

## Semester Thesis

# Crop Row Detection Leading to Autonomous Navigation of a Laser-based Weeding Robot

Autumn Term 2022



Eidgenössische Technische Hochschule Zürich  
Swiss Federal Institute of Technology Zurich

## Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

**Title of work** (in block letters):

Autonomous Crop Row Detection of a Laser-based Weeding Robot

**Authored by** (in block letters):

*For papers written by groups the names of all authors are required.*

**Name(s):**

Mérat

**First name(s):**

Roxane

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the '[Citation etiquette](#)' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

**Place, date**

Zürich, 19/01/2023

**Signature(s)**

*For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.*

# Contents

<b>Abstract</b>	<b>iii</b>
<b>Symbols</b>	<b>iv</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Overview of previous work in the field of crop row detection</b>	<b>3</b>
2.1 LiDAR based techniques . . . . .	3
2.2 Convolutional Neural Networks . . . . .	3
2.3 Classical Machine Vision approach . . . . .	3
2.3.1 Segmentation . . . . .	3
2.3.2 Line-detection . . . . .	4
<b>3 Methodology</b>	<b>7</b>
3.1 Global description of the algorithm . . . . .	7
3.2 Image Preprocessing : sky extraction and vegetation segmentation .	7
3.3 First approximation of the crop rows : Hough Transform . . . . .	11
3.4 Vanishing Point Calculation and outliers elimination . . . . .	12
3.5 Final detection : constrained RANSAC . . . . .	15
<b>4 Evaluation</b>	<b>17</b>
4.1 Datasets Used . . . . .	17
4.2 Results on the Crop Row Benchmark Dataset . . . . .	17
4.2.1 Evaluation Method . . . . .	17
4.2.2 Results . . . . .	18
4.3 Results on Catterra's dataset . . . . .	19
<b>5 Discussion</b>	<b>21</b>
5.1 Failure cases . . . . .	21
5.2 Future Work . . . . .	21
<b>6 Conclusion</b>	<b>23</b>
<b>Acknowledgement</b>	<b>24</b>
<b>Bibliography</b>	<b>26</b>
<b>7 Appendix - Hyperparameters</b>	<b>27</b>

# Abstract

This project aims to develop a computer vision-based system for detecting crop rows in agricultural fields, which is a crucial step in the development of an autonomous agricultural rover robot, enabling the autonomous navigation by providing proprioceptive information. The system utilizes a combination of image processing techniques, including color clustering for vegetation segmentation, Hough transform for crop row detection, as well as vanishing point detection and RANSAC for improved accuracy. Results show that the system is able to detect crop rows with a median accuracy of 82% following the formula of the crop row detection accuracy developed by Vidović et al [1].

# Symbols

## Symbols

$\rho$	distance of a point from the origin
$\theta$	angle between the point and the positive x-axis

## Indices

$x$	x axis
$y$	y axis

## Acronyms and Abbreviations

CNN	Convolutional Neural Networks
LiDAR	Light Detection And Ranging
IMU	Inertial Measurement Unit
UAV	Unmanned Aerial Vehicle
CRBD	Crop Row Benchmark Dataset
CRDA	Crop Row Detection Accuracy
GNSS	Global Navigation Satellite System
RANSAC	Random Sample Consensus
VO	Visual Odometry

# Chapter 1

## Introduction

One of the major challenges facing the agricultural industry today is the problem of weeds. Traditional methods of weed removal include large amounts of herbicides, which are harmful to the environment, or manual labor, which is extremely time-consuming and tiring. To address this issue, Cattera, a project under the Crop Science Laboratory of ETH Zurich, is developing a robotic system that uses a laser module to automatically eliminate weeds.

This project requires the rover robot to navigate autonomously. For now, most agricultural robots use GNSS to accomplish this task. However, GNSS-based navigation is prone to many problems:

- It relies on a fixed infrastructure, such as ground-based reference stations, which may not be available or practical in remote agricultural areas.
- The field conditions, weather, and other environmental factors can affect the accuracy of the GNSS signals and cause errors in navigation.
- The field environment is constantly changing, with crops growing and harvested, making it difficult for the robot to accurately map and navigate the area.

In general, GNSS can provide a rough location of the robot in an agricultural field, but it is not the most robust and reliable solution for autonomous navigation. The need for a more robust method with instant feedback is necessary. Machine vision approaches, specifically visual odometry using an RGB camera, have been extensively researched as an alternative. These methods provide greater robustness and precision and are cost-effective and easy to maintain.

The detection of crop rows is a key aspect in enabling proprioception in agricultural robots and can also be used for other important applications such as crop counting, yield prediction, and field mapping. However, due to the variability in crop type, growth stage, and environmental conditions, detecting crop rows is a challenging task. Many image processing techniques that have been developed are often limited in their ability to detect crop rows robustly under different conditions and their need for a significant amount of manual tuning, and there is a pressing need to develop more robust, adaptable, and automated image processing techniques that can effectively handle the variability of crop rows and environmental conditions.

To develop such an algorithm, we have integrated three key principles: Hough transform to extract the main lines from the image, vanishing point calculation to eliminate outliers, and RANSAC to obtain more precise information about the

center of each crop row while taking into consideration the fundamental characteristics of crop rows. This will enable further research on the autonomous navigation capabilities of the rover robot.



Figure 1.1: Catterra's agricultural robot

## Chapter 2

# Overview of previous work in the field of crop row detection

A lot of research has been done to autonomously detect crop rows using computer vision. The main approaches are LiDAR based techniques, CNN's, and more classical Machine Vision approaches.

### 2.1 LiDAR based techniques

As Zhang et al. [2] and Reiser et al. [3] have shown, height information can be used to detect crop rows. However, while the detection of crops during the late period is successful with LiDAR, this technique is not reliable when it comes to the detection of tiny plants, in the early stages of development. The cost of LiDAR sensors is also high and is not suitable for large-scale configuration of agricultural vehicles.

### 2.2 Convolutional Neural Networks

CNN can also be used to process images of crop rows and detect rows in new images (Bah et al. [4], Doha [5], Pang et al. [6]). However, deep learning methods are limited by the available datasets. As the appearance of the crop row varies a lot depending on the type of crop, the growth stage, the environmental condition and the weather condition, the required dataset must be huge. In our case, we would also need images taken from the same height and angle as our rover. Today, such large-scaled datasets are not available, and acquiring our own would be very difficult, time consuming, and should be done at various seasons of the year and on various crops.

### 2.3 Classical Machine Vision approach

#### 2.3.1 Segmentation

To distinguish crops from the background, segmentation is usually performed using color index-based segmentation followed by threshold-based segmentation ([7]).



Color index segmentation is widely used in agricultural image processing, the most common being the Excess Green Index (ExG, [8]), the Color Index of Vegetative Extraction (CIVE, [9]) and the normalized difference vegetation index (NDVI, [10]). Post-processed, the new gray image has higher-intensity pixels in place of the pixels. Once the image is grayscaled, it is usually binarized. The selection of the threshold has to be very well tuned. A common technique is to use Otsu thresholding (Otsu [11]), which autonomously finds a threshold that maximizes the variance between the foreground (the vegetation) and the background. This works by iterating through all possible threshold values and for each one, calculating the variance (how much the intensities in each group differ from the group's mean intensity) between the two groups of pixels.

As these techniques are highly sensitive to light changes, some use learning-based segmentation, mainly deep learning, to obtain the best segmentation values (Zhang et al. [12]).

Since no large dataset for vegetation segmentation was accessible, here a threshold-independent segmentation method using color clustering was developed, which will be described later.

### 2.3.2 Line-detection

Once the vegetation is segmented and a binary image has been computed, a line detection algorithm is necessary to extract crop rows. Currently, the most commonly used line-fitting techniques are the Hough transform, the least squares method, and the Random Sample Consensus.

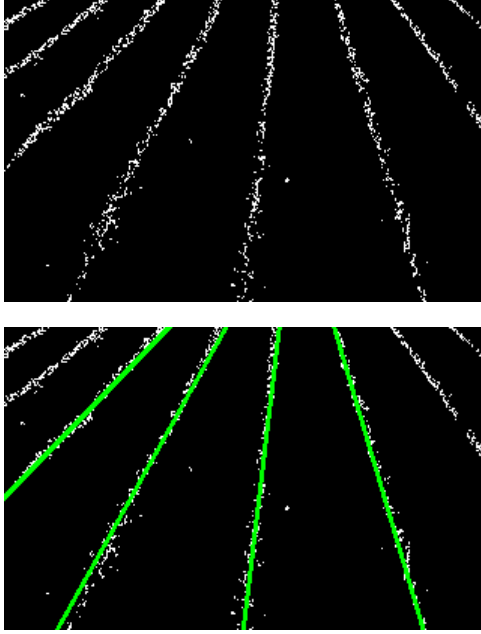
#### Hough Transform

Used to detect lines, circles, or more complex shapes, the Hough transform converts the pixel space to a parameter space, as illustrated in Fig. 2.1 In the case of line detection, each line in the image is described by the parameters  $\rho$  and  $\theta$  as such :

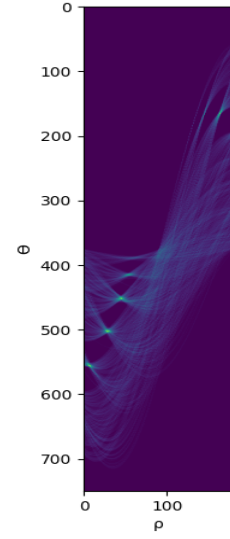
$$\rho(\theta) = x_0 \cdot \cos \theta + y_0 \cdot \sin \theta \quad (2.1)$$

The Hough space is defined as a graph with abscissa  $\theta$  and  $\rho$ , so each point  $(\theta_i, \rho_i)$  describes a line in the image space. Once each pixel of an image is mapped to the Hough space, the intensity values at a certain point  $(\theta_i, \rho_i)$  will be higher when this point describes a prominent line in the image pixels plane.

This technique is one of the most widely used (Marchant and Brivot [13]) but, on its own, it is not sufficient to detect crop rows robustly. It is highly sensitive to noise (lightning, shading, bushiness, various stages of growth), which requires substantial manual tuning and can lead to false detection of lines. In a region of high weed pressure, many lines are feasible. It is also computationally expensive and difficult to use for real-time applications.



(a) Line detected in Image Space by the Hough Process



(b) Accumulator in the Hough Space  
 $\rho$  [pixel count],  $\theta$  [rad]

Figure 2.1: Illustration Hough process

### Least square

Montalvo et al. [14] and Chen et al. [15] use least squares to fit the different rows of crops. Although faster and more robust to weeds, this technique requires prior knowledge to fit the template, such as the number of crop rows to be detected, the expected location of each crop row, and the area to be explored in the image.

### RANSAC

RANSAC (RANdom SAmple Consensus) is an iterative algorithm for estimating the parameters of a mathematical model from a set of observed data that contains outliers (i.e. data that do not fit the model). Here, it is used for line fitting, and the model is defined by the slope and a point on the line. The basic idea behind it is to randomly select a subset of the data points (2 inliers are selected for line fitting) and use them to estimate the parameters of the model. The remaining points (the outliers) are then used to test the validity of the model. This process is repeated multiple times, and the model with the most inliers is chosen as the final solution. The inliers are the points that fall within a certain distance threshold from the line, and the outliers are the points that fall outside of this threshold. RANSAC is a robust algorithm that can handle outliers and still find the correct solution - in our case, it will eliminate weeds or vegetation in between crops. It is faster than the complete Hough transform method and, thus, more adapted for real-time applications. <https://www.overleaf.com/project/6397556aaa04834b75ffb811>

However, as with the least squares method, some prior information on crop rows is necessary: Xie et al. [16] based their RANSAC process on a skeleton of the row features.

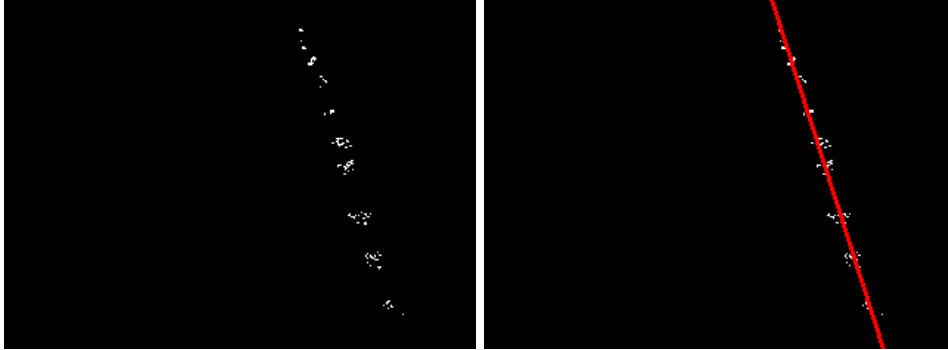


Figure 2.2: Results of the RANSAC algorithm for line fitting

In this project, an algorithm is developed that combines different methods to make each of them more robust and efficient. Hough transform is used as a preprocessing step to calculate the vanishing point and the approximate location of crops, the vanishing point is used to detect outlier lines not parallel in world coordinates, and RANSAC is used to detect the center of each crop rows.

# Chapter 3

## Methodology

### 3.1 Global description of the algorithm

The Algorithm can be divided into four main steps :

1. Preprocessing : segments the vegetation from its background
2. Hough Transform : extracts the main lines of the image
3. Outlier line detection : using the vanishing point, eliminates the lines that are not crossing at that point and replaces them with the next most prominent lines in the image
4. RANSAC : using points around mask corresponding to the regions around the lines previously detected, calculates the center of the crops

The complete algorithm can be visualized in Fig. 3.1.

The algorithm works on single images or sequences of images. As the Hough transform is computationally expensive, the algorithm doesn't use it at each frame : at the end of the RANSAC process, the region of the crops for the next frame is updated : as low changed between each frame was considered, the region where the crops were on the previous frame is used as a reference. As long as the crops detected are considered "valid", only the RANSAC process is runned again, the vanishing point staying the same.

Frames are considered invalid if two crops are too close to each other, if two crops have too similar an angle, if a crop has an angle too close to the horizon, if there are too few vegetation pixels in the area of the crops detected in the previous frame, or if the frame number is a multiple of K (K can be adapted according to the precision needed).

### 3.2 Image Preprocessing : sky extraction and vegetation segmentation

The Preprocessing part of the algorithm serves many purposes:

1. Extracts and resizes images
2. Removes the sky
3. Removes the extremely bushy regions

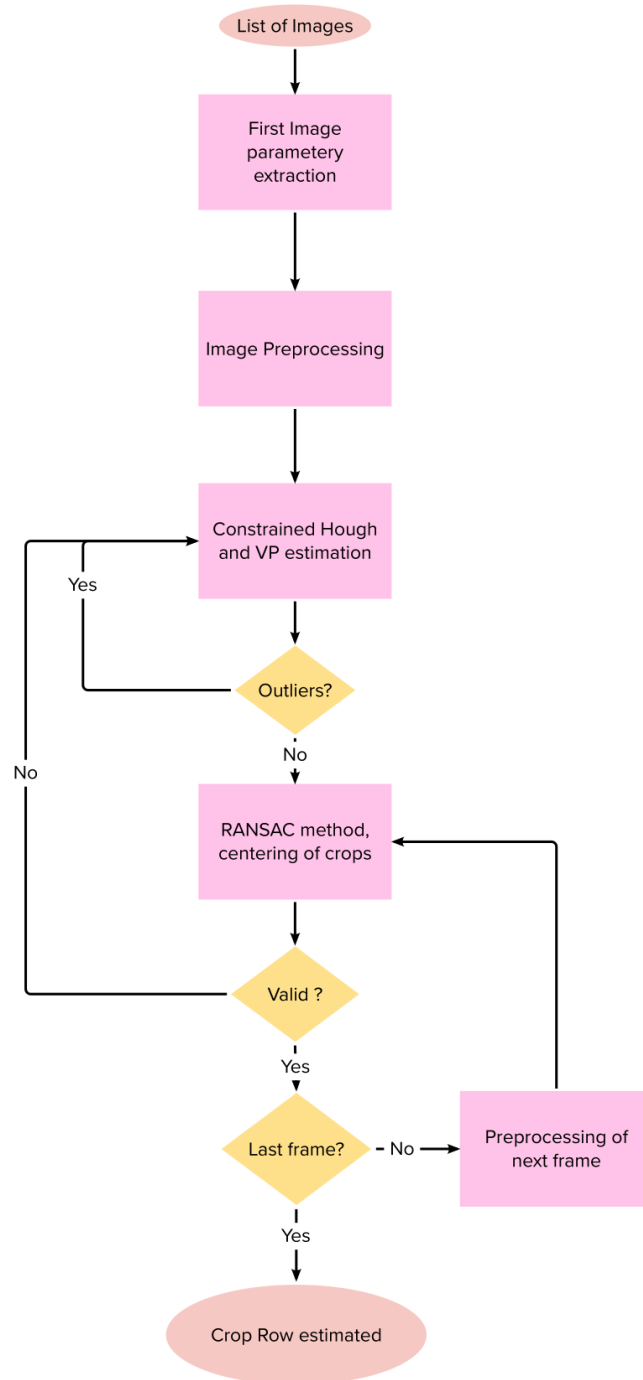


Figure 3.1: Flowchart of the Algorithm

4. Clusters the main colours and extracts the greenest one
5. Masks the image using a custom threshold on this colour
6. Depending on the density of the segmented vegetation, calculates edges of the image or keeps the entire image before proceeding with the first Hough Transform

### Sky Removal

This step reduces the Region of Interest of our image to speed up the algorithm. Results are shown in Fig. 4.2 It consists of three main parts:

- Laplacian Calculation and thresholding : the Laplacian (the second derivative of the image) is calculated to identify edges and fine details. Pixels with low Laplacian are kept -these are the areas of small changes, which should represent the sky.
- Horizon Line Detection : after median filter is applied per column, the first pixels of each column with a high level of change are memorized.
- Elimination of non-sky regions : Starting from the top row, the algorithm eliminates horizontal rows until it finds a row where a line is composed of less than 50% sky (low Laplacian pixels).



Figure 3.2: Results of the sky removal algorithm

### Colour Clustering, vegetation colour extraction

As the vegetation's colour varies a lot depending on the lighting, a threshold for the vegetation index isn't robust enough.

K clusters of colors are created by counting the number of pixels with the same color in an image. The color with the highest pixel count is grouped with all other colors that have a difference in LAB space less than a certain threshold. This process is repeated for the second highest pixel count color, and so on, until K clusters are formed. The outcome can be seen in figure 3.3.

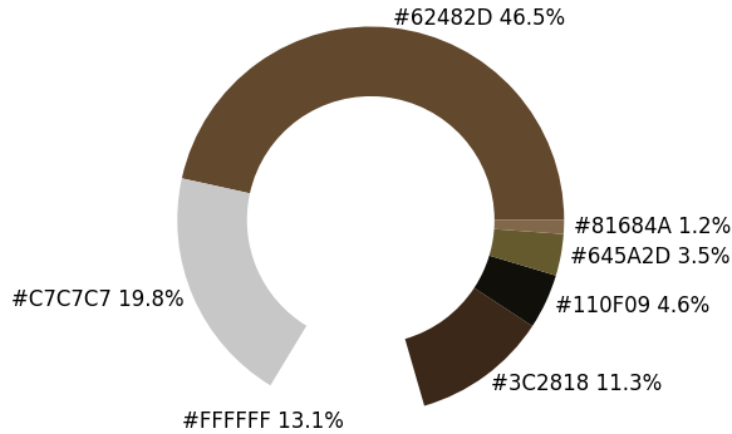


Figure 3.3: Visualisation of the clustered colour

The algorithm compares the clustered colors to the color with the norm closest to  $[0, 255, 0]$  (lime in RGB space) in LAB space. This color is used to create a mask for the image and segment the vegetation by retaining all pixels with a LAB space difference under a certain threshold.

In the video mode of the algorithm, the assumption is made that the color of the vegetation does not change much throughout the video. The reference vegetation color is calculated in frame 0. Although the reference color does not change, the threshold used during masking does, as the goal is to maintain the same level of information throughout the video (the ratio of 1-pixels to 0-pixels in the binary image being used to define "information."). The threshold value is adapted based on this ratio: if the ratio is smaller than the one in frame 0, the threshold is increased, and vice versa. This allows the algorithm to quickly segment the vegetation while maintaining the same level of information throughout the process. Recalculating the reference color at each frame is possible, but this technique has shown satisfactory results, is less computationally expensive, and is more stable.

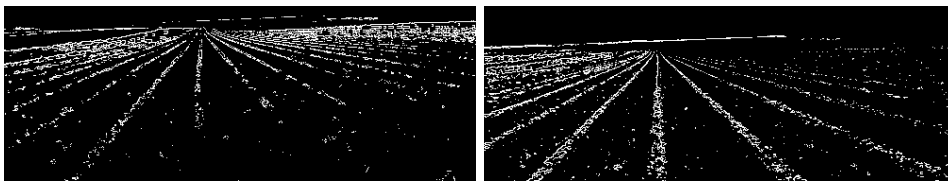


Figure 3.4: Vegetation Image of frame 5 and frame 500

### Bushy Region Removal

The image is eroded by a kernel proportional to its size. This new image is then subtracted from the original image. Results can be shown in Fig. 5.1

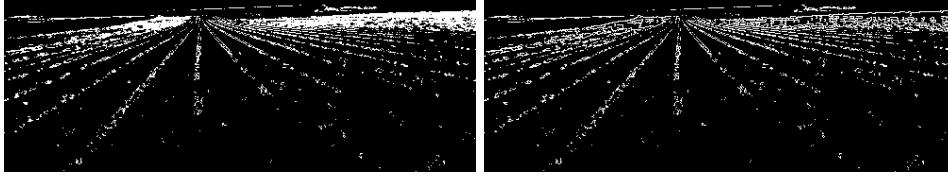


Figure 3.5: Erosion of the bushy regions

### 3.3 First approximation of the crop rows : Hough Transform

Our interest in the Hough Transform is two-fold : the lines are both used to calculate the vanishing point and to isolate the regions where the crop rows should be found in the RANSAC part of the algorithm.

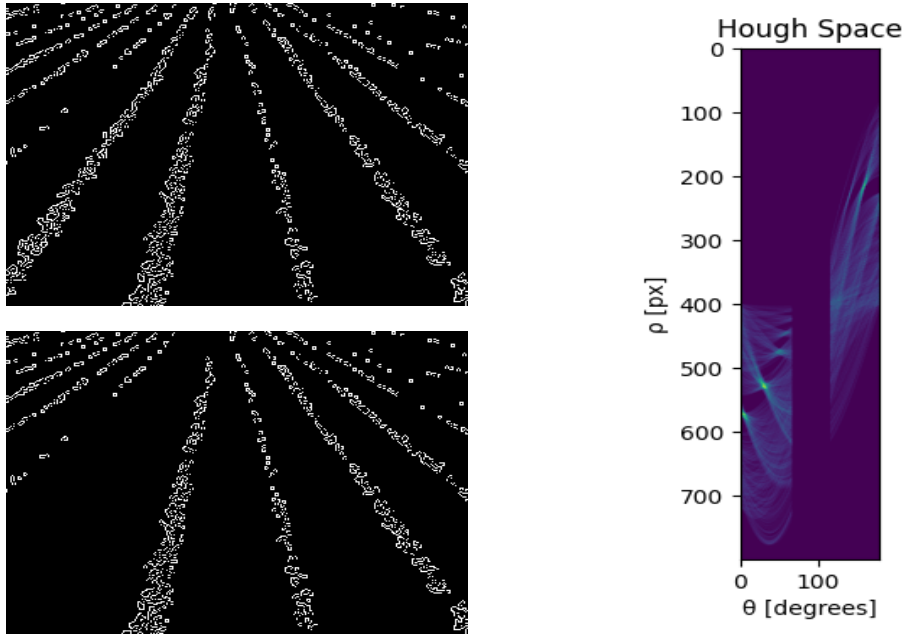


Figure 3.6: Illustration of the Hough process

The Hough Transform described in 2.3.2 finds all the lines in the image with values in the accumulator greater than a certain threshold.

To ensure that the lines detected are spaced by a minimum threshold and no angle is the same, the algorithm implemented in this project uses a slightly different Hough transform.

Depending on the density of the vegetation, either the entire binary image or the edges of the binary image are taken (if the vegetation pixels are more then 10% of the entire pixels) to accelerate the process.

Instead of using a threshold to take all lines, the N-maximums of the Hough space are extracted, where N is the number of desired rows. N is set to a defined number but the algorithm can decrease this number if not enough lines satisfying the "crop



conditions” are detected (as further explained in section 3.4). This approach allows the algorithm to be independent of a threshold.

When in Hough space, the maximum values to be selected must satisfy the following conditions :

1. No crops should have the same angle : this is done by putting a 0-band in the Hough space around each previously detected angle (cf Fig 3.6).
2. Each 1-pixels should be used only once, to detect a line of a single row. This is done by masking the surroundings of where a line has been detected (cf Fig. 3.6).

Deletion of lines and angles is done iteratively.

This process ensures no crops are detected twice, but doesn’t prevent detecting lines that are not going towards the vanishing point.

### 3.4 Vanishing Point Calculation and outliers elimination

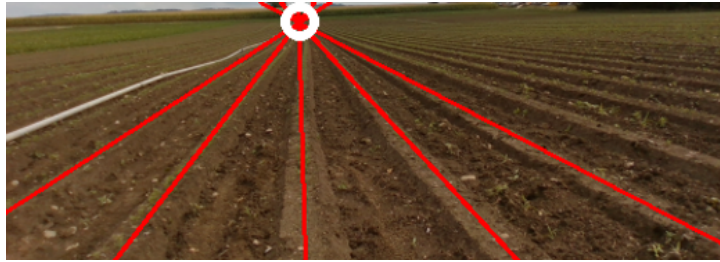


Figure 3.7: Vanishing Point detected

This step makes sure that the lines in camera coordinates detected using the Hough transform all go towards the vanishing point, representing parallel lines in the world coordinates.

It is worth noting that the calculation of the vanishing point could in the future be used for other tasks, such as camera calibration, depth estimation, and perspective distortion correction.

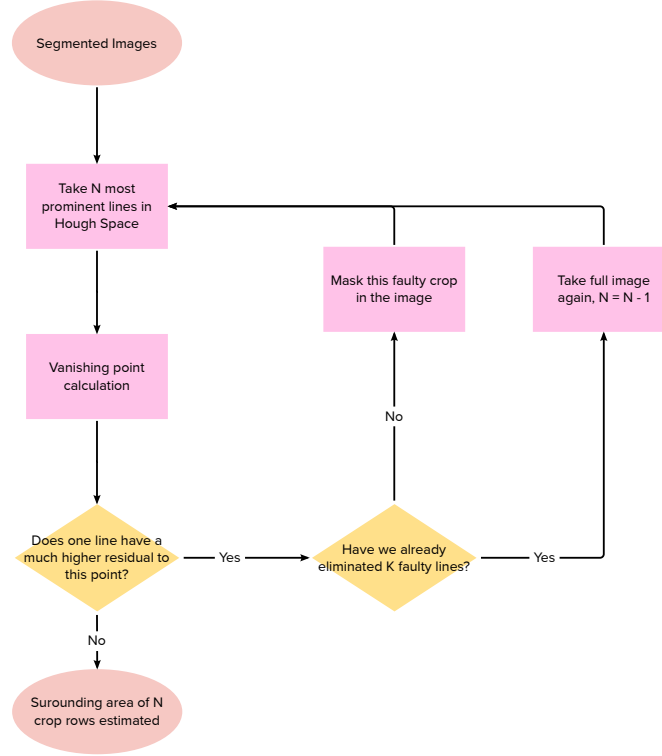


Figure 3.8: Flowchart of the Hough Process and Vanishing Point Calculation

Here, as it was proposed in [17], the vanishing point is determined in the Hough Space and a iterative process is set up to eliminate outlier lines.

As mentioned before, a point in the plane image is represented by a sinusoidal in the Hough space, which corresponds to all possible lines that could pass through this point. Thus, all the parallel lines in the world which pass through the vanishing point in the image plane should be on the same sinusoidal in the Hough space. Finding this sinusoidal should therefore describe the vanishing point in the following way:

$$\rho(\theta) = x_0 \cdot \cos \theta + y_0 \cdot \sin \theta = P(x_0, y_0, \Phi) \cdot \sin \theta + \Phi(x_0, y_0). \quad (3.1)$$

with :

$$\begin{cases} P = x_0 \cos(\Phi) \cdot \sin(1 + y_0^2/x_0^2) \\ \Phi = \arctan y_0/x_0 \end{cases} \quad (3.2)$$

In the developed algorithm, a deterministic method is obviously not enough, as all parallel lines in the real world do not cross exactly at a single point, and as there could be "outlier" lines, not corresponding to parallel crops in the real world. The following statistical method was thus used :

1. Extract the dominant sinusoidal in the Hough space to get a first estimate of the vanishing point. This is done by solving the following minimisation problem :

$$\min_{x_0, y_0} \sum_{i=1}^n W_i (\rho_i - x_0 \cos(\theta_i) - y_0 \sin(\theta_i))^2 \quad (3.3)$$

Solving it leads to:

$$D_{it} = \begin{cases} Ax_0 + Cy_0 = D \\ Cx_0 + By_0 = E \end{cases} \quad (3.4)$$

with :

$$\begin{cases} A = \sum_{i=1}^n W_i \cdot \cos^2 \theta_i \\ B = \sum_{i=1}^n W_i \cdot \sin^2 \theta_i \\ C = \sum_{i=1}^n W_i \cdot \sin \theta_i \cdot \cos \theta_i \\ D = \sum_{i=1}^n W_i \cdot \rho_i \cdot \cos \theta_i \\ E = \sum_{i=1}^n W_i \cdot \rho_i \cdot \sin \theta_i \end{cases} \quad (3.5)$$

2. Once this point is found, compute the total variance of all lines with respect to this point :

$$\sigma^2 = \sum_{i=1}^n W_i \epsilon_i^2 \quad (3.6)$$

with :

$$\epsilon_i = \rho - \bar{\rho}_i \quad (3.7)$$

3. If one of the lines is much more distant than this variance, that is  $\epsilon > k * \sigma$ , classify it as an outlier and remove it from the detected line lists. An example of such a line is shown in Fig. 3.9
4. If more than k lines have already been detected as outliers, start detecting N-1 lines.
5. Repeat the process: take the next dominant line in Hough space if N is unchanged, use the new set of lines to recalculate the vanishing point, and eliminate more outliers if necessary.
6. Stop when no more outliers are detected.

A more concise explanation of this part can be visualized in Fig. 3.8

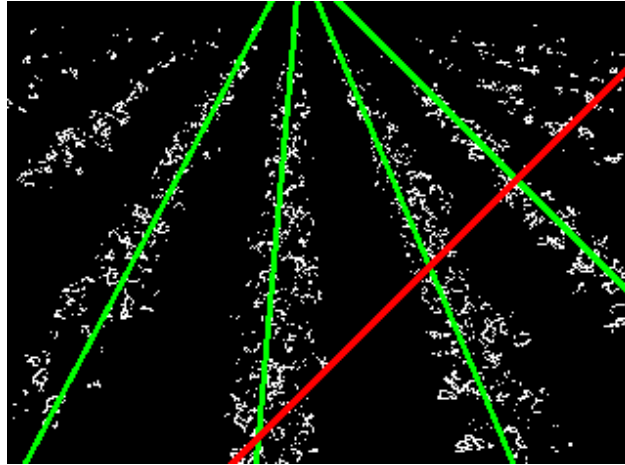


Figure 3.9: Outlier line detected, soon to be eliminated

### 3.5 Final detection : constrained RANSAC

Once the estimate of the crop row position is estimated with the Hough transform and the vanishing point is calculated, RANSAC is used to fit lines in the center of the crop rows. Each fitted line uses a masked version of the vegetation segmented image, where only the regions surrounding the previously detected lines are kept, as shown in Fig. 3.5.

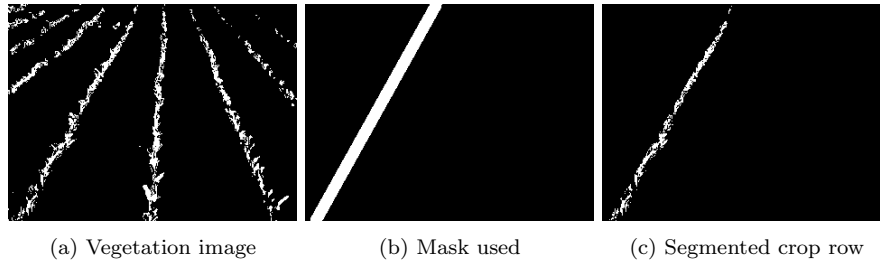


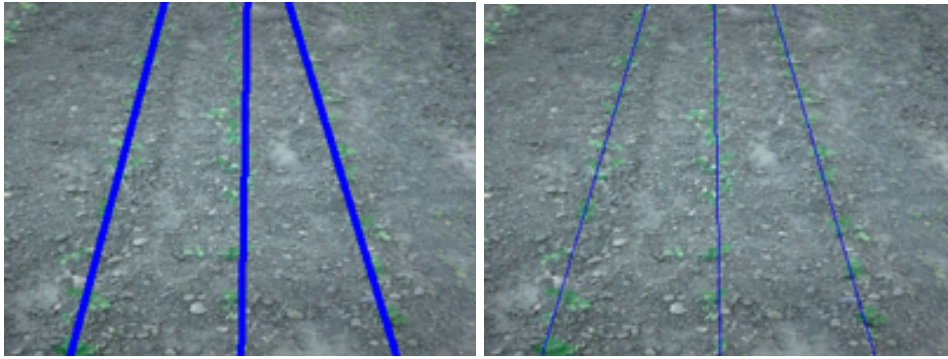
Figure 3.10: Segmentation of a single crop row

In this project, two approaches were tried to fit the rows with RANSAC :

- Fit the crops between the surrounding region of the vanishing point previously calculated and the surroundings of the lines detected with the Hough transform. As a personalized Python version of the RANSAC function was used, the process took a lot of time. It also needed additional iterations to find the best model, as an extra parameter was needed (the vanishing point's surroundings).
- Fit a single crop around the surrounding region of a line detected with the Hough transform, with a heavy bias towards the vanishing point - for this, the data points used for fitting a single line are augmented with multiple times the vanishing point coordinates - this method was selected for the final process, as it was faster and gave similar results.

There are two benefits to using RANSAC instead of the Hough transform :

1. Hough transform is too slow for real-time application, using RANSAC on most frames and Hough only when necessary speeds up the crop row detection process.
2. The lines fitted with the Hough Algorithm are not necessarily the centers of the crops : when working with the edges of the image, they will even be fitted to the side of the crops. The data used for RANSAC being selected randomly from all the points of the crops, the lines detected are likely to be the center of the crops.



(a) Hough detection process, lines not fitted in the center  
(b) Ransac detection process, lines centered on the crops

Figure 3.11: Comparaison Hough and Ransac Process

## Chapter 4

# Evaluation

### 4.1 Datasets Used

To evaluate the developed algorithm, two datasets were used:

- The Crop Row Benchmark Dataset (CRBD) [1] is a collection of images featuring various types of crops that can be used to evaluate crop row detection methods. The dataset includes pictures of crops such as maize, celery, potato, onion, sunflower, and soybeans, with varying levels of weeds and shadows. Some images also contain elements such as grass, sky, or roads. The images were taken at different angles and captured using a Panasonic LUMIX DMC-F2 digital camera in the spring of 2014 in the Croatian region of Slovenia. The dataset has a total of 281 images, including 47 images of straight crop rows, which are the main focus of interest of this project.
- Caterra's own dataset, taken with its front camera intel realsense d435i. It was used to evaluate the algorithm on time-related frames. For this data set that contains no ground truth, a simple visual validation was conducted.

It is important to point out that further evaluations should be conducted on the algorithm, especially to evaluate it on video and later to use it in real time. Weather conditions such as rain, snow, and heavy mist were not evaluated as not part of any dataset.

### 4.2 Results on the Crop Row Benchmark Dataset

#### 4.2.1 Evaluation Method

To compare the results obtained against the ground truth, two matrices are constructed, one for the ground truth and one for the algorithm developed. In this matrix, the number of rows corresponds to the number of image rows ( $h$ ), and each row of the matrix has  $m$  elements, represented by the horizontal coordinates of  $m$  adjacent crop rows detected by the method considered in the  $v$ -th image row. To compute the accuracy, we use the CRDA formula defined in [18] as:

$$CRDA = \frac{1}{m * (h - v_0)} \sum_{v=v_0}^{h-1} \sum_{i=1}^m s(u_{v,i}^*, u_{v,i}, d_v^*) \quad (4.1)$$

with :

$$s(u^*, u, d^*) = \max \left( 1 - \left( \frac{u^* - u}{\sigma d} \right)^2, 0 \right) \quad (4.2)$$

- $m$  — number of crops to be detected
- $v_0$  — first line where evaluation starts (could be bushy with undefined crop position on top of image)
- $h$  — height of the image
- $u^*$  — ground truth on line  $v$
- $u$  — personal results on line  $v$
- $\sigma$  — user defined parameter that is set based on the accuracy needed: in the following evaluation, it is set to 0.3, which means that the matching score is greater than zero only if the horizontal distance between a detected crop row and the corresponding ground truth curve is less than 30% of the distance between adjacent crop rows
- $d^*$  — distance between crops in ground truth

### 4.2.2 Results

Our algorithm achieved a median CRDA of 0.816. Some examples of detected crop rows and their score are shown in Fig. 4.1. The algorithm presents better results for less bushy crop rows, as when bushes are too present, the Hough transform has many lines to choose from. When weeds are far too present in the center of the crop rows, the algorithm may detect an extra row - this is hopefully not the case if an agricultural robot passes regularly. It is also worth noting that some of the crops from the CRDA images are indistinguishable even from the human eye.

Despite the high level of accuracy, the current computational time of the algorithm is too high to be implemented in real-time applications. While the images processed using solely the RANSAC were computed in milliseconds, the average time to process an image using the Hough Transform was 10 seconds. Depending on the accuracy image, this transform can be performed very regularly, which is not practical for real-time applications. As a future work, it would be important to optimize the algorithm's computational time, to enhance the possibility of using it in real-time applications.

Datasets	Mean	Median
All Crop Rows	0.773	0.816
Non bushy crop rows	0.821	0.856
Bushy crop row	0.625	0.689

Table 4.1: Results on the CRBD

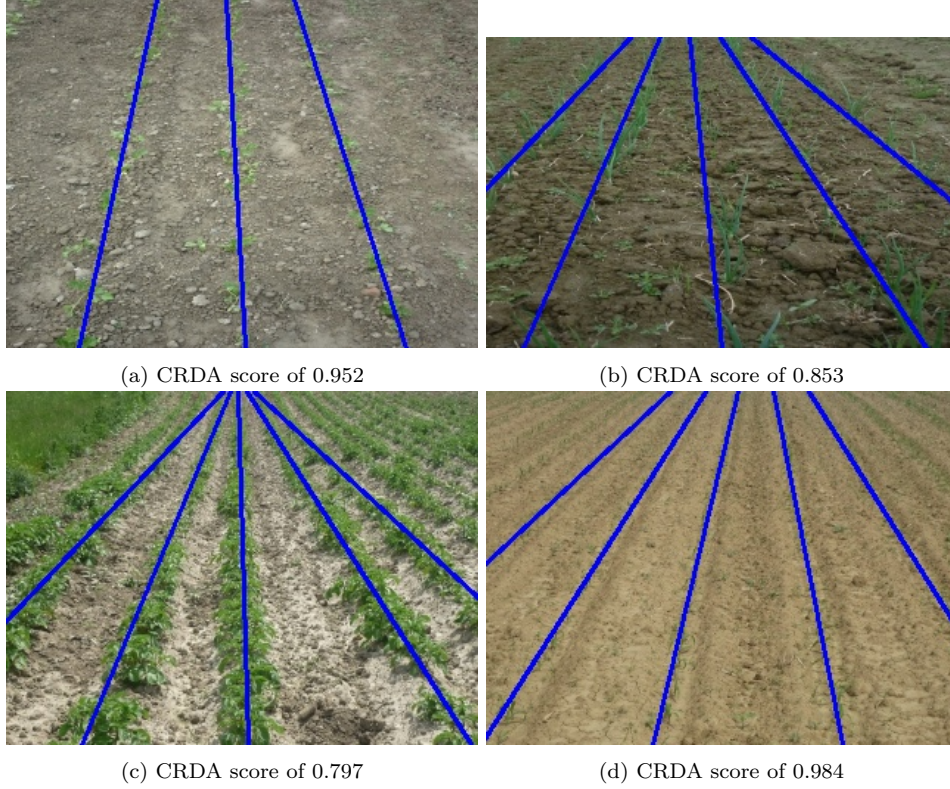


Figure 4.1: Results of the algorithm on different types of crop rows

### 4.3 Results on Catterra's dataset

The Rosbag was captured using the rover's front camera, a realsense d435i. The Rosbag contained 776 images of the rover advancing in the field. For visualization, the annotated video can be seen here : <https://youtu.be/yA1oaAT4Wgg>.

Depending on the accuracy needed, the number of frames before recalculating the Hough transform and the vanishing point varies. For the visualized video, we have chosen  $K = 5$ .

Crops are detected accurately most of the time, and when an error occurs, it is quickly corrected by the algorithm when this one recalculates the hough transform.





Figure 4.2: Annotated frames 0, 200, 400, and 600 of the dataset

# Chapter 5

## Discussion

### 5.1 Failure cases

As the algorithm focuses on fitting straight lines, more work is needed to detect curved crop rows (Fig. 5.1a). The Hough and RANSAC processes should be adapted to use custom models, not just linear models. This would mean complications when it comes to the detection of vanishing points and thus for outlier elimination.

Another issue is the non-ordered detection of the lines : since the lines are detected by order of prominence, the first line is not necessarily the most centered one. This can be a problem if a large number of lines is present in a picture : a central row may not be part of the main N rows and is only detected when we adjust the number of crop to be detected (as shown in Fig. 5.1c) This problem has many solutions : as crops are set in an equidistant way, the distance between two crop rows could be used as an extra constraint. An easier solution would be to perform a search in the middle of two rows when their distance has a high deviation from the average distance between rows in a picture.

Finally, in some extreme cases where bushes are too prominent, the Hough transform detects too many faulty lines and does not manage to calculate a correct vanishing point. The lines kept as "inliers" are the faulty ones.



(a) Curved crop rows

(b) Wrong order detection

(c) Bushy crop rows

### 5.2 Future Work

We have successfully developed an algorithm that detects crop rows in an RGB image or video with high accuracy. Nevertheless, improvements could be made and different cases could be taken into account.

1. Computational time : currently, the algorithm is too slow for real-time applications. A more efficient image processing technique and parallel processing could be implemented.

2. Prior information : Our algorithm could be made even more robust by integrating more prior information, such as the distance between crop rows or the number of crop rows
3. Sensors fusion : integrating additional sensors, such as LiDAR, would enhance the algorithm's accuracy
4. Deep Learning : for even better crop row detection or vegetation segmentation, deep learning could be added
5. Camera Calibration: The algorithm assumes that the camera is calibrated, which means that the intrinsic and extrinsic parameters of the camera are known. However, in real-world applications, the camera may not be calibrated, or the calibration may drift over time. Therefore, it would be beneficial to include an automatic camera calibration module to the algorithm to improve its robustness.

Of course, the main reason for crop row detection is the navigation of the rover : path planning is another important aspect to consider. This project has concentrated on detecting crop rows and does not provide information about the optimal path for the robot to follow. This path could be generated using techniques such as the A\* algorithm, Voronoi diagrams, or other known path planning techniques to ensure that the robot can navigate efficiently and safely between detected crops.

It is also important to note that the dataset used is not representative of all crops: different data sets should be collected to better measure the degree to which the algorithm stays accurate.

## Chapter 6

# Conclusion

In conclusion, the crop row detection algorithm developed in this report utilizes simple image processing techniques such as color clustering, vanishing point calculation, the Hough transform, and RANSAC.

The algorithm is robust to different illuminations, growth stages, and points of view. Since simple computer vision techniques were used, it is also generalizable and is not limited to a single type of plant.

The algorithm was found to be accurate for straight crop rows, especially for images with low to medium vegetation density. It achieved a median accuracy of 82% when tested on a dataset of various images, demonstrating its effectiveness in detecting crop rows in images.

In the future, this algorithm could be further improved by using additional techniques such as machine learning and deep learning models, or LiDAR for very bushy regions. Overall, the algorithm shows promise in being a useful tool for precision agriculture and crop monitoring, but further optimization is needed to make it real-time.

# Acknowledgement

I would like to thank my supervisors Patrick Barton and Cornelius von Einem for their guidance and support throughout my Master's semester project. I am grateful for their mentorship, their time and their expertise.

# Bibliography

- [1] V. et al, “Crop row benchmark dataset,” 2016.
- [2] S. Zhang, Q. Ma, S. Cheng, D. An, Z. Yang, B. Ma, and Y. Yang, “Crop row detection in the middle and late periods of maize under sheltering based on solid state lidar,” *Agriculture*, vol. 12, no. 12, 2022.
- [3] D. Reiser, G. Miguel, M. V. Arellano, H. W. Griepentrog, and D. S. Paraforos, “Crop row detection in maize for developing navigation algorithms under changing plant growth stages,” in *Robot 2015: Second Iberian Robotics Conference*, L. P. Reis, A. P. Moreira, P. U. Lima, L. Montano, and V. Muñoz-Martinez, Eds. Cham: Springer International Publishing, 2016, pp. 371–382.
- [4] M. D. Bah, A. Hafiane, and R. Canals, “Crownnet: Deep network for crop row detection in uav images,” *IEEE Access*, vol. 8, pp. 5189–5200, 2020.
- [5] R. M. Doha, “Master of science in mechanical engineering,” Master’s thesis, Purdue University, may 2022.
- [6] Y. Pang, Y. Shi, S. Gao, F. Jiang, A.-N. Veeranampalayam-Sivakumar, L. Thompson, J. Luck, and C. Liu, “Improved crop row detection with deep neural network for early-season maize stand count in uav imagery,” *Computers and Electronics in Agriculture*, vol. 178, p. 105766, 2020.
- [7] 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.
- [8] G. Ronchetti, A. Mayer, A. Facchi, B. Ortuani, and G. Sona, “Crop row detection through uav surveys to optimize on-farm irrigation management,” *Remote Sensing*, vol. 12, no. 12, 2020.
- [9] A. Mateen and Z. Qingsheng, “Weed detection in wheat crop using uav for precision agriculture,” *Pakistan Journal of Agricultural Sciences*, 2019.
- [10] S. Katari, T. K. Bhowmik, S. S. Nair, A. S, A. R. Nayak, and P. Pankajakshan, “Crop phenology stage forecasting and detection using ndvi time-series and lstm,” in *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, 2022, pp. 6264–6267.
- [11] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [12] Q. Zhang, Y. Liu, C. Gong, Y. Chen, and H. Yu, “Applications of deep learning for dense scenes analysis in agriculture: A review,” *Sensors*, vol. 20, no. 5, 2020.

- [13] J. A. Marchant and R. Brivot, “Real-time tracking of plant rows using a hough transform,” *Real-Time Imaging*, vol. 1, no. 5, pp. 363–371, 1995.
- [14] M. Montalvo, G. Pajares, J. Guerrero, J. Romeo, M. Guijarro, A. Ribeiro, J. Ruz, and J. Cruz, “Automatic detection of crop rows in maize fields with high weeds pressure,” *Expert Systems with Applications*, vol. 39, no. 15, pp. 11 889–11 897, 2012.
- [15] P. Chen, X. Ma, F. Wang, and J. Li, “A new method for crop row detection using unmanned aerial vehicle images,” *Remote Sensing*, vol. 13, no. 17, 2021.
- [16] Y. Xie, K. Chen, W. Li, Y. Zhang, and J. Mo, “An improved adaptive threshold ransac method for medium tillage crop rows detection,” in *2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP)*. IEEE, 2021, pp. 1282–1286.
- [17] A. Matessi and L. Lombardi, “Vanishing point detection in the hough transform space,” in *Euro-Par’99 Parallel Processing*, P. Amestoy, P. Berger, M. Daydé, D. Ruiz, I. Duff, V. Frayssé, and L. Giraud, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 987–994.
- [18] R. A. Arango Quiroz, F. Pereira Guidotti, and A. E. Bedoya, “A method for automatic identification of crop lines in drone images from a mango tree plantation using segmentation over ycrb color space and hough transform,” in *2019 XXII Symposium on Image, Signal Processing and Artificial Vision (STSIVA)*, 2019, pp. 1–5.

## Chapter 7

# Appendix - Hyperparameters

Parameters	Values	Description
N	5	Default number of crop row to be detected
K	5	Default number of frame before executing the complete Hough Transform Process
r	0.1	Ratio vegetative pixels/non vegetation pixels of the image to be considered "bushy"
$\sigma$	0.3	User defined parameter based on the accuracy needed for the CRDA
bw	0.35	Fraction of the bandwidth used for creating the mask over the width of the image
$\alpha$	0.2	Minimum angle for crop row to not be considered as the horizon [rad]
$\beta$	0.1	Difference minimum between angles of two different crop row [rad]
tol	6	Color clustering tolerance for different color to be clustered together
lim	6	Maximum number of color clustered
k	4	Initial tolerance in vegetation segmentation - pixels within a color difference in norm of k to the vegetation color will be considered vegetation pixels
$it_{max}$	5	If $it_{max}$ outlier line were detected in a row, the algorithm start detecting $N - 1$ lines
var	1.25	Maximum ratio difference between the distance of an inlier line to the vanishing point and the standard deviation

Table 7.1: Hyperparameters used throughout the algorithm