agriculture because a large number of agricultural tasks depend on efficient segmentation such as irrigation, density estimation, fruit pruning, disease detection, weeding, among others.

Studies for crop and weed detection have been performed using computer vision techniques, mainly including traditional image processing and artificial intelligence. The next subsections exhibit a general review of the implemented methods for crop/weed detection.

2.2.1 Crop detection

Within the traditional image processing techniques, different indices have been studied to discern between crops and soil background [40],[41],[42]. Color indices present less sensitivity to the lighting variations and the main advantage is that they can accentuate a particular color like green vegetation.

The normalized difference vegetation index (NDVI) is one of the most commonly used remote sensing index for monitoring global vegetation [43]. It is defined as the difference between near-infrared and red reflectance divided by their sum.

$$NDVI = \frac{NIR - RED}{NIR + RED} \tag{2.1}$$

The value of the $NDVI$ will fall between -1 and +1. Values between -1 and 0 indicate dead plants or objects that contain no carbon, while values for live plants range between 0 and 1 (see Figure 2.1). Considering that NDVI is an indicator of the plant's health, a study to predict ground vegetation biomass in an open agricultural area was conducted in [44]. Weather effects like snow reflectance could perturb data collection; to avoid this problem, authors collected data throughout spring and summer.

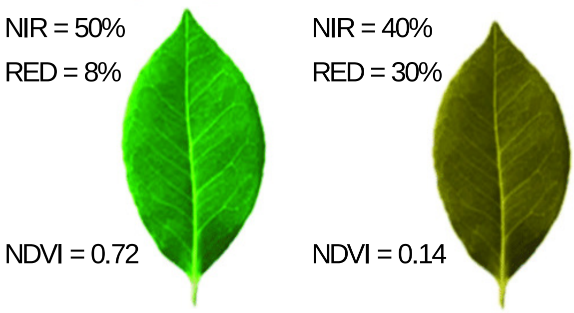

Figure 2.1: Normalized difference vegetation index. Left: Reflectance of healthy vegetation. Right: Reflectance of stressed vegetation.

Thresholding techniques have been applied for plant segmentation. The thresholds can be dynamic or constant. The main disadvantage of these techniques is the difficulty in choosing the proper threshold values [45],[46],[47]. An inadequate value will segment erroneously, that is, a high value will merge crop pixels with other regions which leads to under-segmentation, while a low value may lead to over-segmentation.

Despite the main interest of this review focuses on visual techniques, it is important to mention that there are other alternatives to distinguish crops and weeds. *Andujar et al.* [48] detect and discriminate maize plants and weeds from soil using the light detection and ranging (LIDAR) technology. However, this solution depends on the material of the scanned surface, on the angle of incidence and on the distance at which the LIDAR is located. For these reasons, this kind of sensors can perform well to separate crop/weeds from the ground, but their use for precision agricultural tasks is limited.

A combination between semantic segmentation and various vegetation indices was proposed in [49] for crop classification using RGB images. The results indicate that the proposed solution can operate online at around 20 Hz with precision and recall values of 90%.

2.2.2 Weed detection

In a similar way, it is possible to extend the techniques presented in the previous section to detect weeds [50], [51]. Even so, applying those techniques, crops and weeds are still mixed. For this reason, additional processing has to be considered to discern between them.

In general, the propositions that consider hyperspectral images are based on reflectance and further analysis to classify weeds [52]. The use of spectral features with different leaf colors represent a good indicator to classify crops and weeds. *Cheng et al.* [53] used the mean and standard deviation of each RGB color to distinguish rice plant from euphorbia weeds. The fact that weeds tend to be brown while rice plants are green allows to implement spectral features and the results obtained are promising.

Plant leaves and weeds have diverse textures generated by their veins distribution. Different approaches have been proposed to consider texture as weed discriminator. Wavelet texture features [54] is one of the most employed image processing techniques to extract information from images due to wavelet image decomposition provides a representation that is easy to interpret. This technique transforms images into a representation in which both spatial and frequency information are present [55].

In the same manner, shape features have been successfully applied in weed recognition. Segmented regions in images are compared with previous knowledge of perimeter, diameter, minor and major axis length and area [56], [57]. Other approaches implement descriptors to extract pixel information [56], the mathematical representation of the feature descriptor is a vector with one or more dimensions. It describes elementary characteristics such as color, shape, texture, etc.

2.3 Crop and weed detection based on machine learning

The works presented in sections 2.2.1 and 2.2.2 were one of the first approaches to solve crop/weed detection in agriculture. Their implementation can perform well under certain conditions, but in attempts to extend these solutions to adverse scenarios (irregular illumination, scale factor, occlusion, etc.), learning-based models have been developed. Although these algorithms are more robust, a significant cost of training datasets is the main disadvantage.

In general, a machine learning process is carried out by collecting raw data of the application to be tested. A global description is illustrated in Figure 2.2. Training data is used to make sure the machine recognizes patterns of the data, while validation data injects new data to the model that has not be evaluated before. After the model is built, it is tested once again to validate that it can make accurate predictions, using techniques of cross-validation for checking the model performance. Finally, visualization techniques are used to classify.

The crop and weed classification problem based on their appearance using convolutional neural networks (CNN) and deep learning has been proposed by *Milioto et al.* [58]. A preprocessing to separate vegetation from soil was carried out to only focus on vegetation, followed by a blob segmentation to extract single crops and weeds. However, this proposition is unable to classify when overlapping is present. For this reason, the training was done with crops and weeds at an early growth stage.

Recently, the interest in detecting each plant in an image has incremented due to the potential to isolate each plant for a particular future application. In the machine learning field, instance segmentation allows to achieve this purpose. The use of a Mask R-CNN architecture was presented in [59], the readjusted of the hyperparameters improve the detection but there are some issues to be solved. In the case of weeds, the probability of a false detection increases when small weeds have to be detected. For the training, the crops and weeds are manually segmented and identified by human experts.

The image segmentation algorithms generally produce incomplete masks. In particular, plant segmentation generates deformed masks with some broken leaves. A maize leaf segmentation system under different light conditions was proposed by *Wang*

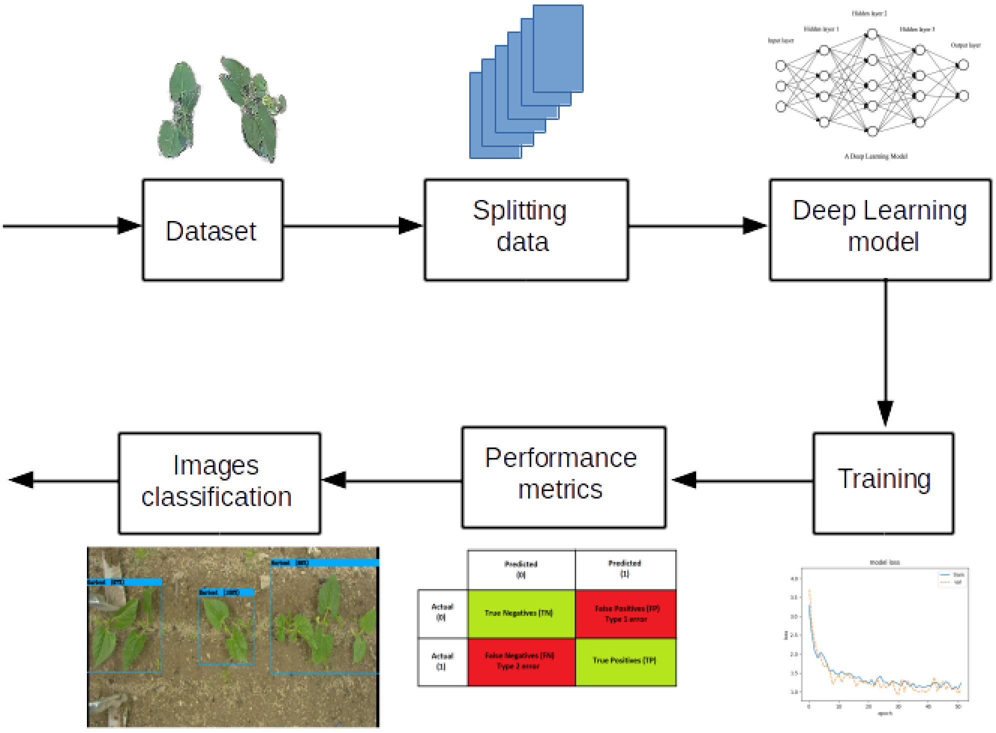

Figure 2.2: Diagram of a general learning-based model.

et al. [60]. To segment the images, a support vector machine was utilized allowing to determine if the pixel belongs to the foreground or not. Then, a maize structure analysis is used to determine the leaf reconstruction area.

A more detailed survey of artificial intelligence and machine learning methods in agriculture is reported in [61].

2.3.1 Plant detection learning-based model

Machine learning is a branch of artificial intelligence (AI) that allows computer systems to infer patterns from data. Machine learning detection models require two fundamentals:

- Data: corresponds to the examples given to the models from which the algorithm will learn.
- Learning algorithm: is the process that, using the previous data will generate a model. In other words, a learning algorithm models a phenomenon from examples.

Both the data and the learning algorithm are strongly linked. On the one hand, the learning algorithm will generate an erroneous model if the data is inconsistent. On the other hand, the performance of the model created with an inefficient algorithm will be unsatisfactory. In machine learning there are three types of learning methods:

1. *Supervised learning*: The model infers the mathematical mapping from inputs to outputs. The training dataset is entirely labeled.
2. *Unsupervised learning*: The model is fed with unclassified training data. The model tries to find similar features between test data to categorize them.
3. *Semi-supervised learning*: The model acquires from both supervised learning and unsupervised learning. The model aims to infer the correct labels for the given unlabeled data.

2.3.2 Convolutional Neural Network structure

Convolutional neural networks (CNN) are inspired by biological processes trying to emulate the neural connectivity of the brain's visual cortex. The overall components of a common CNN are illustrated in Figure 2.3.

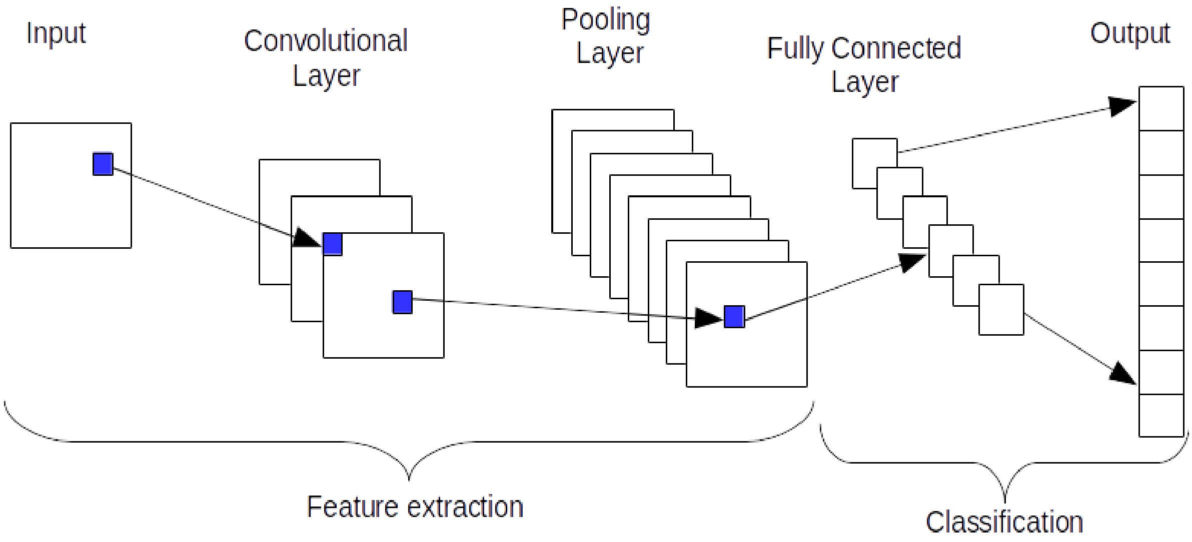

Figure 2.3: A compact representation of a Convolutional Neural Network (CNN)

Essentially, a CNN is composed of three layers [62]. The convolution layer, the pooling layer and the fully connected layer. The convolution layer performs the first activities of the CNN. In this step, multiple mathematical convolutions are carried out between the input image and a sliding window (kernel) to generate feature maps. The convolution task is executed as many times as required.

The pooling layer aims to reduce the size of the feature maps. This is performed by decreasing the connections between layers. Among the techniques to execute this step, Max Pooling is an operation that selects the maximum element from the region of the feature map. The output of the feature map has to be another feature map with smaller dimensions but containing the most relevant features.

The weights of the different layers are varied until the network is able to recognize data. The fully connected layer is directly connected to every node in the previous and in the next layers.

When data is processed by the network, it passes through a series of layers. This is referred to as forward propagation. The network's depth refers to the number of layers within the network. Each layer l possesses a number of nodes, which is known as the layers' width.

The input data a is multiplied by a weight w and a bias b is added to the result

$$z_j^l = \sum_k w_{jk}^l a_k^{l-1} + b_j^l \tag{2.2}$$

where j is the output node and represents the j-th node from the l-th layer, k represents the k-th node from the previous layer and w_{jk}^l is the weight relationship between the two nodes from both layers. w_{jk}^l and b_j^l are randomly initialized.

The weighted input z_j^l for the j-th node in the l-th layer, is fed to an activation function f:

$$a_j^l = f\left(z_j^l\right) \tag{2.3}$$

The activation function f produces an output within a specific range in order to activate the node or not. The values generated by the activation network depend on the type of activation function which is used. Some of the more common activation functions are:

The sigmoid function, which output values range between 0 and 1:

$$f\left(x\right) = \sigma\left(x\right) = \frac{1}{1+e^{-x}} \tag{2.4}$$

The hyperbolic tangent, which output values range between -1 and 1:

$$f\left(x\right) = \tanh\left(x\right) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.5}$$

The Softmax function, which output values range between 0 and 1:

$$f_i\left(\vec{x}\right) = \frac{e^{x_i}}{\sum_N^{j=1} e^{x_j}} \tag{2.6}$$

2.3.3 YOLACT (You Only Look At Coefficient)

The first approach presented in this work to solve crop and weed recognition is based on a fully-connected network, hence the interest in exposing the fundamentals of a CNN in the previous subsection.

Crop and weed classification can be considered as the process of treating an input image and generating a class for each plant and weed in the image. Besides classification, detection provides not only classes of the crops/weeds in the image, but also provides bounding boxes of the classified crops/weeds to know their localization. In the same way, semantic segmentation aims to obtain fine inference by predicting labels. Each pixel is labeled according to the same object class. Finally, instance segmentation allows to separate instances of objects belonging to the same object-class. Figure 2.4 illustrates the preceding explanation.

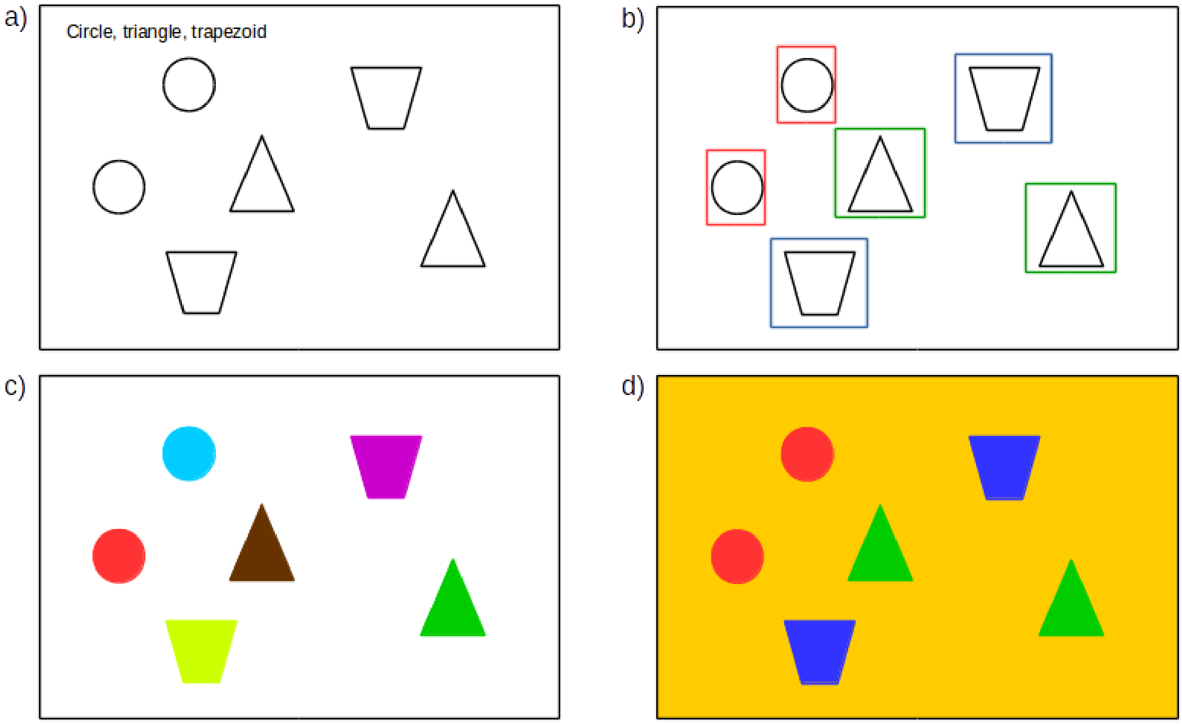

Figure 2.4: Object recognitioairfrn: a) Image classification. b) Object detection. c) Instance segmentation. d) Semantic segmentation.

The present work introduces YOLACT, a real time instance segmentation [63]. Figure 2.5 shows the YOLACT architecture, which is based on a fully-convolutional model. To generate an image with semantically segmented elements, a linear combination is performed between the prototype masks, the predictive coefficients and then crop with a predicted bounding box. The YOLACT approach can perform in real-time

(30 FPS with ResNet-101 convolutional network [64]) due to its parallel architecture and the masks have a higher quality because it considers the full extended image space without loss of quality generating by pooling.

The main goal of YOLACT is to integrate a mask branch to an existing one-stage object detection model. To do this, the instance segmentation task is divided into two parallel tasks. The first task produces a set of image-sized (prototype masks) that do not depend on any instance. The second task adds an extra head to the object detection branch to predict a vector of mask coefficients for each anchor that encode an instance representation in the prototype space as Figure 2.5 shows.

Finally, the anchors are reduced to a single bounding box per instance using the non-maximum suppresion method and a mask is built for each instance by combining linearly the mask coefficients for each instance and the prototype masks. The final mask is cropped using the predicted bounding box and then encoded into binary values using a threshold.

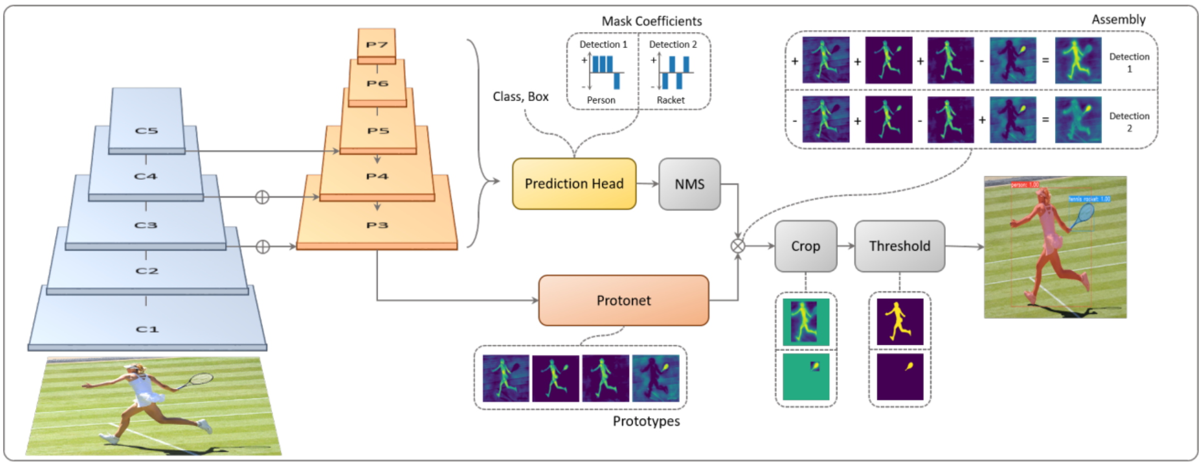

Figure 2.5: YOLACT architecture. Blue/yellow indicates low/high values in the prototype, gray nodes indicate functions that are not trained [63].

YOLACT++ [65] is based on the general structure of YOLACT, a fast mask re-scoring network is added to improve the detection and mask quality. Further, deformable convolutions [66] are set in the backbone. Unlike the YOLACT model, this network is added after the cropped mask and produces the IoU (Intersection over Union (IoU) is used to evaluate the performance of object detection by comparing the ground truth bounding box to the predicted bounding box) mask for each object category. Then, each mask is rescored by considering the product of the predicted IoU mask for the category and the corresponding classification confidence. Deformable convolutional networks increase sample space locations with additional offsets by learning them without additional supervision. This network is added to ResNet at the 3×3 convolution layer without using modulated deformable modules.

2.4 Plant detection based on physical characteristics

This approach aims to detect crops (plants of interest) based on easy-to-extract features without implying machine learning techniques. Any plant can be classified into two categories, either as monocotyledon or dicotyledon, commonly referred to as monocot or dicot. Based on the number of cotyledons present in the seed, each category possesses different physical properties, e.g., monocot leaves are narrow with parallel veins and longer than dicot leaves, while dicot leaves are usually curved with interlace veins as shown in Figure 2.6. A relevant feature to characterize plants is the color, various chromatic coordinate indices have been studied [67], [40] in order to distinguish living plants from soil.

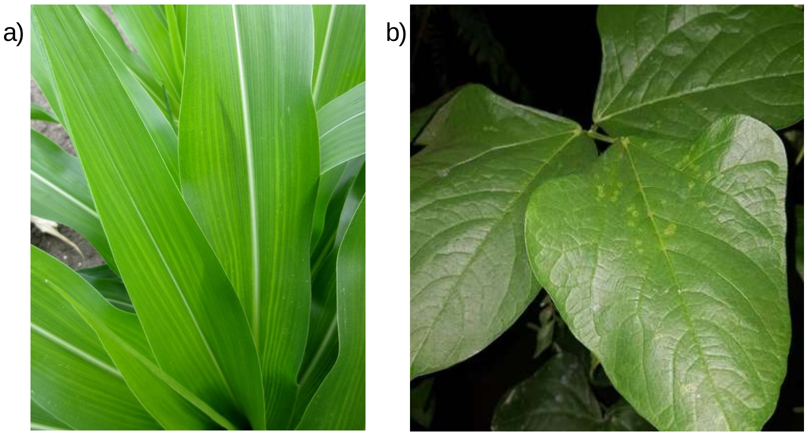

Figure 2.6: Example of monocot and dicot leaves, respectively.

The first stage of the plant classification algorithm is to distinguish vegetation (weeds included) from the background, which is highly dependent on image quality. Image acquisition is influenced by diverse factors including:

- Weather conditions: weather may have a direct or indirect substantial effect on plant identification, cloudy sky during image sensing will reduce natural illumination, decreasing image quality directly. Alternatively, plant growth stage under adverse climatic conditions tends to affect crops.
- Image acquisition system: camera parameters and position have to be chosen depending on the application, e.g., a high fps rate on board a platform moving very slowly becomes inefficient and computationally costly.

- Soil conditions: soils are grouped depending on their chemical composition and their color can be an indicative of certain soil types, for instance, Peat soils are dark while Chalk soils have a white layer of dust.

Different chromatic indices were tested based on the fact that vegetation has larger green indices than others, the Excess Green vegetation (ExG) presents the best performance (see Figure 2.7) and is defined as:

$$ExG = 2g - r - b \tag{2.7}$$

The others tested chromatic indices are:

$$CIVE = 0.441r - 0.881g + 0.385b + 18.78745 \tag{2.8}$$

$$GB = g - b \tag{2.9}$$

$$ExR = 1.4r - g \tag{2.10}$$

$$ExG - ExR \tag{2.11}$$

where

$$r = \frac{R}{R+G+B}\ , g = \frac{G}{R+G+B}\ , b = \frac{B}{R+G+B}$$

with R, G, B are respectively the red, green and blue components of the input images. The resulting near-binary image must provide leaf information to discriminate between crops and weeds.

Due to the fact that agricultural tasks are performed mostly in outdoor environments, it is crucial to deal with changes in illumination. One way to mitigate this adverse effect is to convert the resulting normalized image to hue, saturation, and value (HSV) color space. HSV channels are described as follows:

- Hue: is related to the wavelength of light and represents a pure color. Pure values contain primary colors, secondary colors and mixtures between adjacent pairs of them.
- Saturation: represents color purity. It gives a measure of how much the true color is diluted by white. Decreasing S value, increase whiteness.
- Value: is analog to brightness. Decreasing V value, increase blackness.

The HSV color space is a different representation of the RGB model (Figure

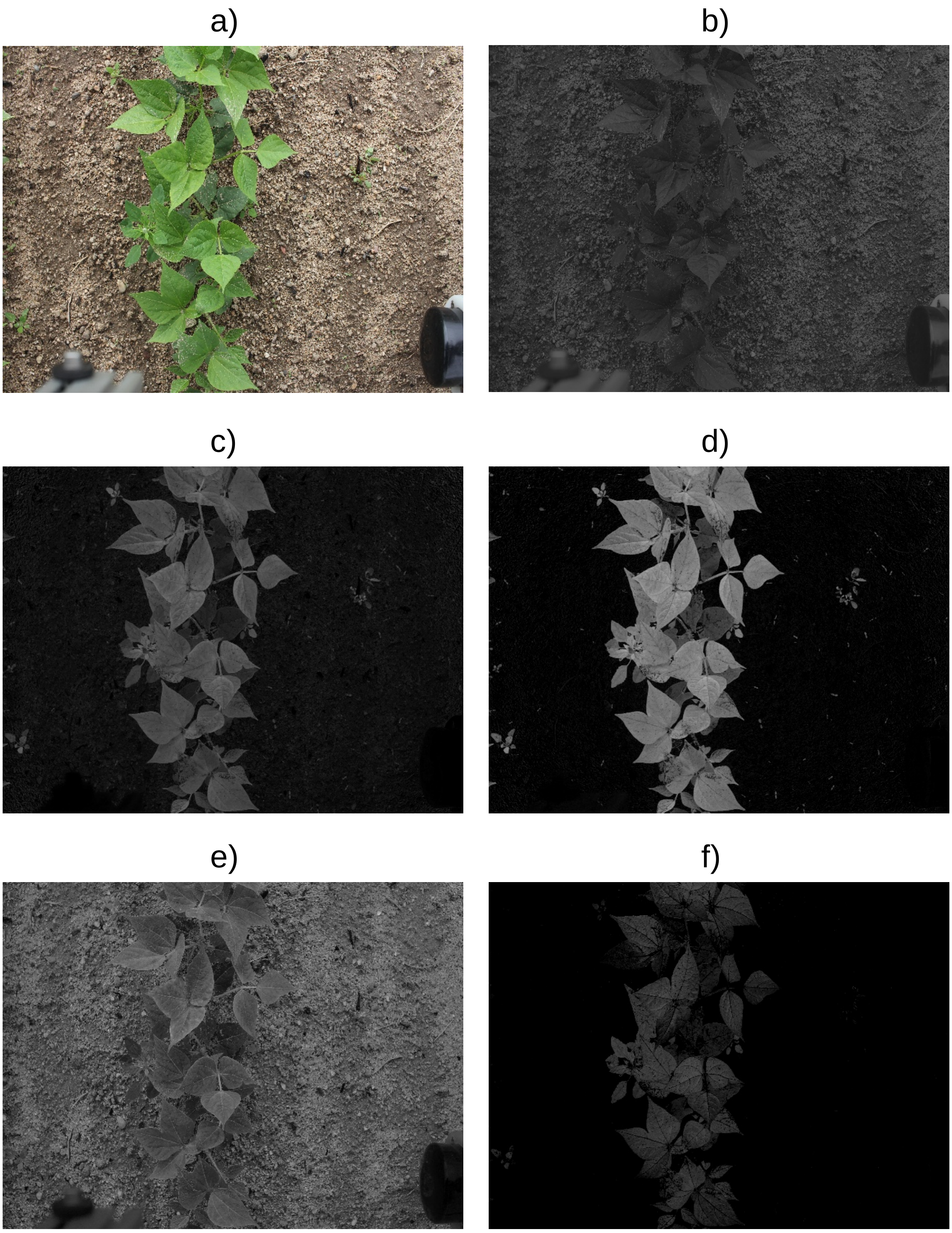

Figure 2.7: a) Original image. b) $CIVE$. c) GB. d) ExG. e) ExR. f) $ExG - ExR$.

2.8). The central vertical axis of the hexcone corresponds to the main diagonal of the color cube.

In the HSV color space, the color distribution of an object is invariant with respect to brightness. For this reason, each HSV channel is filtered to eliminate some regions

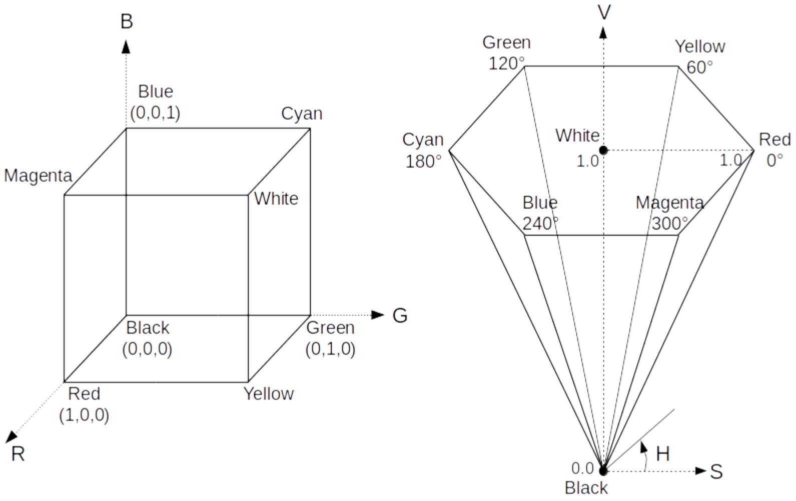

Figure 2.8: RGB cube and HSV hexcone representation.

in the image based on their hue, saturation and value.

$$I_{BIN}\left(x,y\right)=\begin{cases}1 & \text{if } \delta_{min}\leq I_{HSV}(x,y)\leq\delta_{max}\\ 0 & \text{otherwise}\end{cases} \tag{2.12}$$

with δ_{min} and δ_{max} are the (h,s,v) threshold values. Another representation of the equation (2.12) is the HSV decision tree illustrated in Figure 2.9. The objective of the decision tree is to filter an HSV image in multiples subsets: each subset corresponds to a branch of the tree and the discrimination is executed by assigning different thresholds to each channel.

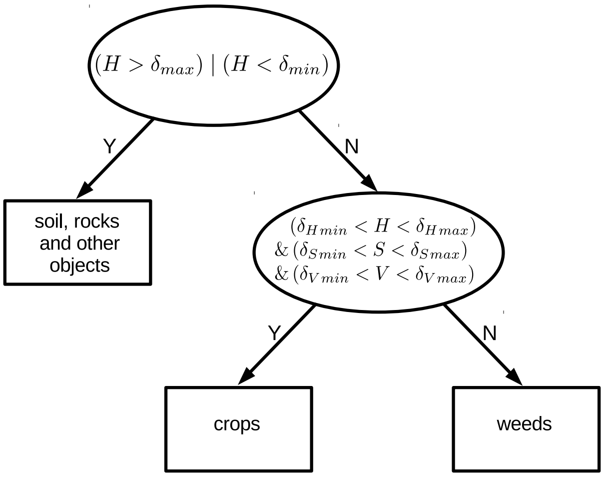

Figure 2.9: HSV decision tree.

For the proposed approach, the generated decision tree contains three branches; the first branch is resulted from the background segmentation. Objects like rocks, soil, etc. are in this branch. The second branch is created by the plants detected and the last branch is formed mostly by weeds.

The transformed HSV image was split into its hue, saturation and value channels as Figure 2.10 shows. Histograms of each channel were calculated (see Figure 2.11). Since H channel represents the pure color, it is mostly utilized to eliminate the background. Which is equivalent to Excess Green vegetation defined in equation (2.7). In the case of S and V channels, multiple combinations of these channels lead to separating crops from weeds.

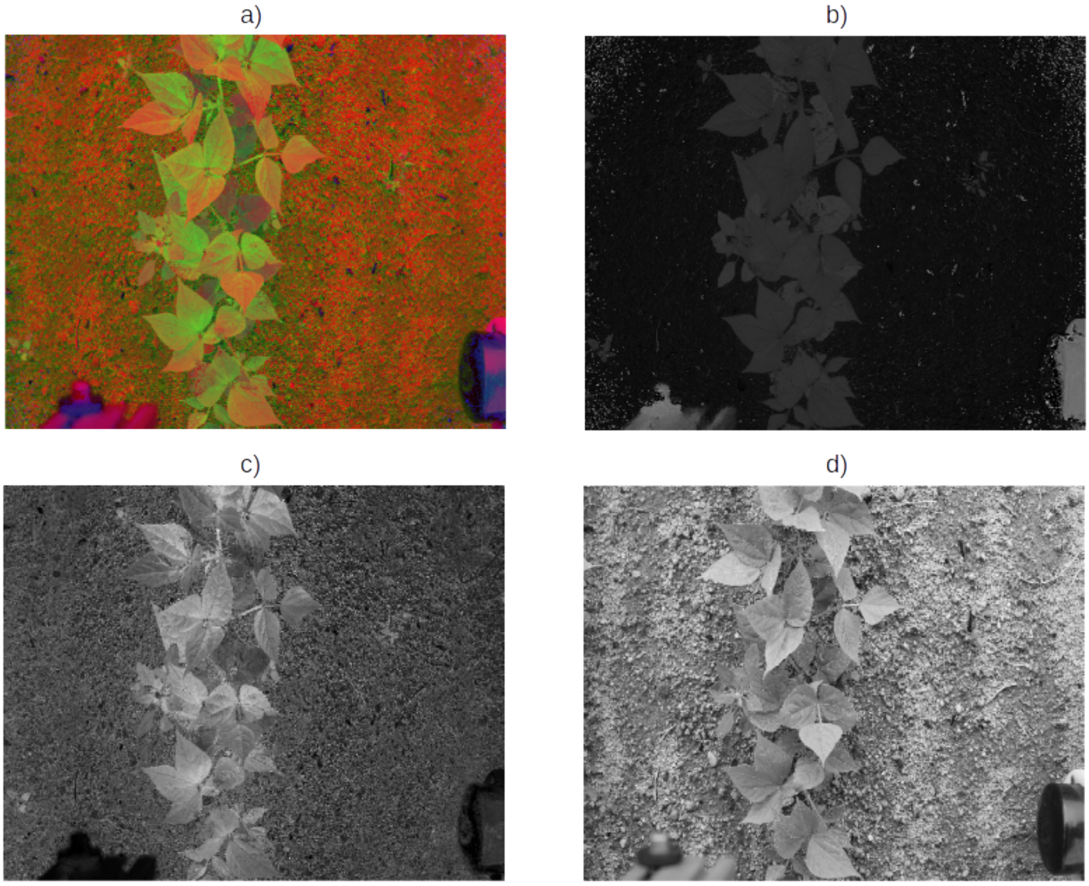

Figure 2.10: HSV space: a) HSV image. b) Hue. c) Saturation. d) Value

To discern between crops and weeds, some physical properties in conjunction with the crop growing stage are considered. In particular, the maize growth stage is divided into two phases: vegetative and reproductive growth (see Figure 2.12). The vegetative growth begins with the emergence of the first leaf (the first leaf in maize has a rounded tip) and ends with the tassel emergence (the last branch of the tassel is visible at the top of the plant). While the reproductive growth begins after tassel emergence (flowering begins when a silk is visible outside the husks) and ends with physiological

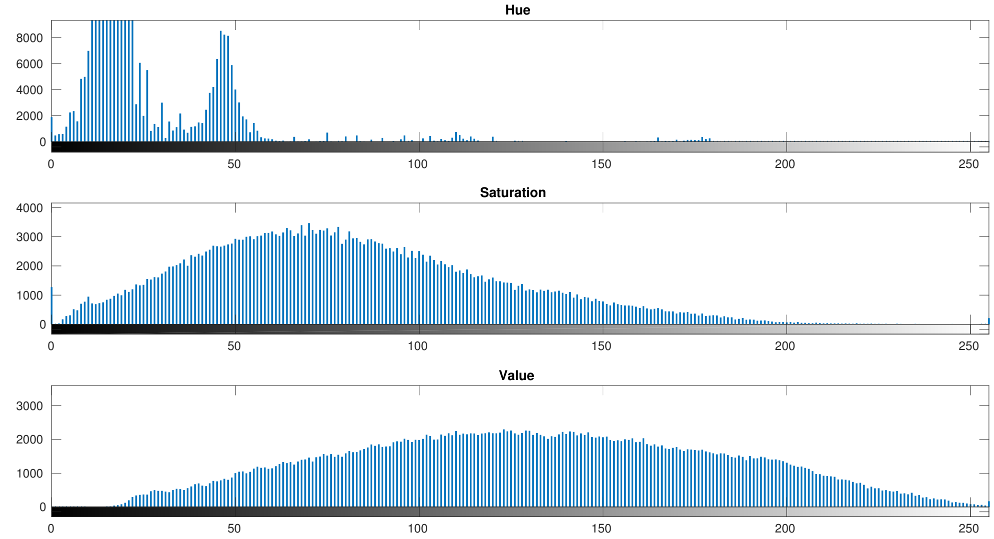

Figure 2.11: The histogram of Hue, Saturation and Value.

maturity (a black layer forms at the base of the kernel).

The maize physical transformation during its growth is shown in Figure 2.12. Its color changes with respect to its stage of growth. Similarly, the number and surface area of its leaves is linked to its growing stage. Parameters that can be used to carry out its detection. In the same manner, the bean growth stages were studied to extract physical characteristics considering the days from planting [68]. As well as maize, bean growth stage is divided into the same categories: vegetative and reproductive, some of its characteristics are detailed in Appendix A.1.

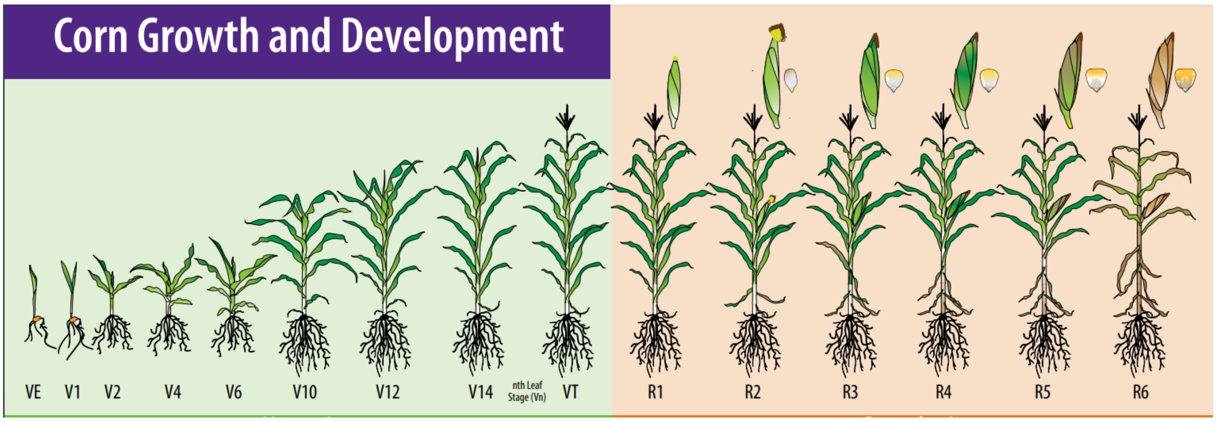

Figure 2.12: Maize growing stage [69].

Afterwards, by implementing a morphological operation, small plants (part of weeds) and segmentation noise are removed, refining the region of interest.

$$A \ominus B = \{z \mid B_z \subseteq A\} \tag{2.13}$$

Equation (2.13) implies the erosion of A by the structuring element B. In this case, A corresponds to the binary image (I_{BIN}) and B is a mask of size 4×4 pixel. After labeling the N regions from the resulting binary image, moment calculation is applied to determine the mass distribution and the location of each region's center $(\overline{x}, \overline{y})$ in order to estimate the exact location of the crops.

$$\overline{x} = \frac{M_{10}}{M_{00}}, \quad \overline{y} = \frac{M_{01}}{M_{00}} \tag{2.14}$$

$(\overline{x}, \overline{y})$ are given in pixels and $M_{ij} = \sum_x \sum_y x^i y^j I(x, y)$ are the raw region's moments.

Finally, the area of each detected plant is compared with a threshold δ_a to keep plants above that value, δ_a depends on the crop growth stage.

2.5 YoCo-v1: plant detection based on learning-model and physical characteristics

The previous subsections detailed two different alternatives to carry out plant and weed recognition. However, as was explained, each proposition has some drawbacks, i.e., the detection based on physical characteristics fulfills crop detection by generating finely refined masks; but some weeds are segmented incorrectly and they remain in the generated masks (see Figure 2.13 - c)). On the other hand, YOLACT++ performs well on weeds and partially well on crops, but imperfect masks are generated due to inaccurate bounding box detection as Figure 2.13 - b) shows.

In this section, a novel and computationally lightweight algorithm based on the fusion between YOLACT++ and the easy-to-extract characteristics algorithm is presented. The refined masks generated by the algorithm based on physical characteristics are used to complete the fragmented masks generated by YOLACT++. Similarly, the incorrectly detected weeds are removed from masks generated by the algorithm based on physical characteristics using the YOLACT++ weed segmentation. The global architecture of the proposed approach is shown in Figure 2.14.

The improvements resulting of this fusion are:

- A solution to the issue of mask leakage resulting from YOLACT++ semantic segmentation of crops/weeds.

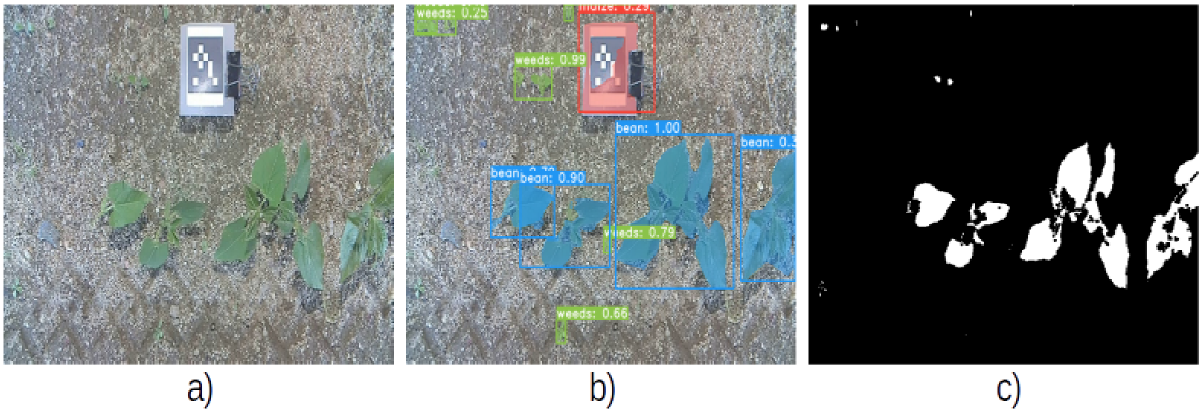

Figure 2.13: Crop/weed detection. a) Input image. b) YOLACT++ segmentation . c) Characteristics-based segmentation.

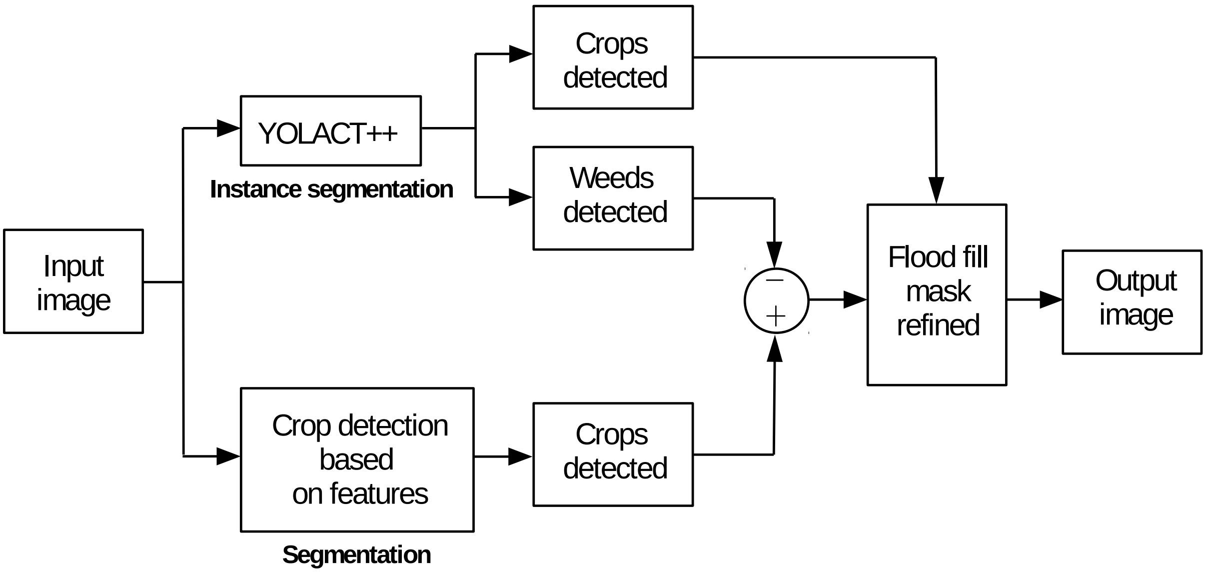

Figure 2.14: Overview of the proposed crop/weed segmentation network YoCo-v1.

- Reassigning the position of the bounding boxes on objects (plants/weeds) regarding the location failure issue.
- Improvement of the mask quality of each instance by refining fragmented masks and reducing false positives and negatives, increasing the mean intersection over union (mIoU) and pixel accuracy (PA) scores.

According to the scheme presented in Figure 2.14, the input image is used by both detection methods: YOLACT++ and the algorithm based on feature extraction. In the case of YOLACT++, a previous training stage is necessary to execute the algorithm.

2.5.1 Construction of synthetic images

Traditionally, training a network to perform object segmentation requires a substantial dataset containing image samples with their corresponding ground truths. This task requires labeling and usually manual segmentation in each image sample causes labeling errors due to inadequate information used, e.g., when pixels are labeled in image data, visual input is used instead of numeric values corresponding to the observation.

To reduce the number of image samples to carry out training, the initial stage involves creating synthetic RGB images to build a synthetic image dataset. The proposed methodology consists in designing an element of the crop/weed classes to create "collages" from natural images. To design an image, using a cropped element from our dataset of crop fields, a random rotation, a bicubic scaling interpolation, and a brightness and contrast adjustment are considered by a linear multiplier. Similarly, a random background coincident to the experimental soil is set to be segmented. Then, the generated element is proportionally adjusted in brightness and overlapped between elements to create samples from images obtained by the camera in natural conditions (Figure 2.15). Finally, a mask polygon description per crop/weed is associated with generating a properties schema regarding each image in a JavaScript Object Notation (JSON) file, it is a data format that stores structures and objects and it is easy to integrate with different interfaces.

Figure 2.15: Samples of generated synthetic images.

2.5.2 Instance segmentation

According to the pipeline shown in Figure 2.14, each input image $\mathcal{I}$ is processed to produce instance segmentation. In terms of performance, the best ranked is the Mask R-CNN [70] based on a two-stage detector. The processing speed is also essential for real-time execution. The YOLACT++ method solves this issue by using a single-stage detector. The result is a combination of the instance prototypes and associated predicted coefficients. For training, a classification loss, a box regression loss, and a mask loss are used.

Deficiencies can be found in incomplete and overlapping masks [71]: location failure and leakage. The first refers to partial object localization in scenarios with many elements, with a partial segmentation. Leakage occurs when the bounding box is not accurate because the masks are cropped after being assembled without considering noise removal outside the cropped region. In the same way, if the bounding box is larger than the desired detected object, the masks of neighboring elements will also be added to the current mask.

2.5.3 Head

Considering the architecture diagram Figure 2.14, every machine learning model might be thought of as a backbone plus a head. The head performs the processing strategy of the images generated by both segmentations. In particular, the head is executed sequentially, starting with image segmentation and then mask refined.

Mask refinement

In the initial stage of the crop mask refinement, a processing path for YOLACT++ segmented instance $\mathcal{I}_Y$ and another processing path given by the feature extraction algorithm for segmentation $\mathcal{I}_s$ is presented.

For each input image, the first path computes $\mathcal{N}_c$ bounding boxes and $\mathcal{N}_c$ masks of the detected crops and $\mathcal{N}_w$ bounding boxes and $\mathcal{N}_w$ masks of the detected weeds, generating the images $\mathcal{I}_{Y_1}$ and $\mathcal{I}_{Y_2}$, respectively. The second path processes the segmentation $\mathcal{I}_s$ using the method explained in subsection 2.4. The previous masks are illustrated in Figure 2.16.

To remove weeds that remain in the $\mathcal{I}_s$ segmentation, the $\mathcal{I}_{Y_2}$ weed instance segmentation is used:

$$\hat{\mathcal{I}}_s = \mathcal{I}_s - \mathcal{I}_{Y_2}. \tag{2.15}$$

The results of (2.15) implies that some detected weeds by YOLACT++ were already removed in $\mathcal{I}_s$.

For now, and regarding the pipeline shown in Figure 2.14, there are three images.

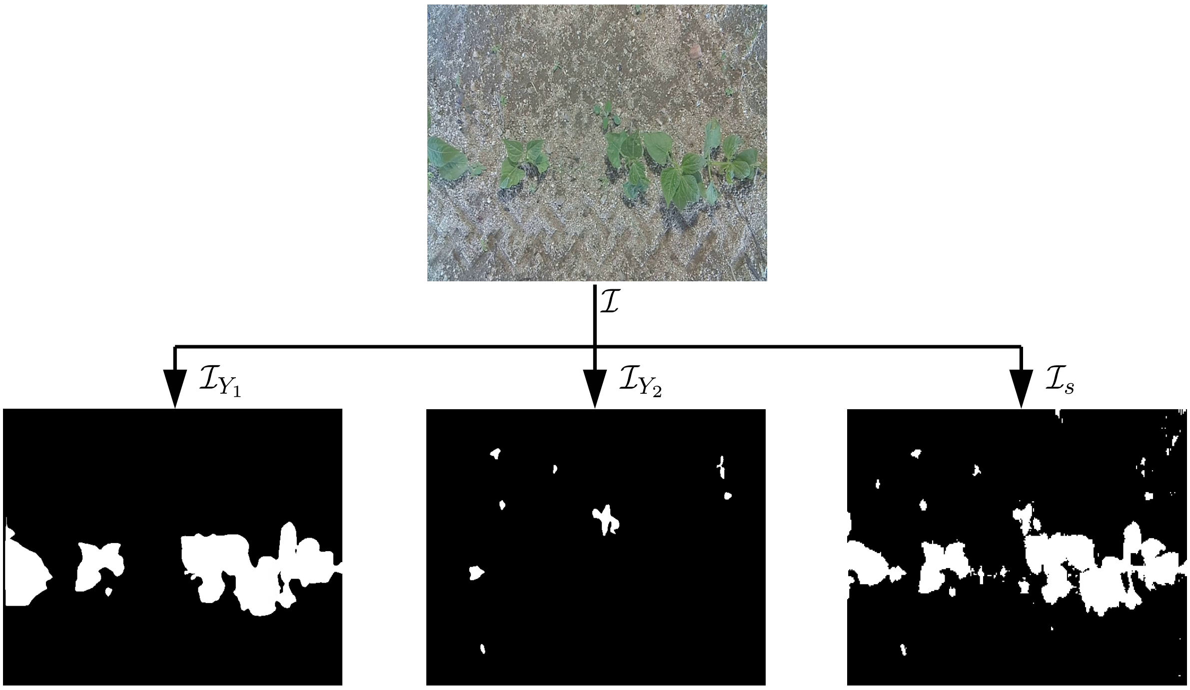

Figure 2.16: Obtained masks by YOLACT++ and the crop detection algorithm based on physical characteristics.

Figure 2.17: Improved image $\hat{\mathcal{I}}_s$.

The first image corresponds to the crop instance segmentation $\mathcal{I}_{Y_1}$: for each detected crop, the masks generated in this image are generally fragmented or incomplete. The second image corresponds to the weed instance segmentation $\mathcal{I}_{Y_2}$: these masks mostly correspond to the true weeds; the third image is an improved image $\hat{\mathcal{I}}_s$ of $\mathcal{I}_s$ as a result of the weeds detected by $\mathcal{I}_{Y_2}$ were removed. However, some weeds not detected in $\mathcal{I}_{Y_2}$ may still remain in this improved image.

To fill the incomplete crop masks in $\mathcal{I}_{Y_1}$, the improved image $\hat{\mathcal{I}}_s$ is used as a reference to delimit the area to be filled. Thus, the mask's contours are computed (Figure 2.18) to produce a crop contour image $\mathcal{I}_c$.

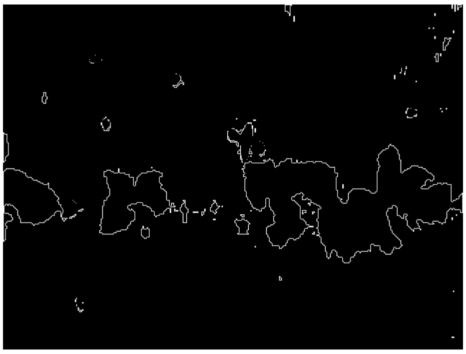

Figure 2.18: Mask's contours of $\hat{\mathcal{I}}_s$.

After defining the borders of each mask to be filled, the mask refinement is performed using the flood fill process described in Figure 2.19.

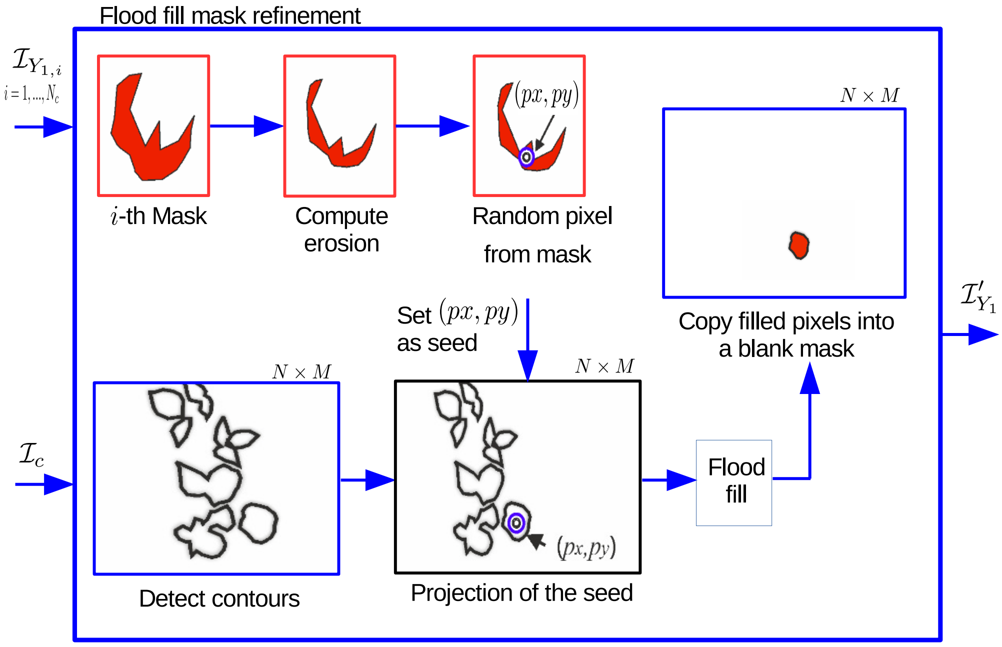

Figure 2.19: Flood fill mask refinement process

The i-th mask $\mathcal{I}_{Y_{1,i}}$ of instance $\mathcal{I}_{Y_1}$ has a set of pixel coordinates $\mathcal{P}_i$ associated with the the i-th mask $\mathcal{I}_{c_i}$ of crop contour image $\mathcal{I}_c$:

$$\mathcal{P}_i = \{p | p \in I_{Y_{1,i}} \cap \mathcal{I}_{c_i}\} \tag{2.16}$$

where p is a pixel coordinates (p_x, p_y). In other words, it indicates the position and the class of the detected crops in the associated contour image. The idea is therefore to fill the incomplete i-th mask $\mathcal{I}_{Y_{1,i}}$ delimited by the the i-th contour mask $\mathcal{I}_{c_i}$. A randomly selected pixel $p \in \mathcal{P}_i$ is set as a seed to fill the incomplete mask.

Due to pixel leakage of $\mathcal{I}_Y$, it may contain pixels out of the contour mask, i.e., pixels such that $\mathcal{O} = \{p | p \notin \mathcal{I}_c\}$ (Figure 2.20).

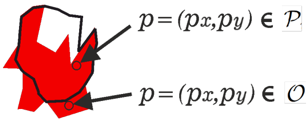

Figure 2.20: The red mask represents the i-th mask $\mathcal{I}_{Y_{1,i}}$, while the black contour is the i-th contour mask $\mathcal{I}_{c_i}$.

Hence, to simplify the selection of a seed in $\mathcal{P}$, the mask $\mathcal{I}_{Y_{1,i}}$ is eroded to eliminate the set $\mathcal{O}$ as Figure 2.21 shows.

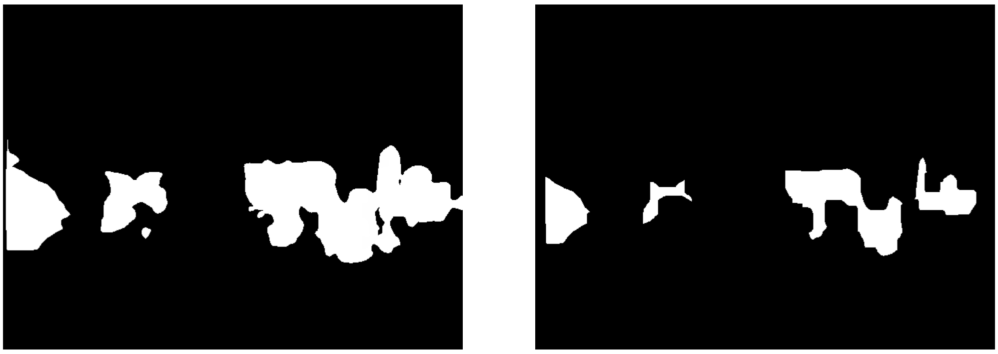

Figure 2.21: On the left, crop instance segmentation $\mathcal{I}_{Y_1}$ and on the right, $\mathcal{I}_{Y_1}$ eroded.

The well-known flood fill queue algorithm [72], initialized by a random seed p, is used to fill the associated class delimited by the contours $\mathcal{I}_c$. Finally, each filled mask $\mathcal{I}_{Y_{1,i}}$ is copied to a blank image to remove undetected elements and unwanted surrounding objects of $\mathcal{I}_c$:

$$\mathcal{I}'_{Y_1} = \bigcup_{i=1}^{\mathcal{N}_c} \text{floodfill}_{seed_i}(\mathcal{I}_c) \tag{2.17}$$

The refined mask $\mathcal{I}'_{Y_1}$ shown in Figure 2.22 includes the crop class instances. Each crop class of the mask is associated with colorimetry improvement by feature-based segmentation. Therefore, the output is a semantic mask defined for the crops.

Instance processing

The last step consists in instance processing. The refined mask $\mathcal{I}'_{Y_1}$ corresponds to a semantic segmentation because each pixel of the mask is classified to a category. However, the refined mask cannot distinguish every crop detected in the image. The

Figure 2.22: Refined mask $\mathcal{I}'_{Y_1}$

goal is to segment the refined mask $\mathcal{I}'_{Y_1}$ to generate an instance segmentation mask. Thus, the crop instance segmentation $\mathcal{I}_{Y_1}$ given by YOLACT++ is used. To define the k-th instance of $\mathcal{I}'_{Y_1}$, the k-th instance is removed from $\mathcal{I}_{Y_1}$. To the left of the Figure 2.21, the first k-th instance is eliminated generating the mask shown in Figure 2.23.

Figure 2.23: The k-th instance removed from $\mathcal{I}_{Y_1}$.

This mask is subtracted from the semantic refined mask $\mathcal{I}'_{Y_1}$. The selection of the k-th instance followed by the subtraction, is performed for each instance detected by $\mathcal{I}_{Y_1}$. Figure 2.24 shows the results just for the first k-th instance subtraction.

Figure 2.24: Instance segmentation mask.

Then, in the same way described above, a stage of contour extraction is performed followed by the flood fill stage. Finally, each k-th instance of the refined mask is cropped to generate the crop instance segmentation mask M_c presented in Figure 2.25.

The instances generated in Figure 2.25, may in some cases not correspond to the detected crops. However, in most of the tested images, the results validate the proposed approach. The few cases where the instances are calculated erroneously, correspond to adverse situations where the crops are close to each other causing overlapping of the crop leaves, as is the case of the obtained results in Figure 2.25.

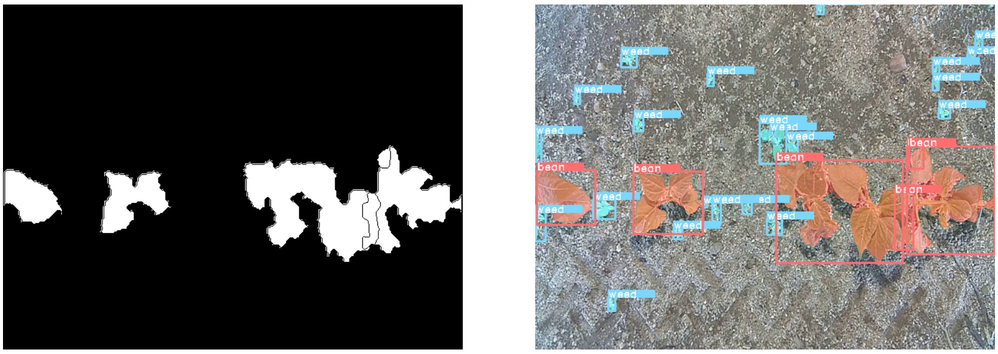

Figure 2.25: Yoco-v1 instance segmentation.

Additional classes

To find the refined mask of the remaining elements (weeds), M_c is subtracted from $\mathcal{I}_s$ when two detected classes are set:

$$M_w = M_c - \mathcal{I}_s \tag{2.18}$$

where $\{M_w\}_r, r = 1 \cdots \mathcal{N}_{w'}$. For additional classes, the model is easily extended in its architecture by adding a new head per sequentially connected class, representing a context similar to the layers of a deep network.

Bounding boxes

For each class detected and refined (crop and weed), a bounding box is re-calculated. The $\mathcal{Q}$ pixels of each u-th $\in M_c$ or $\in M_w$ instance contours, form a set of coordinates $D =$

$\{a_u, b_u\}_{u=1}^{Q}$ where four characteristic pixels $\amalg_u$ are set to design the bounding box. Then:

$$\begin{aligned}\amalg_u =& \{(\min_u(a_u), b_u), (a_u, \min_u(b_u)),\\ &(\max_u(a_u), b_u), (a_u, \max_u(b_u))\}_{u=1}^{4}\end{aligned} \tag{2.19}$$

2.6 Experimental results

A comparison between the proposed approach (Yoco-v1) and some state-of-the-art semantic segmentation methods (Mask R-CNN, YOLACT and YOLACT++) varying the backbone network is presented. Two metrics are introduced in order to evaluate object detection: the mean Average Precision (mAP) and the Intersection over Union (IoU).

The mean Average Precision is calculated by finding the Average Precision (AP) for each class and then averaging over the number of classes.

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{2.20}$$

The Average Precision (AP) summarizes a precision-recall curve as the weighted mean of precisions achieved at each threshold, with the increase in recall from the previous threshold used as the weight

$$AP = \sum_{n} (R_n - R_{n-1}) P_n \tag{2.21}$$

where P_n and R_n are the Precision and Recall. 'Precision' is defined as the number of true positives over the number of true positives plus the number of false positives Eq. (2.22), while 'Recall' is defined as the number of true positives over the number of true positives plus the number of false negatives Eq. (2.23).

$$P = \frac{TP}{TP + FP} \tag{2.22}$$

$$R = \frac{TP}{TP + FN} \tag{2.23}$$

The intersection over union is obtained by

$$IoU = \frac{Area\ of\ intersection\ of\ two\ boxes}{Area\ of\ union\ of\ two\ boxes} \tag{2.24}$$

A comparison between computational efficiency rates was made on a 1-GPU NVIDIA GeForce RTX 3060 laptop and on a Jetson Xavier NX embedded system without TensorRT. The comparative models were trained on the laptop with a batch size of 8 (the batch size is a number of samples processed before the model is updated) with the ImageNet pre-trained weights, the stochastic gradient descent was used in 20×10^3 iterations with a weight decay of 5×10^{-4}. Note that the head attached to YOLACT++ is only activated during execution. Therefore, the behavior of YoCo-v1 was analyzed during inference in two situations: laboratory and field experiments.

To evaluate the proposed architecture, ResNet-50-FPN and ResNet-101-FPN were used as the benchmark for their highest fps according to [63]. An AP benchmark was performed using a laptop and an embedded system for the considered state-of-the-art segmentation methods. The analysis was carried out only for the evaluation dataset since there is no comparison with YoCo-v1 in training. The evaluation of the fps during the inference of maize and beans was done because it will be used to control the action of the robot. Hence, Table (2.1) summarizes the compiled dataset on backbone variation, inference method and AP results for maize and beans.

Method	Backbone	FPS	mAP	AP_{75}	AP_{50}	mAP	AP_{75}	AP_{50}	FPS	mAP	AP_{75}	AP_{50}	mAP	AP_{75}	AP_{50}
		NVIDIA GeForce RTX 3060							Jetson Xavier NX						
			Maize			*Bean*				*Maize*			*Bean*		
Mask R-CNN	R-101-FPN	15	0.67	0.75	0.75	0.69	0.77	0.73	7	0.58	0.67	0.59	0.55	0.68	0.66
	R-50-FPN	17	0.68	0.75	0.70	0.79	0.76	0.71	8	0.59	0.63	0.65	0.57	0.64	0.61
YOLACT	R-101-FPN	29	0.58	0.63	0.61	0.70	0.69	0.63	19	0.52	0.61	0.55	0.61	0.62	0.58
	R-50-FPN	32	0.50	0.60	0.57	0.64	0.61	0.66	18	0.50	0.50	0.53	0.55	0.52	0.60
YOLACT++	R-101-FPN	27	0.61	0.71	0.69	0.68	0.64	0.67	17	0.63	0.67	0.69	0.60	0.67	0.68
	R-50-FPN	31	0.63	0.69	0.68	0.69	0.61	0.65	18	0.61	0.68	0.68	0.55	0.59	0.64
YoCo v1	**R-101-FPN**	**27**	**0.80**	**0.80**	**0.82**	**0.80**	**0.81**	**0.78**	**17**	**0.77**	**0.81**	**0.79**	**0.73**	**0.81**	**0.75**
	R-50-FPN	29	0.75	0.79	0.77	0.84	0.81	0.79	18	0.75	0.76	0.78	0.79	0.80	0.80

Table 2.1: Comparative results of instance segmentation methods for maize and bean. The mAP was calculated by taking the mean AP over all classes and overall IoU thresholds.

For the Mask R-CNN, a $mAP =$ 0.67 for maize and $mAP =$ 0.75 for bean were obtained using the computer system, while a $mAP =$0.58 for maize and $mAP =$ 0.55 for bean were obtained using the embedded system. Nevertheless, the fps are not satisfactory for a real-time application. Regarding YOLACT, the difference is 0.09 mAP for maize and 0.012 mAP for bean on the laptop. The difference is not relevant concerning the implementation in the embedded system. However, processing speed is relevant for Mask R-CNN since a rate of 19 fps was achieved using the deeper ResNet-101-FPN network. YOLACT++ improves these results over the mAP.

On the other hand, the proposed approach results in a $mAP =$0.80 for maize and $mAP =$0.80 for bean using the computer system, at 27 fps. For the embedded system, $mAP =$0.77 for maize and $mAP =$0.73 for beans at 17 fps.

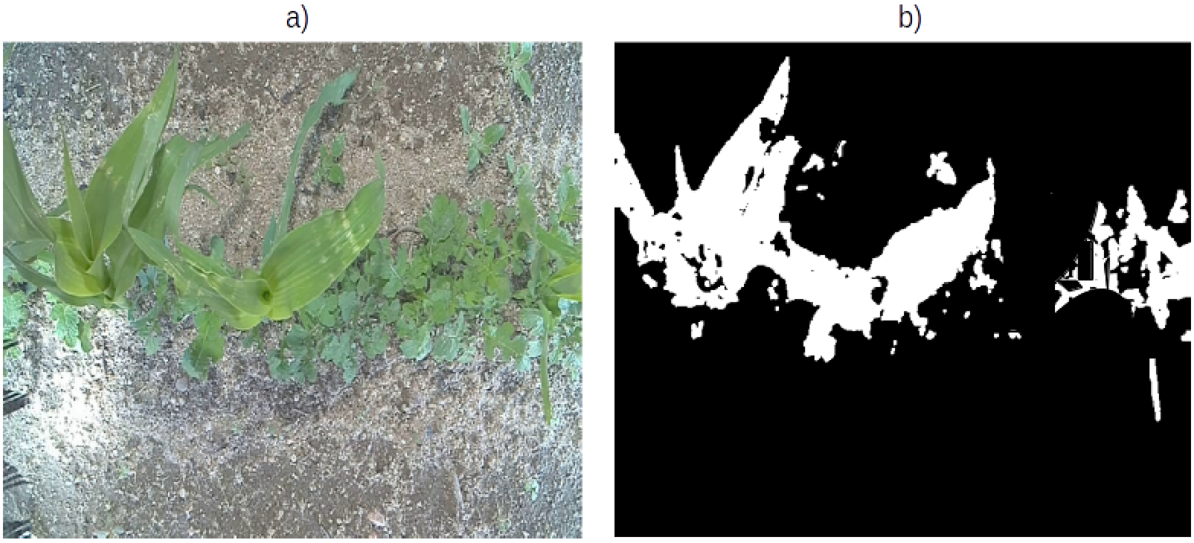

Figure 2.26: Maize mask generated by the algorithm based on physical characteristics. a) Original image. b) Maize mask.

Figure 2.27: Bean mask generated by the algorithm based on physical characteristics. a) Original image. b) Bean mask.

Figures 2.28 and 2.29 shows a comparison between YOLACT++ and the proposed approach Yoco-v1. The segmentation given by YOLACT++ tends to create incomplete masks in maize, it can be attributed to its advanced growing stage. As was explained and illustrated in Figure 2.12, the physical characteristics varied from one week to the other. Similarly, the reduced intra-row distance between crops and weeds perturbed their detection. The research done on physical properties for both maize and bean was utilized to adjust the parameters of the algorithm based on physical characteristics. The threshold

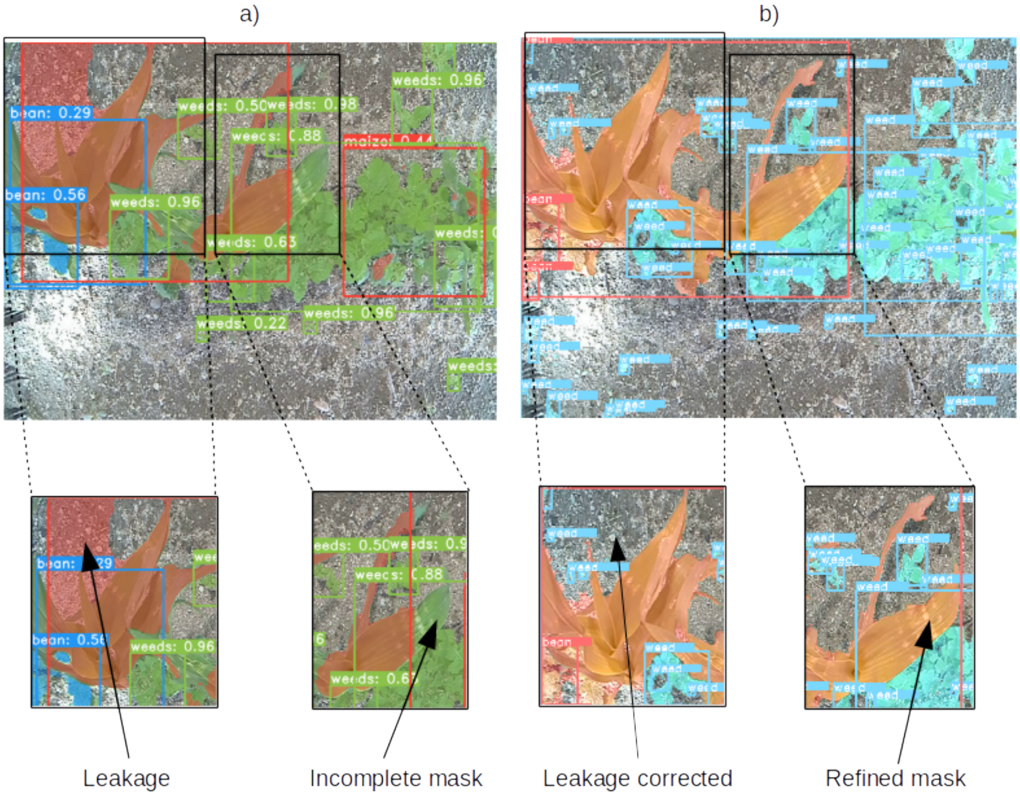

Figure 2.28: Maize segmentation. Comparison between YOLACT++ and the proposed approach Yoco-v1. a) Segmentation given by YOLACT++ with leakage and incomplete mask problems. b) Segmentation given by Yoco-v1 with leakage corrected and refined mask.

values selected for bean segmentation were δ_{min} =(30,35,100) and δ_{max} =(96,255,255), while maize threshold values were δ_{min} =(25,52,72) and δ_{max} =(102,255,255). The size of the element to perform the morphological operation was 4×4 for bean and 3×3 for maize. Figure 2.26 shows the mask obtained. It is a refined mask that performs well for maize detection. However, some weeds remain in the image.

In the case of bean, the segmentation given by YOLACT++ presents problems of leakage which, as can be observed in Figure 2.29, the leakage is around the plant leaf and also in the space formed in the center of the plant leaves. The results given by the algorithm based on physical characteristics are similar to those obtained in the maize crop. The algorithm can extract a well delimited bean mask (Figure 2.27).

The crop/weed segmentation given by the proposed approach improves the results given by the algorithms operating independently. For the case of incomplete masks, the detection given by the algorithm based on physical characteristics fills the plant leaf as can be seen in Figure 2.28, this is because the masks generated segment crops perfectly. Equivalently, the leakage problem was reduced substantially using

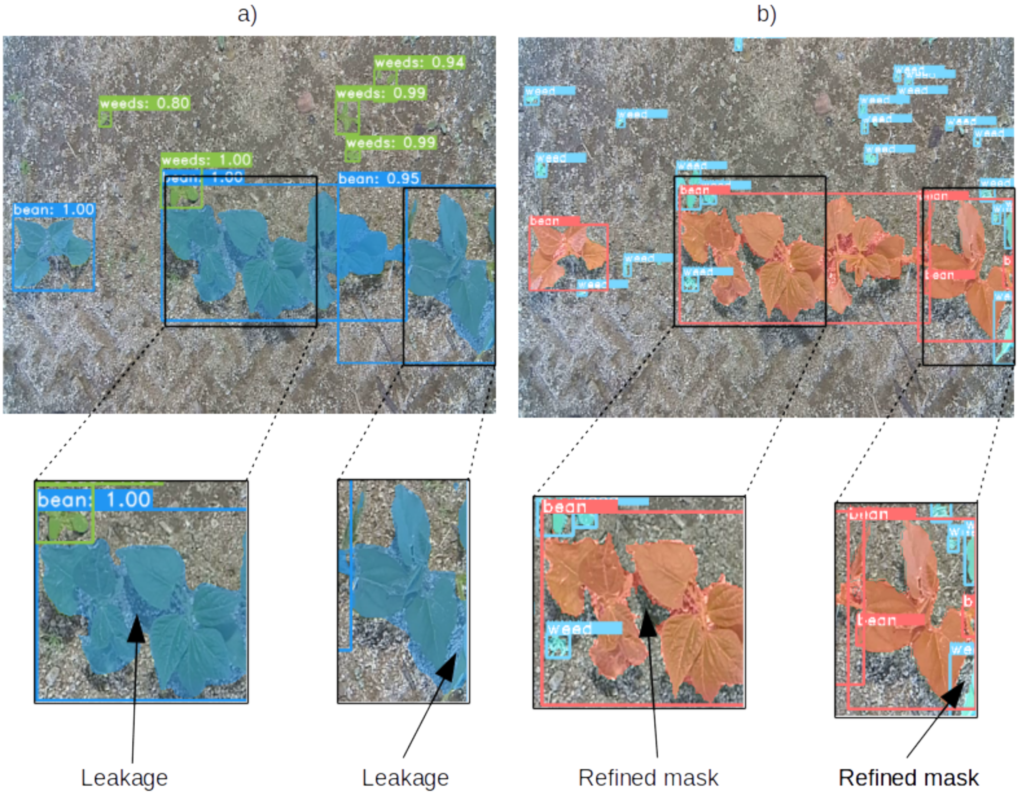

Figure 2.29: Bean segmentation. Comparison between YOLACT++ and the proposed approach Yoco-v1. a) Segmentation given by YOLACT++ with leakage problem. b) Segmentation given by Yoco-v1 with leakage corrected.

Yoco-v1. In the case of the remaining weeds in masks generated by the algorithm based on physical characteristics, when fusion was carried out, the weeds detected successfully by YOLACT++ allowed to remove them.

2.7 Conclusion

Three algorithms for crop and weed segmentation were exposed in this chapter: the first one is a learning-based fully connected network, the second is based on physical characteristics and the last algorithm resulting from the fusion of the first two algorithms.

The generation of synthetic images to train the learning-based algorithm allowed to reduce the exhausting and tedious process of labeling the dataset. A portion of the collected dataset was utilized to extract some crops and plants to create samples under different conditions (multiple rotations, contrasts, scaling, background, etc.). The segmented mask generated by this algorithm performs well on weeds. However, in the case of crop segmentation some problems were observed: leakage, location failure and incomplete masks.

In the case of the algorithm based on physical characteristics, it has been shown that the use of artificial intelligence is not mandatory to carry out crop/weed identification. The physical characteristics chosen to execute crop/weed detection represent a trade-off between robustness and processing time. The higher the number of physical characteristics, the lower the estimation rate. The results given by the algorithm suggest that it can be a solution to separate crops from different weeds. To choose the parameter values, a compilation was conducted to describe the physical changes of maize and bean during their whole growing stage. However, the methodology can be extended to other crops and weeds.

The fusion between the learning-based algorithm and the algorithm based on physical characteristics improved the results of each algorithm performed independently. On one side, the masks generated by YOLAC++ are incomplete, the bounding box localization sometimes does not correspond with the real object position and erroneous elements are considered during the mask creation, resulting in the leakage problem. On the other hand, the algorithm based on physical characteristics segments better vegetation (crops) than weeds. The results obtained by Yoco-v1 proved that the fusion of these algorithms can improve crop/weed segmentation. The crop masks were refined using the physical characteristics segmentation and the bounding box position was corrected.

The results obtained during the evaluation campaigns validate the proposed algorithms, several experiments were conducted under different weather conditions and distinct crop/weed growth stages.

3

Autonomous navigation based on visual perception

Chapter summary

The dual-purpose plant recognition system was used in the previous chapter to obtain the segmentation that allows to discern between crops and weeds. This information is given by the proximal detection and it will be used to manipulate the weeding tool in Chapter 4. In this chapter, the secondary purpose of the plant recognition system is exploited: autonomous vision-based navigation. The proposed approach considers the data provided by the frontal detection and it is carried out without relying on GPS data. The solution is based on a visual servoing technique that uses the vanishing point estimation formed by the crop line detection. This approach allows to navigate through the parcel, make transitions between parcels and change from one parcel to another one (half-turn). Divers scenarios were created in the Gazebo simulator as close as possible to real-agricultural conditions to validate the proposed approach. Likewise, the results presented in this chapter were obtained in the ROSE challenge and the ACRE competition.

Table of contents

3.1 Introduction

Autonomous navigation is one of the most challenging tasks in robotics. In particular, agricultural robotics imposes hard restrictions. On the one hand, agriculture is done in outdoor environments where conditions are always changing and, moreover, can not be controlled. On the other hand, if the robot carries out imprecise navigation, crops can be damaged. For these reasons, navigation algorithms are still an open problem for the robotics community.

To provide the capacity for autonomous navigation to a robot, the robot needs to know its exact position. A variety of sensors can provide information about the environment to position the robot on the experimental field. Two categories of sensors can be identified, those that provide the absolute position and those that report the relative position. The former refers to a global position, the most utilized sensor for this purpose is the global positioning system (GPS). The latter focuses on local positions, examples of these sensors are: gyroscopes, encoders, cameras, accelerometers, etc.

At the same time, localizing a robot in an urban terrain where GPS signal is frequently lost requires robust algorithms to estimate its pose. Normally, the more robust the algorithms, the processing rate tends to decrease, which dictates the compromise between accurate pose estimation and the robot's velocity.

Having solved the pose estimation problem, the next factor to be considered is the type of controller to be implemented as well as the locomotion to be used. The proposed approach in this chapter allows to navigate in the parcels of the experimental field. The plant detection presented in the previous chapter is used to provide visual information to control the vehicle using visual servoing. The fact that autonomous navigation is an element of the intra-row autonomous weeding task, it is assumed that the platform can move at low speed, which implies some considerations in the vehicle's wheeled locomotion. These considerations are described now on, but before a review of navigation systems in agriculture is presented.

3.2 Navigation methods in agriculture

Nowadays, robots carry out diverse missions in agricultural fields as was exposed in subsections 1.2.1-1.2.5. Among the tasks that they share in common, navigation is one of the most important. As was mentioned before, two types of farming can be distinguished: indoor and outdoor.

One of the contributions that corresponds to the first category is presented in

[15]. It exposes the use of steel pipes as a method of guidance. To navigate along the pipes, the authors chose nylon wheels because they can withstand heavy loads and high temperatures. In general, the implementation of rails as navigation systems is widely used in indoor farming due to the workspace is known and the dimensions of these scenarios are reduced.

In the case of outdoor farming, as a consequence of most crops are sowed in rows, different algorithms for row detection have been proposed [29], [73]. The main problem in the row detection process is to find multiple features that allow to identify crop lines under different conditions and in a constant way. The features extracted from crops are used to compute line estimation. Among the first efforts, the Hough transform was the most implemented technique. However, it performs well under favorable conditions, which is not the case in real agricultural implementations. For example, the quantity of the detected features is not constant and false positive detections are always present in this kind of application. Consequently, line estimation is not robust and the processing time to estimate the Hough space accumulator can be a factor to be considered. To mitigate the disadvantages just mentioned, the fusion between the Hough transform and the robot's model was proposed in [74].

GPS technology is another alternative to perform guiding in farming. Considering that GPS provides absolute position measurements, their values can fluctuate between 4 to 8 m accuracy, which represents an ambiguous reference in precision agriculture. To increase the accuracy of conventional GPS systems, fusion with other sensors has been investigated: the use of a GPS base station resulting in a Real-Time Kinematic Global Positioning System (RTK-GPS) [75], [76], the inertial measurement unit (IMU) and gyroscope sensors [77], [78]. The drawbacks of the GPS technology are the obstruction of the line-of-sight to satellites, radio frequency interference and poor signal in some regions, especially in farming environments where robots can operate under obstacles like trees, objects, solar panels, etc.

3.3 Autonomous vision-based navigation

In this section, an autonomous vision-based navigation is presented. When the workspace allows to extract visual information, it can be used for different purposes. In particular, when the frontal perception system detects crops, this detection is utilized to guide the platform through the parcel. The proposed approach is based on a visual servoing technique for trajectory correction. The goal is to center the platform in the lane formed by the crop lines.

3.3.1 Camera modeling

The simplest camera model to describe the mathematical relationship of the projection of points from 3-D space to an image plane is the pinhole model.

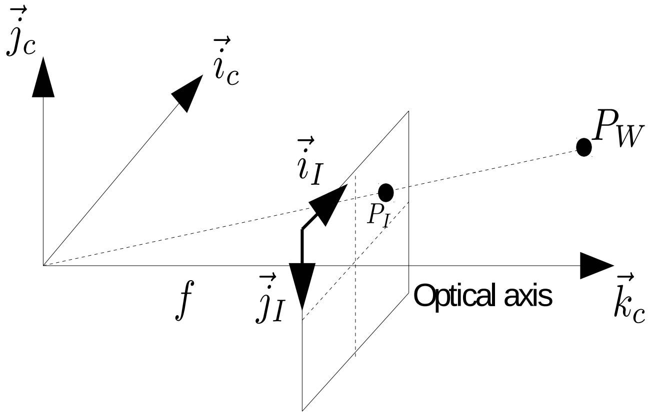

Figure 3.1: Pinhole model with associated coordinate systems. $\left(\vec{i}_W, \vec{j}_W, \vec{k}_W\right)$ is the world coordinate frame. $\left(\vec{i}_C, \vec{j}_C, \vec{k}_C\right)$ is the camera coordinate frame and $\left(\vec{i}_I, \vec{j}_I, \vec{k}_I\right)$ is the image coordinate frame.

Considering the pinhole camera model shown in Figure 3.1, a point in space is represented by the coordinates $P_W = (X_P, Y_P, Z_P)$. When projecting this point on the normalized image plane (in metric units)

$$\begin{aligned} x_P &= f\frac{X_P}{Z_P} \\ y_P &= f\frac{Y_P}{Z_P} \end{aligned} \tag{3.1}$$

where f is the focal length of the camera lens. It is expressed in homogeneous coordinates:

$$\begin{pmatrix} fX_P + Z_P p_x \\ fY_P + Z_P p_y \\ Z_P \end{pmatrix} = \begin{bmatrix} f_x & 0 & p_x & 0 \\ 0 & f_y & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X_P \\ Y_P \\ Z_P \\ 1 \end{pmatrix} \tag{3.2}$$

where (p_x, p_y) are the coordinates of the principal point, f_x and f_y represent the pixel ratios in the x-axis and the y-axis. Finally, the camera coordinate and the world coordinate frames are related by the extrinsic matrix , $\boldsymbol{X_c} = \begin{bmatrix} R & t \end{bmatrix} \boldsymbol{X}_w$, where R is 3×3 rotation matrix and t is 3×1 translation matrix (Figure 3.2).

Thus, the general mapping is given by:

$$x = K\begin{bmatrix} R & t \end{bmatrix} = \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_1 & r_2 & r_3 & t_1 \\ r_4 & r_5 & r_6 & t_2 \\ r_7 & r_8 & r_9 & t_3 \end{bmatrix} \tag{3.3}$$

The variations of P on the normalized image plane when the robot moves and considering f =1, we have:

$$\begin{aligned} \frac{d}{dt}(x_P) &= \frac{d}{dt}\left(\frac{X_P}{Z_P}\right) \\ &= \frac{1}{Z_P}\dot{X}_P - \frac{X_P}{Z_P^2}\dot{Z}_P \end{aligned} \tag{3.4}$$

$$\begin{aligned} \frac{d}{dt}(y_P) &= \frac{d}{dt}\left(\frac{Y_P}{Z_P}\right) \\ &= \frac{1}{Z_P}\dot{Y}_P - \frac{Y_P}{Z_P^2}\dot{Z}_P \end{aligned} \tag{3.5}$$

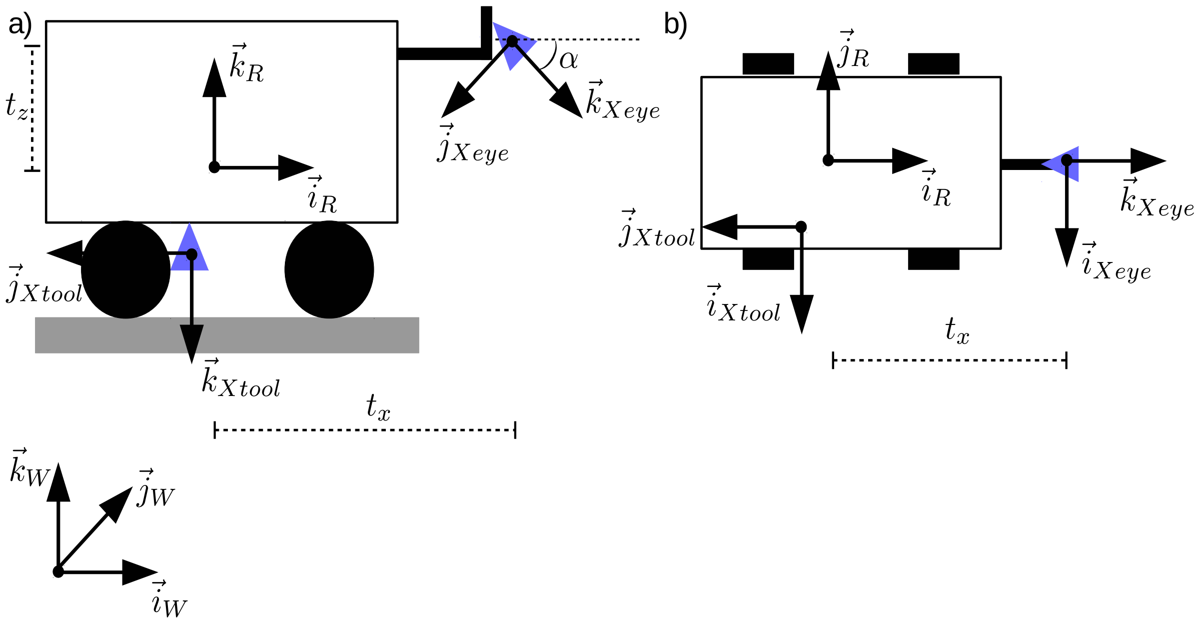

Figure 3.2: Platform description. a) Side view. b) Top view. t_x and t_y are the translations with respect to the robot frame and α is the camera orientation.

and the relative velocity of the point P with respect to the camera is

$$\dot{X} = -\begin{pmatrix} \upsilon_x \\ \upsilon_y \\ \upsilon_z \end{pmatrix} - \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \times \begin{pmatrix} X_P \\ Y_P \\ Z_P \end{pmatrix} \implies \left\{ \begin{array}{c} \dot{X} = -\upsilon_x - \omega_y Z_P + \omega_z Y_P \\ \dot{Y} = -\upsilon_y - \omega_z X_P + \omega_x Z_P \\ \dot{Z} = -\upsilon_z - \omega_x Y_P + \omega_y X_P \end{array} \right\} \tag{3.6}$$

inserting (3.6) in (3.4) and (3.5) and grouping terms

$$\dot{x}_P = \begin{pmatrix} -\frac{1}{Z_P} & 0 & \frac{x_P}{Z_P} \end{pmatrix} \begin{pmatrix} \upsilon_x \\ \upsilon_y \\ \upsilon_z \end{pmatrix} + \begin{pmatrix} x_P y_P & -(1+x_P^2) & y_P \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \tag{3.7}$$

$$\dot{y}_P = \begin{pmatrix} 0 & -\frac{1}{Z_P} & \frac{y_P}{Z_P} \end{pmatrix} \begin{pmatrix} \upsilon_x \\ \upsilon_y \\ \upsilon_z \end{pmatrix} + \begin{pmatrix} 1+y_P^2 & -x_P y_P & -x_P \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \tag{3.8}$$

which can be written

$$\dot{x} = L_s \boldsymbol{\upsilon}_c \tag{3.9}$$

where L_s is the interaction matrix and is given by

$$L_s = \begin{pmatrix} -\frac{1}{Z_P} & 0 & \frac{x_P}{Z_P} & x_P y_P & -(1+x_P^2) & y_P \\ 0 & -\frac{1}{Z_P} & \frac{y_P}{Z_P} & 1+y_P^2 & -x_P y_P & -x_P \end{pmatrix} \tag{3.10}$$

The coordinates of the vanishing point on the normalized image plane are

$$x_{vp} = 0 \tag{3.11}$$

$$y_{vp} = 0 \tag{3.12}$$

Equations (3.11) and (3.12) result from the vanishing point projection to infinity, $Z = \infty$. Considering the vanishing point coordinates and using the equations (3.7) and (3.8)

$$\dot{x}_{vp} = \begin{pmatrix} x_{vp} y_{vp} & -\left(1+x_{vp}^2\right) & y_{vp} \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \tag{3.13}$$

$$\dot{y}_{vp} = \begin{pmatrix} 1 + y_{vp}^2 & -x_{vp}y_{vp} & -x_{vp} \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \tag{3.14}$$

Which corresponds to the vanishing point movements in the image. From Figure 3.2, the transformation matrix between the world frame and the robot frame is given by

$$T_{W\longleftarrow R} = \begin{pmatrix} \cos\theta_R & -\sin\theta_R & 0 & x_R \\ \sin\theta_R & \cos\theta_R & 0 & y_R \\ 0 & 0 & 1 & z_R \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.15}$$

with x_R, y_R, z_R and θ_R are the robot absolute localization. The transformation matrix between the robot frame and the camera frame is given by

$$T_{R\longleftarrow C} = \begin{pmatrix} 0 & -1 & 0 & -t_y \\ \sin\alpha & 0 & -\cos\alpha & t_x\sin\alpha - t_z\cos\alpha \\ \cos\alpha & 0 & \sin\alpha & t_x\cos\alpha + t_z\sin\alpha \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.16}$$

where α, t_x, t_y and t_z correspond with the extrinsic camera parameters. The camera linear velocity with respect to the world frame is

$$\boldsymbol{\upsilon}_c = \begin{pmatrix} \upsilon_c - t_y\dot{\theta}_R \\ t_x\dot{\theta}_R \\ 0 \end{pmatrix}_R = \begin{pmatrix} -t_x\dot{\theta}_R \\ \left(\upsilon_c - t_y\dot{\theta}_R\right)\sin\alpha \\ \left(\upsilon_c - t_y\dot{\theta}_R\right)\cos\alpha \end{pmatrix}_C \tag{3.17}$$

and its angular velocity written as

$$\boldsymbol{\omega}_c = \begin{pmatrix} 0 \\ 0 \\ \dot{\theta}_R \end{pmatrix}_R = \begin{pmatrix} 0 \\ -\dot{\theta}_R\cos\alpha \\ \dot{\theta}_R\sin\alpha \end{pmatrix}_C \tag{3.18}$$

Then,

$$\boldsymbol{\upsilon}_c = \begin{pmatrix} \upsilon_x \\ \upsilon_y \\ \upsilon_z \end{pmatrix}_C = \begin{pmatrix} -t_x\omega_c \\ (\upsilon_c - t_y\omega_c)\sin\alpha \\ (\upsilon_c - t_y\omega_c)\cos\alpha \end{pmatrix}_C \tag{3.19}$$

$$\boldsymbol{\omega} = \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}_C = \begin{pmatrix} 0 \\ -\omega_c \cos\alpha \\ \omega_c \sin\alpha \end{pmatrix}_C \tag{3.20}$$

with $(\boldsymbol{v}_c, \boldsymbol{\omega}_c)$ the linear and angular velocities of the platform.

3.3.2 Modeling of the platforms

Two different platforms were utilized in this work. The first operates using a differential drive and the second has an Ackermann steering mechanism.

The differential drive under consideration (Figure 3.3) assumes that the robot wheels do not slide. The platform movements are controlled by the speed of its wheels, forward motion is achieved when both wheels are driven at the same rate and turning movements are achieved by controlling each wheel at a different rate.

The kinematic model is given by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v_c \cos\theta \\ v_c \sin\theta \\ \omega_c \end{bmatrix} \tag{3.21}$$

where $q = \begin{bmatrix} x & y & \theta \end{bmatrix}^T$ is the platform pose with respect to the world frame, v_c and ω_c are the linear velocity and the angular velocity control inputs respectively.

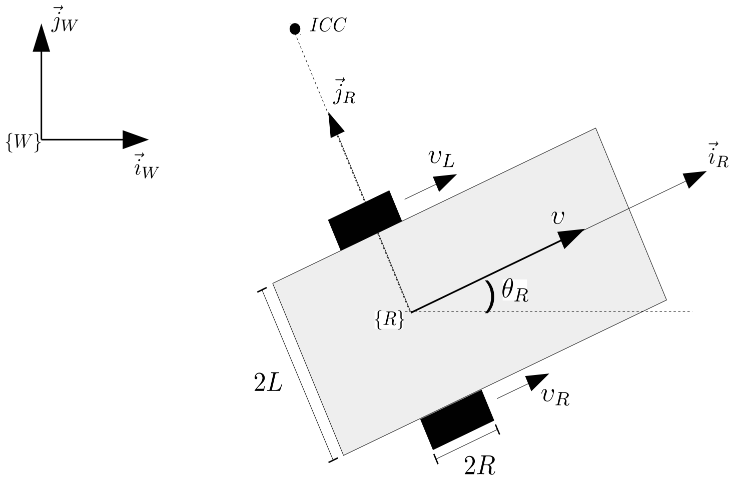

Figure 3.3: Differential drive.

For rear-wheel driving, the vehicle configuration is given by $q = \begin{bmatrix} x & y & \theta & \delta \end{bmatrix}^T$, with (x, y) the position and θ the orientation of the platform with respect to the world frame $\{W\}$, and δ the steering angle of each front wheel as is depicted in Figure 3.4. the kinematic model is derived as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ \frac{\tan\phi}{l} \\ 0 \end{bmatrix} \upsilon_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \upsilon_2 \tag{3.22}$$

where υ_1 and υ_2 are the linear and the steering velocity, respectively. When the front wheel is normal to the longitudinal axis of the platform, the robot becomes jammed, which corresponds to the model singularity at $\phi = \pm\frac{\pi}{2}$. To avoid this singularity, the platforms have restricted range of the steering angle.

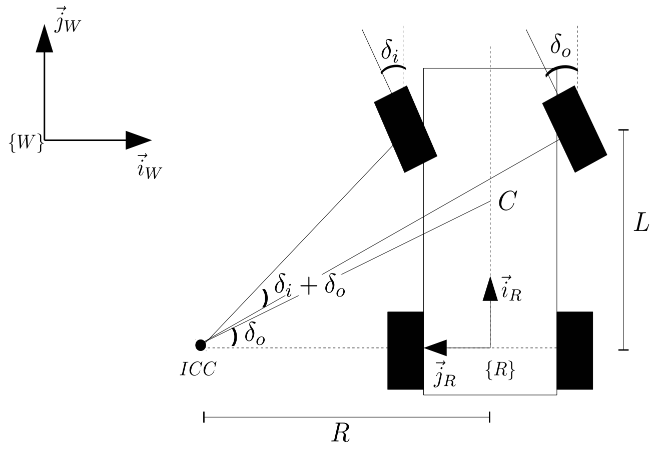

Figure 3.4: Ackermann steering geometry with multiple reference frames.

In the model (3.22), vertical movements are ignored, roll motion is not considered and rear steering angles are null. Due to the platform navigates at low speed, the simplified model (3.22) was utilized.

3.3.3 Features extraction from frontal detection

In order to carry out an autonomous navigation, sensors have to provide information to the robot of its environment. The detection described in Subsection 2.4 aims to detect only the plants of interest in the presence of a wide assortment of weeds.

The choice of this proposition is justified by:

1. It can detect multiple crops considering their physical characteristics.
2. It can be easily configured to adapt to different growing stages of the plant.

3. The required processing time allows to implement the algorithm on an embedded system.
4. The position and orientation of the frontal sensor can be modified (see Figure 3.5).

Figure 3.5: Multi-modal frontal detection onboard the experimental platform.

The points described above distinguish the proposed approach from the works reported in the literature. First, machine learning algorithms classify plants based on the data with which they were trained. This implies new training for different crops or considering a dataset big enough that can integrate a wide variety of crops. The parameters of the proposed algorithm can be modified experimentally or take into account the physical characteristics of the plants of interest to be detected.

Then, similar to the first point, crops change their physical characteristics as they grow. Dataset availability is hard to find for different growing stages. Normally, the success classification rate tends to decrease when experimental conditions differ from training data.

Although it is true that the higher the number of extracted features, the better the algorithm, its computational time increases significantly, and in some cases, their use in real-time applications is not feasible. Moreover, the experimental field conditions considered in this thesis impose a challenging scenario where plants are not scattered and they are mixed with weeds, ground, and vegetation as is illustrated in Figure 3.6.

Figure 3.6: Example of RGB images given by frontal detection. a)Bean. b)Maize.

3.3.4 Plant detection using the industrial sensor AQiT

The AQiT solution is a system that analyses hyperespectral data, RGB images and, using CNN technology, it can classify crops and weeds. It is developed and marketed by Carbon Bee [79]. As part of the consortium of this project, this company presents another alternative to detect plants of interest. Due to is a commercial solution, detailed information of its operation is limited. However, the scheme shown in Figure 3.7 describes in a general way the main elements of the sensor.

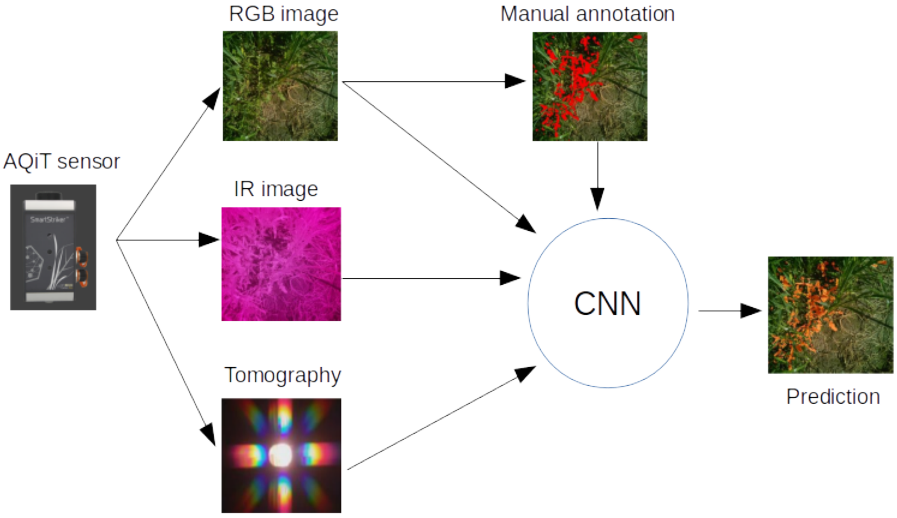

Figure 3.7: General AQiT sensor scheme

The detection of the AQiT sensor is given in .json files format which stores the coordinates of each polygon that represents the plant or plants detected in each image. The purpose of integrating this sensor and its respective results is to evaluate an industrial solution for rapid implementation in challenging detection conditions and, in the context of this thesis to make a comparison with the proposed approach as can be seen in the results section below.

3.3.5 Line extraction and vanishing point estimation

The autonomous navigation strategy proposed in this work is based on vanishing point estimation. The vanishing point is the point at which the projection of parallel lines intersect (see Figure 3.8). These lines are either extracted directly from the scene or they have to be estimated given some image features. In this work, the image features are given by the previous plant of interest detection.

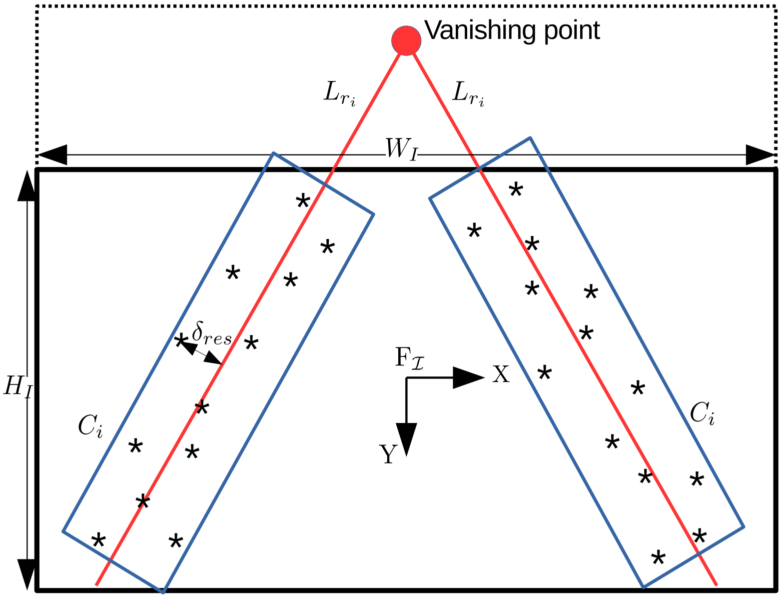

Figure 3.8: Vanishing point estimation given the coordinates of the detected plants. The vanishing point lies outside the image, its projection on the image plane is used for platform trajectory correction.

From Figure 3.8, H_I and W_I are the height and width of the image, respectively. The clusters are denoted by C_i with $i = 1, 2$ for maize crop and $i = 1, 2, 3$ for bean crop. L_{r_i} is the reference line i used to generate the cluster C_i, δ_{res} is the distance between the reference line and the edge of the cluster and $F_{\mathcal{I}}$ is the image frame.

To perform line extraction, the initial position of the platform has to be near the crop rows. That is, the platform may or may not be aligned with respect to the crops, but they have to be inside the camera's field of view.

Considering that the coordinates of the detected plants are known a priori given by the equation (2.14). The initial estimations of the reference lines L_r are initialized by (3.23)

$$L_{r_i} : a_{r_i}x + b_{r_i}y + c_{r_i} = 0 \tag{3.23}$$

where a_{r_i}, b_{r_i} and c_{r_i} are the parameters of the reference lines, with $i \in \{1, 2\}$ or $i \in \{1, 2, 3\}$ for maize and bean plants respectively.

The reasons for not applying the line detection techniques reported in the literature, like Hough transform [80], random sample and consensus (RANSAC) [81], are:

- The plant classification is not perfect and some false positives are detected, resulting in some outliers which perturb the precision of the line extraction.
- In some situations, very few plants are detected. For some algorithms, the more features the better the line detection.
- The processing time is related with the navigation velocity, most of the reported algorithms require substantial computing resources.

The reference lines are used to cluster the plants of interest detected by the frontal multimodal sensor. To eliminate the detected false positives, the residual is calculated. This difference between the data point and the line is compared with the threshold value δ_{res} to create each cluster. If the residual is either less than or equal to δ_{res}, the plant is added to the cluster Eq. (3.32); otherwise, the plant is considered as a false detection and is ignored.

Given a point P which corresponds with the coordinates of a detected plant of interest, and a reference line L, the line that covers the shortest possible distance from the point to the line is the perpendicular segment PA, of distance S (see Figure 3.9).

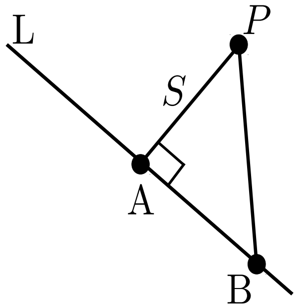

Figure 3.9: Right triangle formed by the line L, the segment perpendicular PA and the segment PB.

The distance between the point P with coordinates (x_0, y_0) and the line L is the segment PA, so that, $S = \|PA\|$. The general form of L is $ax + by + c = 0$ and its slope is $\frac{-a}{b}$. Due to S is perpendicular to L, its slope is $\frac{b}{a}$. To find the distance S, another line L' is added, L' is parallel to L and it passes through P. Similarly, S' is the parallel line to S and it passes through the origin (Figure 3.10).

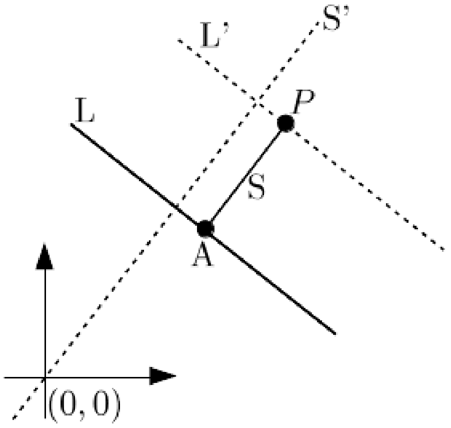

Figure 3.10: General scheme with the additional lines L' and S' to estimate the distance S.

Due to L' is parallel to L, they have the same slope. Given that L' passes through P, its equation is

$$y - y_0 = -\frac{a}{b}(x - x_0) \implies y = \frac{-ax + ax_0 + by_0}{b} \tag{3.24}$$

The equation of S' is

$$y = \frac{b}{a}x \tag{3.25}$$

As Figure 3.10 shows, lines L' and S' intersect, using equations (3.24) and (3.25)

$$\frac{-ax + ax_0 + by_0}{b} = \frac{b}{a}x \implies x = \frac{a(ax_0 + by_0)}{a^2 + b^2} \tag{3.26}$$

The intersection point of L' and S' is

$$P_{L'-S'}\left(\frac{a(ax_0 + by_0)}{a^2 + b^2}, \frac{b(ax_0 + by_0)}{a^2 + b^2}\right) \tag{3.27}$$

Now, the intersection between L and S'

$$\frac{-ax - c}{b} = \frac{b}{a}x \implies x = \frac{-ac}{a^2 + b^2} \tag{3.28}$$

using x to solve for y

$$y = \frac{b}{a}\left(\frac{-ac}{a^2+b^2}\right) \implies y = \frac{-bc}{a^2+b^2} \tag{3.29}$$

The intersection point of L and S' is

$$P_{L-S'}\left(\frac{-ac}{a^2+b^2}, \frac{-bc}{a^2+b^2}\right) \tag{3.30}$$

Finally, using the distance formula

$$\begin{aligned} d &= \sqrt{\left(\frac{-ac}{a^2+b^2} - \frac{a(ax_0+by_0)}{a^2+b^2}\right)^2 + \left(\frac{-bc}{a^2+b^2} - \frac{b(ax_0+by_0)}{a^2+b^2}\right)^2} \\ &= \frac{|ax_0+by_0+c|}{\sqrt{a^2+b^2}} \end{aligned} \tag{3.31}$$

Thus, to consider a detected plant with coordinates (x_k, y_k) as an element of the cluster C_i, a threshold δ_{res} is defined, which is the distance in pixels to cluster the detected crops.

$$C_i \equiv \frac{|\overline{x}_k a_{r_i} + \overline{y}_k b_{r_i} + c_{r_i}|}{\sqrt{a_{r_i}^2 + b_{r_i}^2}} \leq \delta_{res} \tag{3.32}$$

where $i = \{1, 2\}$ for maize crop or $i = \{1, 2, 3\}$ for bean crop and $(\overline{x}_k, \overline{y}_k)^\top$ are the coordinates of the plant k with $k = 0, 1, ..., N-1$. The parameters of the reference line are a_{r_i}, b_{r_i} and c_{r_i}.

The number of clusters is proportional to the number of crop lines. For each cluster, a line is fitted using a linear regression analysis which leads to define the line parameters $\hat{m}_i$ and $\hat{b}_i$ as:

$$\forall(\overline{x}_k, \overline{y}_k) \in C_i : \begin{cases} \hat{m}_i = \dfrac{\sum_k \overline{x}_k \overline{y}_k - \frac{1}{N}\sum_k \overline{x}_k \sum_k \overline{y}_k}{\sum_k \overline{x}_k^2 - \frac{1}{N}\left(\sum_k \overline{x}_k\right)^2} \\ \\ \hat{b}_i = \dfrac{\sum_k \overline{y}_k - \hat{m}_i \sum_k \overline{x}_k}{N} \end{cases} \tag{3.33}$$

where $\hat{m}_i$ is the slope, $\hat{b}_i$ is the y-intercept and N is the number of plants detected. $\hat{m}_i$ and $\hat{b}_i$ are the parameters of the extracted line $\hat{L}_i$ from the cluster C_i.

The reference lines are updated with the parameters of the extracted lines $L_{r_i} = \hat{L}_i$ and an accumulator is set to zero $R = 0$. This update is carried out for each new observation. When the plants are not detected or their detection does not belong to any cluster, the accumulator R is increased. This accumulator allows the platform to navigate in the absence of new detections as long as $R \leq \delta_R$. During this period the navigation is carried out considering virtual lines L_{v_i} created from the propagation of the reference lines.

When the number of elements in a cluster is very small with respect to the other cluster, it could be associated with a poor detection or a reduction of the number of plants. In that case, the reliability of the line extraction has to be considered. The empirical entropy is introduced to measure the cluster's heterogeneity and is given by:

$$H_T = -\sum_{i=1}^{N_C} P\left(C_i\right) \log_2 P\left(C_i\right) \tag{3.34}$$

where N_C is the number of clusters, $P\left(C_i\right)$ is the probability of a plant being classified as an element of the cluster C_i and is defined as $P\left(C_i\right) = \dfrac{N_{C_i}}{N}$, with N_{C_i} is the number of crops in the cluster C_i and N is the total number of crops. Finally, if $H_T \leq \delta_E$, the reference lines obtained in the previous iteration will be considered. The constant value δ_E is defined to allow for maximum disparity between clusters.

Now that lines have been extracted from the environment, the next step is to use them for navigation purposes. For any set of parallel crop lines in the scene, when projected on the image frame, there exists a point where the lines intersect. This point is called vanishing point, it can be located inside or outside the image as Figure 3.8 shows.

The proposed navigation approach estimates the vanishing point to navigate through the parcels; the projection of the estimated vanishing point has to remain in the center of the image to maintain the platform in the center of the lane formed by the crop lines.

To estimate the vanishing point, the system of equations formed by the lines has to be solved. In the case of maize crops (two rows), the linear system of equations could be solved using one of the various algebraic methods. In contrast, the system of equations for bean crops (three rows) will generally have no exact solution when distortion or other error factors are present. A fast and robust least squares approximation is presented which includes the solution for both systems of equations.

The proposed method consists in finding the least squares solution of the system of equations given by (3.33), this is, minimizing the distance between the set of lines

$$(\widehat{x}_{vp}, \widehat{y}_{vp}) = \min_{x,y} J_{LLS}(x, y) = \sum_{i=1}^{M} \sum_{j=1, j\neq i}^{M} e_{ij}^2(x, y) \tag{3.35}$$

where $(\widehat{x}_{vp}, \widehat{y}_{vp})^T$ are the coordinates in pixels of the vanishing point, M is the number of rows and $e_{ij}(x,y) = L_{ij} = a_{ij}x + b_{ij}y + c_{ij}$, L_{ij} is obtained by evaluating and subtracting crop line $\hat{L}_j$ from line $\hat{L}_i$.

The solution of the equation is given by

$$\begin{bmatrix} \widehat{x}_{vp} \\ \widehat{y}_{vp} \end{bmatrix} = A^{+}c \tag{3.36}$$

$[\widehat{x}_{vp} \;\; \widehat{y}_{vp}]^T$ are the coordinates in pixels of the vanishing point. A^{+} is the pseudo-inverse of matrix A with

$$c = [-c_{12}]\,, \quad A = \begin{bmatrix} a_{12} & b_{12} \end{bmatrix} \tag{3.37}$$

for maize and

$$c = [-c_{12} - c_{13}]^T\,, \;\; A = \begin{bmatrix} a_{12} & b_{12} \\ a_{13} & b_{13} \\ a_{23} & b_{23} \end{bmatrix} \tag{3.38}$$

for bean.

The Algorithm 1 allows to estimate the vanishing point given the coordinates of the detected crops.

3.3.6 Visual servoing for trajectory correction

Visual servoing control utilizes vision data to control the motion of a robot [82]. The data is acquired from a camera and using techniques of image processing, computer vision or any other solution to this purpose, features are extracted from the image provided by the camera. The aim of the vision-based control is to minimize an error $\boldsymbol{e}(t)$, defined by

$$\boldsymbol{e}(t) = \boldsymbol{s}\left(m(t), a\right) - \boldsymbol{s}^{*} \tag{3.39}$$

the vector $\boldsymbol{m}(t)$ is a set of image measurements that are computed to generate the vector $\boldsymbol{s}\left(m(t), a\right)$ of k visual features, in which a can represent additional knowledge about the system and $\boldsymbol{s}^{*}$ are the desired values of the features. The visual features $\boldsymbol{s}$ consist of the coordinates of the vanishing point projection in the image $\mathrm{s} = (\widehat{x}_{vp}, \widehat{y}_{vp})^{\top}$. The proposed approach assumes that the platform is moving at constant linear velocity. Consequently, the controller regulates the angular velocity ω to position the platform in the lane's center formed by line detection, i.e., $\boldsymbol{s}^{*} = \left(\frac{W_I}{2}, 0\right)^{\top}$, W_I is the image width.

Algorithm 1 Vanishing point estimation

```
Require: Coordinates of detected plants $(x_k, y_k)\ k \in N$
Ensure: Vanishing point coordinates $(\widehat{x}_{vp}, \widehat{y}_{vp})$
Initialization:
$L_{r_i} : a_{r_i}x + b_{r_i}y + c_{r_i}$
$C_i \leftarrow L_{r_i}$                          ▷ A cluster generated for each line initialized
while $(x_k, y_k)$ available do
    if $\left(d_{(x_k,y_k)} \leq \delta_{res}\right)$ then
        $C_i \leftarrow (x_k, y_k)$
    else
        $(x_k, y_k)$ ignored
    end if
    Extraction of line $\hat{L}_i$ from the cluster $C_i$
    Virtual lines generation $L_{v_i}$ from $L_{r_i}$
    Entropy estimation $H_T$
    if $(H_T \leq \delta_E)$ then
        $\hat{L}_i \leftarrow L_{v_i}$
    else
        $\hat{L}_i \leftarrow \hat{L}_i$
    end if
    Update $L_{r_i} \leftarrow \hat{L}_i$
    Compute LLS to obtain $(\widehat{x}_{vp}, \widehat{y}_{vp})$
end while
```

The relationship between the time variations of $\boldsymbol{s}$ and the camera velocity $\boldsymbol{v_c} = (v_c, \omega_c)$ is given by

$$\dot{\boldsymbol{s}} = L_s \boldsymbol{v_c} \tag{3.40}$$

where L_s is the interaction matrix related to $\boldsymbol{s}$. From equations (3.39) and (3.32) the time variation of the error is obtained

$$\dot{\boldsymbol{e}} = L_e \boldsymbol{v_c} \tag{3.41}$$

Considering a locally asymptotically stable control scheme that exponentially regulates $\boldsymbol{e} \rightarrow 0$ (i.e.,$\dot{e} = -\lambda \boldsymbol{e}$) such that

$$\boldsymbol{v_c} = -\lambda \widehat{L_e^+} \boldsymbol{e} = -\lambda \widehat{L_s^+} \left(\boldsymbol{s} - \boldsymbol{s}^*\right) \tag{3.42}$$

with $\widehat{L_e^+}$ is the approximation of the pseudo-inverse of the interaction matrix and λ is a constant parameter to be defined according to the desired speed of convergence.

3.4 Autonomous navigation without crop detection

The previous section described the navigation approach using crop detection given by the frontal sensor. However, when crop detection is not available, other alternatives have to be used in order to perform autonomous navigation. The absence of crops can be found in the transition zone and in any position of the field far from the parcels.

In this section, two alternatives are proposed to carry out autonomous navigation without considering crop detection: the first is based on visual detection and the second is a model predictive control to guide the platform.

3.4.1 Bird's-eye view transformation

The objective of using bird's-eye view perception is to produce a space representation to navigate when there are not crops to be detected, like in the case of the transition zones and the last segment of each parcel (see Figure 1.12). Bird's-eye view transforms the image captured by the frontal sensor into an equivalent image captured by the virtual camera as is illustrated in Figure 3.11.

Figure 3.11: Bird's-eye view transformation (on the right) of the image given by the frontal sensor (on the left).

Let $P = (x_1, y_1, z_1)^T$ the projected point on the image plane of the real camera and $P' = (x_2, y_2, z_2)^T$ the projected point on the image plane of the virtual camera. The homographic mapping H [83] of the two points is the planar projective transformation expressed by the equation (3.43)

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} = H \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} \tag{3.43}$$

The 3×3 matrix H is called the homogeneous matrix and $H = (h_{ij})$

$$\begin{aligned} \widehat{x}_2 = \frac{x_2}{z_2} = \frac{h_{11}x_1 + h_{12}y_1 + h_{13}}{h_{31}x_1 + h_{32}y_1 + h_{33}} \\ \widehat{y}_2 = \frac{y_2}{z_2} = \frac{h_{21}x_1 + h_{22}y_1 + h_{23}}{h_{31}x_1 + h_{32}y_1 + h_{33}} \end{aligned} \tag{3.44}$$

In the new image generated by the virtual camera, the depth of the image captured by the frontal sensor is easily distinguishable, which corresponds with the width of the parcel.

Using the same approach of transforming the image to the HSV color space and after applying thresholds to separate the vegetation from the rest of the image, a binary image is obtained. Then, a sliding window $\mathcal{W}_s$ centered in the image is used for two purposes: the first is to determine if vegetation is detected inside the window. Due to the characteristics of the experimental field, at the end of each parcel there is an area without vegetation of approximately 5 m long (see Figure 3.12). After this area, there is again vegetation in the center and sides of the image as Figure 3.12 shows. Thus, it is possible to distinguish when the end of a parcel has been reached. Otherwise, when the platform crosses the transition zone, at the center of the image there is a region without vegetation; and the second purpose is to extract a line in the non-vegetation zone of the image. This line is used to center the platform in the lane formed by the non-vegetation area.

The color segmentation allows to distinguish the lanes formed in the regions where there is no vegetation, as Figure 3.14–c shows. To avoid processing the entire image, a region of interest is defined which corresponds to the pixels that are inside the window $\mathcal{W}_s$. The dimensions of this window are 1.5×3 m.

Due to multiple lanes are detected in the bird's-eye view (see Figure 3.14), this introduces an ambiguity in identifying the lane that the platform has to follow. The window $\mathcal{W}_s$ reduces the regions of the image and the extraction of the line will be carry out inside this window, removing the other lanes.

To perform line extraction, the vegetation segmentation generates a mask where white pixels correspond to the detected vegetation and the black pixels are related to the non-vegetation area. Both regions are clustered and a contour detection is performed

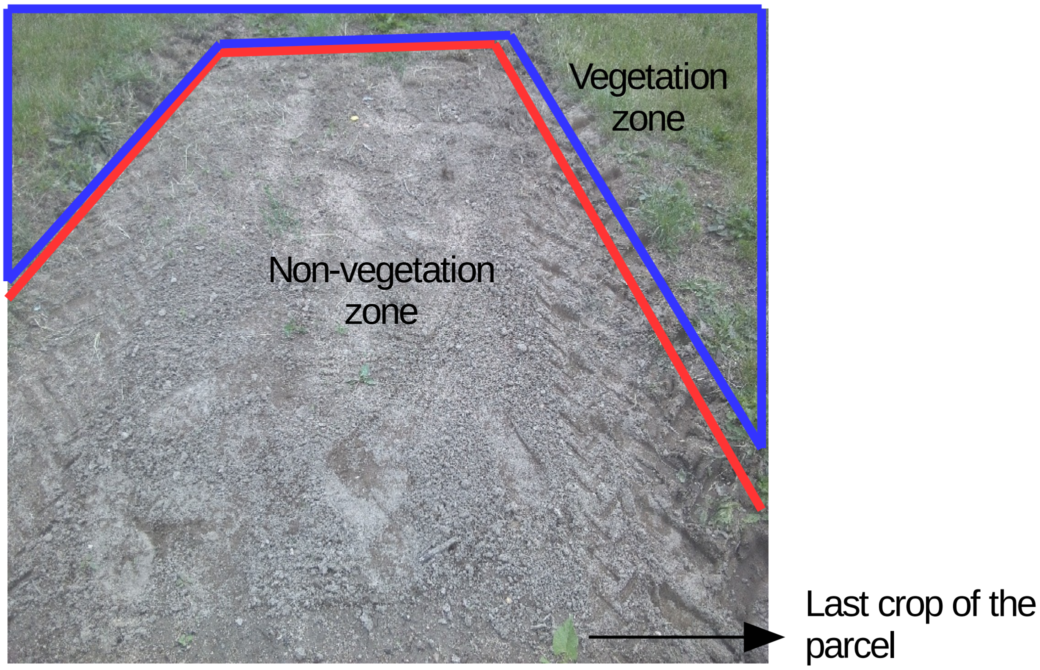

Figure 3.12: Image taken from autonomous navigation to illustrate the end of a parcel.

over the black pixels in order to find the lane's border. Finally, the middle line is obtained from the contour detection. Figure 3.13 shows a scheme of the previous description.

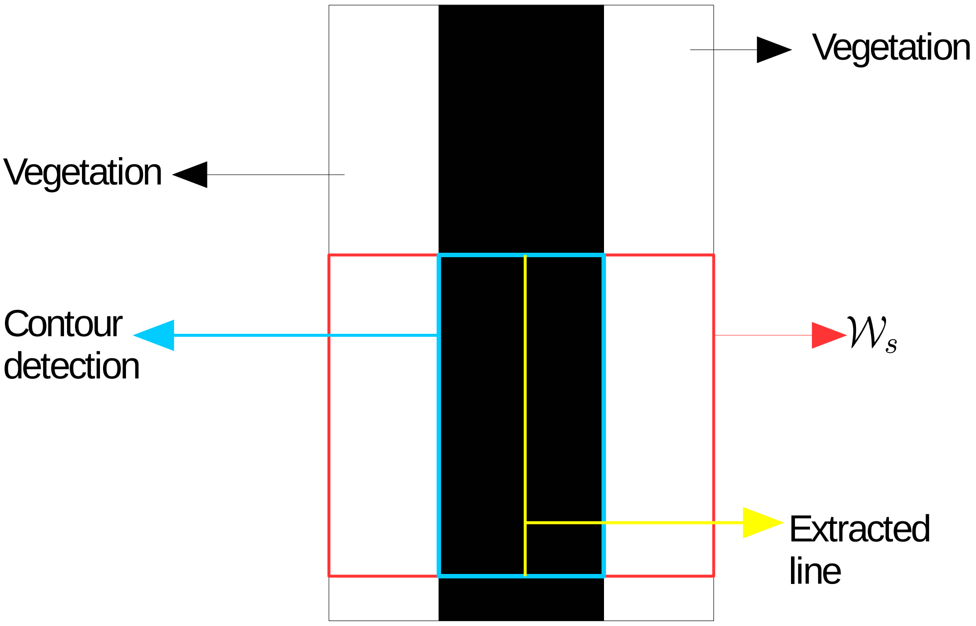

Figure 3.13: Bird's-eye view scheme.

For every new image, the position of $\mathcal{W}_s$ is updated by centering on the position of the parcel. Once the line has been obtained, the image features are set to $\boldsymbol{s} = (x, y)$

where x and y are the coordinates of the first point of the line and $s^* = \left(\frac{W_I}{4}, 0\right)$

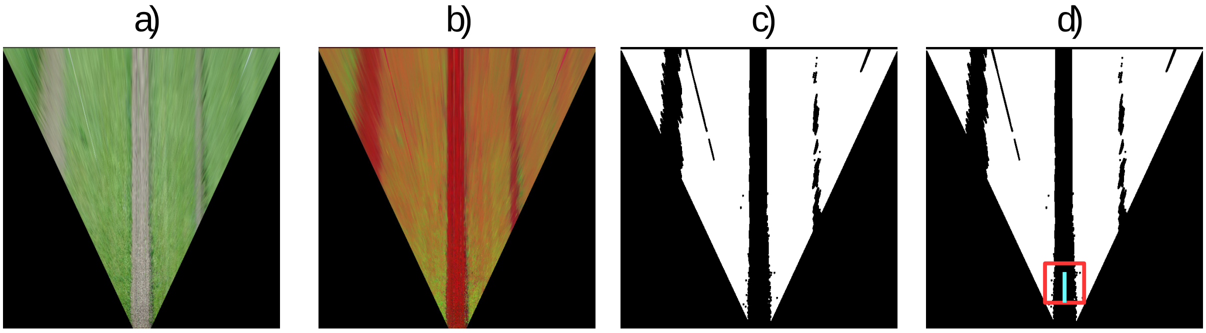

Figure 3.14: Top-view image processing. a) Top-view image captured by Xeye. b) HSV image. c) Green vegetation filtered. d) Line fitted within the sliding window.

3.4.2 Model predictive control

The model predictive control (MPC) [84] is the second proposed approach to perform autonomous navigation when crop detection is not available. This methodology guides the platform without relying on any visual detection and is utilized to move the platform from any initial point to any desired point. This approach can be used to bring the platform closer to the crop's position or to change between parcels (half-turn).

The MPC predicts the future behavior of the system over a finite time horizon (Figure 3.15). Based on the system model and the estimates of the current state of the system, the optimal control inputs with respect to a defined control objective and subject to system constraints is computed. However, only the first control input of the optimal control sequence is executed. This process is repeated at the next time instant using new available information on the system state.

In the following, some of the key concepts and techniques in MPC are presented. Considering an arbitrary system:

$$x_{k+1} = Ax_k + Bu_k \tag{3.45}$$

$$y_k = Cx_k \tag{3.46}$$

where $x_k \in \mathbb{R}^{n_x}$, $u_k \in \mathbb{R}^{n_u}$, $y_k \in \mathbb{R}^{n_y}$ are the system state, the control input and the system output, respectively, and k is the discrete time index.

The MPC minimizes a cost function J to drive the system state to some desired reference point. The controller performance is described by a quadratic cost index of the form:

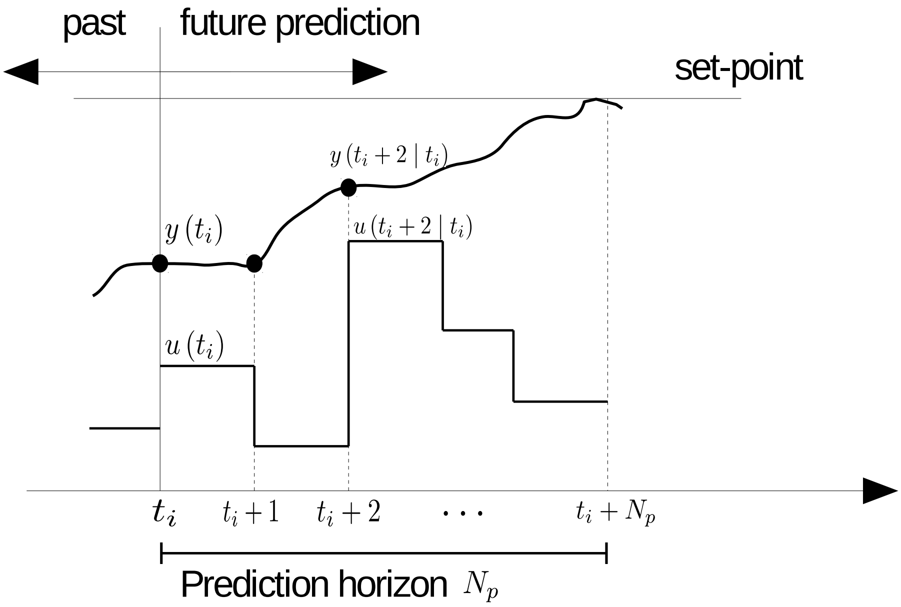

Figure 3.15: Construction of the control action using model predictive control.

$$J\left(x_0,\{u_0,u_1,u_2,...\}\right)=\sum_{k=0}^{\infty}\left(\|x_k\|_Q^2+\|u_k\|_R^2\right) \tag{3.47}$$

with $\|\cdot\|^2$ is the quadratic form and Q, R are the weighting matrices on states and inputs. It is assumed that R is a symmetric positive-definite matrix and that Q is symmetric and positive semidefinite [84]. The optimal value of the cost (3.47) with respect to minimization over admissible control sequences $\{u_k,u_{k+1},u_{k+2},...\}$ is

$$J\left(x_k\right)=\min_{u_k,u_{k+1},u_{k+2},\ldots} J\left(x_k,\{u_k,u_{k+1},u_{k+2},...\}\right) \tag{3.48}$$

This formulation leads to an optimal control problem, the controller is required to minimize at time k the performance cost 3.47 subject to the constraints imposed in the states and the inputs. More details on MPC can be found in [84] and [85].

As mentioned at the beginning of this section, the MPC is used to guide the platform to a desired position, which corresponds to the point-stabilization or regulation problem. To perform this task, the differential drive model similar to the presented in Eq. (3.21) is considered:

$$\dot{\boldsymbol{x}}=\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix}=\begin{bmatrix}v_c\cos\theta\\ v_c\sin\theta\\ \omega_c\end{bmatrix}=\begin{bmatrix}\cos\theta & 0\\ \sin\theta & 0\\ 0 & 1\end{bmatrix}\boldsymbol{u} \tag{3.49}$$

with the linear velocities of the right υ_R and the left υ_L wheels define the linear velocity $\upsilon_c = \frac{\upsilon_R + \upsilon_L}{2}$ and the angular velocity $\omega_c = \frac{\upsilon_R + \upsilon_L}{2L}$. The state and control vectores are $\boldsymbol{x} = (x, y, \theta)^T$ and $\boldsymbol{u} = (\upsilon_c, \omega_c)^T$, respectively.

A reference differential drive model is defined, it possesses the desired values for the state and control vectors. In the case of regulation problem, the values of $\boldsymbol{x}_r$ and $\boldsymbol{u}_r$ are constants. For the trajectory tracking problem, the state $\boldsymbol{x}_r$ and control $\boldsymbol{u}_r$ vectors have time varying values.

$$\dot{\boldsymbol{x}}_r = \begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} \upsilon_r \cos\theta_r \\ \upsilon_r \sin\theta_r \\ \omega_r \end{bmatrix} = \begin{bmatrix} \cos\theta_r & 0 \\ \sin\theta_r & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{u}_r \tag{3.50}$$

where $(x_r, y_r, \theta_r)^T$ is the reference state vector and the reference control vector $\boldsymbol{u}_r = (\upsilon_r, \omega_r)^T$. The goal is to control (3.49) to the reference (3.50). Thus the error state vector can be defined as:

$$\boldsymbol{x}_e = \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \tag{3.51}$$

The error dynamic model is obtained by differentiating (3.51)

$$\begin{aligned} \dot{x}_e &= \omega_c y_e - \upsilon_c + \upsilon_r \cos\theta_e \\ \dot{y}_e &= -\omega_c x_e + \upsilon_r \sin\theta_e \\ \dot{\theta}_e &= -\omega_r - \omega_c \end{aligned} \tag{3.52}$$

The linearized version of (3.52)

$$\dot{\boldsymbol{x}}_e = \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} 0 & \omega_r & 0 \\ -\omega_r & 0 & \upsilon_r \\ 0 & 0 & 0 \end{bmatrix} x_e + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{u}_e \tag{3.53}$$

where

$$\boldsymbol{u}_e = \begin{bmatrix} -\upsilon_c + \upsilon_r \cos\theta_e \\ \omega_r - \omega_c \end{bmatrix} \tag{3.54}$$

Having defined the controlled and the reference models, a control input $\boldsymbol{u}(t)$ has to be computed minimizing a weighted cost function given by:

$$J\left(t, \boldsymbol{x}_e\left(\tau\right), \boldsymbol{u}_e\left(\tau\right)\right) = \int_t^{t+\tau} l\left(\tau, \boldsymbol{x}_e\left(\tau\right), \boldsymbol{u}_e\left(\tau\right)\right) d\tau \tag{3.55}$$

where $l\left(\tau, \boldsymbol{x}_e\left(\tau\right), \boldsymbol{u}_e\left(\tau\right)\right) = \boldsymbol{x}_e\left(\tau\right)^T Q \boldsymbol{x}_e\left(\tau\right) + \boldsymbol{u}_e\left(\tau\right)^T R \boldsymbol{u}_e\left(\tau\right)$. With Q and R are the weight matrices and T is the prediction horizon. Which can be written as:

$$\min_{\boldsymbol{u}} J\left(t, \boldsymbol{x}_e\left(\tau\right), \boldsymbol{u}_e\left(\tau\right)\right) \tag{3.56}$$

$$\begin{aligned} \text{subject to:} \quad & \dot{\boldsymbol{x}}\left(\tau\right) = f\left(\boldsymbol{x}\left(\tau\right), \boldsymbol{u}\left(\tau\right)\right) \\ & \boldsymbol{u}\left(\tau\right) \in U, \left(\tau \in \left[t, t+T\right]\right) \\ & \boldsymbol{x}_e\left(t+T\right) = 0 \end{aligned} \tag{3.57}$$

3.5 Simulation results

Digital twins are being adopted in robotics to recreate conditions of real-life scenarios. In particular, these tools are very useful in agriculture, where access to experimental fields involves high costs of displacement which could be further complicated by unpredictable weather and crop conditions.

To validate the proposed algorithms in this chapter, multiple scenarios were built using the high-fidelity Gazebo simulator and the ROS middleware to emulate the real platforms, its sensors and the crop fields. The Figure 3.16 shows the platforms employed in this work. For more details of the platforms and their instrumentation, see Appendix A.2.

Multiple scenarios were created to replicate the conditions of the ROSE challenge (see subsection 1.4) and the ACRE competition (see subsection 1.5). Moreover, to validate the robustness of the navigation algorithm, some parcels were modified adding some noise in the plant position, so that crops were not perfectly aligned (see Figure 3.17). In the same way, a portion of parcels were created with a distinct crop distribution.

One of the first tests carried out in the simulator was to evaluate the effect of changing the position and orientation of the frontal sensor. As expected, the number of visible plants projected on the camera will differ according to the camera orientation. Similarly, the illumination perceived by the camera will be different depending on its position and orientation.

The simulation suggests that the perception system is able to detect the num-

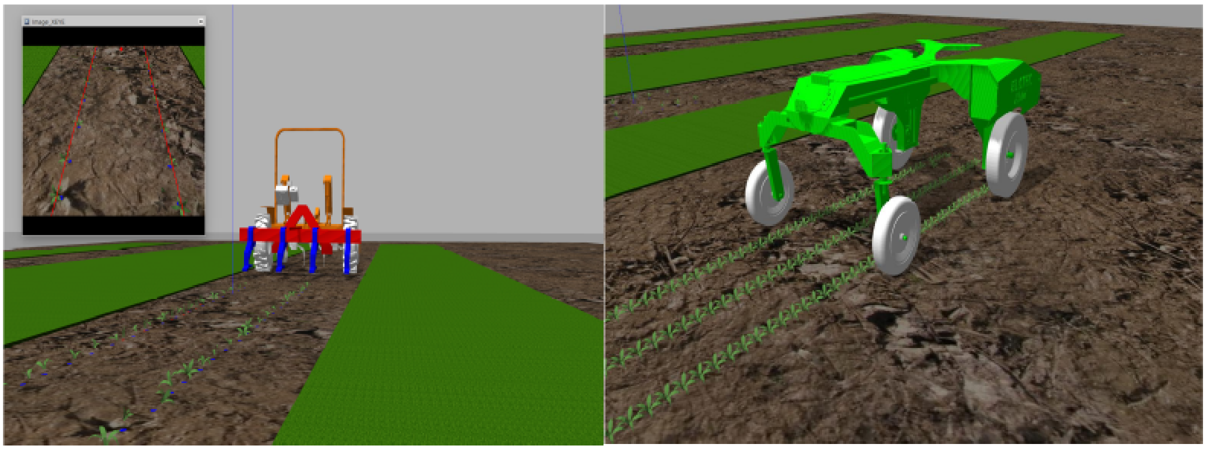

Figure 3.16: Platforms on Gazebo simulator. On the left side of the image, the differential drive configuration and on the right, the Ackerman steering platform.

ber of plants needed to perform the navigation at an orientation between 30° and 60° with respect to the ground. For the altitude, a distance of 1.2 m with respect to the ground was considered.

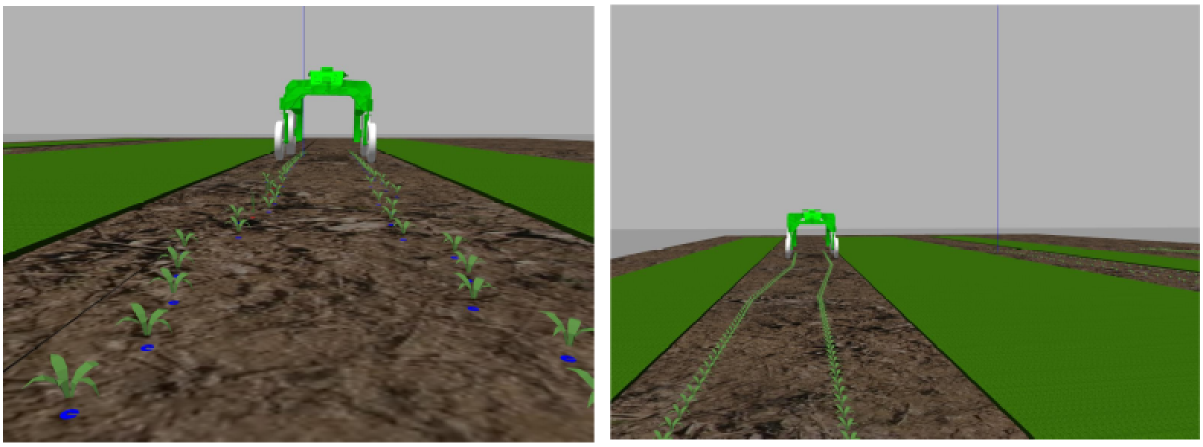

Figure 3.17: Examples of parcels built in Gazebo. On the left: a Maize parcel with artificial markers (part of ROSE challenge scenarios). On the right: a Maize parcel without artificial markers (part of ACRE competition).

Once the frontal sensor orientation and position have been determined, the navigation was tested. Four situations were considered:

1. The initial position of the platform is aligned with respect to the crops as is illustrated to the left of the Figure 3.17.
2. The initial position of the platform is not aligned with respect to the crops, but crops remain into the field of view of the camera.
3. The initial position of the platform is far away from crops and, without relying on

visual perception, the platform is guided at the beginning of one of the parcels using the MPC approach.

4. The complete navigation scenario of the ACRE competition, including maize and bean crops, transition zones and half-turn.

All the results presented below were obtained considering a constant linear velocity $v_c = 0.4$ m/s, $\delta_{res} = 80$ pixels for the maize parcel and $\delta_{res} = 60$ pixels for bean parcel.

The results of the first scenario (initial position of the platform aligned with respect the crops) are shown in Figure 3.18. The vanishing point estimation for both crops: maize and bean remains around the reference during the autonomous navigation, the reference being at the image center with time constant $\lambda = \dfrac{1}{\tau}$ and $\tau = 2.5$.

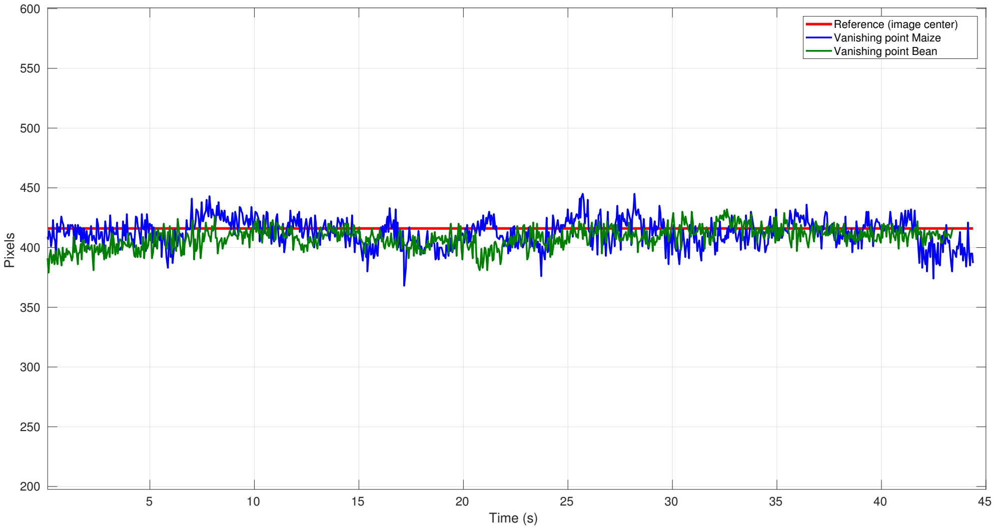

Figure 3.18: Vanishing point estimation during autonomous navigation in maize and bean crops. The initial position of the platform was aligned with respect to the crops.

The second scenario was planned to prove the robustness of the proposed approach. Due to the platform is not aligned with respect to the crops at the beginning of the simulation, a larger initial error between the reference and the vanishing point estimation is present, the value of time constant is $\tau = 3$ However, the results indicate that the trajectory correction algorithm is able to rectify the position of the platform and navigate through the parcel (see Figure 3.19).

To solve the cost function J in the model predictive control, the open source tool for nonlinear optimization CasADi [86] was used. The prediction horizon was selected $T = 10$, and the considered constraints were $v_{c_{max}}$ = 0.2 m/s and $\omega_{c_{max}} = \dfrac{\pi}{8}\ rad/s$.

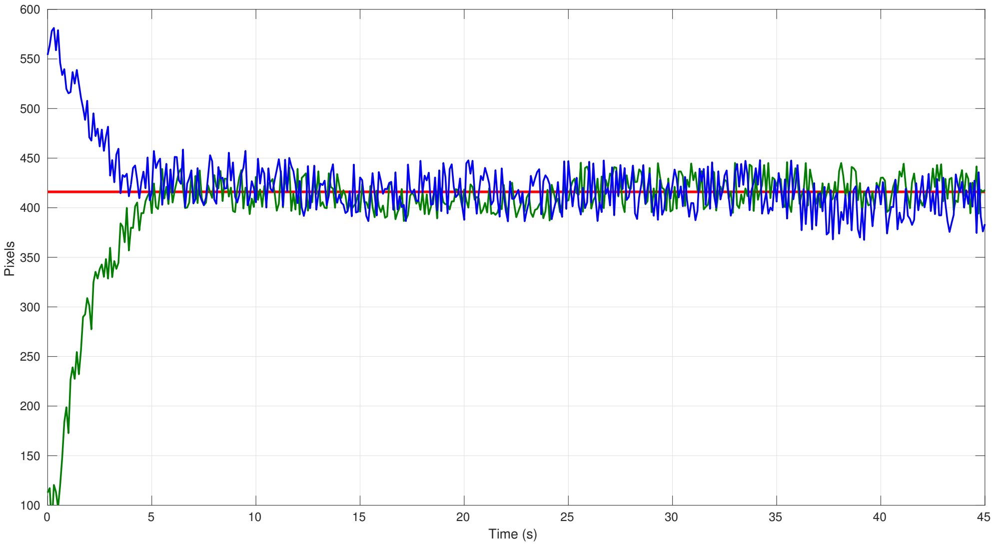

Figure 3.19: Vanishing point estimation during autonomous navigation in maize and bean crop. The initial position of the platform was not aligned with respect to the crops.

The weight matrices were chose:

$$Q = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0.5 \end{bmatrix} \quad , \quad R = \begin{bmatrix} 0.8 & 0 \\ 0 & 0.1 \end{bmatrix}$$

The initial position of the platform was selected at the origin of the World coordinate frame. However, the initial orientation of the platform was not perfectly aligned with the crops, that is $x_0 = \left(0, 0, \frac{\pi}{8}\right)$. The desired pose corresponds with the beginning of one of the parcels $x_d = (15, 5, 0)$. The results of Figure 3.20 show the trajectory performed by the platform to reach the beginning of the parcel considering the velocity constraints imposed.

The values of the input vector generated by the model predictive control are shown in Figure 3.21. Both velocities are restricted in magnitude due to the considered constraints during the optimization of J.

Finally, the last results presented in this subsection correspond to the field navigation benchmark of the ACRE competition. The initial position of the platform is centered with respect to the crops. The platform carried out the autonomous navigation through four parcels (two maize parcels and two bean parcels) according to the Algorithm 2.

This scenario allows to test all the techniques explained in the previous subsections simultaneously. As is illustrated in Figure 1.12, the first section corresponds

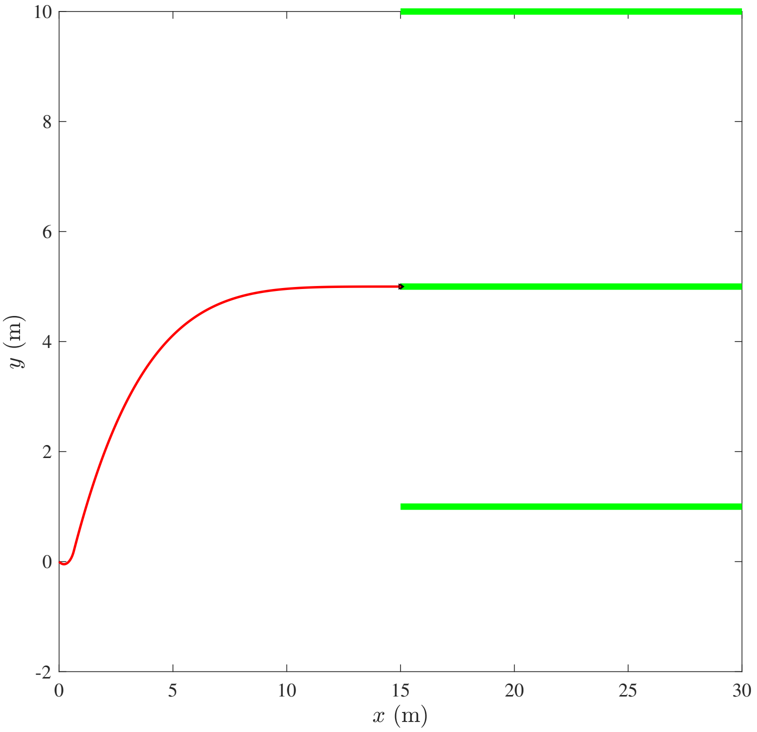

Figure 3.20: Platform's trajectory to reach one of the parcels based on model predictive control approach.

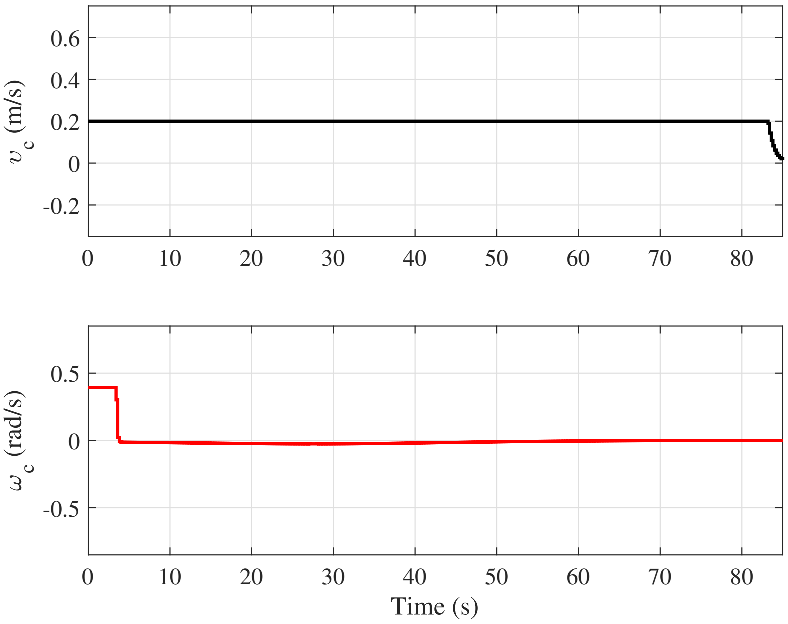

Figure 3.21: Linear and angular velocities generated by the model predictive control.

Algorithm 2 Autonomous navigation

```
Require: RGB image
Ensure: $(\upsilon_c, \omega_c)^T$
Initialization:
Accumulator $R = 0$
Accumulator $T = 0$
$\mathcal{W}_s$
Iteration:
Plant detection
Vanishing point estimation
if (Vanishing point is available) then
    Navigation using vanishing point
    Accumulator $R = 0$
else
    Navigation using virtual lines
    Accumulator $R++$
    if ($R > \delta_R$ and line fitted by $\mathcal{W}_s$) then
        Navigation using bird's-eye view
        Accumulator $T++$
    end if
    if ($T > \delta_T$ and vegetation into $\mathcal{W}_s$) then
        Half-turn
        Accumulator $R = 0$
        Accumulator $T = 0$
    end if
end if
```

to a maize parcel (two crop lines) which uses the same navigation task presented in the ROSE challenge. Then, a transition zone without crops divides the first and the second parcel, the second parcel was sowed with beans in three rows. Crops were sowed in parallel lines in these parcels.

During the transition zone, the vanishing point detection is not available, the bird's-eye view transformation is used to guide the platform until the crops are again observed and the vanishing point can be estimated. At the end of the second parcel, a half turn was executed to change row. Due to the experimental field conditions, during the half turn visual information is not available. For this reason, the platform uses the absolute position given by the GPS guidance system and, using the MPC approach, it can change to a new parcel. At the moment the vanishing point is available again, the platform will navigate based on visual perception. The vanishing point estimation for each parcel can be seen in Figure 3.22.

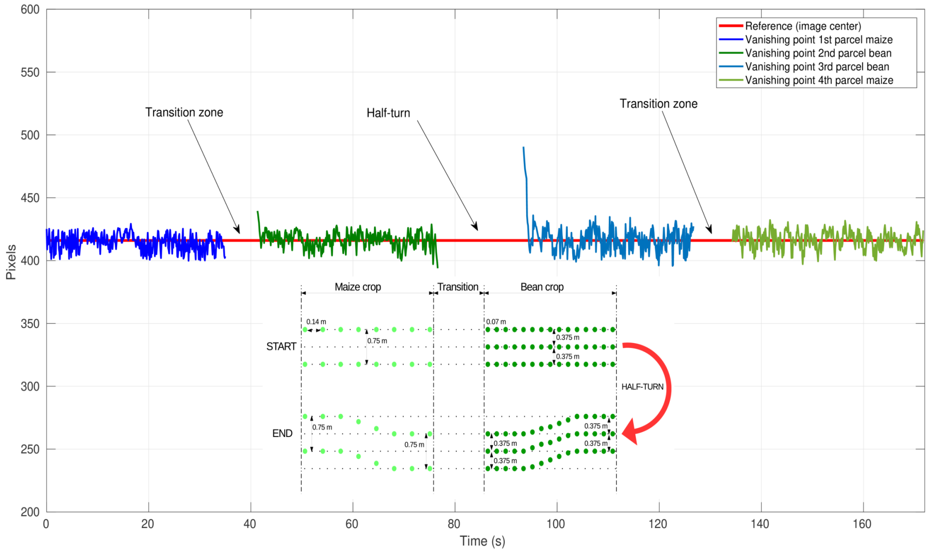

Figure 3.22: Vanishing points estimation in Gazebo simulator within the ACRE scenario.

3.6 Experimental results

The results presented in this section were obtained during the ROSE challenge (subsection 1.4) evaluation campaigns and the ACRE competition (subsection 1.5). For a detailed description of the experimental platforms and their instrumentation, see Appendix A.2.

The classification algorithm was tested in two different campaigns; the first took place in September 2019 under clear sky and sunny weather, and the second in September 2020 under cloudy sky and cold drizzle.

The growing stages of the crops for each evaluation were not the same, they were more advanced in the evaluation campaign of 2019 in comparison to 2020 (see Figure 3.23).

To evaluate the classifier output quality, the Precision and Recall performance metrics introduced in Chapter 2 are used. It is based on the well-known *confusion matrix* shown in Table 3.1, which expresses all the combinations between the predicted and actual values.

The classification decision is determined by comparing the output of the proposed classification algorithm against a human expert, that is, for each image, the values of TP, FP and FN were counted.

The results obtained in Figure 3.24 show a better performance of the proposed algorithm in the maize parcel with respect to the bean parcel, this is because the

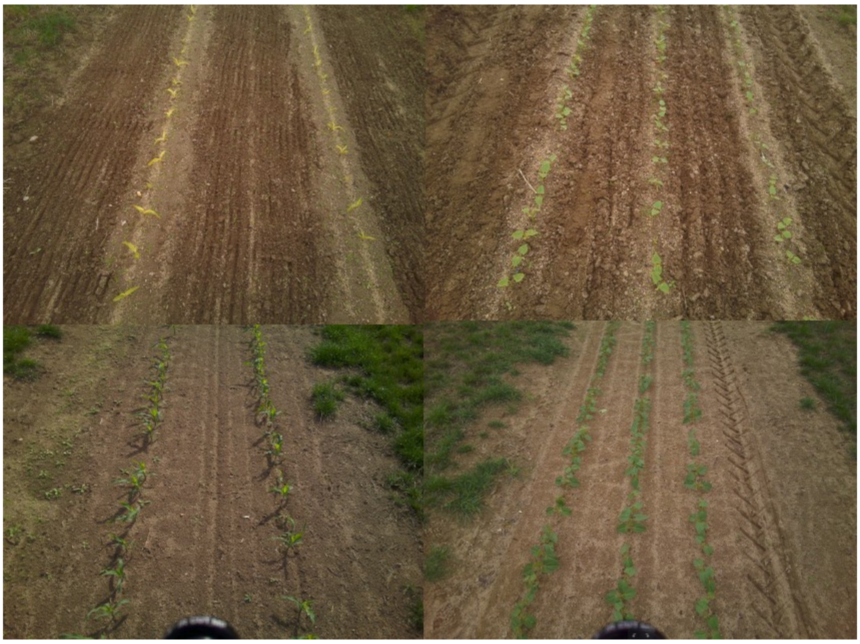

Figure 3.23: Example of images extracted from datasets recorded during the challenge campaign September 2019 (bottom) and September 2020 (top).

Table 3.1: Confusion matrix.

Actual values	
True Positive (TP) *Plant of interest identified correctly*	**False Negative** (FN) *Plant of interest not identified*
False Positive (FP) *Plant of interest identified incorrectly*	**True Negative** (TN) *Not identified as plant of interest*

intra-row distance is longer in maize and the weed density is more significant for beans. In the case of beans, crops are closely spaced and phenomena such as occlusion disturb the classification. Another point to note is that the precision was bigger in the second evaluation campaign for both crops, this is explained by in the first evaluation campaign, crops' growing stage was more advanced and their physical characteristics were less distinguishable, as Figure 3.23 illustrates.

In order to measure the performance of the proposed approach with respect to other solutions, an accuracy comparison between the algorithm based on feature characteristics and the AQiT sensor is presented. The same dataset obtained during the evaluation campaigns (September 2019 and September 2020) was utilized. The obtained results are shown in Table 3.2.

Among the points to highlight from Table 3.2, the first and very important is the operation rate. The AQiT sensor performs crop detection at $f = 5$ Hz at best, which implies some restrictions for real-time applications. The results given by the proposed

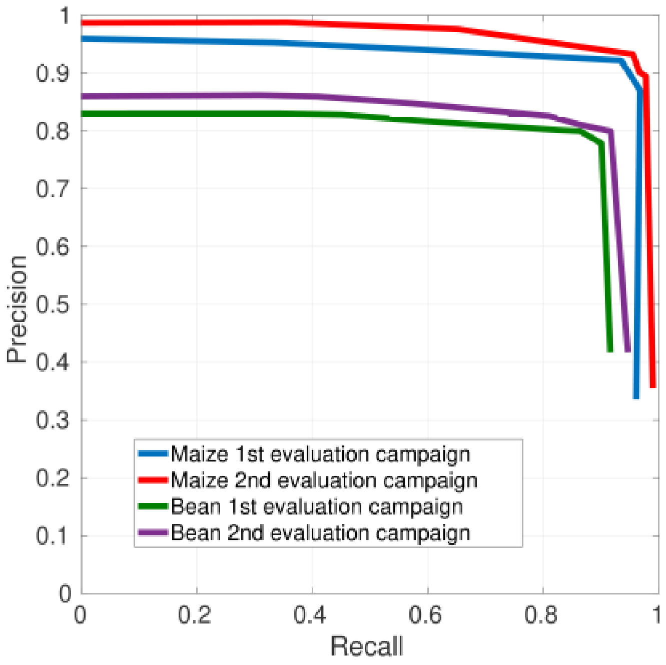

Figure 3.24: Precision-Recall curves for maize and bean crop classification.

Table 3.2: Crop detection accuracy: feature-based and AQiT sensor.

Solution	**Operation rate** (Hz)	**Maize**(1st campaign)	**Bean** (1st campaign)	**Maize** (2nd campaign)	**Bean** (2nd campaign)
AQiT sensor	5	76%	74%	63%	62%
Proposed feature extraction detection	20	94.33%	81.75%	97.09%	83.8%

approach were obtained at $f = 20$ Hz, allowing to implement the algorithm in real-time onboard an embedded system. In the maize case, the differences between the obtained results were even more noticeable in the second evaluation campaign. It is associated with not retraining the AQiT sensor and like plant characteristics were not the same, the accuracy was disturbed.

The autonomous navigation was tested with two different platforms and the navigation task was not the same. Figure 3.25 shows the errors between the estimated vanishing points and the reference. They were obtained within the framework of the ROSE challenge (September 2021) where the task aimed at navigating autonomously one full parcel. The linear velocity of the platform was constant $\upsilon = 0.2$ m/s. The

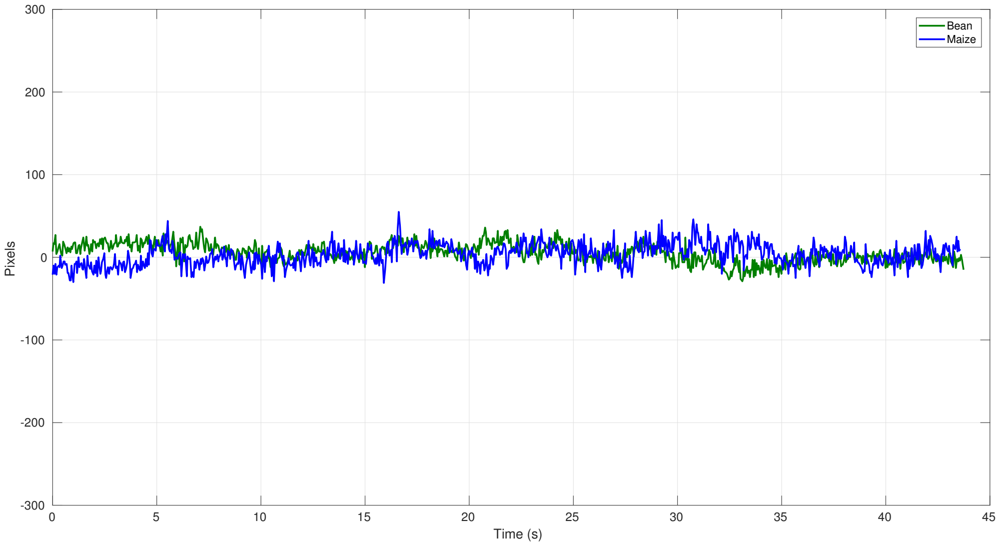

Figure 3.25: Error evolution with differential platform during the ROSE challenge.

task started with the platform centered at the beginning of the parcel and ended after crossing the parcel until its final point.

Although GPS was not used for the navigation task in this work, to have a reference of the trajectory carried out by the platform, two GPS-RTK were mounted in front and at rear of the platform. With the GPS data, the position and the orientation of the platform during the navigation task were estimated and used as ground truth to evaluate the vision-based navigation. The platform pose was overlapping with a seed map as Figure 3.26 shows.

The second autonomous navigation test was carried out with an Ackermann steering configuration platform. The crops and weeds were the same as in the previous scenario, but the plant's sow distribution was modified for the ACRE competition.

To cross from the first parcel to the second one when the absence of plants does not allow to estimate the vanishing point, the platform automatically switches to virtual visual servoing using the lane formed by the non-vegetation generated by the bird's-eye view. When plant detection becomes available again, the vanishing point estimation leads the platform to be aligned with respect to the plants, reducing the error tracking as Figure 3.27 shows.

For the half-turn at the end of the second parcel, as was mentioned, the proposed approach does not rely on any GPS reference point. To substitute the reference points, the previous platform characterization allows to known the steering angle for approximating the platform to the third parcel until vanishing point estimation was

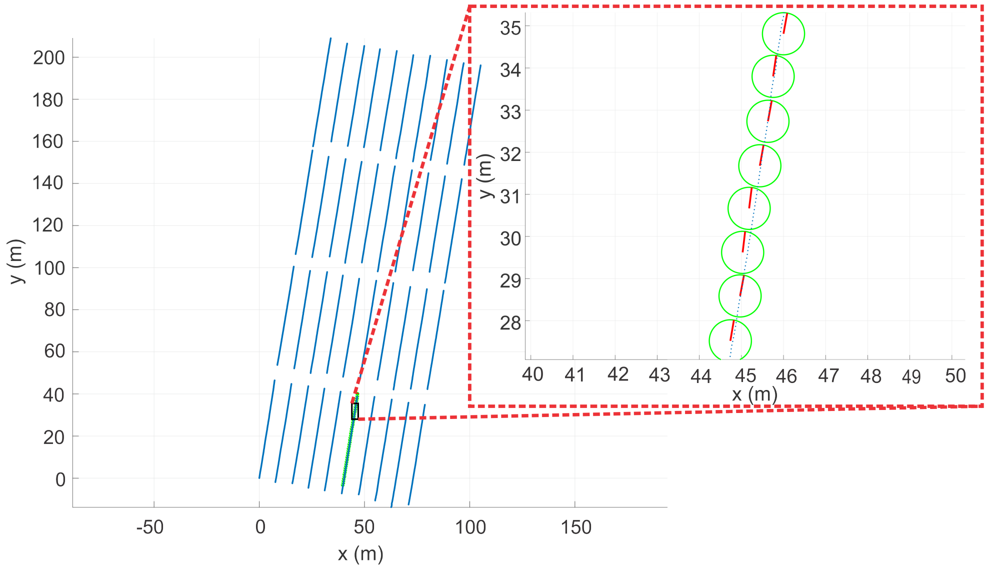

Figure 3.26: Trajectory estimation on one crop parcel using the onboard GPS and projected on the crop field. The display included (right) is a zoom of the position and orientation of the platform on a part of the trajectory. The red lines correspond with the platform's orientation.

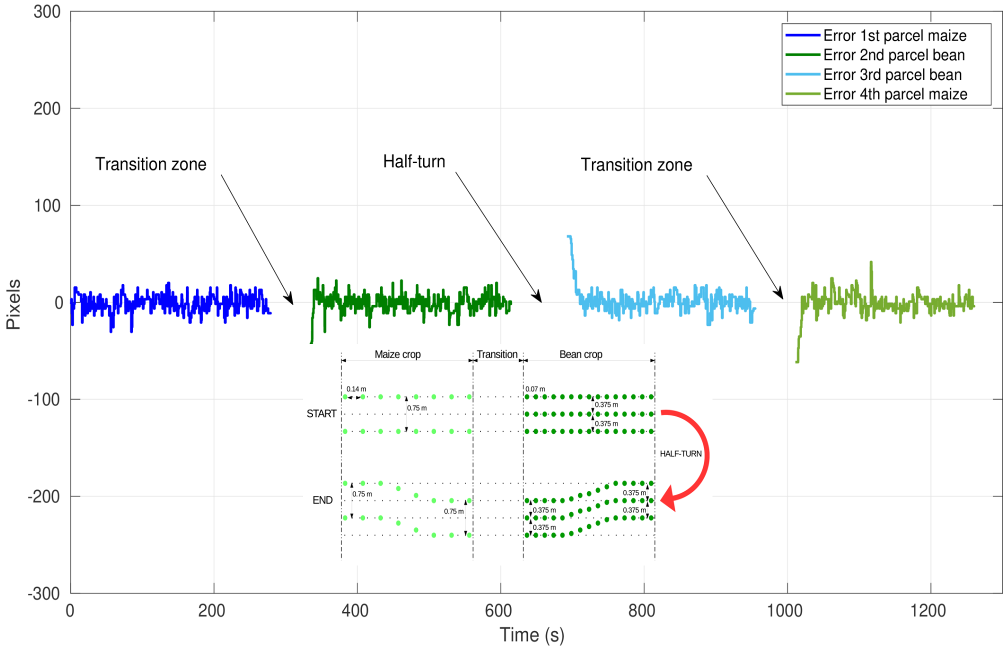

Figure 3.27: Error tracking during the experimental campaign of the ACRE competition.

available again. In Figure 3.27, the larger error at the beginning of the third parcel (light blue color) is because after the half-turn of the platform, the vanishing point appears at the edge of the images. As the robot approaches to the center of the crops, the error decreases.

The final segment of the navigation benchmark allows to verify the scope of the proposed approach. The navigation algorithm not only performs on straight paths but on curved paths. The orientation of the frontal sensor (Xeye) plays a crucial role. The more perpendicular the angle formed by the optical axis of the camera $\vec{k}_{Xeye}$ and the ground, the effect of not straight path is mitigated. However, the vanishing point estimation tends to disappear as line projection differs from the projection assumed in this work. In the opposite case, the reliability of the line extraction reduces considerably. An adequate orientation allows to perceive locally straight paths even when it is not true, as Figure 3.28 shows.

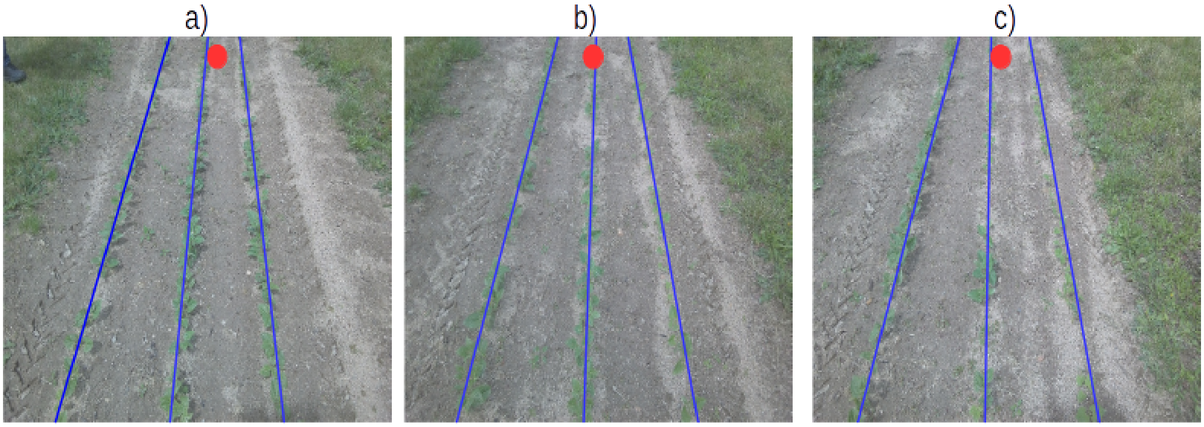

Figure 3.28: Images extracted from the autonomous navigation. a) At the beginning of the parcel (straight path). b) At the middle of the parcel (not straight path). c) At the last segment of the parcel (straight path).

The same idea to integrate the two GPS-RTK to the platform was carried out for evaluation purposes. The full trajectory accomplished by the platform can be seen in Figure 3.29. The zooms presented to the right of the Figure 3.29 corresponds to the first and second parcels (straight path formed by the crops) and to the left of the image, are the final parcels, which sow was modified to create a curved path. The zoom at the bottom of the image illustrates the half-turn.

3.7 Conclusion

An autonomous vision-based navigation method was exposed in this chapter. The plant classification algorithm based on physical features was successfully tested with higher accuracy in maize (93%) with respect to bean crops (83%). Considering that the detection presented in this chapter is used for navigation purposes, the plant classification results are good enough to achieve this purpose. The proposed approach

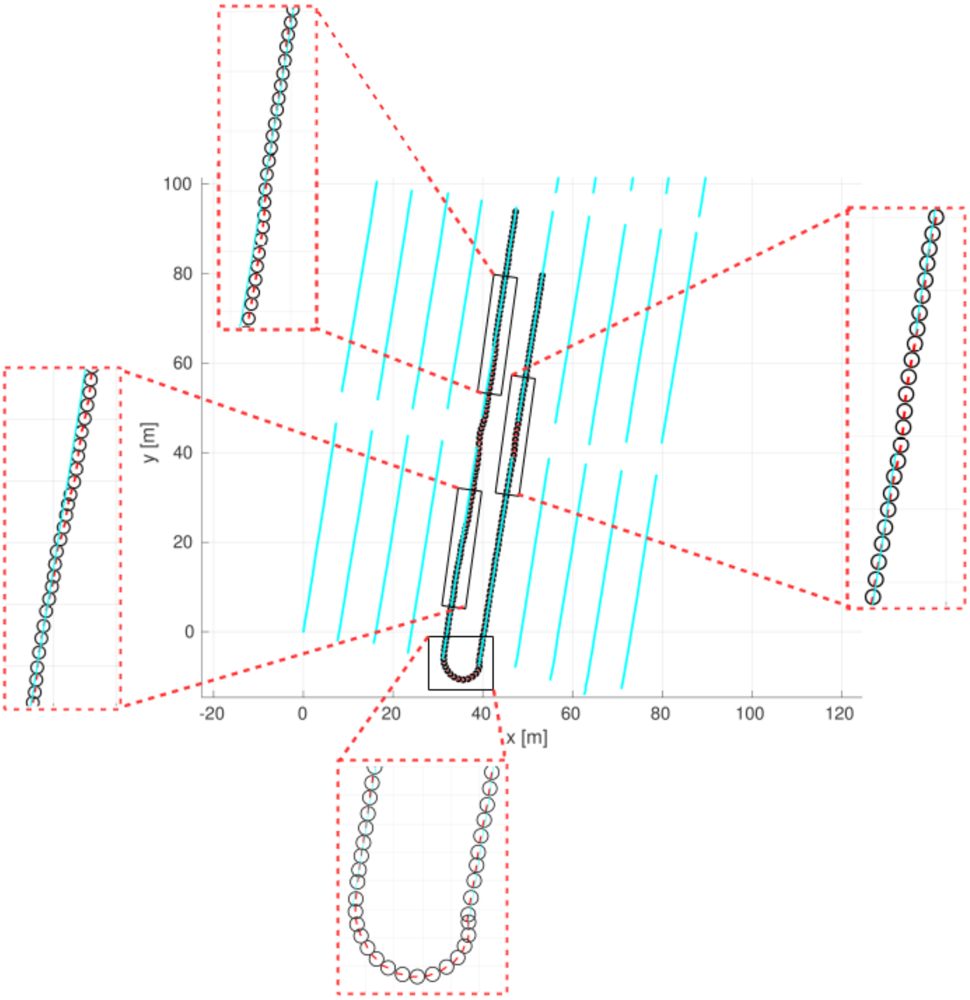

Figure 3.29: Field navigation benchmark during the ACRE competition. The dotted rectangles are a zoom of some regions to highlight the platform orientation. The vehicle's pose is given by the RTK-GPS system.

is based on crop physical characteristics which provide significant advantages: it allows to adapt the algorithm to a great variety of plants. Early information about the growing stage of the crops was considered to adjust the algorithm parameters and it can be implemented in real-time onboard an embedded system.

The obtained results suggest that vanishing point estimation can be utilized to perform autonomous navigation in agricultural scenarios. The line extraction presented in this chapter is able to extract crop lines under diverse adverse scenarios: in the presence of false plant detection, change of illumination during experimental tests, at different camera positions, to mention a few. In the same way, the least squares solution proposed to estimate the vanishing point is a fast solution that allows to find the position of the vanishing point for any number of crop lines.

The transition to the bird's-eye view guidance for navigation allows the platform to navigate in the absence of the vanishing point. Furthermore, the use of the Gazebo simulator to create realistic scenarios enabled to execute multiple tests to validate the proposed approaches considering the difficulties of finding a real experimental field.

Based on the results shown in this chapter, it might be concluded that visual information can be extracted without using techniques of artificial intelligence to perform autonomous navigation. However, in the zones where crop extraction is not available, the model predictive control is an alternative to perform this task. Moreover, MPC allows to integrate different constraints to estimate the control vector.

4

Weeding actuation and robot localization

In the previous chapters, features extraction given by the dual-purpose plant recognition system and the autonomous navigation of the platform were presented. To complete the chain: detection-navigation-action, the weeding actuation is exposed in this chapter. The proposed approaches are based on proximal and frontal detection to accomplish a mechanical weeding. The first one controls the weeding tool locally using the segmentation algorithm executed in the proximal multimodal sensor (Xtool). The second one proposes a plant propagation system to activate the weeding tool, given the plants detected by the frontal sensor (Xeye), they are propagated from the camera's field of view to the weeding tool activation. Both approaches operate under the principle to protect the plants of interest and eliminate everything else that is not detected. An Extended Kalman Filter is conceived as data fusion algorithm to estimate the robot's localization when needed. Simulations and experimental results validate the proposed approach to carry out intra-row autonomous weeding.

Table of contents

4.1 Introduction

Weeds are plants that grow in an unwanted place. When they are located close to the crops, the crop growing stage is perturbed due to part of the natural resources are absorbed by the weeds, for instance: the amount of water during irrigation, solar energy, surface where they are, among others. In the same way, weeds harbor species that can contaminate or even kill crops. For these reasons, weeding has been carried out since the beginning of agriculture to remove undesired plants. Until not long ago and even nowadays, it is performed by manpower.

World population growth forces to increase substantially food manufacturing. Consequently, the eradication of weeds plays a fundamental role in crop production. To achieve efficient weeding control, classic methods such as hand weeding by farm workers or the use of chemical substances that affect health and the environment must be replaced by new technologies that increase weeding speed and limit repercussions on the environment or on human health.

Among the factors to be considered in the weeding task are: crops and disease detection, the characteristics of the parcel and the type of mechanisms to be deployed. In this chapter, a mechanical weeding is exposed. The mechanical weeding tools are located under the experimental platform. The proposed approach protects crops identified as plants of interest and eliminates weeds.

4.2 Weeding strategies

The weeding task can be divided into two categories: inter-row [87] and intra-row [75], [87]. The first consists of removing the weeds that are between the crop rows, which is not very difficult since the distance between crop rows is usually large enough to use a tool (mechanical, electrical, laser device, etc.) and crossing the parcel to eliminate them. However, the intra-row weeding is a great challenge because this task is devoted to eliminating weeds located between crop rows and, according to the kind of crops, the distance between two consecutive plants can be very small, increasing the difficulty to eliminate weeds without damage the plant of interest.

The research platform WeedElec [88] is a weeding solution based on high-voltage probe. The contactless discharge is generated between the probe and the weed detected. The current induced by the discharge flows through the whole weed. The spark duration can be between 0.25 and 1 second, the electrode material is stainless steel anode and the voltage is around 60 kV pulses. Although the weeding task has begun to be carried out using electrical approaches, mechanical weeding predominates among the solutions to this problem. Diverse mechanical tools have been implemented to compare the results obtained for each one of them.

An evaluation of the performance of some mechanical weeding mechanisms was proposed in [89]. These are the harrow, sweep, ducksfoot, rotary powered hoe and rotary brush devices. Harrow weeders (Figure 4.1-a) can operate for both inter and intra-row weeds but when weeds have deep roots they are ineffective. Weeders such as sweep and ducksfoot (Figure 4.1-b and d) are used to eliminate inter-row weeds. The sweep tools consist of a set of blades and the weed control is carried out by cutting their roots below the surface. The duckfoot operates in a similar way to the sweep, but the working depth increases with respect to the sweep, from 25 mm to 40 mm. Rotary brush and hoe (Figure 4.1-c and e), are weeding tools that rotate around an axis, this axis can be vertical or horizontal and they can be used for both types of weeding. These rotary tools operate better during the first weeks of the growth stage.

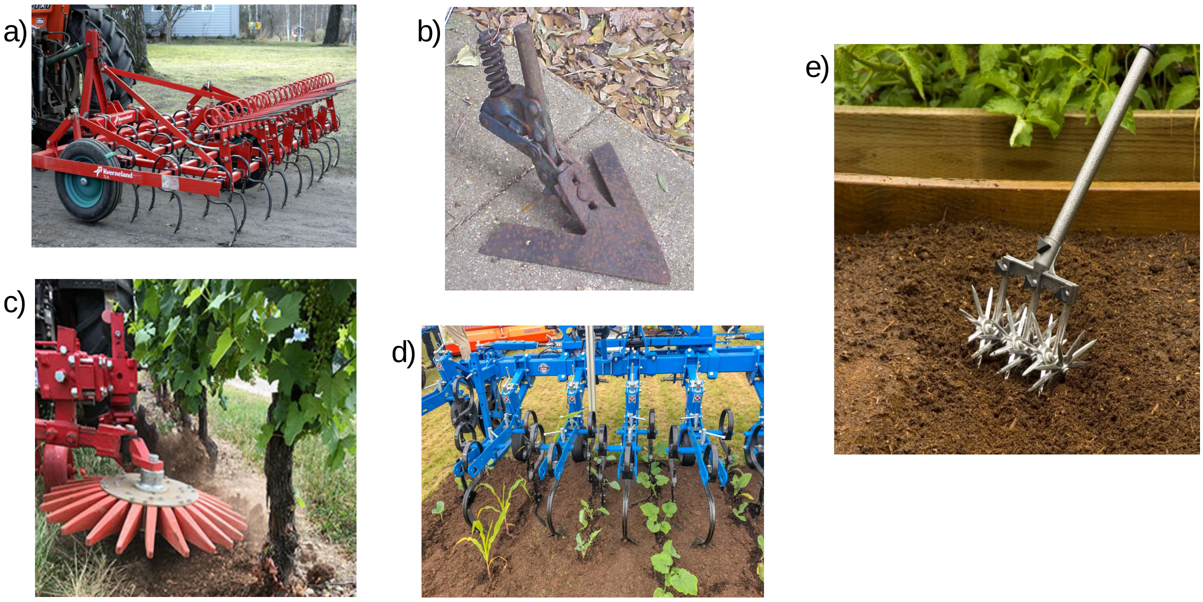

Figure 4.1: Mechanical weeding tools. a) Harrow. b) Sweep. c) Rotary brush. d) Ducksfoot. e) Rotary hoe.

A comparison between three mechanisms was presented by *McCool et al.* [90]: an arrow hoe for tilling below surface, tines for tilling above surface and a cutting mechanism. The authors applied the three mechanisms over a period of eight weeks. The results show that as the plants grow, the effectiveness of the mechanisms tends to decrease and the cutting mechanisms presented the lowest success rate among them.

4.3 Weeding actuation

This work proposes a mechanical weeding mechanism to carry out inter and intra-row autonomous weeding. The weeding tool used corresponds to the sweep shown in Figure 4.1-b, despite the fact this type of tool is mainly used to perform inter-row weeding, the approach presented in this chapter allows to carry out both types of weeding.

To carry out inter-row weeding, the tools are placed in front of the platform with their tips pointing in the same direction of the platform movement. While to carry out intra-row weeding, the tool is placed at the rear of the platform with the tip pointing perpendicular to the direction of the platform movement as shown in Figure 4.2.

Figure 4.2: Mechanical tools placed in front and at the rear of the platform for inter-row and intra-row weeding.

The weeding tools at the front of the platform are always on (below the ground). So, as the platform moves, the tools remove any weed located between the crop rows. The intra-row weeding is carried out by activating and deactivating the weeding tool intermittently: as previously mentioned, the proposed approach aims to eliminate everything that is not detected as a plant of interest. Similar to inter-row weeding, the weeding tool remains active while the platform is moving (weeding tool below the ground) but when a plant of interest is detected, the tool is disabled to protect the detected crop (by disabling the tool, its position is above ground). The Figure 4.3 illustrates the previous explanation.

To achieve the weeding, some assumptions were made:

- During the displacement through the parcel, the autonomous navigation algorithm allows the platform to remain close enough to the crops and weeds, so that the weeding tool when activated will be below ground level, allowing the removal of the weeds.

- The angle β formed by the position of the weeding tool (Figure 4.3-b), allows crossing the crop line from one side to the other, ensuring intra-row weeding.
- The weeding tool is able to penetrate different types of soil under different weather conditions: dry ground, wet ground, etc.

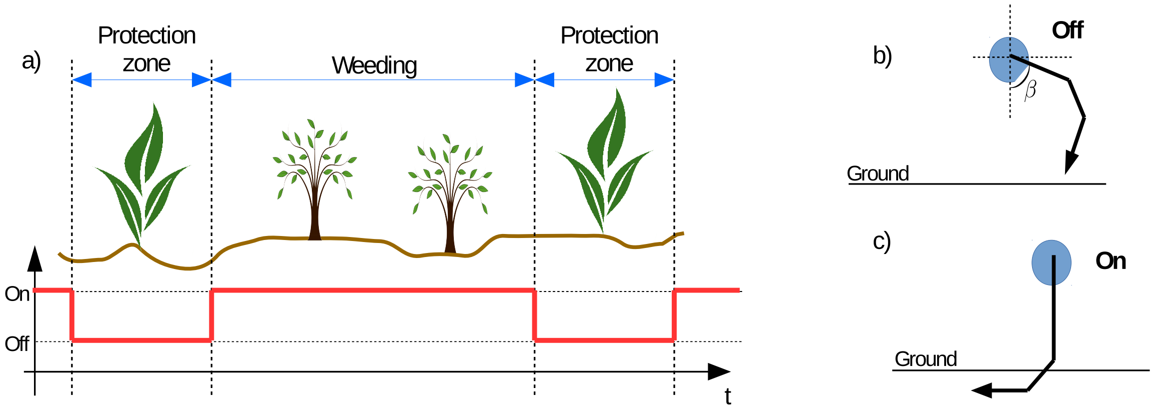

Figure 4.3: Intra-row weeding scenario. a) The weeding tool is disabled when a plant of interest is detected. b) Weeding tool disabled. c) Weeding tool activated.

In order to properly control the weeding tool, its kinematic needs to be estimated. This is accomplished using the parameter estimation approach based on the step response for the open loop system [91]. Among the different systems, the first order systems with time delay are described by a first order differential equation and their main dynamic is damped.

Let a first order system $G(s)$ given by:

$$G\left(s\right)=\frac{k}{Ts+1}e^{-\tau s} \tag{4.1}$$

with k the steady state gain, T the time constant and τ the time delay. These three parameters are determined by drawing a tangent line at the inflection point of the system step response and its intersection with the coordinate axes. The steady state gain k is obtained from the intersection of the tangent line with the y-axis, the time constant T is given by the double intersection of the tangent with the x-axis and the time delay τ is the difference between the time at which the step is applied and the time when the system reacted.

The values of the estimated parameters are given in (4.2). The Figure 4.4 shows the step response of the weeding tool as well as its identified dynamics.

$$G\left(s\right)=\frac{-70}{0.04s+1}e^{-0.015s} \tag{4.2}$$

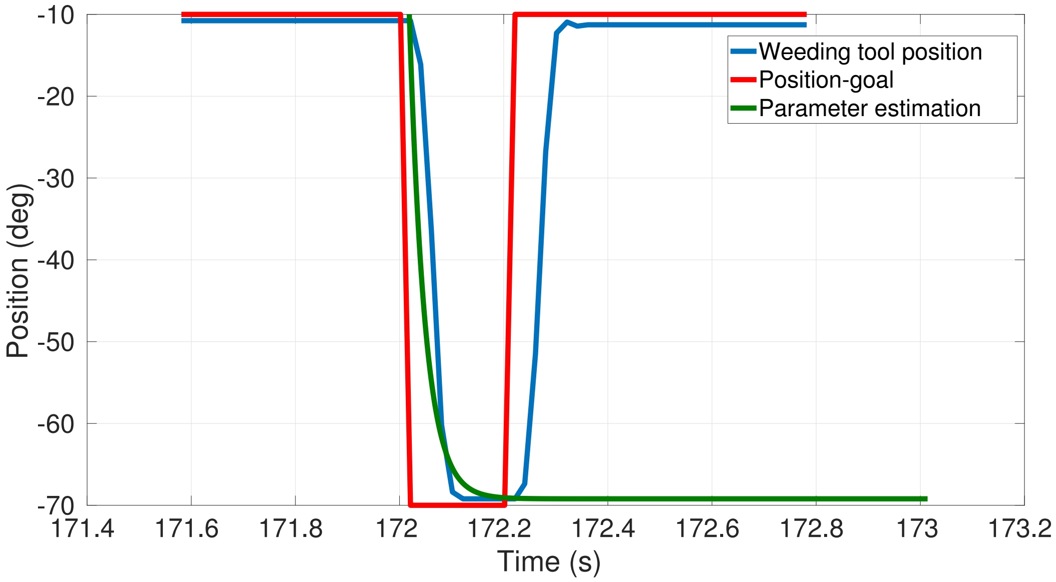

Figure 4.4: The step response of the weeding tool and its parameter estimation.

Once the parameters of the tool have been obtained, the height at which the tool will be fixed must be determined. To ensure that weeds will be destroyed not only superficially but also from their roots, the weeding tool must penetrate the soil at a certain depth. Figure 4.5 shows the size of the roots of some extracted weeds. The depth range is between 1 and 5 cm. These values correspond to the evaluation campaign where plants and weeds had the highest growth stage.

Due to weeds absorbing part of the resources necessary for crop growth (water, sunlight, space, etc.), weeding should be performed after the first weeks of crop planting to prevent its degradation. Depending on the weed to be removed, the depth that the weeding tool can reach is adjusted. In the case of an intervention during the first growing stages, the depth of the weeding tool is between 1 and 2 cm.

On the other hand, if the weeding is carried out at an advanced growing stage, firstly, the crops could present a significant decomposition as a result of the pests formed by weeds. Then, the intra-row distance reduces due to the advanced growing stages of crops and weeds complicating the weeding tool intervention. Finally, when the weeding tool cuts a dense weed population, crops can be damaged due to reduced space for manoeuvre or when removing weeds. For these reasons, weeding is recommended during the early stages of crop growth.

After explaining how the weeding tool operates and obtained its parameters, the two activation methods to manipulate the weeding tool for intra-row weeding are exposed.

Figure 4.5: Root size of different weeds. a) Matricaria. b)Ryegrass. c) Moutarde and d) Matricaria.

4.3.1 Weeding actuation based on proximal detection

The first approach to control the weeding tool is based on proximal detection. The detection is given by the multimodal sensor Xtool which is located under the platform (see appendix A). Most of the proposed solutions [92] work in this way; weeds are constantly detected as long as they are detected in the camera's field of view.

The proposed autonomous weeding approach performs row-wise. The camera's field of view distinguishes only one crop row and the weeding tool can be observed at the border of the image. However, this proposal can be extended to operate two or more weeding tools simultaneously either by adding an additional sensor for each crop row or with a single sensor increasing its height with respect to the ground to observe the number of crop rows to be weeded.

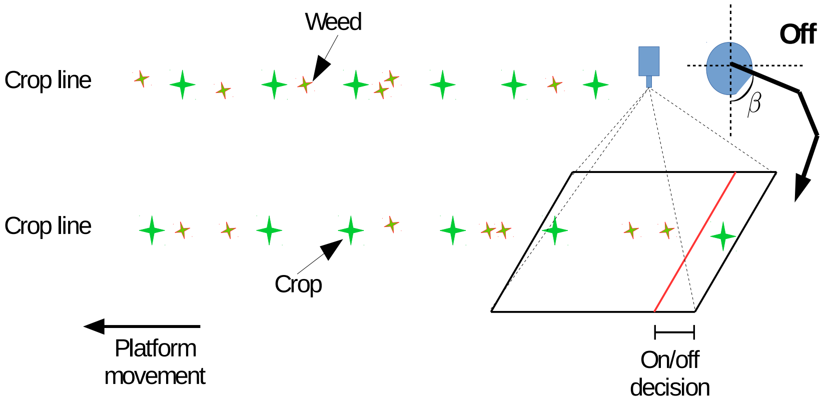

Figure 4.6: Weeding actuation based on proximal detection.

As the platform moves, crops and weeds are captured by the camera. Using the algorithm based on physical characteristics presented in Chapter 2, the coordinates of the plants of interest are estimated using Eq. (2.14). Then, a zone denominated the on/off decision zone (Figure 4.6) is defined and when a plant of interest is inside the decision zone, the weeding tool is disabled (Eq. 4.3).

$$Weeding\_tool = \begin{cases} 1 & \text{if } \bar{y}_i > \delta_{Itool} \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

where $\bar{y}_i$ is the y coordinate of the i-th detected plant and δ_{Itool} is the size of the decision zone in pixels.

Despite that throughout this chapter the weeding task protects the plant of interest, with the on/off decision zone approach, it can be observed that this proposition can be reversed. That is, considering the weed detection and when they are inside the decision zone, the weeding tool will be activated to eliminate them, which would imply navigating with the weeding tool deactivated.

The weeding task in the ROSE challenge was divided into two stages: during the first stage, markers were used to easily differentiate crops and weeds, and in the second stage, the markers were removed to perform the autonomous weeding without any visual indicators. The description of each marker is detailed in Table 4.1 and are illustrated in Figure 4.7.

	Diameter	**Color**	**Center diameter**	**Width**
Crop markers	2.5 cm	Blue	8 mm	3 mm
Weed markers	2 cm	Red	5mm	6 mm

Table 4.1: Markers description.

The markers indicate the position and the class of plants, they are placed on the ground and the plant stem is in the center of the marker as Figure 4.7 shows.

To identify the red markers (weeds), a color segmentation is performed. First, the image is converted to the HSV color space for the reasons explained in Chapter 2.

$$I_{RGB}\left(x,y\right) \longrightarrow I_{HSV}\left(x,y\right) \tag{4.4}$$

Then, the red color is around 180 in the H channel of HSV color space. Two thresholding operations are conducted to segment it:

$$I_{i_{BIN}}(x,y) = \begin{cases} 1 & \text{if } \delta_{i_{min}} \leq I_{HSV}(x,y) \leq \delta_{i_{max}} \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

with $\delta_{i_{min}}$ and $\delta_{i_{max}}$ are the lower and upper boundaries respectively, and $i = 1, 2$ corresponds to each segmentation mask resulting of the two thresholding operations. The segmented images are merged using the logical OR operator. Finally, an opening morphological operation deletes any undesirable segmented region.

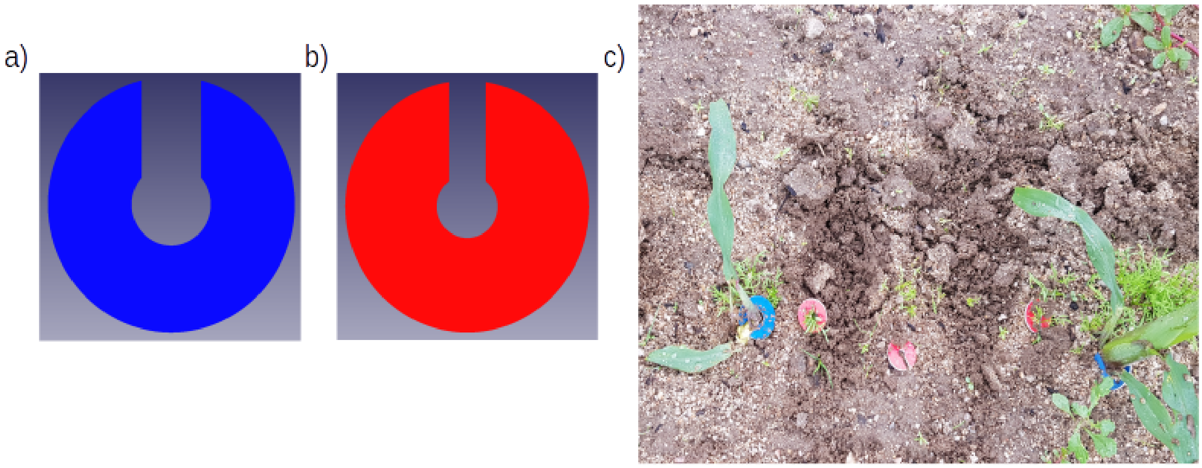

Figure 4.7: Markers placed at the crop and weed stem.

When the intra-row weeding is performed based on proximal detection, some points must be mentioned:

- If a weed is too close to a plant of interest, it cannot be eliminated due to the principle of protection of plants of interest.
- If a plant of interest is occluded (by other crops or weeds, lighting disturbance, etc.) inside the on/off decision zone, it will be destroyed even though the plant could be detected in other regions of the image.
- If the crops are in an advanced growth stage, they can obstruct the acquisition of the images (Figure 4.8-a). This limitation was partially solved by repositioning the detection system and using the detection algorithm based on physical characteristics allowing to detect the plants of interest from different positions and angles as Figure 4.8-b shows.

Figure 4.8: Xtool multimodal sensor. a) Obstruction of image acquisition. b) Xtool repositioning.

4.3.2 Weeding actuation based on frontal detection

The second proposed method to carry out the autonomous weeding utilizes the frontal perception system to detect crops using the algorithm based on physical features. However, other alternatives like the AQiT sensor, the learning-based models and the fusion algorithm presented in this work can be used in order to perform crop detection.

When crops are detected, they will be projected to be propagated in the weeding tool frame (Figure 4.9). As the platform moves, the weeding tool remains activated to eliminate weeds and when a plant of interest is propagated to the same weeding tool frame, it is deactivated to protect the plant.

The considered environment is defined by a 1×4 m grid that covers the vehicle as Figure 4.9 shows. The grid is partitioned into grid cells c_i at coordinates (x_i, y_i, z_i), with $i = 1, 2, \cdots, N$ and $z_i = 0$. Every grid cell is assigned a state between two possible values $Free$ (F) and $Occupied$ (O), which corresponds with the projected crop on the occupancy grid [93], [94].

$$b_i \in \{0, 1\} \tag{4.6}$$

with $b_i = 1$ indicating that cell c_i is occupied, and $b_i =$0 indicating that cell c_i is empty. The occupancy grid can be seen as the vector

$$b = [b_1, b_2, \cdots, b_N] \in \{0, 1\}^N \tag{4.7}$$

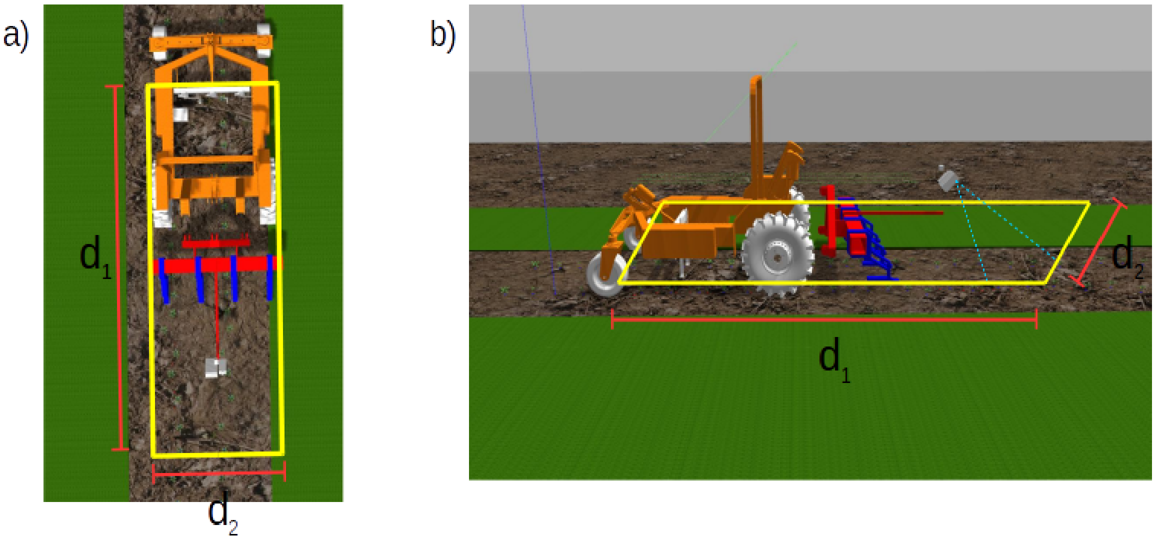

Figure 4.9: Occupancy grid. a) Top view. b) Side view.

The set b is commonly referred to a map. It is attached to the vehicle and it is defined on a sensor independent coordinate system where the origin is defined with respect to a fixed location on the platform such as the center of the rear axis. The platform's pose is assumed known and will be explained in the next subsection.

The problem is then to estimate a specific map within the set of all possible maps with N elements given a sequence of measurement vectors

$$M_T = [m_1, m_2, \cdots, m_t, \cdots, m_T] \tag{4.8}$$

with T representing the current sensing time. That is, $1 \leq t \leq T$ refers to the measurement index, and $m_t \in \{0, 1\}$ is a binary vector.

The goal is to develop a framework for updating the posterior probability of the occupancy of each grid cell c_i associating the random variable b_i, given the sequence of measurements M_T.

To estimate the vector of cell posterior probabilities p, some proposed works factor the joint distribution of the cell occupancy as

$$p_b(b) = \prod_i p_b(b_i) \tag{4.9}$$

In the same way, the collection of measurement vectors J_T is assumed to be factored as

$$p_{b|M}\left(b \mid M_T\right) = \prod_i p_{b|M}\left(b_i \mid M_T\right) \tag{4.10}$$

which implies that the elements in the cell occupancy are conditionally independent. Then, given the collection of T sequential observations M_T, to estimate the probability occupancy of each cell using Bayes' rule.

$$p\left(b \mid M_{1:T}, x_{1:T}\right) = \prod_i p\left(b_i \mid M_{1:T}, x_{1:T}\right) \tag{4.11}$$

where x corresponds to position of the platform, the platform position estimation will be expose in the next subsection.

$$p\left(b_i \mid M_{1:T}, x_{1:T}\right) = \frac{p\left(M_T \mid b_i, M_{1:T-1}, x_{1:T}\right) p\left(b_i \mid M_{1:T-1}, x_{1:T}\right)}{p\left(M_T \mid M_{1:T-1}, x_{1:T}\right)} \tag{4.12}$$

$$p\left(b_i \mid M_{1:T}, x_{1:T}\right) = \frac{p\left(M_T \mid b_i, x_T\right) p\left(b_i \mid M_{1:T-1}, x_{1:T-1}\right)}{p\left(M_T \mid M_{1:T-1}, x_{1:T}\right)} \tag{4.13}$$

where $p\left(M_T \mid b_i, x_T\right) = \dfrac{p\left(b_i \mid M_T, x_T\right) p\left(M_T \mid x_T\right)}{p\left(b_i \mid x_T\right)}$

After applying Bayes' rule and Markov

$$p\left(b_i \mid M_{1:T}, x_{1:T}\right) = \frac{p\left(b_i \mid M_T, x_T\right) p\left(M_T \mid x_T\right) p\left(b_i \mid M_{1:T-1}, x_{1:T-1}\right)}{p\left(b_i\right) p\left(M_T \mid M_{1:T-1}, x_{1:T}\right)} \tag{4.14}$$

and doing the same for the free cell probability

$$p\left(\neg b_i \mid M_{1:T}, x_{1:T}\right) = \frac{p\left(\neg b_i \mid M_T, x_T\right) p\left(M_T \mid x_T\right) p\left(\neg b_i \mid M_{1:T-1}, x_{1:T-1}\right)}{p\left(\neg b_i\right) p\left(M_T \mid M_{1:T-1}, x_{1:T}\right)} \tag{4.15}$$

Computing the radio of both probabilities:

$$\frac{p\left(b_i \mid M_{1:T}, x_{1:T}\right)}{p\left(\neg b_i \mid M_{1:T}, x_{1:T}\right)} = \frac{p\left(b_i \mid M_T, x_t\right)}{1 - p\left(b_i \mid M_T, x_T\right)} \frac{p\left(b_i \mid M_{1:T-1}, x_{1:T-1}\right)}{1 - p\left(b_i \mid M_{1:T-1}, x_{1:T-1}\right)} \frac{1 - p\left(b_i\right)}{p\left(b_i\right)} \tag{4.16}$$

The update Eq. (4.16) depends on the current measure, which is the first term of the equation, the previous estimate related with the second term and a prior probability which corresponds with the third term of the equation and by using the log-odds notation can be rewritten as

$$(b_i \mid M_{1:T}, x_{1:T}) = L\left(b_i \mid M_T, x_T\right) + L\left(b_i \mid z_{1:T-1}, x_{1:T-1}\right) - L\left(b_i\right) \tag{4.17}$$

with

$$L\left(x\right) = \log \frac{p\left(x\right)}{1 - p\left(x\right)}$$

Since multiplications are replaced by additions, Eq. (4.17) allows faster updates.

4.3.3 Robot pose estimation

This work considers two different approaches for autonomous navigation and weeding. The former is based on relative navigation and weeding (visual servoing and weeding based on proximal detection) and the latter considers absolute positioning to navigate in the absence of visual information and to carry out weeding task based on propagation. The estimate of the platform's pose is presented using a data fusion algorithm.

4.3.4 Kalman filter: preliminaries

The Kalman filter [95] aims to estimate the state $x \in \mathbb{R}^n$ of a discrete-time process that is described by the linear stochastic difference equation

$$x_{k+1} = F_k x_k + G_k u_k + v_k \tag{4.18}$$

with a measurement $z \in \mathbb{R}^m$

$$z_k = C_k x_k + w_k \tag{4.19}$$

The random variables v_k and w_k are the process and measurement noise, respectively. They are assumed to be independent between them, white and with normal probability distribution. The $n \times n$ state matrix F, relates the state at time step $k-1$ to the state at the step k. The $n \times l$ matrix G relates the control input $u \in R^l$ to the state x. The $m \times n$ observation matrix C, relates the state to the measurement z_k.

Defining $\hat{x}_{k|k-1}$ the a priori state at step k given knowledge of the process prior to step k, and $\hat{x}_{k|k}$ the posteriori state estimate at step k given measurements z_k. The a priori and a posteriori estimate errors are

$$\begin{aligned} e_k^- &= x_k - \hat{x}_{k|k-1} \\ e_k &= x_k - \hat{x}_{k|k} \end{aligned} \tag{4.20}$$

The a priori and a posteriori estimate error covariances are:

$$P_k^- = E\left[e_k^- e_k^{-T}\right] \quad , \quad P_k^- = E\left[e_k e_k^T\right] \tag{4.21}$$

$$\widehat{x}_k = \hat{x}_{k|k-1} + K\left(z_k - C_k \hat{x}_{k|k-1}\right) \tag{4.22}$$

The Eq. (4.22) computes an a posteriori state estimate $\widehat{x}_k$ as a linear combination of an a priori estimate $\hat{x}_{k|k-1}$ and a weighted difference between the actual measurement z_k and a measurement prediction $C_k\hat{x}_{k|k-1}$. The $n \times m$ matrix K is the Kalman gain and is given by

$$K_k = P_k^- C_k^T \left(C_k P_k^- C_k^T + R_k\right)^{-1} \tag{4.23}$$

Equation (4.32) is utilized when all the measurements are available When the system dynamics are nonlinear as is the case, the Extended Kalman filter (EKF) is used, it is not an optimal filter, it gives an approximation of the optimal estimate. The non-linearities of the system's dynamics are approximated by a linearized version of the non-linear system.

The Extended Kalman Filter algorithm is stated below:

EKF time update equations:

$$\widehat{x}_{k+1}^- = f\left(\widehat{x}_k, u_k\right) \tag{4.24}$$

$$P_{k+1}^- = A_k P_k A_k^T + Q_k \tag{4.25}$$

Similar to the Kalman Filter, the time update equations project the state and covariance from time step k to step $k+1$, A_k and W_k are the process Jacobians at step k, and Q_k is the process noise covariance at step k.

EKF measurement update equations:

$$K_k = P_k^- C_k^T \left(C_k P_k^- C_k^T + R_k\right)^{-1} \tag{4.26}$$

$$\widehat{x}_k = \widehat{x}_k^- + K\left(z_k - h\left(\widehat{x}_k^-\right)\right) \tag{4.27}$$

$$P_k = \left(I - K_k H_k\right) P_k^- \tag{4.28}$$

The measurement update equations correct the state and covariance estimates with the measurement z_k, H_k and V_k are the measurements Jacobians at step k, and R_k is the measurement noise covariance at step k.

4.3.5 Vehicle localization

The architecture of the proposed extended Kalman filter is shown in Figure 4.10. The two GPS onboard the platform are utilized to estimate the orientation and the position of the platform. A moving average window was used to filter the measurements given by the inertial measurement unit, and as a result of changing the direction of motion, the platform's odometry was modified.

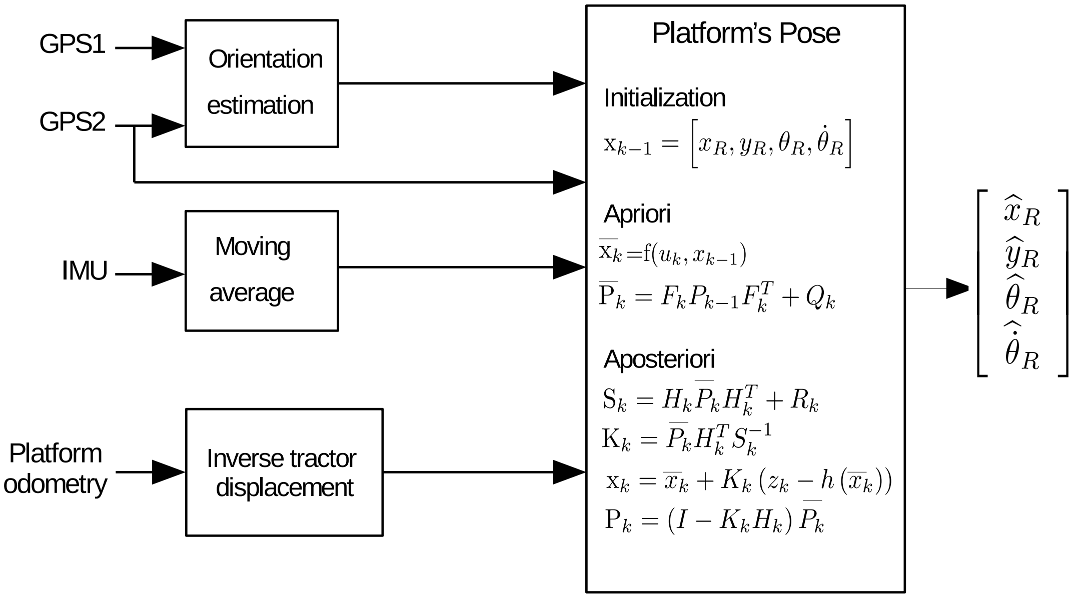

Figure 4.10: Extended Kalman filter to estimate platform's pose.

In order to merge different sensors, we proposed the state vector defined as:

$$\boldsymbol{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta \\ \dot{\theta} \end{bmatrix} \tag{4.29}$$

where:

$$
\begin{aligned}
\dot{x}_1 &= \cos(x_3)\left(\frac{R_L}{2}U_1 + \frac{R_R}{2}U_2\right) \\
\dot{x}_2 &= \sin(x_3)\left(\frac{R_L}{2}U_1 + \frac{R_R}{2}U_2\right) \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= 0
\end{aligned}
\tag{4.30}
$$

where R_L and R_R are the radius of left and right wheels, and U_1, U_2 are the left and right wheel velocities. The following initial conditions were considered:

$$
\boldsymbol{x}_0 = \begin{bmatrix} 0 \\ 0 \\ \frac{\pi}{2} \\ 0 \end{bmatrix} \quad , \quad P_0 = \begin{bmatrix} \left(\frac{20}{3}\right)^2 & 0 & 0 & 0 \\ 0 & \left(\frac{20}{3}\right)^2 & 0 & 0 \\ 0 & 0 & \left(\frac{120\pi}{180}\right)^2 & 0 \\ 0 & 0 & 0 & \left(\frac{10\pi}{180}\right)^2 \end{bmatrix}
\tag{4.31}
$$

The observation matrix is defined as:

$$
C_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}
\tag{4.32}
$$

4.4 Simulations results

The simulation scenarios used in this chapter correspond to those described in the ROSE challenge. The crop distribution was made considering the inter and intra-row distances specified in subsection 1.4. However, the position of the weeds was set randomly. This means that the weeding task was validated under different conditions: weeds close to the crops, two or more weeds sowed close to each other, scattered weeds, etc. In the same way, markers were added in some parcels to easily recognize crops and weeds. The rest of the parcels were designed without visual markers as Figure 4.11 shows.

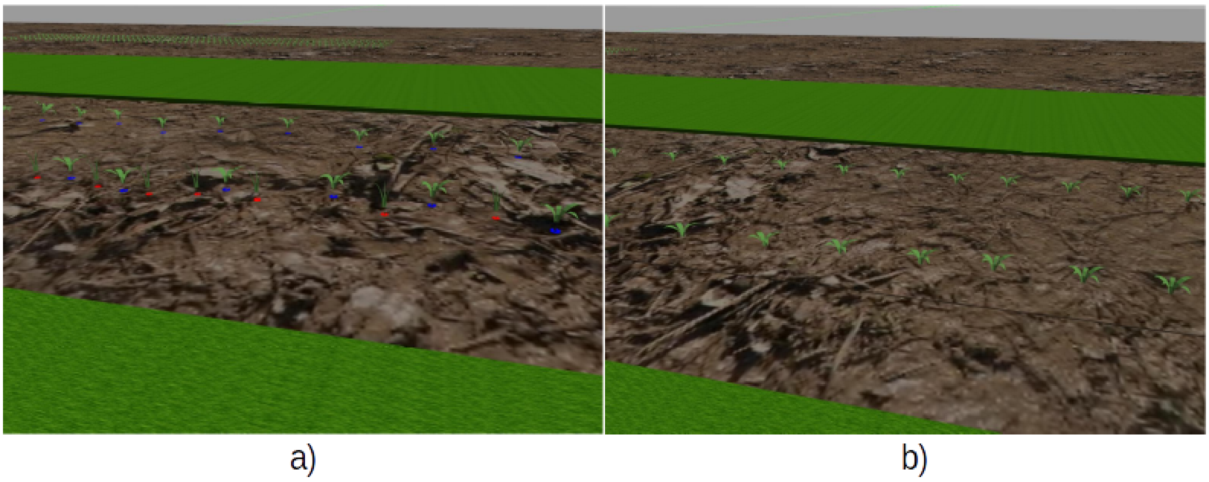

Figure 4.11: a) Parcel with markers to recognize crops and weeds. b) Parcel without markers.

The first proposed approach considering the proximal detection was tested with and without markers. For the case of parcel with markers, the weeding configuration was as follows: the tool remained deactivated while the platform was moving, the marker detection algorithm was designed to detect only the red markers which correspond to the weeds. As the tractor moves, markers are captured by the perception system. When the red markers are detected within the decision zone, the weeding tool is activated.

For the parcel without markers, only the crops were detected. The weeding tool operating mode was inverted. That is, when a plant of interest is detected in the decision zone, the tool is deactivated to avoid damaging the crops. The linear velocity of the platform is constant and $v_c =$ 0.2 m/s and the on/off decision zone has a width of 60 pixels for both cases (parcel with and without markers) which corresponds to 30 mm.

The number of weeds planted in each row was counted to make a comparison with the number of weeds that were eliminated after the intervention of the platform. The obtained results are presented in Table 4.2.

Crop	Weeds (with markers)	Weeds (without markers)	Weeds after weeding (with markers)	Weeds after weeding (without markers)
Maize	30	30	4	6
Bean	30	30	6	7

Table 4.2: Weeding results considering the proximal detection.

The highest percentage of weeding efficiency was 86.6% and it was obtained in maize with markers, while the lowest was 76.6% for beans without markers. The beans with markers and the maize without markers presented the same percentage of weeding effectiveness equal to 80%. The weeds that remained after the intervention reported in Table 4.2, most correspond to weeds located next to a crop.

The second proposed approach was tested in the same scenarios previously described. However, the weeding could not be carried out considering the markers due to their dimensions they are not detected by the frontal sensor. In the case of the parcel without markers, crop detection was performed. The image given by the camera and the crop projection on the occupancy grid is shown in Figure 4.12.

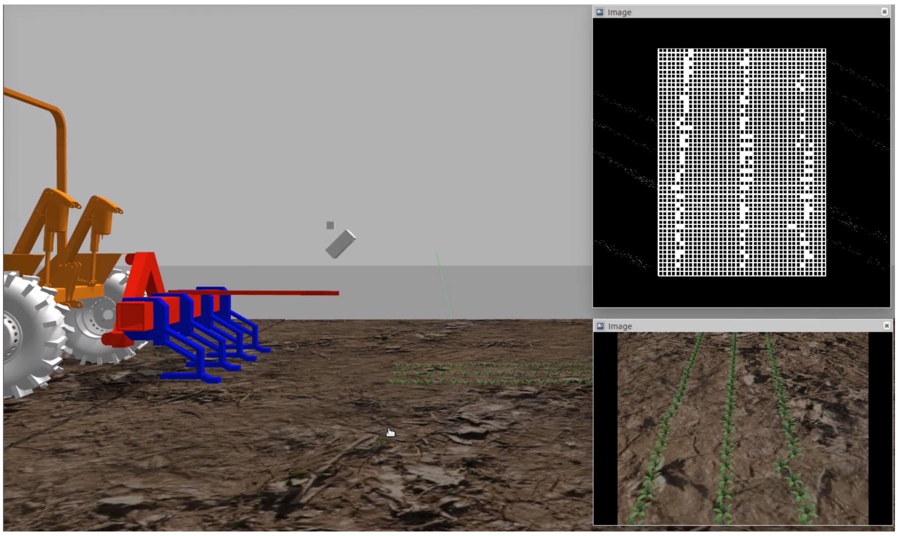

Figure 4.12: Bean propagation over the occupancy grid.

To perform the autonomous weeding, the plants detected by the perception system were propagated on the occupancy grid. The velocity of the platform remained constant $v_c = 0.2$ m/s, the weeding tool was deactivated when the crops were detected in the cells corresponding to the decision zone. Regardless of the parcel to be weeded, the number of crop rows detected is propagated in the occupancy grid. Figure 4.13 shows the weeding task on a maize parcel formed by two crop rows.

In the same way, the results reported in Table 4.3 show the number of weeds planted and the number of weeds that remained after performing the weeding task. Unlike proximal detection, this approach allows weeding to be performed simultaneously in each crop row as shown by the results in Table 4.3.

The highest weeding efficiency was in the maize parcel with 93.3% and 86.6% for the first and second row, respectively. These efficiency rates are higher than those obtained in the bean parcel (73.3%, 70% and 80% for the first,second and third row, respectively) as they are associated with the greater intra-row distance which allows better detection and the larger working space of the weeding tool.

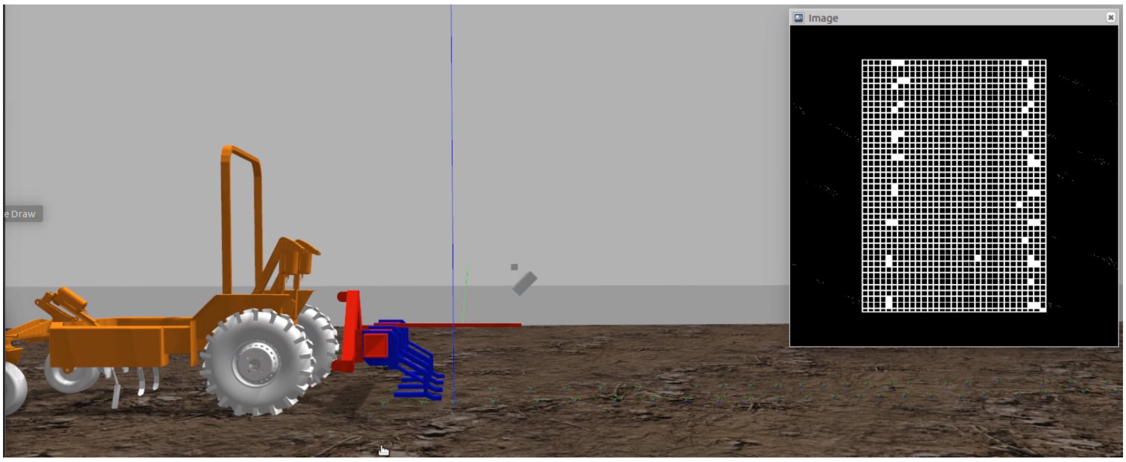

Figure 4.13: Maize propagation over the occupancy grid.

Crop	Weeds (with markers)	Weeds (without markers)	Weeds after weeding (with markers)	Weeds after weeding (without markers)
Maize 1st row	30	30	-	2
Maize 2nd row	30	30	-	4
Bean 1st row	30	30	-	8
Bean 2nd row	30	30	-	9
Bean 3rd row	30	30	-	6

Table 4.3: Weeding results considering the frontal detection.

On the other hand, in the bean parcel, the crops are sowed close to each other at an intra-row distance of 8 cm, which reduces the execution time and the work space of the weeding tool. As a result of weeding was carried out under the premise of protecting crops, the number of preserved weeds was higher.

4.5 Experimental results

The weeding approach based on proximal detection was tested in two different evaluation campaigns of the ROSE challenge. With the crops and weeds at different growth stages:in the first evaluation campaign, the crops presented a growth stage VE according to Figure 2.12, and in the second evaluation campaign, the crops were in a growth stage between V4 and V6 according to Figure 2.12. While the second weeding approach based on frontal detection was tested only in the last evaluation campaign.

The two proposed approaches were tested. To remove weeds with markers, the weeding tool remained deactivated while the platform was moving on the parcel and when a red marker was detected in the activation zone, the tool was activated.

In the parcel without markers, the algorithm for detection of plants of interest based on physical characteristics was used to perform the weeding. With this approach, the weeding tool remains active while the platform moves and deactivates when detecting a plant of interest in the decision zone to avoid damaging the crops and eliminate everything that is not detected as a plant of interest. Figure 4.14 shows two images taken from both methods of weeding tool operation: the first focuses on protecting crops when they are located in the activation zone of the weeding tool and the second is devoted to eliminating weeds when red markers are detected in the activation zone. The on/off decision zone has a width of 80 pixels.

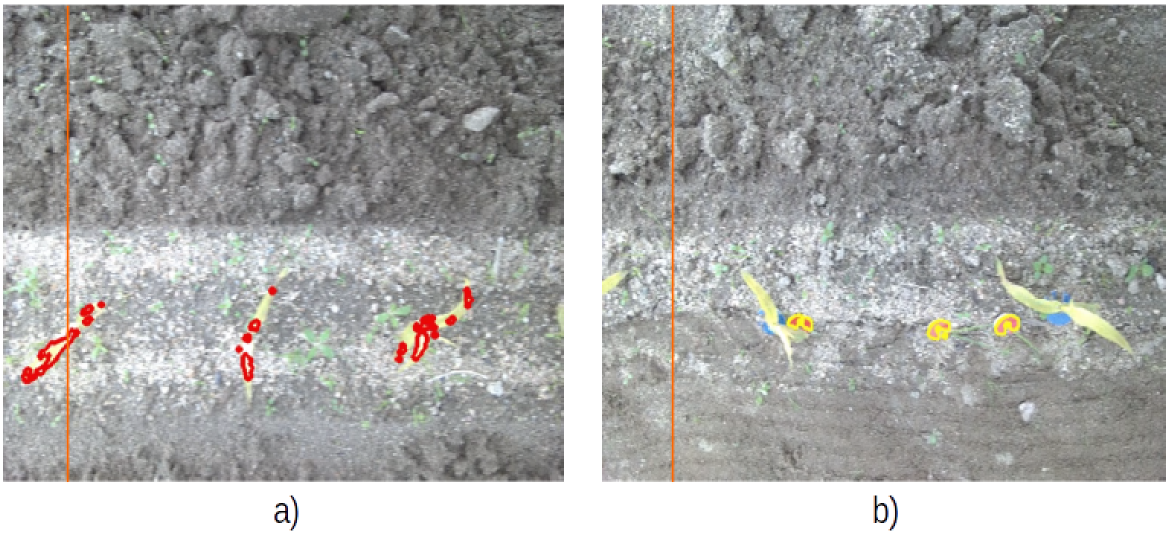

Figure 4.14: Autonomous weeding task. a) Detected crops to be protected. b) Weeds marker detection.

The obtained results after the intervention in the parcels with and without markers are shown in Table 4.4. The weeding efficiency for the maize parcel with markers was 85.7%, while the maize parcel without markers and the bean parcel with markers presented the same efficiency as 75.8%. The bean parcel without markers could not be weeded due to an electronic failure of the weeding tool, which was unable to execute the command estimated by the activation algorithm.

The weeds that are preserved after the weeding task, both in the simulation subsection as well as in this experimental subsection, are mostly weeds that are close to the crops as shown in Figure 4.15. The weeds that are not close to the crops can be eliminated using the proposed approach.

Crop	Weeds (with markers)	Weeds (without markers)	Weeds after weeding (with markers)	Weeds after weeding (without markers)
Maize	14	14	2	3
Bean	14	14	3	-

Table 4.4: Weeding results considering the proximal detection.

Figure 4.15: Weed examples. a) Weed positioned far enough to be weeded by the proposed approach. b) Weed positioned very close to the crops.

The second proposed approach could not be validated in the experiments as a result of the crop's advanced stage of growth. When executing the crop detection algorithm based on the physical features, crops are captured by the frontal sensor. Due to the leaf dimensions, the algorithm can differentiate crops from the rest of the image but it is unable to isolate each plant. The detected crops projected on the occupancy grid form constant cell patterns that do not allow to perform weeding as Figure 4.16.

Although the weeding considering the frontal detection could not be validated experimentally, the results of the platform position estimation are presented. The Extended Kalman filter allows to estimate the platform's pose during its transition through the experimental parcel. Figure 4.17 corresponds to the estimation of the position in the plane $x - y$. It can be verified that the estimated states by the Kalman filter for the position of the platform converge quickly to the real trajectory.

The other two states estimated by the Kalman filter correspond to its orientation and its angular velocity. At the end of the parcel, the platform performed a full turn to

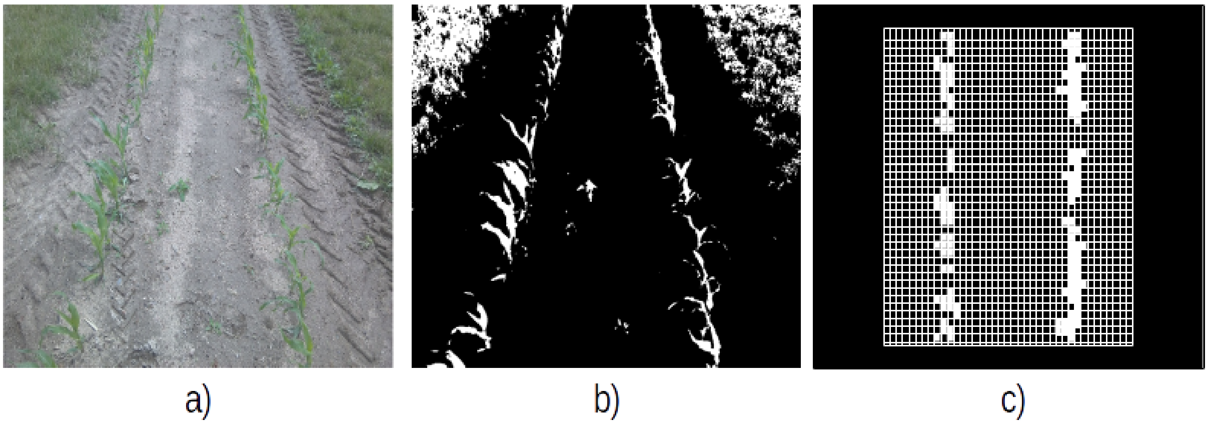

Figure 4.16: Autonomous weeding considering frontal detection. a) RGB image. b) Crops detection. c) Detected crops projected on the occupancy grid.

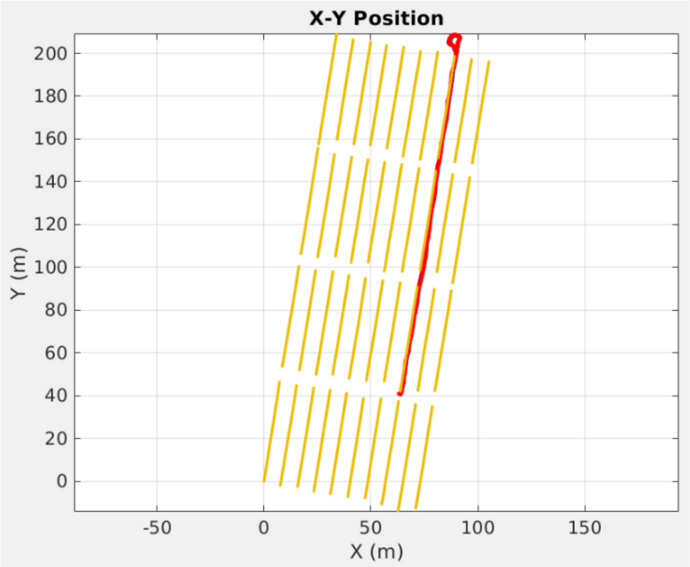

Figure 4.17: Position estimation of the platform using the Extended Kalman filter.

return to the beginning of the parcel, which can be seen in Figure 4.18.

The initial position of the platform is located at the beginning of one of the parcels with an orientation of 82°. This orientation corresponds to the alignment of the platform with respect to the crop lines. As the tractor moves, its orientation estimation remains around the same value. At the same time, the estimated angular velocity remains close to zero, which corresponds to the displacement of the platform in the straight line.

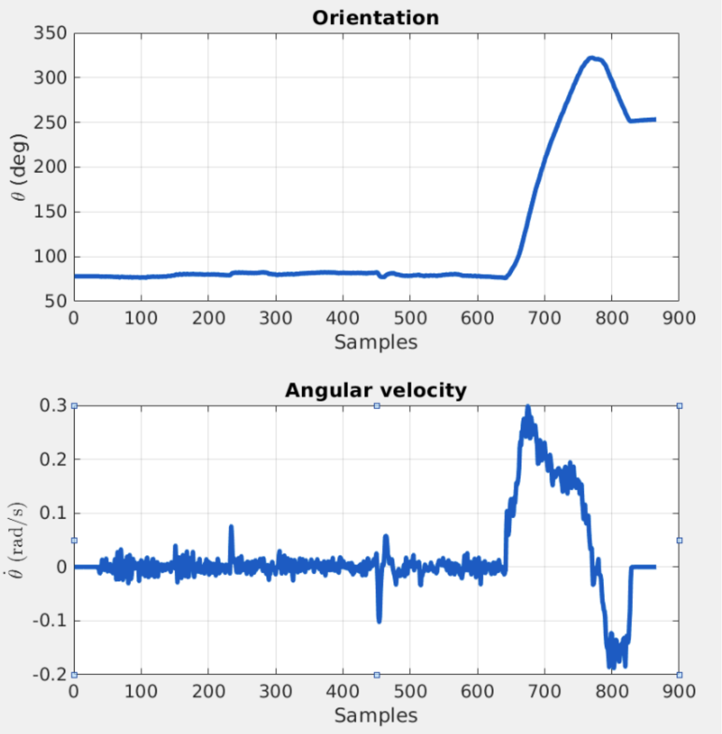

Figure 4.18: Orientation and angular velocity estimation of the platform using the Extended Kalman filter.

4.6 Conclusion

Two alternatives to carry out autonomous mechanical weeding were presented in this chapter. The difference between them is the acquisition system considered: the first proposition uses a proximal detection and the second approach is based on a frontal detection. Due to the characteristics of the considered mechanical weeding tool, the weeds that are located very close to the crops can not be eliminated to avoid damaging them, regardless of either of two proposed solutions.

The first weeding approach allows to identify crops/weeds in a close manner. Most published solutions work in this way. However, when crops and weeds are in an advanced growing stage, the detection system could be obstructed by the plants and, consequently, the weeding task can not be carried out. The proposed approach deals with this problem by allowing modifying the position and orientation of the detection system to perform weeding under different situations. On the other hand, the drawback is that a plant can be detected when it is projected in the image, but if it is obstructed in the activation area, it will be eliminated since the tool activation system only considers the plants detected in the on/off decision zone of the image.

The second weeding approach differs from the methods commonly found in the literature. The plants detected by a frontal acquisition system are propagated from the front of the platform to the weeding tool position. With this approach, the occlusion problem present in the first contribution is solved because once the crops are detected they are propagated on an occupancy grid up to the position of the weeding tool. One of the main tasks to solve when using this approach is the platform location. It was performed using the extended Kalman filter to estimate the position of the platform. The constraint with this method is when crops are in a very advanced growing stage, when projected on the occupancy grid, they occupy more than one cell, generating a pattern of multiple occupied cells without the capacity to isolate each plant.

5

Conclusions and outlook

Table of contents

5.1 Summary

This thesis has proposed a solution to carry out autonomous mechanical weeding. Multiple tasks have been solved to achieve this goal. First of all, the ability to identify the working environment of the platform was developed. Due to this thesis is focused on agricultural robotics, the experimental validation was carried out in outdoor environments where working conditions are constantly changing, which increases the complexity of this work.

The proposed approach is based on crop and weed detection to protect the former and eliminate the latter. Thus, their identification implied a fundamental stage in this work. The proposed algorithms based on physical characteristics, machine learning and the fusion of both allowed to identify crops/weeds efficiently. Despite the fact that both algorithms operated correctly separately, their fusion allowed to improve the deficiencies of each one of them performing independently.

Crop detection was used to solve other autonomous weeding subtasks such as autonomous navigation. The proposed approach estimates the vanishing point formed by crop lines to use it in platform trajectory correction (the vanishing point must remain in the image center as long as the platform is moving). The use of visual information discards the dependence on global positioning systems normally used to perform autonomous navigation.

To provide complete autonomous navigation, the platform must be able to navigate not only in the parcel, but also to make the transition from one parcel to another. Both scenarios were solved in this work; to navigate through the parcels, the estimation of the vanishing point was used as a reference and to carry out the transition, the bird's eye view transformation allowed transit between parcels.

The inter and intra-row weeding were performed with the same category of mechanical weeding tool. Two solutions were proposed: the first was carried out using the proximal detection and the second considering the frontal detection. The weeding actuation principle was the same in both cases: the weeding tool was activated while the platform was moving, eliminating any weeds and, when a plant of interest was detected, the tool was disabled to avoid damaging crops. However, the crop identification was different. In proximal detection, it only focuses on the detection of plants projected in the camera's field of view. On the other hand, considering the frontal detection, the detected crops were propagated from the front of the platform to the position of the weeding tool.

Numerous simulation scenarios were created in the Gazebo simulator to validate each one of the subtasks and the global autonomous weeding task. Usually, the availability of an agricultural experimentation space is challenging. For this reason, the use of digital twins allowed generating the experimentation scenarios as close as the experimental validation site.

The reported results in this work were obtained within the framework of two competitions: the ROSE challenge and the ACRE international competition. During the ROSE challenge, various evaluation campaigns were organized which allowed to test the algorithms under diverse weather conditions and different growth stages of crops and weeds.

The proposed detection systems (Xeye and Xtool) developed for this project allowed efficient data acquisition under various conditions. Furthermore, the conception of the detection system allowed to protect the multiple sensors during the experimental validation as well as its easy installation on different experimental platforms.

5.2 Outlook

Based on the proposed approach, several extensions can be planned, which are listed below.

5.2.1 Visual perception

First, the proposed crop detection algorithm was tested considering maize and beans, which suggests contemplating more crops to be detected by the algorithm. In the same way, despite multiple weeds were considered during the experiments, the integration of new ones can be conceivable. Similar to the scenarios presented in the ROSE challenge, the detection of new crops must be carried out during their multiple stages of growth. In this way, a database can be generated relating the parameters needed by the algorithm to their respective growth stage for a variety of crops.

Another study that can be carried out considering the detection algorithm is to modify the position of the data acquisition systems and compare the obtained results for each position and orientation. In this way, according to the physical characteristics of each crop and weed, certain characteristics can be perceived easily from different positions and orientations according to the crop/weed to be detected.

Finally, the use of the frontal and proximal perception systems leads to merging both sources of information. This fusion should compensate for some characteristics not perceived by each of them operating independently. The problems mentioned in the proximal detection (occlusion, false detection, etc.) could be improved with additional information. In this context, for proximal detection, more cameras can be added to have more images from different positions and not just from one sensor, which will allow to perform tasks such as reconstruction of the plant of interest, a robust identification of the plant stem, etc.

5.2.2 Autonomous navigation

Regarding navigation, a free-obstacle environment was considered. However, in order to increase the platform's autonomy, an obstacle detection system could be integrated. The first proposal to develop consists of the detection of statical obstacles. Some realistic objects frequently found in farms that can be associated to this category of obstacles are: signposts, parcel delimitation indicators, tools, objects lying on the floor, etc. Subsequently, an extension to dynamic obstacle detection could be considered. In this way, the platform could interact in parallel with humans in agricultural tasks and avoid causing any harm to people.

Once the obstacles are detected, the platform has to decide what to do. Commonly, the easiest decision is to stop the system until receiving a new command from the operator. However, a supplementary system that allows the platform to perceive its environment will allow it to make its own decisions. That is, a trajectory replanning strategy without damaging the crops, returning to the beginning of the mission to be able to find a route to navigate. If the obstacle is dynamic, the robot will be able to estimate the dynamic of the obstacle to avoid collision, etc.

Despite the fact that the navigation proposal is based mainly on the extraction of visual information, in some cases the available information may not be enough to allow the robot to navigate autonomously. To solve this problem, a data fusion algorithm can be programmed. In particular, during the transition zones between parcels where there is an absence of crops or weeds, in this case, classical positioning systems such as GPS can provide greater robustness to the system.

5.2.3 Weeding tool

Despite having carried out some experimental tests of autonomous weeding, the failure of the weeding tool restricted increasing the number of validation tests, so the first perspective regarding weeding is to increase the number of field tests carried out. Furthermore, the autonomous weeding considering frontal detection could not be validated experimentally, which implies carrying out these tests.

The mechanical weeding tool can be replaced by any other weeding alternative to compare the obtained results with the proposed approach and the obtained results with the new weeding methods. The fact of working with a weeding tool that only has one degree of freedom considerably limits the weeding task. The conception of a weeding tool with higher performance is under development.

The proposal of new algorithms to control the weeding tool must be carried out. An analysis to know the execution time of the weeding under different meteorological conditions and different characteristics of the soil should be considered. This will determine the maximum speed at which the platform can be moved and carry out in parallel the weeding task with the new weed mechanism.

Similar to the proposal made in the detection, the new weeding tool must be tested with different crops and weeds. According to the physical properties of the plants and their growth state, the weeding system will obtain different success rates for autonomous weeding. With this information, a database can be generated in which will be displayed the periods for each plant in which weeding should be performed to obtain optimal performance.

5.2.4 Platform

Although in the present work two different experimental platforms were used: an Ackermann configuration and a differential vehicle. The proposed approach can be extrapolated to other configurations of experimental platforms with an adaptation stage considering the dynamics of the new platform, its control inputs, and its constraints.

To improve the performance of the experimental platforms, dynamic models that consider all the factors like the platforms sliding, the lateral forces actuating over the platform, better mass distribution of the platform, etc. have to be considered.

The lighting system can be replaced with another more sophisticated, instead of having a constant intensity as provided by the system used in this work, the light emitted by the lamps can be gradually controlled, taking into account to the experimental conditions, which will improve the crop/weed detection.

5.2.5 Scalability

Agricultural tasks are carried out in open environments and frequently correspond to large tracts of land that can restrict the operation range of a single robot, since it depends on its autonomy. In order to increase the range of operations and coverage in agricultural robotics, a multi-agent system can be implemented which will allow multiple robots to operate in a greater geographical area.

The same task can be carried out by multiple robots in parallel, reducing operation times in the case of large agricultural areas. Task optimization algorithms can be implemented to increase the effectiveness of the mission. However, collaboration to perform the same task is not the only one that can be done. The use of unmanned aerial vehicles can be a supplementary source of information to be exploited. In the case of crop detection, the larger-scale images provided by the aerial system can reinforce the detection. For autonomous navigation, the UAV will be able to communicate additional navigation routes due to the fact that it has a parcel wide field perception of the work environment.

5.2.6 Simulations

The scenarios generated in the Gazebo simulator considered the real conditions of the experimental field. However, if the proposals described above are implemented, the scenarios must be modified to integrate the suggested perspectives. In particular, static and dynamic obstacles must be integrated.

5.3 Publications

- Avilés-Mejia, J. E., Soto, D., Stéphant, J., & Labbani-Igbida, O. (2022, August). *Autonomous Vision-Based Navigation and Control for Intra-Row Weeding.* In 2022 IEEE 18th International Conference on Automation Science and Engineering (CASE) (pp. 575-582). IEEE.

A

Appendices

Table of contents

A.1 Bean growth stages

Bean germination is the procedure in which a seed, spore or begins growth. It is affected by divers factor like: water, light, soil conditions, etc. As bean germinates, its physical characteristics change according to the number of days elapsed since planting. Tables A.1 and A.2 show some characteristics during each stage of growth.

Stage	General description	Days from planting
VE	Hypocotyl emergence (crook stage)	7-8
VC	Cotyledon (seed leaves) and unifoliolate leaves visible	8-9
V1	First fully developed trifoliolate at the third node	10
V2	Second trifoliolate (count when leaf edges no longer touch)	19
V3	Third trifoliolate (secondary branching begins to show in leaf axils)	29
V(n)	N-th trifoliolate, but with blossom clusters still not visibly opened	A new node every 3 to 5 days
V5	Bush (determinate) plants may vegin to exhibit blossom and become stage R1	50
V8	Vine (indeterminate) plants bay begin to exhibit blossom and become stage R1	40

Table A.1: Bean vegetative stages [68].

The bean growth stages can be classified in four categories:

- Emergence and early vegetative growth
- Branching and rapid vegetative growth
- Flowering and pod formation
- Pod and maturation

The emergence and early vegetative category corresponds to the growth stages between VE and V3 shown in Table A.1. While the rest of the growth stages in Table A.1 correspond to the branching and rapid vegetative category. From Table A.2, the flowering and pod formation starts at R1 and ends at R4 stage. Finally, the last category is developed two months after sowing.

The information shown in Tables A.1 and A.2, may differ if a weed control is not carry out. For this reason, the elimination of weeds is suggested to be performed during early stages of bean growth to prevent weeds absorbing nutrients, damaging crop development.

Stage	General description	Days from planting
R1	One blossom open at any node	50
R2	Pods 1/2 inch long at first blossom position (usually node 2 to 3)	53
R3	Pods 1 inch long at first blossom position; secondary branching at all nodes, so plant is becoming denser but not taller, 1/2 bloom	56
R4	Pods 3 inches long (seeds not discernible; bush types may be shorter)	59
R5	Pods 3 to 4 inches (seed discernible)	64
R6	Seeds at least 1/4 inch over long axis	66
R7	Oldest pods have developed seeds (other parts of plant will have full-length pods with seeds almost as large as first pods; pods will be developed over the whole plant)	72
R8	Leaves yellowing over half of plant, very few small pods may be drying (point of maximum production has been reached)	90
R9	Mature, at least 80% of pods showing yellow and mostly ripe; only 40% of leaves still green	105

Table A.2: Bean reproductive stages [68].

To better interpret the previous tables, a description of the bean plant is shown in Figure A.1.

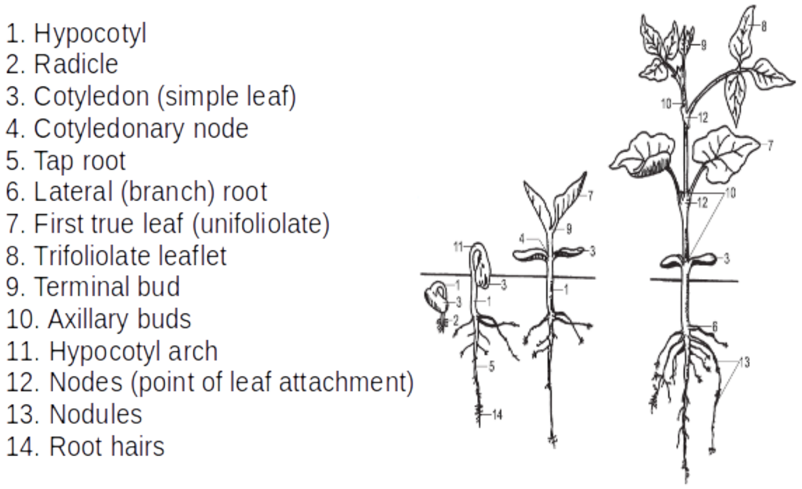

Figure A.1: Bean description [68].

A.2 Platform instrumentation

The experimental platforms utilized in this work consist of two compact electric tractors: the first one was developed by Sabi Agri[1] as part of the consortium created to participate in the ROSE challenge, this tractor has a differential drive configuration (Figure A.2). The second one is an Ackermann steering vehicle designed by XLIM and manufactured by Elatec[2], it was employed during the ACRE competition and it has an Ackermann steering mechanism (Figure A.3).

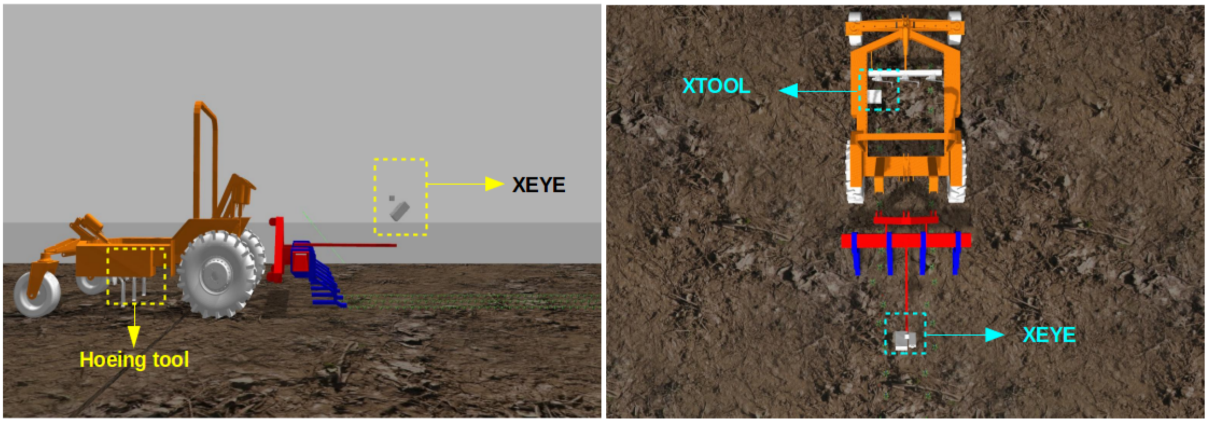

Figure A.2: Differential drive configuration platform with multimodal sensors attached.

Each platform was equipped with the same multimodal sensors developed by XLIM for this project. The multimodal sensors named Xeye and Xtool were fixed in front and under the platforms respectively. In addition, for ground truth, two GPS-RTK were installed on the platform as Figure A.3 shows to record its trajectory and orientation and they could be used in other applications.

The multimodal sensors Xeye and Xtool including:

- A nanocomputer NVIDIA Jetson Xavier
- An 8 MP Raspberry Pi Module
- A 9-axis inertial measurement unit MPU-9250
- An Intel RealSense TM D435 camera
- A lidar-lite v3 Garmin

The lighting system was installed under the platform to control the illumination conditions. This system consists of two 20 WLED projectors inside a black box. The data flow and the platforms were controlled using the Robot Operating System (ROS) middleware. To provide internet connection to the ROS-network, a Router RUTX11 was utilized.

---

[1] https://www.sabi-agri.com/en/home/

[2] https://www.elatec.fr/

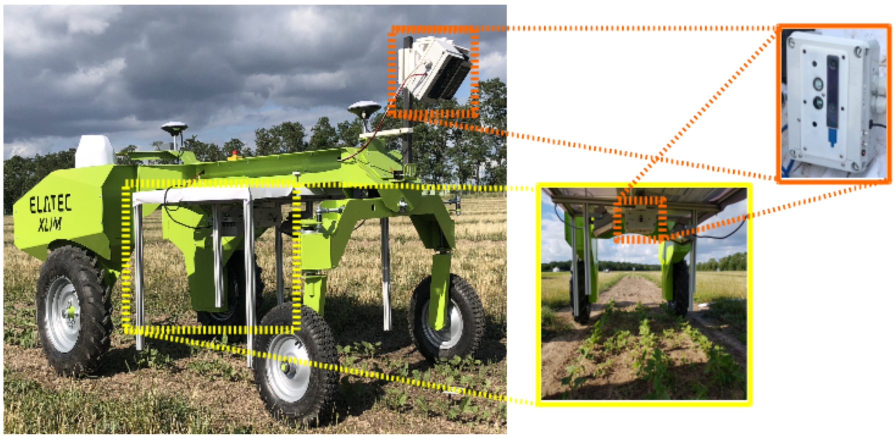

Figure A.3: Cinteo platform during the ACRE competition.

The low level control of the platform is handled by a Raspberry Pi 4. The detailed dimensions of the Cinteo platform are indicated in Figure A.4.

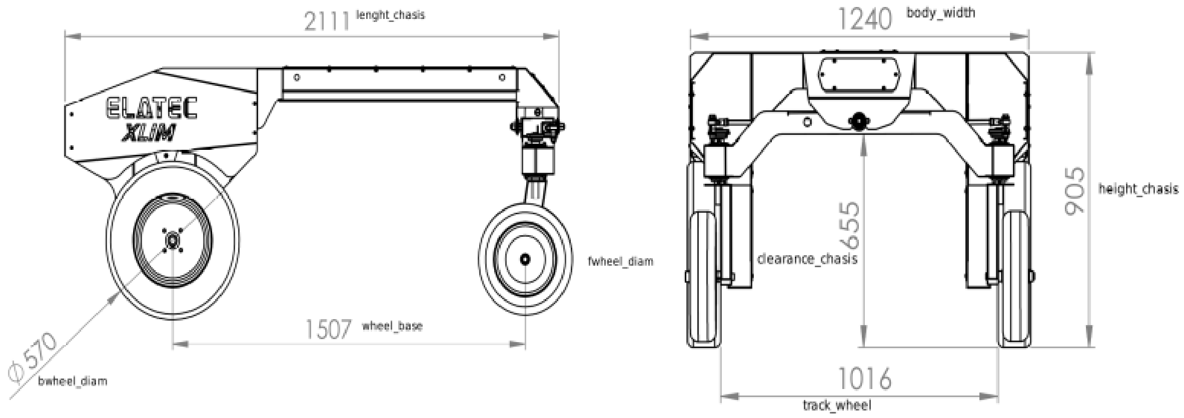

Figure A.4: Cinteo platform description.

B

References

Table of contents

References

[1] *World population prospects 2022*, https://www.un.org/development/desa/pd/sites/www.un.org.development.desa.pd/files/wpp2022_summary_of_results.pdf, Accessed: 2022-12-14.

[2] *Autonomous system for agricultural purposes such as spraying, tillage, fertilization, contour cut, harvest, and transportation*, https://www.raussendorf.de/en/fruit-robot.html, Accessed: 2022-07-08.

[3] *Greenbot*, https://precisionmakers.com/en/greenbot, Accessed: 2022-07-08.

[4] X. He, Y. Ding, D. Zhang, L. Yang, T. Cui, and X. Zhong, "Development of a variable-rate seeding control system for corn planters part ii: field performance," *Computers and Electronics in Agriculture*, vol. 162, pp. 309–317, 2019.

[5] A. Cay, H. Kocabiyik, and S. May, "Development of an electro-mechanic control system for seed-metering unit of single seed corn planters part i: design and laboratory simulation," *Computers and Electronics in Agriculture*, vol. 144, pp. 71–79, 2018.

[6] N. S. Naik, V. V. Shete, and S. R. Danve, "Precision agriculture robot for seeding function," in *2016 international conference on inventive computation technologies (ICICT)*, IEEE, vol. 2, 2016, pp. 1–3.

[7] Z. M. Khazimov, G. Bora, K. Khazimov, M. Khazimov, I. Ultanova, and A. Niyazbayev, "Development of a dual action planting and mulching machine for vegetable seedlings," *Engineering in Agriculture, Environment and Food*, vol. 11, no. 2, pp. 74–78, 2018.

[8] R. Oberti, M. Marchi, P. Tirelli, A. Calcante, M. Iriti, and A. N. Borghese, "Automatic detection of powdery mildew on grapevine leaves by image analysis: optimal view-angle range to increase the sensitivity," *Computers and Electronics in Agriculture*, vol. 104, pp. 1–8, 2014.

[9] *Robot one*, https://pixelfarmingrobotics.com/robot-one/, Accessed: 2022-07-08.

[10] S. Von Bueren and I. Yule, “Multispectral aerial imaging of pasture quality and biomass using unmanned aerial vehicles (uav),” *Accurate and Efficient Use of Nutrients on Farms, Occasional Report*, vol. 26, pp. 1–5, 2013.

[11] A. S. Natu and S. Kulkarni, “Adoption and utilization of drones for advanced precision farming: a review,” *Int. J. Recent Innov. Trends Comput. Commun*, vol. 4, no. 5, pp. 563–565, 2016.

[12] V. Singh and A. K. Misra, “Detection of plant leaf diseases using image segmentation and soft computing techniques,” *Information processing in Agriculture*, vol. 4, no. 1, pp. 41–49, 2017.

[13] M. M. Ozguven and K. Adem, “Automatic detection and classification of leaf spot disease in sugar beet using deep learning algorithms,” *Physica A: statistical mechanics and its applications*, vol. 535, p. 122 537, 2019.

[14] D. Haboudane, N. Tremblay, J. R. Miller, and P. Vigneault, “Remote estimation of crop chlorophyll content using spectral indices derived from hyperspectral data,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 2, pp. 423–437, 2008.

[15] P. J. Sammons, T. Furukawa, and A. Bulgin, “Autonomous pesticide spraying robot for use in a greenhouse,” in *Australian Conference on Robotics and Automation*, Commonwealth Scientific and Industrial Research Organisation Canberra, Australia, vol. 1, 2005.

[16] *Xag r150 unmanned ground vehicle*, `https://www.xa.com/en/xauv_r150`, Accessed: 2022-07-08.

[17] *Agras mg-1 dji*, `https://www.dji.com/fr/mg-1`, Accessed: 2022-07-08.

[18] F. Ahmad, A. Khaliq, B. Qiu, M. Sultan, and J. Ma, “Advancements of spraying technology in agriculture,” *Technology in Agriculture*, pp. 1–19, 2021.

[19] *Agrobot*, `https://www.agrobot.com/e-series`, Accessed: 2022-07-08.

[20] Z. De-An, L. Jidong, J. Wei, Z. Ying, and C. Yu, “Design and control of an apple harvesting robot,” *Biosystems engineering*, vol. 110, no. 2, pp. 112–122, 2011.

[21] B. Arad, J. Balendonck, R. Barth, *et al.*, “Development of a sweet pepper harvesting robot,” *Journal of Field Robotics*, vol. 37, no. 6, pp. 1027–1039, 2020.

[22] E. Van Henten, B. v. Van Tuijl, J. Hemming, J. Kornet, J. Bontsema, and E. Van Os, “Field test of an autonomous cucumber picking robot,” *Biosystems engineering*, vol. 86, no. 3, pp. 305–313, 2003.

[23] K. Tanigaki, T. Fujiura, A. Akase, and J. Imagawa, “Cherry-harvesting robot,” *Computers and electronics in agriculture*, vol. 63, no. 1, pp. 65–72, 2008.

[24] A. Leu, M. Razavi, L. Langstädtler, *et al.*, “Robotic green asparagus selective harvesting,” *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 6, pp. 2401–2410, 2017.

[25] W. Lili, Z. Bo, F. Jinwei, *et al.*, “Development of a tomato harvesting robot used in greenhouse,” *International Journal of Agricultural and Biological Engineering*, vol. 10, no. 4, pp. 140–149, 2017.

[26] *Naïo technologies. oz - weeding, transportation and harvest assistance robot*, `https://www.naio-technologies.com/wp-content/uploads/2019/04/brochure-OZ-ENGLISH-HD.pdf`, Accessed: 2022-07-08.

[27] *Naïo technologies. dino - autonomous mechanical weeding robot*, `https://www.naio-technologies.com/wp-content/uploads/2019/04/brochure-DINO-ENGLISH-HD.pdf`, Accessed: 2022-07-08.

[28] *Farming revolution weeding robot*, `https://farming-revolution.com/`, Accessed: 2022-07-08.

[29] B. Åstrand and A.-J. Baerveldt, “An agricultural mobile robot with vision-based perception for mechanical weed control,” *Autonomous robots*, vol. 13, no. 1, pp. 21–35, 2002.

[30] X. Wu, S. Aravecchia, P. Lottes, C. Stachniss, and C. Pradalier, “Robotic weed control using automated weed and crop classification,” *Journal of Field Robotics*, vol. 37, no. 2, pp. 322–340, 2020.

[31] A. Sharma, V. Kumar, B. Shahzad, *et al.*, “Worldwide pesticide usage and its impacts on ecosystem,” *SN Applied Sciences*, vol. 1, no. 11, pp. 1–16, 2019.

[32] *Pesticides, our world in data*, `https://ourworldindata.org/pesticides`, Accessed: 2022-07-08.

[33] D. Pimentel and M. Burgess, “Environmental and economic costs of the application of pesticides primarily in the united states,” in *Integrated pest management*, Springer, 2014, pp. 47–71.

[34] M. Eddleston, N. A. Buckley, P. Eyer, and A. H. Dawson, “Management of acute organophosphorus pesticide poisoning,” *The Lancet*, vol. 371, no. 9612, pp. 597–607, 2008.

[35] B. Alewu and C. Nosiri, “Pesticides and human health,” *Pesticides in the modern world–effects of pesticides exposure. InTech*, pp. 231–50, 2011.

[36] T. B. Hayes, P. Case, S. Chui, *et al.*, “Pesticide mixtures, endocrine disruption, and amphibian declines: are we underestimating the impact?” *Environmental health perspectives*, vol. 114, no. Suppl 1, pp. 40–50, 2006.

[37] A. Nougadère, V. Sirot, A. Kadar, *et al.*, “Total diet study on pesticide residues in france: levels in food as consumed and chronic dietary risk to consumers,” *Environment international*, vol. 45, pp. 135–150, 2012.

[38] *The rose challenge - robotics and sensors at the service of ecophyto*, `https://www.challenge-rose.fr/en/home/`, Accessed: 2022-07-08.

[39] *Agri-food competition for robot evaluation*, `https://metricsproject.eu/agri-food/acre-competition/`, Accessed: 2022-07-08.

[40] G. E. Meyer and J. C. Neto, “Verification of color vegetation indices for automated crop imaging applications,” *Computers and electronics in agriculture*, vol. 63, no. 2, pp. 282–293, 2008.

[41] H. Yao, Y. Huang, Z. Hruska, S. J. Thomson, and K. N. Reddy, “Using vegetation index and modified derivative for early detection of soybean plant injury from glyphosate,” *Computers and Electronics in Agriculture*, vol. 89, pp. 145–157, 2012.

[42] M. Á. Castillo-Martınez, F. J. Gallegos-Funes, B. E. Carvajal-Gámez, G. Urriolagoitia-Sosa, and A. J. Rosales-Silva, “Color index based thresholding method for background and foreground segmentation of plant images,” *Computers and Electronics in Agriculture*, vol. 178, p. 105 783, 2020.

[43] T. N. Carlson and D. A. Ripley, “On the relation between ndvi, fractional vegetation cover, and leaf area index,” *Remote sensing of Environment*, vol. 62, no. 3, pp. 241–252, 1997.

[44] T. Borowik, N. Pettorelli, L. Sönnichsen, and B. Jędrzejewska, “Normalized difference vegetation index (ndvi) as a predictor of forage availability for ungulates in forest and field habitats,” *European journal of wildlife research*, vol. 59, no. 5, pp. 675–682, 2013.

[45] J. A. Marchant, R. D. Tillett, and R. Brivot, “Real-time segmentation of plants and weeds,” *Real-time imaging*, vol. 4, no. 4, pp. 243–253, 1998.

[46] S. Gebhardt, J. Schellberg, R. Lock, and W. Kühbauch, “Identification of broad-leaved dock (rumex obtusifolius l.) on grassland by means of digital image processing,” *Precision Agriculture*, vol. 7, no. 3, pp. 165–178, 2006.

[47] M. Guijarro, G. Pajares, I. Riomoros, P. Herrera, X. Burgos-Artizzu, and A. Ribeiro, “Automatic segmentation of relevant textures in agricultural images,” *Computers and Electronics in Agriculture*, vol. 75, no. 1, pp. 75–83, 2011.

[48] D. Andújar, V. Rueda-Ayala, H. Moreno, *et al.*, “Discriminating crop, weeds and soil surface with a terrestrial lidar sensor,” *Sensors*, vol. 13, no. 11, pp. 14 662–14 675, 2013.

[49] A. Milioto, P. Lottes, and C. Stachniss, “Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in cnns,” in *2018 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2018, pp. 2229–2235.

[50] M. H. Siddiqi, I. Ahmad, and S. B. Sulaiman, “Weed recognition based on erosion and dilation segmentation algorithm,” in *2009 International Conference on Education Technology and Computer*, IEEE, 2009, pp. 224–228.

[51] A. I. Tejeda and R. C. Castro, “Algorithm of weed detection in crops by computational vision,” in *2019 International Conference on Electronics, Communications and Computers (CONIELECOMP)*, IEEE, 2019, pp. 124–128.

[52] A. AlSuwaidi, C. Veys, M. Hussey, B. Grieve, and H. Yin, “Hyperspectral selection based algorithm for plant classification,” in *2016 IEEE International Conference on Imaging Systems and Techniques (IST)*, IEEE, 2016, pp. 395–400.

[53] B. Cheng and E. T. Matson, “A feature-based machine learning agent for automatic rice and weed discrimination,” in *International Conference on Artificial Intelligence and Soft Computing*, Springer, 2015, pp. 517–527.

[54] A. Bakhshipour, A. Jafari, S. M. Nassiri, and D. Zare, “Weed segmentation using texture features extracted from wavelet sub-images,” *Biosystems Engineering*, vol. 157, pp. 1–12, 2017.

[55] W. LanLan and W. Youxian, “Weed/corn seedling recognition by support vector machine using texture features,” *African Journal of Agricultural Research*, vol. 4, no. 9, pp. 840–846, 2009.

[56] P. J. Herrera, J. Dorado, and Á. Ribeiro, “A novel approach for weed type classification based on shape descriptors and a fuzzy decision-making method,” *Sensors*, vol. 14, no. 8, pp. 15 304–15 324, 2014.

[57] S. K. Mathanker, P. R. Weckler, R. K. Taylor, and G. Fan, “Adaboost and support vector machine classifiers for automatic weed control: canola and wheat,” in *2010 Pittsburgh, Pennsylvania, June 20-June 23, 2010*, American Society of Agricultural and Biological Engineers, 2010, p. 1.

[58] A. Milioto, P. Lottes, and C. Stachniss, “Real-time blob-wise sugar beets vs weeds classification for monitoring fields using convolutional neural networks.,” *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 4, 2017.

[59] J. Champ, A. Mora-Fallas, H. Goëau, E. Mata-Montero, P. Bonnet, and A. Joly, “Instance segmentation for the fine detection of crop and weed plants by precision agricultural robots,” *Applications in plant sciences*, vol. 8, no. 7, e11373, 2020.

[60] P. Wang, Y. Zhang, B. Jiang, and J. Hou, “An maize leaf segmentation algorithm based on image repairing technology,” *Computers and electronics in agriculture*, vol. 172, p. 105 349, 2020.

[61] G. Bannerjee, U. Sarkar, S. Das, and I. Ghosh, “Artificial intelligence in agriculture: a literature survey,” *International Journal of Scientific Research in Computer Science Applications and Management Studies*, vol. 7, no. 3, pp. 1–6, 2018.

[62] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in *2017 international conference on engineering and technology (ICET)*, Ieee, 2017, pp. 1–6.

[63] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, “Yolact: real-time instance segmentation,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9157–9166.

[64] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[65] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, “Yolact++: better real-time instance segmentation,” *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[66] J. Dai, H. Qi, Y. Xiong, *et al.*, “Deformable convolutional networks,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 764–773.

[67] D. M. Woebbecke, G. E. Meyer, K. Von Bargen, and D. A. Mortensen, “Color indices for weed identification under various soil, residue, and lighting conditions,” *Transactions of the ASAE*, vol. 38, no. 1, pp. 259–269, 1995.

[68] *Bean production guide*, `https://library.ndsu.edu/ir/bitstream/handle/10365/9510/a1133_1997.pdf?sequence=1%26isAllowed=y`, Accessed: 2022-07-08.

[69] *Corn growth and development*, `https://bookstore.ksre.ksu.edu/pubs/MF3305.pdf`, Accessed: 2022-07-08.

[70] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[71] Y. Lee and J. Park, “Centermask: real-time anchor-free instance segmentation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 13 906–13 915.

[72] P. Shirley, M. Ashikhmin, and S. Marschner, *Fundamentals of computer graphics*. AK Peters/CRC Press, 2009.

[73] G. Jiang, X. Wang, Z. Wang, and H. Liu, “Wheat rows detection at the early growth stage based on hough transform and vanishing point,” *Computers and Electronics in Agriculture*, vol. 123, pp. 211–223, 2016.

[74] B. Åstrand and A.-J. Baerveldt, “A vision based row-following system for agricultural field machinery,” *Mechatronics*, vol. 15, no. 2, pp. 251–269, 2005.

[75] M. Pérez-Ruiz, D. Slaughter, C. Gliever, and S. Upadhyaya, “Automatic gps-based intra-row weed knife control system for transplanted row crops,” *Computers and Electronics in Agriculture*, vol. 80, pp. 41–49, 2012.

[76] T. Bakker, K. van Asselt, J. Bontsema, J. Müller, and G. van Straten, “Autonomous navigation using a robot platform in a sugar beet field,” *Biosystems Engineering*, vol. 109, no. 4, pp. 357–368, 2011.

[77] R. Eaton, J. Katupitiya, K. W. Siew, and B. Howarth, “Autonomous farming: modelling and control of agricultural machinery in a unified framework,” *International journal of intelligent systems technologies and applications*, vol. 8, no. 1-4, pp. 444–457, 2010.

[78] K. Michio, N. Noguchi, K. Ishii, and H. TERAO, “The development of the autonomous tractor with steering controller applied by optimal control,” in *Automation technology for off-road equipment proceedings of the 2002 conference*, American Society of Agricultural and Biological Engineers, 2002, p. 367.

[79] *Agthech carbon bee*, `https://carbonbee-agtech.fr/`, Accessed: 2022-07-08.

[80] A. S. Hassanein, S. Mohammad, M. Sameer, and M. E. Ragab, “A survey on hough transform, theory, techniques and applications,” *arXiv preprint arXiv:1502.02160*, 2015.

[81] K. G. Derpanis, “Overview of the ransac algorithm,” *Image Rochester NY*, vol. 4, no. 1, pp. 2–3, 2010.

[82] B. Espiau, F. Chaumette, and P. Rives, “A new approach to visual servoing in robotics,” *ieee Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 313–326, 1992.

[83] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[84] E. F. Camacho and C. B. Alba, *Model predictive control*. Springer science & business media, 2013.

[85] F. Allgöwer and A. Zheng, *Nonlinear model predictive control*. Birkhäuser, 2012, vol. 26.

[86] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “CasADi – A software framework for nonlinear optimization and optimal control,” *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[87] N. S. Chandel, A. K. Chandel, A. K. Roul, K. R. Solanke, and C. Mehta, “An integrated inter-and intra-row weeding system for row crops,” *Crop Protection*, vol. 145, p. 105 642, 2021.

[88] F. Abdelghafour, F. Rançon, S. Liu, *et al.*, “Weedelec: a robotic research platform for individual weed detection and selective electrical weeding,” *Delta*, vol. 1, p. C2, 2021.

[89] D. Pullen and P. Cowell, “An evaluation of the performance of mechanical weeding mechanisms for use in high speed inter-row weeding of arable crops,” *Journal of Agricultural Engineering Research*, vol. 67, no. 1, pp. 27–34, 1997.

[90] C. McCool, J. Beattie, J. Firn, *et al.*, “Efficacy of mechanical weeding tools: a study into alternative weed management strategies enabled by robotics,” *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1184–1190, 2018.

[91] H. Niemann and R. Miklos, “A simple method for estimation of parameters in first order systems,” in *Journal of Physics: Conference Series*, IOP Publishing, vol. 570, 2014, p. 012 001.

[92] Y. Li, Z. Guo, F. Shuang, M. Zhang, and X. Li, “Key technologies of machine vision for weeding robots: a review and benchmark,” *Computers and Electronics in Agriculture*, vol. 196, p. 106 880, 2022.

[93] A. Elfes, “Using occupancy grids for mobile robot perception and navigation,” *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[94] S. Thrun, “Learning occupancy grid maps with forward sensor models,” *Autonomous robots*, vol. 15, pp. 111–127, 2003.

[95] M. S. Grewal, A. P. Andrews, and C. G. Bartone, “Kalman filtering,” 2020.

Robotique mobile autonome en milieu ouvert

Résumé : L'agriculture joue un rôle vital dans le développement de l'humanité pour répondre à une demande alimentaire croissante. Pour cette raison, des produits chimiques comme les engrais, les herbicides, les fongicides, etc., sont utilisés pour protéger les cultures contre les maladies, les ravageurs et les mauvaises herbes. Cependant, l'utilisation de ces produits implique la dégradation des écosystèmes ainsi que des effets néfastes sur la santé humaine. Cette thèse propose une solution de désherbage mécanique autonome afin de remplacer/réduire l'utilisation d'herbicides. Dans un premier temps, la détection des cultures et des mauvaises herbes s'effectue à l'aide de techniques d'apprentissage automatique et de traitement d'image pour protéger les premières et éliminer les secondes. Ensuite, la détection des cultures est utilisée pour naviguer de manière autonome sans s'appuyer sur les données GPS en estimant le point de fuite formé par les rangées de cultures. Enfin, la commande de l'activation de l'outil de désherbage est proposée en s'appuyant sur la modélisation de la plateforme et la détection temps réel en proposant plusieurs variants. Les résultats rapportés dans cette thèse ont été obtenus en conditions agricoles réelles dans le cadre de deux challenges internationaux : le challenge ANR ROSE et la compétition européenne ACRE. De la même manière, ces conditions ont été recréées à l'aide du simulateur Gazebo et des jumeaux numériques afin de valider les approches proposées.

Mots clés : Robotique agricole, Désherbage autonome, Détection des cultures/mauvaises herbes , Navigation basée sur la vision, Vision et apprentissage, Challenge ROSE, Compétition ACRE.

Autonomous mobile robotics in open enviroments

Abstract: Agriculture plays a vital role in the development of humanity to meet a growing demand for food. For this reason, chemicals like fertilizers, herbicides, fungicides, etc., are used to protect crops from diseases, pests and weeds. However, the use of these products implies the degradation of ecosystems as well as adverse effects on human health. This thesis proposes an autonomous mechanical weeding solution in order to replace/reduce the use of herbicides. Firstly, the detection of crops and weeds is carried out using machine learning and image processing techniques to protect the former and eliminate the latter. Then, crop detection is used to navigate autonomously without relying on GPS data by estimating the vanishing point formed by the crop rows. Finally, the activation of the weeding tool is based on the modeling of the platform and real-time detection by proposing several variants.

The results reported in this thesis were obtained in real agricultural conditions within the framework of two international competitions: the ANR ROSE challenge and the European ACRE competition. In the same way, these conditions were recreated using digital twins and the Gazebo simulator in order to validate the proposed approaches.

Keywords: Agricultural robotics, Autonomous weeding, Crop/Weed detection, Vision-based navigation, Vision and machine learning ,ROSE challenge, ACRE competition.

Université de Limoges, CNRS, XLIM, UMR 7252, F-87000 Limoges, France