ROW FOLLOWING AND ALTITUDE ESTIMATION WITH UAV IMAGES FOR AGRICULTURAL FIELDS

A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in Computer Engineering

by
Burak YÖRÜK

July 2023
İZMİR

We approve the thesis of **Burak YÖRÜK**

Examining Committee Members:

Prof. Dr. Devrim ÜNAY
Department of Department of Electrical and Electronics Engineering, İzmir Democracy University

---

Assoc. Prof. Dr. Mustafa ÖZUYSAL
Department of Department of Computer Engineering, Izmir Institute of Technology

---

Prof. Dr. Yalın BAŞTANLAR
Department of Department of Computer Engineering, Izmir Institute of Technology

18 July 2023

Prof. Dr. Yalın BAŞTANLAR
Supervisor, Department of Computer Engineering
Izmir Institute of Technology

Prof. Dr. Cüneyt F. BAZLAMAÇCI
Head of the Department of Computer Engineering

Prof. Dr. Mehtap EANES
Dean of the Graduate School of Engineering and Sciences

ACKNOWLEDGMENTS

I would like to thank everyone who helped and supported me, especially:

To Prof. Dr. Yalın Baştanlar, my supervisor during the thesis writing process, for giving me this opportunity and supporting me with his comments, ideas and calculations. I would also like to thank the venerable Assoc. Prof. Dr. Mustafa Özuysal and Prof. Dr. Devrim Ünay for accepting to be part of my thesis presentation jury and for taking the time.

I would like to express my sincere gratitude to my family for their unwavering support and encouragement throughout the thesis writing process. Their invaluable assistance in helping me balance my daily responsibilities enabled me to devote my full attention to this thesis.

Meshine Swarm Technologies company, for encouraging me to carry out studies that led me to the thesis by revealing the problem experienced in the field and allowing me to use the computers, drone and other technical equipments.

Zeki GÜL, Abdulkadir Şehmus ÖZGÜN and Yunus Emre AŞKIN for the patience they showed by standing side by side during the work that will cause us to identify the problems in the field as a startup and during the dissertation process.

ABSTRACT

ROW FOLLOWING AND ALTITUDE ESTIMATION WITH UAV IMAGES FOR AGRICULTURAL FIELDS

Traditional methods in agriculture involve the use of tractors; however, more than 10% of the planted fields suffer from harvest losses due to these vehicles. Moreover, tractors cannot enter all agricultural lands, thus reducing the available field for planting. After heavy rainfall, mud and other effects prevent these vehicles from accessing arable field, and processes such as crop spraying take significantly longer. In the past, aerial spraying methods using high altitude aircraft were attempted to overcome these problems; however, this method was banned in many areas due to the insufficient altitude and the harmful effects of chemical dispersion outside the fields. Nowadays, UAVs present a better alternative, and aerial spraying methods are regaining popularity. However, these vehicles can still cause errors when flying with a human operator, and their flight times are limited due to inadequate battery capacity. Therefore, the development of UAVs capable of autonomous flight reduces operator costs. However, during flight, liquid changes in the pesticide tanks hinder the UAV's ability to spray pesticides autonomously at a fixed altitude and prevent unwanted pesticide dispersion in undesirable rows. The thesis study provides following of plant rows on UAV images and making altitude estimation from camera images. In this way, it ensures that the UAVs in agricultural areas can stay at a fixed altitude for appropriate spraying and irrigation and prevents the spread of pesticides to unwanted rows.

ÖZET

TARIM ALANLARI İÇİN İHA GÖRÜNTÜLERİNDEN SIRA TAKİBİ VE İRTİFA KESTİRİMİ

Mevcut tarımsal alanlarda geleneksel yöntem traktör kullanımı olmakta, yer araçlarının girdiği tarlaların önemli bir kısmında ekili arazilerin %10 ve üstünde hasat kaybına yol açmaktadır. Ayrıca yine traktör girebilmesi amacıyla tarım arazilerinin tamamı bitki ekimi için kullanılamamaktadır. Yağmur sonrası çamur vb.etkilerin de oluşması, eğimli araziler bu araçların tarlaya girmesini önlemekte ve bitkilerin ilaçlanması gibi işlemler çok daha uzun vakitler alabilmektedir. Bu sorunları aşmak amacıyla geçmiş yıllarda denenen havadan uçak ile ilaçlama yöntemleri maalesef yeterli irtifanın sağlanamaması ve ilacın tarla dışına yayılması sonucu çevreye zararlı etkiler bırakması nedeniyle pek çok yerde yasaklanmıştır. Günümüzde İHA'lar daha iyi bir alternatif oluşturabilmekte ve havadan ilaçlama yöntemleri tekrar popülerlik kazanmaktadır. Ancak bu araçların da insan operatör eşliğinde uçmaları hatalar meydana getirebilmekte ve batarya kapasitelerinin yetersizliği nedeniyle uçuş süreleri kısalmaktadır. Dolayısıyla otonom özellikte uçuş yapabilen araçların geliştirilmesi operatör maliyetlerini düşürmektedir. Ancak bu araçların uçarken ilaç haznelerindeki sıvı değişimleri aracın sabit irtifada otonom şekilde ilaçlama yapılmasının önünde engel oluşturmaktadır. Tez çalışması, İHA görüntüleri üzerinden bitki sıralarının takibinin yapılması ve kamera görüntülerinden yükseklik tahmini yapılmasını sağlanmaaktadır. Bu şekilde tarım alanlarındaki İHA'ların uygun ilaçlama ve sulama çalışmaları için sabit irtifada kalabilmelerini sağlamakta ve istenmeyen sıralara ilaç yayılmasını engellemektedir.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

Table **Page**

LIST OF ABBREVIATIONS

CNN . . . . . Convolutional Neural Network
FOV . . . . . Field of View
GPS . . . . . Global Positioning System
GPU . . . . . Graphics Processing Unit
HD . . . . . High Definition
HSV . . . . . Hue, Saturation, Value
IMU . . . . . Inertial Measurement Unit
LIDAR . . . . . Laser Imaging Detection and Ranging
RGB . . . . . Red Green Blue
SLAM . . . . . Simultaneous Localization and Mapping
UAV . . . . . Unmanned Air Vehicle
VPS . . . . . Visual Positioning System

CHAPTER 1

INTRODUCTION

Although Turkey has the 18th most crowded population, the 14th largest agricultural economy,[1] and the 15th largest agricultural land area in the world, it struggles to cope with decreasing agricultural lands. The agricultural sector, which struggles to provide enough food for the increasing population (and more than 40 million tourists annually), is located in a geography that is extremely vulnerable to the adverse effects of climate change and geopolitical tensions. Therefore, digitization in agriculture and precision and restorative agricultural methods are crucial.

UAVs have become one of the most effective technologies developed for spraying and remote monitoring purposes in recent years. In China, UAVs are used to monitor, protect against pests and harmful insects, apply fertilizers and agricultural chemicals on a 20 million-hectare area of cotton cultivation. They also provide information about irrigation and harvest timing.[2] UAVs are generally used and becoming widespread in digital farming applications such as production, distribution, and waste management. Research continues on the use of nano UAVs throughout the entire supply chain except for processing and packaging.[3]

However, different problems have emerged with the usage of UAVs depending on the characteristics of the agricultural land. The terrain conditions for field crops, garden plants, trees, forests, vineyards, and other plants that grow in different environments vary. Although UAVs have made progress in pesticide spraying for crops grown in field lands, they have fallen short in tree plants and vineyards.

The subject of this thesis, altitude estimation and row following for agricultural UAVs, was determined by examining the tests conducted in 2019 under the Meshine Swarm Technology company in Manisa Vineyards Fig.1.1. Existing visual positioning sensors, pressure altimeters and GPS systems were insufficient in capturing fixed low-altitude in vineyards, and low-precision GPS systems caused location shifting, leading to pesticide spreading to other rows. Therefore, these problems aim to be solved to achieve precision in uneven terrains.

Figure 1.1. Agricultural UAV used in our study

Commonly used techniques for altitude measurement today are altimeter (laser or pressure), IMU, radar and range meter sensors GPS, or computer vision.[4] However, with standard GPS, vertical accuracy can vary between 25-50 meters. Also, transmission interruptions may occur in an urban environment. IMUs provide indications of velocity, acceleration, attitude, and orientation, but they are subject to drift and error accumulation. Altimeters are widely used, but they rely on pressure variations and have an accuracy error of 6-7%. Laser altimeters are more sensitive, but reflections can affect the measurement. Radar sensors provide altitude and mapping simultaneously, but they are active systems, energy consumption is high.[5] In addition, there are solutions with stereoscopic approaches from the camera to find the altitude.[5] and stereo vision (disparity estimation).[6] Although it is a useful approach in sloping lands, it cannot be said that it is a directly optimized approach for agricultural application.

In recent years, ultrasonic and pressure sensor-based studies have been put forward with sensor fusion applications, to solve previous signaling problems.[7] However, the problems experienced in practice have shown that these studies are insufficient in agricultural applications at low altitudes.

As a result, although there are approaches such as ultrasonic and laser for high precision, it is not possible to use them in data entry due to variable heights in their practical application in the vineyard area. Otherwise, fluctuating movements are seen in the UAV flight and unsuccessfully affect the operation.This brings us back to image-to-height detection algorithms that can operate at low power.

The aim of the study is to develop a fleet of autonomous vehicles that can fly and follow rows, thereby reducing operator costs. The current methods employed in agricultural areas rely heavily on the use of tractors, which can lead to significant harvest

loss in some fields where ground vehicles cannot enter. Although aerial spraying methods have gained popularity in recent years, the use of UAVs with human operators can lead to errors.

This thesis study seeks to enable UAVs to both follow the rows on their images and estimate the altitude by detecting the distance from the camera images. The thesis material and methods include sensor usage, running deep learning models over embedded boards that can work in the UAV, robot simulation tests, and real-time testing with edge computing methods on the drone if possible. Furthermore, the study aims to explore the potential of integrating additional sensors, such as Lidar, to improve the accuracy of the altitude estimates.

Through this study, we hope to demonstrate the feasibility and effectiveness of autonomous UAVs for agricultural applications, with the potential to significantly reduce costs and increase efficiency in the industry. To achieve the objectives of the study, various approaches will be developed and implemented. The first method will involve the use of stereo cameras and depth estimation sensors such as Lidar to estimate the altitude of the UAV relative to the plants. This approach will leverage deep learning models to analyze the sensor data and provide accurate altitude estimates.

Running deep learning models on embedded boards for real-time analysis of image data and conducting robot simulation tests in Linux environments are important approaches for improving the accuracy of altitude estimates and row following in UAVs for agricultural applications. However, technical limitations or budget constraints may prevent their implementation.

Real-time testing with edge computing methods on the drone will be conducted if possible. This will involve analyzing the image data on the UAV itself, reducing the need for data transfer and improving the accuracy and speed of the analysis. The thesis study also involves the development of custom datasets for training and testing the deep learning models. These datasets will be collected from vineyards and other agricultural areas, using various sensors and cameras to capture images and altitude data from different perspectives.

Overall, this study represents an exciting opportunity to explore the potential of autonomous UAVs for agricultural applications. A solution is aimed to be developed that can significantly improve efficiency in the agricultural industry. By leveraging cutting-edge

technologies such as deep learning and computer vision, the goal is to develop vehicles that can fly autonomously and follow rows, reducing operator costs.

This study is highly relevant in today's context, as the world faces increasing pressure to feed a growing population while also addressing sustainability concerns. The use of autonomous UAVs in agriculture has the potential to improve crop yields, reduce expenses, and minimize the environmental impact of traditional farming methods. Moreover, the study aligns with the United Nations Sustainable Development Goals, particularly goal 2, which aims to end hunger, achieve food security and improved nutrition, and promote sustainable agriculture.

This study has commercial potential in the agricultural industry. A solution that reduces costs and improves efficiency can attract investors and companies looking to innovate. However, the team, including the thesis owner, must address several challenges. These include developing reliable deep learning models, integrating Lidar sensors, optimizing row following and creating a testing framework. Regulatory requirements must also be met for UAV safety during operation.

This thesis study focuses on comparing and utilizing various techniques for row following and altitude estimation using images obtained from unmanned aerial vehicles in agricultural fields. The potential benefits of this study include precise positioning for operations, filling gaps in existing altitude and row following studies, and serving as a backup system. This study aims to develop a solution that uses advanced technologies to increase efficiency, reduce operator expenses, and promote sustainability by preventing unwanted pesticide applications in vineyards. We are excited to share the results of our research and contribute to the advancement of autonomous unmanned aerial vehicles in the agricultural industry.

1.1. Contributions of this thesis

1 - Altitude Estimation using Camera Geometry: The camera geometry and calibration matrix were utilized to determine the distance to objects in the captured images and subsequently calculate the altitude of camera. By employing the camera calibration matrix, which contains the optical properties of the camera, along with known real-world distances, accurate measurements of object distance and altitudes were achieved. This approach provided a reliable method for estimating altitudes based on camera calibration.

2 - Altitude Estimation using a Trained Deep Learning Model: As an alternative height measurement technique, a pre-trained deep learning model for depth estimation was employed. The model utilized the captured images to predict depth values, which were then used to calculate the corresponding altitudes. This approach offered a different perspective on altitude estimation by leveraging the capabilities of deep learning and computer vision techniques.

3 - Morphological Operations for Plant Row Detection: In addition the morphological operations performed during the analysis process had the additional benefit of detecting plant rows. This information was utilized to develop a row following algorithm, which facilitated the tracking of plant rows in the images. The integration of morphological operations and the row following algorithm added value to the thesis by providing a comprehensive solution for plant row following and altitude estimation.

Overall, this thesis made contributions in the field of altitude measurement using camera calibration, depth estimation with deep learning in vineyards and plant row detection. The combination of these techniques enhanced the understanding and analysis of altitudes in various applications, particularly in the context of agricultural practices in vineyard fields.

1.2. Organization of this thesis

The thesis begins by providing an overview of the current state and background of altitude estimation studies in fields around the world. The proposed method, which is an alternative to the current methods and its development stages, is then presented. The visual sets collected during the development stage are examined, and the row following results obtained through image processing methods are shared. The necessary calculations for altitude estimation from images are also presented, along with the use of deep models in the study. Finally, in the experimental section, the authors observe the results and outputs of their experiments. The thesis concludes by explaining the real-life application of the proposed solution and outlining potential areas for improvement.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

Although the use of computer vision in the agricultural field has become widespread for the past 10 years, especially the studies given as a review of the literature mainly include methods for disease detection, grain quality and phenotype characteristics of the plant.[8]

There are generally four stages in the studies. These are image acquisition, segmentation, extraction of characteristics and comparison with standards.[9] However, these are steps of methods focused entirely on measurements on plants. Therefore, it is necessary to look at the applications of computer vision in agricultural areas for robotic navigation. It is seen that computer-vision based autonomous navigation studies are applied in various fields in the agricultural field. For example, it is frequently encountered in steering support systems for autonomous land vehicles such as tractors to find their way. However, when we look at navigation solutions for aircraft, one of the remarkable studies has been the support of an autonomous drone from the air to provide the navigation of an airbot on the ground for paddy fields.[10] Similarly, VPS used because sufficient sensitivity cannot be obtained from the existing GPS and IMU sensors. However, as a further study, they also stated that it can be used to avoid the herbicide spray drift problem. In another study, flight plan change and navigation guidance were made according to the detection system in the image for precision agriculture application.[11] As can be seen, Vision-based navigation methods are also very diverse, but they are basically designed for guidance for precision agriculture solutions, and it is necessary to narrow the subject in particular for altitude estimation.

Regarding the datasets, it can be challenging to find appropriate data for row following and height estimation in agricultural land. One study addressed this issue by utilizing the Crop Row Benchmark Dataset, which is a collection of images of field rows of various crops, including maize, celery, potato, onion, sunflower, and soybean. The images, each with a 320x240 pixel resolution, were captured in Croatia during the spring of 2014 using a Panasonic LUMIX DMC-F2 digital camera. This dataset is commonly used for analyzing field rows using image processing techniques, as reported in.[12]

2.1. Altitude Estimation of Camera

Altitude estimation of a camera refers to the process of determining the vertical position of a camera with respect to a reference plane. This is an important task in computer vision and robotics, as it is often necessary to know the camera's height above ground to perform tasks such as autonomous navigation, object recognition, and scene understanding.

Various sensors can be used to determine altitude, including IMU, GPS, altimeter, and other approaches like monocular vision, stereo vision, ultrasound, lidar, and radar range meter sensors. However, raw GPS has poor accuracy and is sensitive to transmission interruptions in urban environments. IMUs are subject to drift and error accumulation, while altimeters have an accuracy error and rely on pressure variations.

Radar can estimate altitude and velocity with greater accuracy, but it requires active sensors that consume power. Lasers are highly accurate, but they have specific requirements for reflecting surfaces. In this case, cameras with rates up to 200Hz, low energy consumption, and compact structures come to the rescue. However, most of the currently used image-based applications are based on optical flow sensors.[13]

At the latest stage, a real-time hybrid stereo vision-based altitude estimation study was conducted for the complete control of UAV.[14] However, row tracking is not included in this study and it has not been used for low altitude stabilization in agricultural areas.

There are many different methods available for achieving this using either monocular or stereo cameras. One method involves using an additional camera to optimize and accurately estimate the scale of the monocular visual odometry.[15] In monocular vision, the camera's height can be estimated by analyzing the perspective distortion in the image caused by the camera's position relative to the ground plane. This can be done by detecting features such as angle between horizon lines in the image, and using geometric methods to calculate the camera's height.

Another approach is to use vanishing points in the image to estimate the camera's height. A vanishing point is a point in the image where parallel lines appear to converge. In a typical outdoor scene, the vanishing point for the ground plane is at infinity, meaning that parallel lines on the ground appear to converge at a point that is infinitely far away. By detecting the vanishing point for the ground plane in the image, the camera's height can be estimated by using the known size of an object in the scene and the angle between the camera's optical axis and the ground plane.

While yet another approach is to estimate the absolute scale of monocular structure from motion for a multi-modal stereo camera.[16] Stereo cameras can also be used to estimate the pitch and roll angles as well as the height of the camera positions.[17] The camera's height can be estimated by triangulating the position of corresponding image points in two or more images taken from different viewpoints. By knowing the baseline distance between the cameras and the pixel coordinates of corresponding image points, the camera's height can be calculated using trigonometry.

Other methods include monocular depth estimation, which predicts the distance of each pixel relative to the camera from a 2D image,[18] and depth map estimation using stereo cameras, which is simpler than computing disparity and faster than other monocular methods.[19] Since deep learning methods became prevalent in many computer vision tasks,[20] unsurprisingly deep neural network based mono and stereo depth estimation methods were also proposed.[21]

Stereo vision cameras can also be used to measure the height of various field crops,[22] and analytical expressions can be derived to determine the absolute scale using a monocular camera.[23] Compensation for camera tilt effect is necessary for stereo-vision-based crop height measurement.[24] Monocular and stereo cameras are also used in SLAM systems.[25]

Other sensors such as lidar or sonar can also be used to estimate camera height by measuring the distance between the camera and the ground plane. This can be done by emitting a laser or sound wave and measuring the time it takes for the signal to bounce back, which can then be used to calculate the distance.

Overall, altitude estimation is a significant task in computer vision and robotics, and there are various approaches that can be used depending on the available sensors and the specific application requirements.

2.2. Background of Proposed Methods of this Thesis

In this study, monocular RGB images were collected over the vineyard and progressed over it. Height estimation with monocular RGB vision involves using a single camera to estimate the vertical position of the camera with respect to a reference plane. This can be accomplished by analyzing the perspective distortion in the image caused by the camera's position relative to the ground plane, but in the thesis it was achieved by analyzing

the perspective, pitch angle in the image and actually known metric measurements.

Other methods for estimating camera height with monocular RGB vision include using machine learning algorithms to detect features in the image, such as road markings or buildings, and using geometric methods to calculate the camera's height based on the size and position of those features.

Altitude estimation using a camera calibration matrix involves using a mathematical model to relate the position of objects in a 3D world to their corresponding positions in a 2D image captured by the camera. This mathematical model is known as the camera calibration matrix, which describes the intrinsic parameters of the camera such as its focal length, principal point, and image distortion coefficients.

The camera calibration matrix can also be used to correct image distortion, which can affect the accuracy of the height estimation. This is typically done using a process called camera calibration, which involves capturing images of a calibration target with known geometry, and using the resulting image correspondences to estimate the camera calibration matrix.

2.2.1. Image Processing Techniques for Color Representation and Analysis

In digital images, there are various color representations, or color spaces, that can be used. These representation types can be chosen not only for efficient storage purposes but also for enabling different types of analyses and manipulations. Before being considered for inclusion in the study, it is important to define the image processing techniques used. The following techniques are presented in the order in which they were employed:

Color Spaces

- RGB

  The RGB color model is an additive system that utilizes red, green, and blue light to generate a wide range of colors.[26] In this color space, combinations of these primary colors are used to represent colors, with each color component ranging from 0 to 255. By adjusting the intensity of each color component, it is possible to create millions of unique colors. This color model is widely used in digital imaging, computer graphics, and display technologies due to its ability to closely mimic the

way that displays emit light to produce colors.[27]

- HSV

  HSV is a color space commonly used in image processing and computer vision. Hue, saturation, and value are the three components that make up the HSV color space. Hue represents the dominant wavelength of a color and is measured in degrees from 0 to 360. Saturation represents the purity of a color and is measured as a percentage from 0 to 100, while value represents the brightness of a color and is measured as a percentage from 0 to 100. By utilizing the HSV color space, it becomes easier to manipulate and analyze color information in an image, such as detecting specific colors or adjusting the brightness and saturation of an image.[28]

- Grayscale

  Grayscale is a color space commonly used in digital image processing. In this color space, each pixel is represented by a single intensity value, ranging from 0 to 255, with 0 being black and 255 being white. Grayscale images are essentially black and white images, but with varying shades of gray in between. By converting a color image to grayscale, it becomes possible to simplify the image and reduce the amount of data required to represent it.[29] This can be useful in applications such as image compression, edge detection, and image enhancement.

Image Filtering

- Blurring

  Convolution is a mathematical operation that can be performed on an image using a kernel that corresponds to its elements. This operation is performed on each pixel in the source image using a 3x3 kernel. The resulting filtered image appears blurred due to the averaging effect of the convolution operation, which tends to smooth or blur the image.[30]

Image Thresholding

Thresholding is a fundamental technique in digital image processing used to segment images. Thresholding involves setting a threshold value and then assigning pixel values above or below this threshold to different classes.[31]

- Binary

  A binary image is a type of image that consists of only two colors, typically black and white, and is often referred to as a bi-level or two-level image. Each pixel in this format is stored as a single bit, represented as either a 0 or 1. The terms black-and-white, B&W, monochrome, or monochromatic are frequently used in reference to this concept, although they may also describe any image with only one sample per pixel, such as grayscale images. Binary images are commonly used in image segmentation, skeletons, and interpretation operations.[32]

- Inverse Binary

  Inverse-binary is a technique used to reverse the dynamic of an image containing two different grayscale populations. This process essentially inverts the pixel intensity values, such that the lighter pixels become darker and the darker pixels become lighter.[33] This technique has been used in various image processing applications, such as segmentation and object recognition. By reversing the dynamic of the image, it becomes possible to highlight specific features or regions of interest that may not have been visible in the original image Fig.2.1.

Figure 2.1. Binary and Inverse-binary image comparison

- Otsu Thresholding

  Otsu's method is a popular technique used in computer vision and image processing for automatic image thresholding.[34] The method is named after Nobuyuki Otsu and involves finding a single intensity threshold that divides pixels into two classes: foreground and background. This threshold is determined by minimizing the intra-class intensity variance or maximizing the inter-class variance.[35] Otsu's method is easy to implement and is effective in various applications, including object recognition and image segmentation.

Feature detection

- Blob Detection

  A blob refers to a group of connected pixels in an image that share a common property, typically grayscale value, and can be identified and marked based on their differing properties from surrounding regions, such as brightness or color. In essence, a blob is a region of an image where certain properties are constant or nearly constant, such that all points within the region can be considered similar in some respect.[36]

- Contour Detection

  Inverse-binary is a technique used to reverse the dynamic of an image containing two different grayscale populations. This process essentially inverts the pixel intensity values, such that the lighter pixels become darker and the darker pixels become lighter.[33] This technique has been used in various image processing applications, such as segmentation and object recognition. By reversing the dynamic of the image, it becomes possible to highlight specific features or regions of interest that may not have been visible in the original image.[37]

- Hough Transform

  The Hough transform is a prevalent image processing technique that can extract features of a specified shape from an image, such as lines, circles, and ellipses.[38] This transform requires the desired features to be defined in a parametric form, and it can tolerate gaps in feature boundary descriptions while being relatively immune to image noise. Due to these advantages, the Hough transform technique is widely used for detecting feature boundaries in computer vision, image analysis, and digital image processing[39]

  A line segment can be analytically described in various ways. However, a parametric or normal representation provides a convenient equation for describing a set of lines. In this representation, the length of a normal from the origin to this line is denoted as r, and the orientation of r with respect to the X-axis is denoted as θ. For any point (x, y) on this line, r and θ remain constant. (Refer to Fig.2.2. for a visual illustration and Eq.2.1)

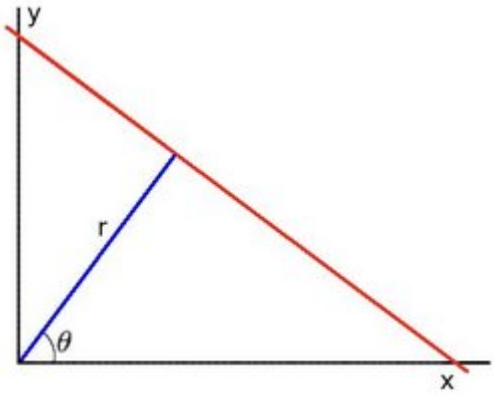

Figure 2.2. Hough transform - parametric description of a straight line

$$r = x\cos\theta + y\sin\theta \tag{2.1}$$

Morphological Operations

In the realm of mathematical morphology, the term "morphology" often refers to the study of form and structure in animals and plants. As a tool for extracting image components that aid in the representation and description of region shape, such as boundaries, skeletons, and the convex hull, mathematical morphology is of particular interest. Moreover, morphological techniques for pre- or post-processing, such as morphological filtering, thinning, and pruning, are also of interest.[40]

- Erosion

  Erosion is a mathematical morphology operation that reduces the size of shapes in an input image by replacing each pixel with the minimum value of the pixels in its neighborhood defined by a structuring element B.[40]

$$\text{Erosion}(A, B) = A \ominus B$$

- Dilation

  Dilation is a mathematical morphology operation that enlarges the shapes in the input image by replacing each pixel with the maximum value of the pixels in its neighborhood defined by a structuring element B.[40]

$$\text{Dilation}(A, B) = A \oplus B$$

- Opening and Closing

Opening and closing operations in mathematical morphology have distinct effects on image components. Opening smooths the contour of an object, breaks narrow isthmuses, and eliminates thin protrusions. It eliminates the thin protrusions of the obtained image. On the other hand, closing tends to smooth sections of contours, fuses narrow breaks and long thin gulfs, eliminates small holes, and fills gaps in the contour. It eliminates the small holes from the obtained image.[40]
The opening of set A by structuring element B, denoted $A \circ B$, is defined as

$$\text{Opening}(A, B) = (A \ominus B) \oplus B$$

Thus, the opening A by B is the erosion of A by B, followed by a dilation of the result by B (Fig.2.3.). Similarly, the closing of set A by structuring element B, denoted $A \bullet B$. is defined as

$$\text{Closing}(A, B) = (A \oplus B) \ominus B$$

which, in words, says that the closing of A by B is simply the dilation of A by B. followed by the erosion of the result by B.

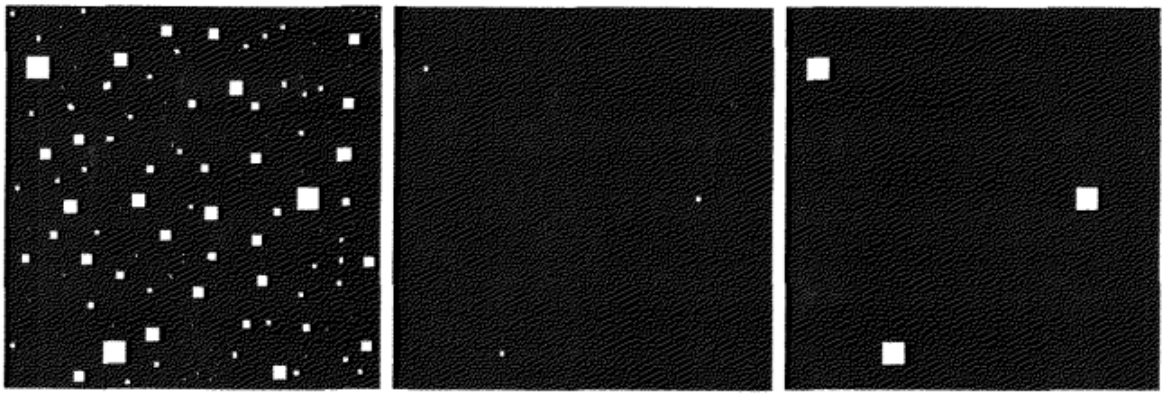

Figure 2.3. Opening operation from left to right.[40]

CHAPTER 3

METHODOLOGY

The basic studies and methods used are mentioned here. It is divided into sub-sections and starts from the dataset creation phase. It continues with image processing and row detection. Altitude estimation is deduced from the camera calibration matrix and depth map. The chapter is concluded with a discussion on UAV navigation.

3.1. Generating the Dataset

In order to determine the field rows and conduct altitude estimation as part of our study, we generated a dataset using unmanned aircraft flight in the Manisa Turkey region. However, it is unfortunate to note that we were unable to obtain a fully overlapping dataset for the vineyard area.

The initial images captured after harvest proved to be unusable, necessitating re-shoots before the harvest. The images depict the state of the field leaves after greening. As this study focuses on viticulture, the image data was personally collected via UAV flights at different times of the year in the Manisa-Alaşehir region.

In total, the image set contains 162 images and 15 videos. It offers different views for post-harvest and actual harvest time for the vineyard land. Our final dataset that was used in our experiments contains only harvest period with 5 videos and a separate set of 85 photographs. Video resolutions are 1344x720p HD quality, photos are 1240x720p HD quality. Images were taken from a height of 4m at a speed of 1 m/s at 13.8" and 12.4" camera pitch angle.

DJI Phantom 4 Pro model drone was used for shooting, and the sensor and lens information are as follows.

Table 3.1. Camera specifications

Sensor	1” CMOS Effective pixels - 20M
Lens	FOV 84° 8.8 mm/24 mm (35 mm format equivalent)
Aperture	f/2.8 - f/11
Focus Range	1 m - ∞

(a) 3m Flight

(b) 3m Flight-2

(c) 4m Flight-1

(d) 4m Flight-2

(e) 5m Flight-1

(f) 5m Flight-2

Figure 3.1. Harvest period of vineyard

Detailed information about camera shots, such as frame numbers, shooting times, etc. is given in Tab.4.1. After the image acquisition process was completed, software tools were used for data processing and analysis.

3.2. Row Detection

Inspired by,[41] we developed a crop row detection and row following algorithm. The row following study involves converting RGB images into grayscale and enhancing the green values. The study then takes the grayscale images as inputs and applies morphological operations including binarization, erosion, dilation and gap filling. Later, line extraction with Hough Transfrom is employed. The process involves iterating through

Hough thresholds until the optimal value is found for the desired number of crop rows, and includes filtering. After this point, the binary image is further processed to be used in row following step as detailed in Section 3.3. (Fig.3.2.).

Sample input images are displayed in Fig.3.3.

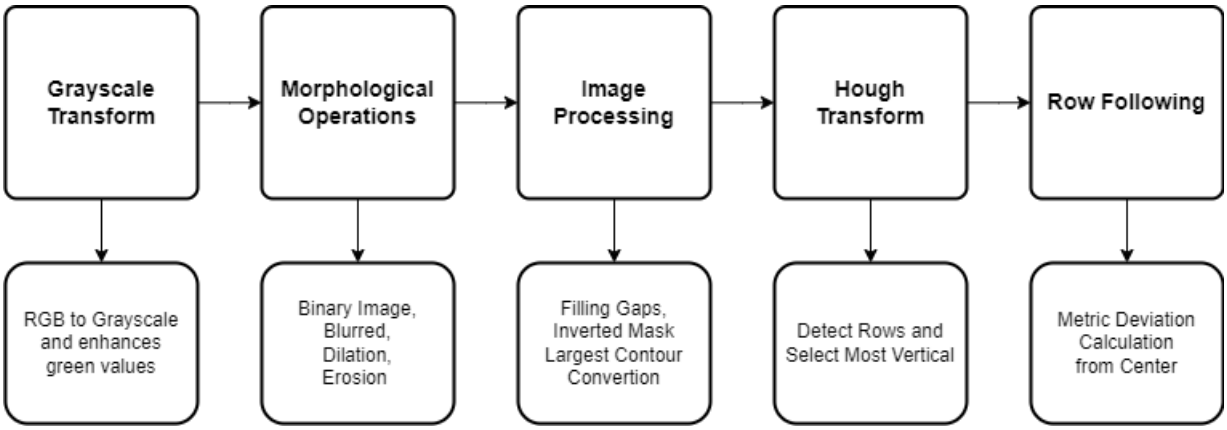

Figure 3.2. Overview of row detection algorithm

Figure 3.3. Vineyard reduced images

In the initial stage, the existing images were reduced to 320x180 dimensions. A grayscale transform function was then applied to convert an RGB image to grayscale while enhancing the green values. By leveraging the HSV color space, a binary mask was created to isolate the green color range.

The function then extracted the green pixels and discarded the red and blue channels, resulting in a grayscale image with amplified green components (Fig.3.4.).

To eliminate noise, specific regions in the upper and lower halves of the grayscale image were selectively modified. Subsequent steps involved blurring, thresholding, dilation,

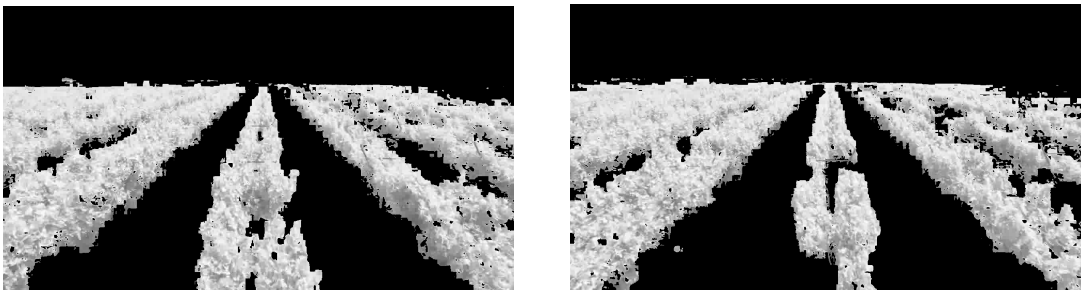

Figure 3.4. Grayscale Transform sub-outputs

and erosion to enhance blob-like structures and minimize disturbances. More consistent inferences were obtained with the post-binary blur method. By blurring the image, the row became clearer. Then, by applying dilation and erosion, usable outputs for the skeletonization stage were obtained. In addition, the image output was cleaned by cropping from the middle height where the row line is evident and the edges that cause distortions (Fig.3.5.).

After converting the images into binary and clarifying the rows with image processing, sending them directly to the Hough transform increased the errors in finding the line. Therefore, filling the holes in the image and processing an extra layer is important before Hough transform. The output is shown in (Fig.3.6.) .

The skeletonize function takes a grayscale image as input and applies erosion to progressively thin down the objects in the image. The resulting eroded image is returned as the output. Additionally, a Sobel edge detection operation is performed on the eroded image to produce the final binary skeleton image(Fig.3.7.).

The Hough transform with varying thresholds to extract lines from a set of crop points. It iteratively adjusts the threshold to find the optimal value for the desired number of crop rows. The function includes filtering steps to remove duplicate and faulty lines based on similarity thresholds. The extracted lines are visualized in two images, crop_lines and crop_lines_hough. If the desired number of rows is found, it is printed, and the function returns the filtered line visualization and the Hough transform line visualization (Fig.3.8.).

Figure 3.5. Binary morphological outputs

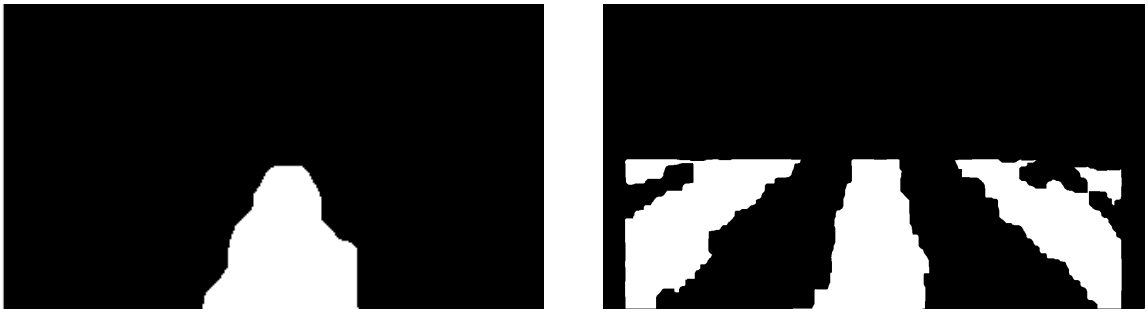

Figure 3.6. Filled holes outputs

Figure 3.7. Skeletonization processes

Figure 3.8. Hough Transform results

3.3. Row Following

After completing the Hough transform stage, the metric deviation from the center was calculated in order to keep the main crop row (that the UAV is following) in the center of the image.

The developed code implements an image processing algorithm capable of analyzing a series of images to detect vertical lines and calculate the horizontal distance between the *Selected Hough Line* and the *Vertical Center Line* . The algorithm employs the OpenCV library for computer vision operations and the NumPy library for numerical computations Fig3.9.).

An alignment-based approach has been demonstrated for row detection. Firstly, the origin point (x_0, y_0) is assumed to be at the exact center of the image. Then, lines are determined from the Hough transform result in the image, and the line with the narrowest angle with the vertical is selected. The coordinate (x_1, y_0) corresponding to the half of the image height is defined on the selected line. Finally, the distance in pixels between the selected line's center (x_1, y_0) and the center of the image (x_0, y_0) is calculated. It's illustrated in Fig.3.10.

Horizontal difference between two point coordinates produces the deviation value $(x_1 - x_0)$. Vertical lines that center on these points are constantly being aligned, and the pixel difference between them is converted into a metric form to determine the direction of the UAV.

In Fig. 3.11., the green line represents the *Vertical Center Line* of the image, while the red line shows the *Narrowest Hough Line* aligned as exact vertical. In each iteration, the average of the selected Hough lines is calculated for a repeated 5-point moving average. The average deviation line is updated at each step. If we take the y-axis as $y_0 = 0$, the horizontal difference (referred to as deviation) between the two points is only dependent on the x-axis and is as follows Eq.3.1.

$$x_A - x_0 = ((x_n - x_0) + (x_{n-1} - x_0) + (x_{n-2} - x_0) + (x_{n-3} - x_0) + (x_{n-4} - x_0))/5 \quad (3.1)$$

If $(x_{n+1} - x_0) < (x_A - x_0)$, the distance of (x_{n+1}) is selected for navigation, and a blue *Deviation Line* is drawn through that point. Conversely, if $(x_{n+1} - x_0) > (x_A - x_0)$, the average distance of (x_A) is selected as the Deviation, and a blue line is drawn through the

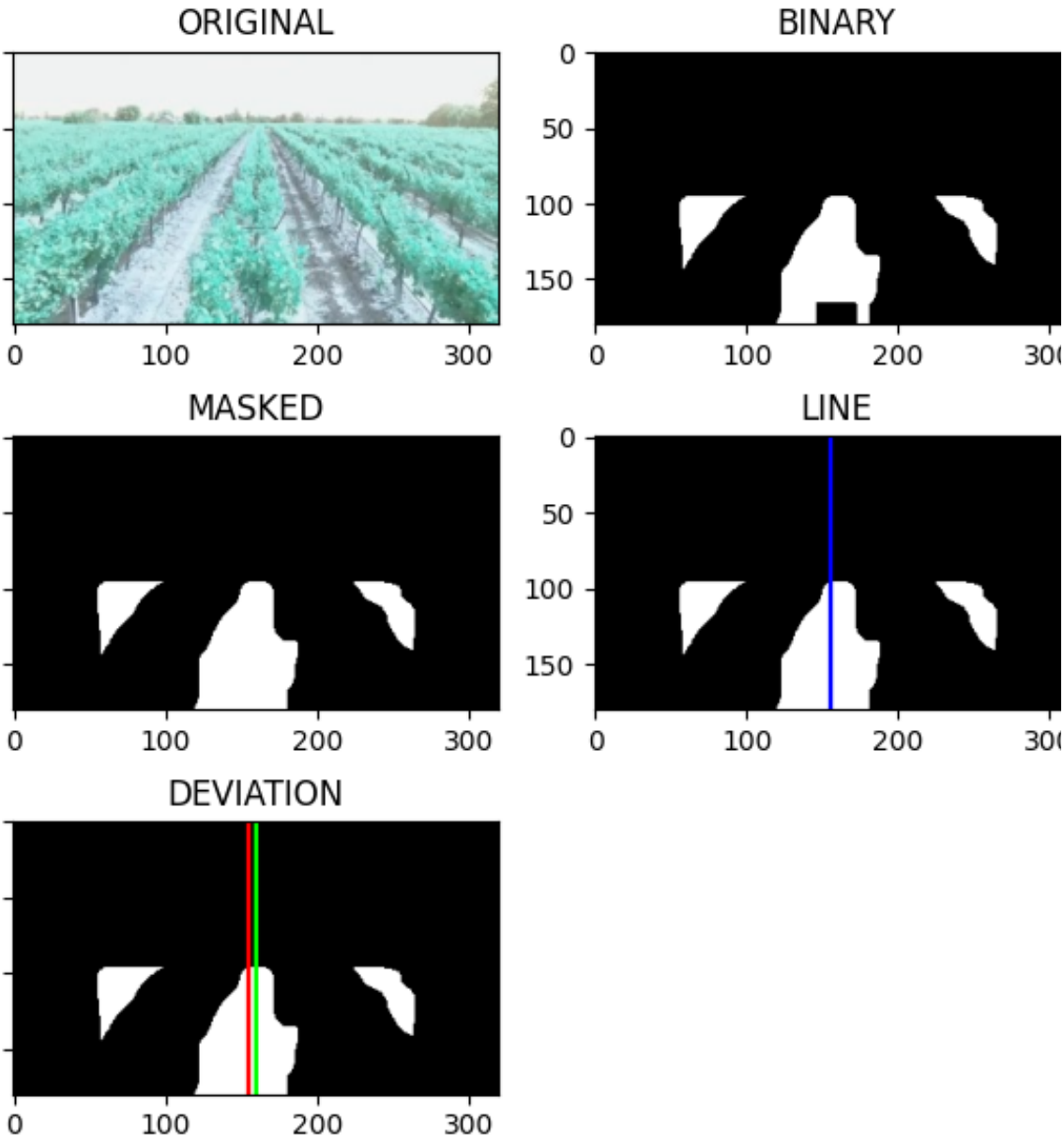

Figure 3.9. Row following phases step-by-step

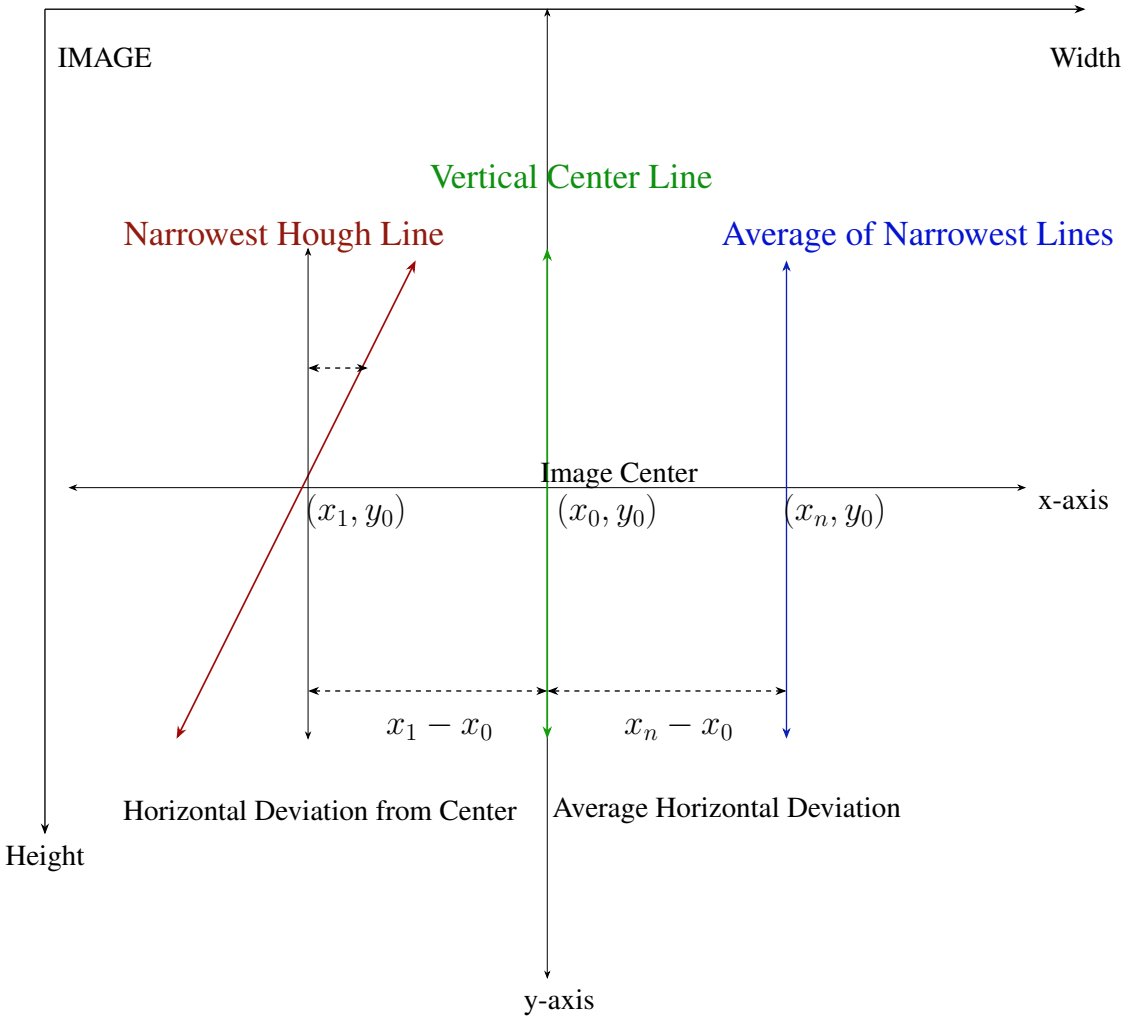

Figure 3.10. Row following explanation on the image plane

point at (x_A, y_0). In the second image of Fig.3.11., it can be observed that the red line is horizontally farther from the green center line than the blue line. Therefore, the average line will be selected for the deviation as horizontal distance data.

(a) Deviation Line(Blue) has lower gap than Average to the Center Line(Green)

(b) Selected Line(Red)'s horizontal gap higher than Deviation Line(Blue) to the Center Line(Green)

Figure 3.11. Detected rows and deviation from visual center

The pseudo code of our row following algorithm is as follows:

1. The code initializes libraries and defines blob width global metric parameters.
2. Hough Transform is applied to the images to detect lines.
3. The narrowest angled line is selected in the vertical direction.
4. Horizontal average of last 5 selected lines is computed and named as Average Line.
5. The horizontal pixel distance between the Narrowest line(in step 3) and Center line (vertical line passing through the image center) is computed and defined as "Deviation".
6. If the "Deviation" is lower than the deviation of the Average line, **then the output is metric and pixel equivalence of this deviation**.
7. If the "Deviation" is higher than the deviation of the Average line, **then the output is the deviation of the average line.**

Metric results are obtained by converting pixel distances to actual crop width distance. If we take half width of the row at the midpoint of the image, called as **half of the blob width**, as 40 pixels, then the metric distances are as follows.

- Deviation between Center line and Narrowest line:
  - Pixels: 5.5999755859375
  - Metric: 8.40 cm
- Deviation between Center line and Average line:
  - Pixels: 3.20001220703125
  - Metric: 4.80 cm

Although the algorithm has not been tested on the aircraft yet, its inputs are set to capture the incoming video stream. Half of the blob width value is ideally calling variable from automated pixel detection function. It returns the metric deviation statement required to steer the autonomous vehicle. Thanks to this algorithm, various applications such as row following or object tracking with computer vision can be realized in autonomous vehicles, especially in UAVs.

3.4. Altitude estimation via Camera Calibration

3.4.1. Calibration matrix

The calibration matrix, also known as the camera matrix or intrinsic matrix, plays a crucial role in camera calibration by encoding the intrinsic parameters of a camera. It is a 3x3 matrix denoted as K, encompassing focal length (f_x, f_y) representing scaling factors for converting world coordinates to image coordinates, principal point coordinates (c_x, c_y) indicating the optical center of the camera, aspect ratio and distortion parameters. Determined through a calibration process using known calibration patterns, the calibration matrix enables tasks like image undistortion, camera pose estimation, and 3D reconstruction, by accurately mapping between 3D world coordinates and 2D image coordinates.

The camera calibration matrix plays a crucial role in determining the camera's optical properties, such as focal length, which are fundamental for various image analysis tasks. It is provided by the camera manufacturer and encodes intrinsic parameters that enable accurate mapping between 3D world and 2D image coordinates.

By leveraging the calibration matrix and known image distances, researchers can estimate the altitude at which the camera captured the image. The calculation involves the focal length and angle of view to establish a relationship between the distance in the image plane and the actual distance in the real world. Accurate height estimation contributes to applications in computer vision, remote sensing, and photogrammetry.

It is represented as Eq.3.2 :

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{3.2}$$

Here, f_x and f_y represent the focal lengths of the camera. c_x and c_y represent the coordinates of the camera center in the image plane.[42]

3.4.2. Camera calibration

Camera calibration is the process of measuring and correcting the geometric features of a camera system. This process removes distortions in an image, scales the image properly, and helps to predicts the real world positions of objects in the image. Camera calibration is required for many 3D computer vision applications such as Structure-from-motion,[43] SLAM (simultaneous localization and mapping),[44] and visual localization and photogrammetry.[45]

In general, the parameters obtained from camera calibration include camera matrix, image distortion coefficients, and external rotation and translation parameters. With these parameters, it is possible to estimate the real-world coordinates of objects in the image, calculate the distances in the image, and measure the dimensions of objects in the real world.

To calibrate the camera, a Chess Board image was first used on the computer screen. This image had 2048x2048 pixel dimensions, in the form of a checkerboard, and had a distance of 38 mm between the frames on a 27' full screen. However, due to resolution incompatibility of the captured images and the inability to obtain different angles when shooting from the screen, it was decided to take photos again.

The settings for shooting the Vineyard images were adjusted to the same drone camera and the necessary controls were made. The calibration model had to be re-photographed with the collected image set conjugate features. For this, the DJI Phantom 4 Pro camera was used and the image features were set to 1280x720p. Then, a checkerboard printout was taken and the images were shot again on an appropriate background. The output images are shared below Fig.3.12..

Figure 3.12. Calibration pattern shoots

Revealing the calibration matrix:

Several applications have been tested for generating the calibration matrix, including the Camera Calibration toolbox using the Python-openCV repository[46] and the Camera Calibration Toolbox.[47] However, the first method, which uses the notebook camera, encountered a problem with camera access when working on a virtual machine due to the operating system. While the second method successfully ran and the photos were uploaded, the correct output could not be produced. After several attempts, the Matlab Computer Vision Toolbox Camera Calibrator tool was used.[48]

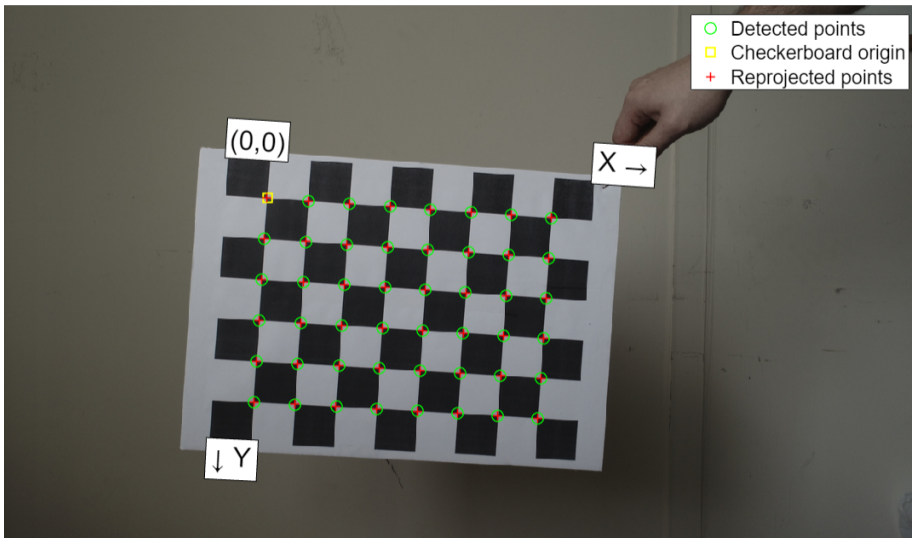

Figure 3.13. Detected points on the calibrated image

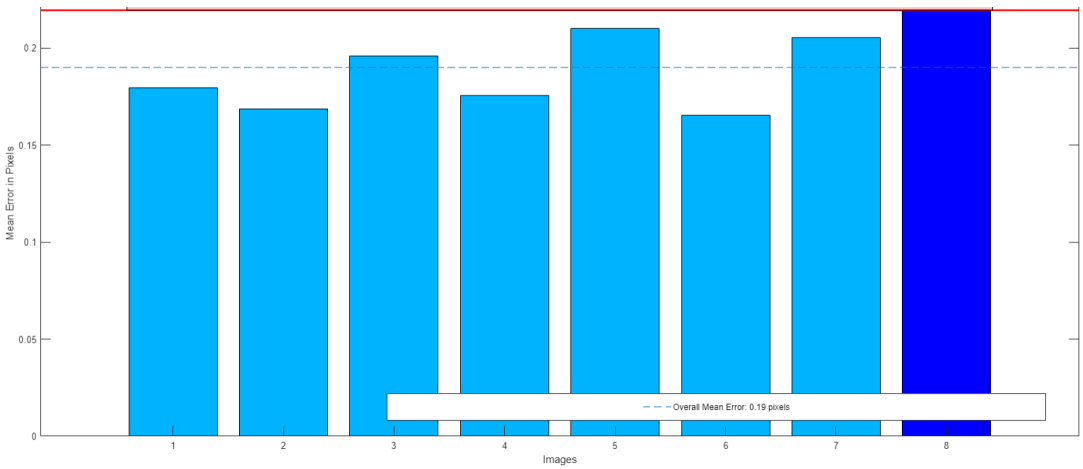

Figure 3.14. Camera calibration projection errors

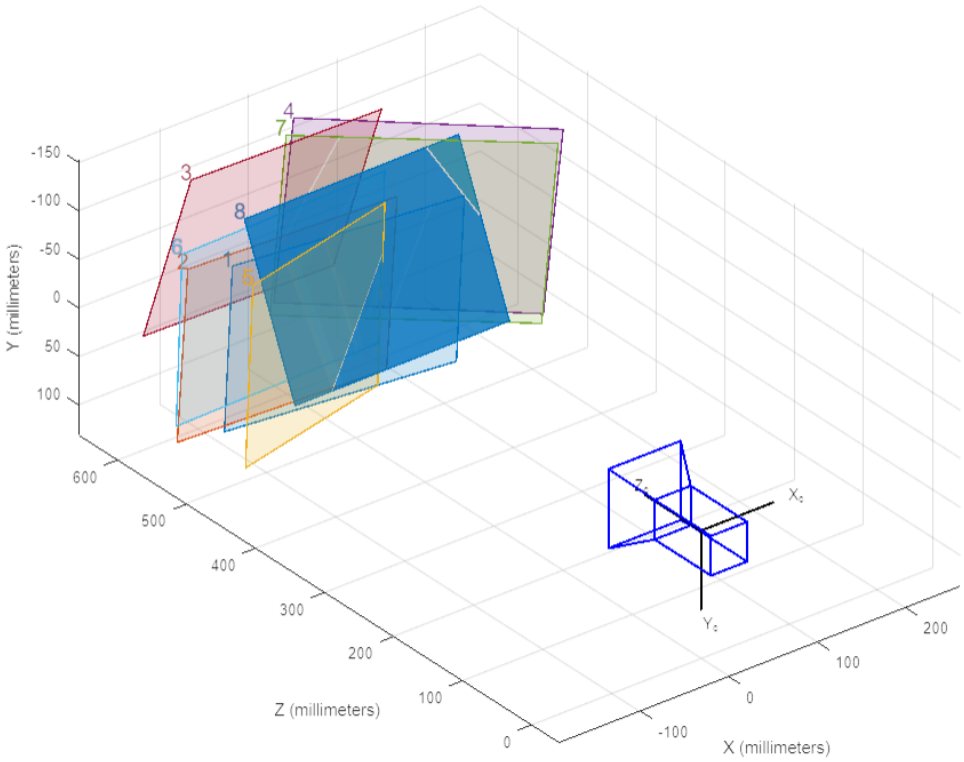

Figure 3.15. Calibration camera-centric pattern

Twenty-six photos were added initially, then the appropriate threshold was identified and optimized for eight photos. The Fig.3.13. presents the visual output of the MATLAB Calibration toolbox, which is accompanied by the mean error value in pixels, as illustrated in Fig.3.14.. As a result of the projections, the overall mean error value is idealized as 0.19. The camera-centric pattern, in which the shooting angles are calculated, can also be observed in Fig.3.15.. Calibration matrix was obtained at the optimum point as follows Eq.3.3.

$$K = \begin{pmatrix} 874.33 & 0 & 640.49 \\ 0 & 872.39 & 360.81 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.3}$$

The calibration matrix gives values f_x=874.33, f_y=872.39, c_x=640.49, and c_y=360.81. This approach has yielded the calibrated value of the focal length in pixels. However, to employ it for calculations, the metric value must be ascertained. To this end, the following formula was utilized. Upon conducting a thorough analysis of the data, the focal length in millimeters was determined to be 9.016 mm, which deviates from the stipulated 8.8 mm focal length in the camera specifications.

$$\frac{focal\_length\_mm}{sensor\_size\_mm} = \frac{focal\_length\_pixels}{image\_width\_pixels} \tag{3.4}$$

$$pixel\_size = \frac{sensor\_size\_mm}{image\_width\_pixels} = \frac{13.2}{1280} = 0.0103 \tag{3.5}$$

$$pixel\_size = 10.3\mu m$$

$$focal\_length\_mm = focal\_length\_pixels * pixel\_size \tag{3.6}$$

$$= 874.33 * 0.0103 = 9.016$$

3.4.3. Altitude estimation

In order to determine the distance between the center point of an image and the camera, it is necessary to take into account several important parameters. These include the pixel distance, image size, image center, and distance calculation, all of which are crucial factors in determining the distance of an object from a camera in an image. By knowing the actual distance between pixels in the image, it becomes possible to calculate the actual size of each pixel. If the size of the image is known, the pixel distance can then be used to calculate the actual size of the image. The center point of the image is calculated by dividing the width and height of the image by two. The angle made by this center point and half of the crop row in Horizontal FOV is shown as pixel distance pix and is the pixel value of the crop width. Actual crop half width is shown as dm. As a result of multiplying focal length f with diagonal length L is found. Finally, the distance between the image center and the camera can be calculated using an appropriate formula Eq.3.7 and Fig.3.16..

$$L = \frac{dm * f}{pix} \tag{3.7}$$

To calculate the vertical distance between the camera and the crop, Eq. 3.8 is used (G_p is the gimbal pitch angle) and the relation is shown in Fig.3.17.:

$$H = L * sin(G_p) \tag{3.8}$$

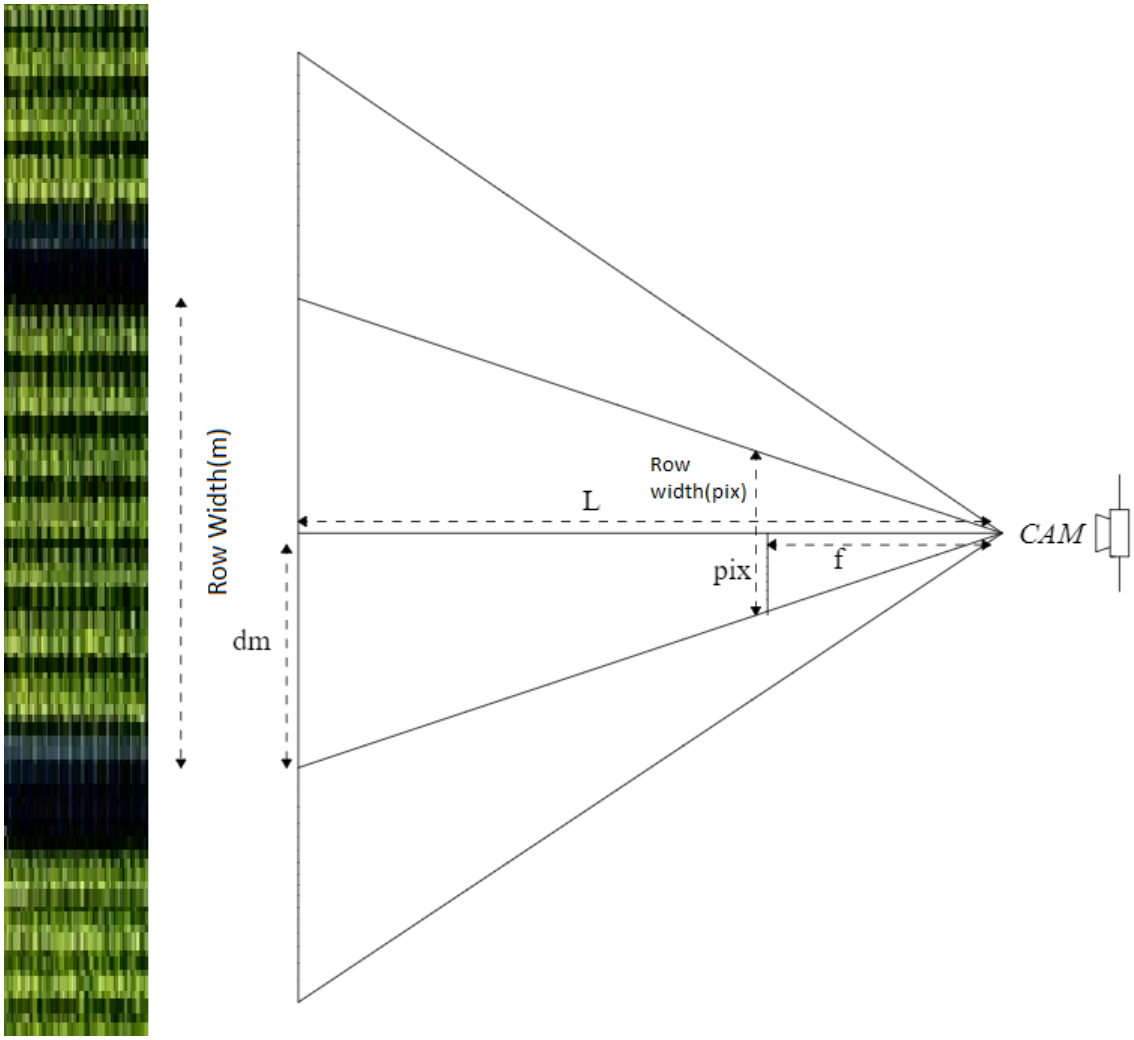

Figure 3.16. Bird's Eye View of the scene

Various methods have been tried for camera altitude calculation. Before these methods could be tested, the camera calibration matrix had to be revealed Eq.3.3. Apart

from that, the FOV angles were calculated according to the known camera parameters as follows.

- Diagonal FOV: 84 degrees,
- Vertical FOV: 53.13 degrees,
- Horizontal FOV: 73.74 degrees,

  If the sensor parameters are
- 1’ CMOS = 0.0254 m,
- Height = 8.8mm
- Diagonal =15.86 mm

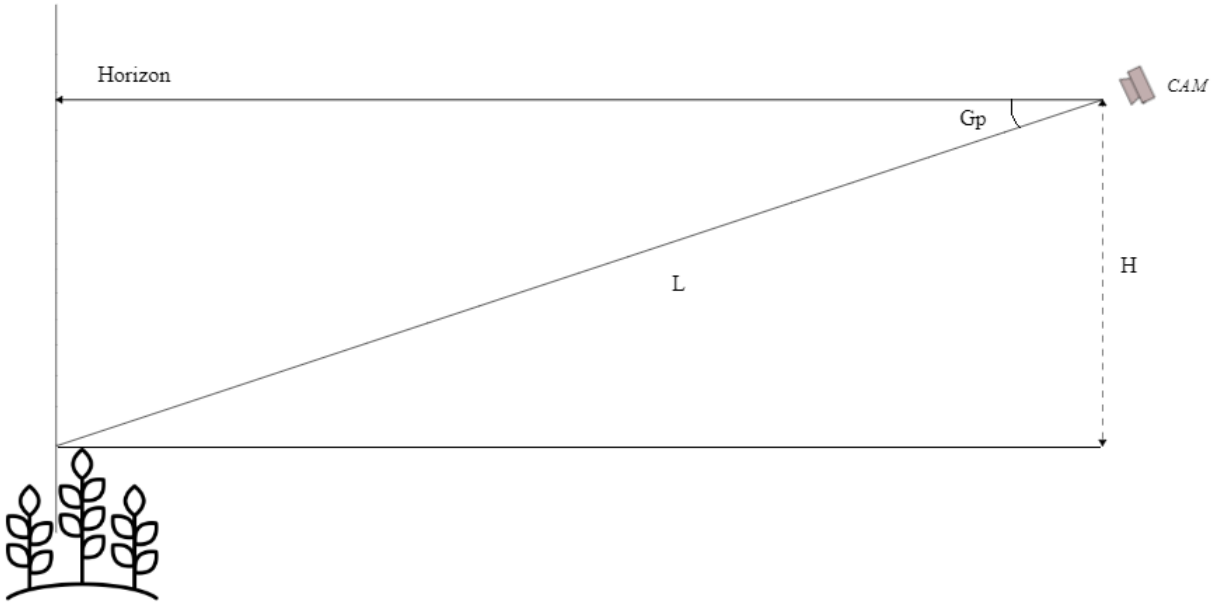

Figure 3.17. Camera altitude from side view

At the same time, it is necessary to know the actual distances in the images in order to calculate the vineyard images. The distances between the rows in the images are 3 meters and the vine plants are wrapped around the V-shaped bars on the stakes. The distances between these V-shaped irons correspond to 2.2 meters. When the vine leaves in the image are taken into account, it has been calculated that there is a spacing of about 2 meters between rows (Fig.3.18.).

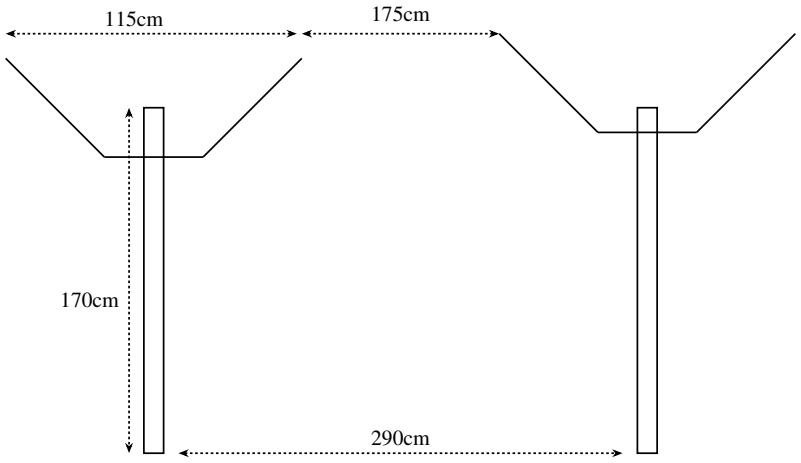

Figure 3.18. Vineyard post dimensions and row spacing

Although the flight heights vary from place to place, it varied between 4.1 and 4.9 meters according to the values obtained from the drone telemetry logs. In the study, an example will be shown between two images, one of which is far and the other is closer (Fig.3.3.).

Example Altitude Estimation from Camera parameters

In Fig.3.16. and Fig.3.17., the geometric steps of the height calculation were tried to be expressed. First of all, the row half length is 95 pixel manually calculated as shown in the Fig.3.19. and this half length actually corresponds to 60 cm. Using $f = 873.35$ value from the calibration matrix, Fig.3.17., the diagonal distance L from the similarity between the triangles in Eq.3.7 was found and this value corresponds to 5.5 meters.

According to the received telemetry log data, our $Gp = 13.8^{\circ}$ in drone flight, which means gimbal pitch angle. The vertical distance can be calculated from the L and Gp information (Fig. 3.17.). In addition, since the calculation is made from the top of the plant leaves in the image, it is necessary to add the plant height. According to the values taken from the vineyard, this height is approximately 1.7 meters. Hence row height $R_h = 1.7\ mt$. With this information we have obtained, the H value has been calculated as Eq. 3.9 and Eq. 3.10 :

$$H = L \times sin(Gp) = 5.5 \times sin(13.8) = 1.31\ mt. \tag{3.9}$$

$$H + R_h = 3\ mt. \tag{3.10}$$

Figure 3.19. 3m altitude measurement values

According to the Fig.3.20., the half width of the row at the midpoint of the image is 35 pixels, equivalent to 0.6 meters. When the proportion is established according to figure, the $L = 14.97\ meters$. When the same calculation is applied, H is equal to $15 \times sin(13.8) = 3.57\ mt$.

If the row height is added, the real camera height is equal to $5.27\ mt$.

Figure 3.20. 5m altitude measurement values

It has been found that the logs received from telemetry are compatible with the findings, and an important step has been taken for visually calculating the height in the field. The applied method takes a known metric length and pixel value as input and generates a close output by using the focal length and gimbal pitch input from the camera, apart from barometer and GPS measurement.

3.5. Altitude estimation via Depth Map

Depth estimation is the process of estimating the distances or depths of objects in an image. This process is typically achieved through the acquisition and analysis of a pair of images taken with either a stereo camera or a single camera. Process with a stereo camera is used to find the overlapping points of the same objects viewed from two different angles and to calculate the distance between these points, taking into account factors such as the position of the camera, the difference in image planes, and the camera's intrinsic and extrinsic parameters. Depth estimation process with a single camera is typically achieved by using shadows, perspective, and different colors of objects. Factors such as the size and position of objects, lighting conditions, and surface properties of objects are also taken into account during the calculation process. It is still an active research topic and ongoing research in this area is enabling the development of more accurate, fast, and reliable depth estimation algorithms and the discovery of new methods for broader application areas. In this study, Monodepth2 was used which is one of the state-of-art method in this area. Monodepth2 is a deep learning framework for estimating the depth of a scene from a single monocular image.[21]

Monodepth2 uses a deep CNN to learn the relationship between an input image and its corresponding depth map. The network is trained on a large dataset of monocular images and their corresponding depth maps, allowing it to learn a mapping between the two. One of the key feature is the ability to estimate the depth of objects in real-time on a single GPU. This is achieved through a multi-scale depth estimation approach that allows the network to estimate the depth of objects at different scales in the image. It also incorporates a number of additional features, including the ability to estimate the depth of moving objects and the ability to generate dense depth maps with fine-grained details. Due to the benefits it provides, tests have been carried out to use this method for altitude estimation from vineyard images. By subtracting the depth of the vineyard

rows and detecting the height, a platform has been tried to be provided for flight from a fixed altitude. In this way, it will be possible to provide a stable altitude for tracking and spraying on the row.

Figure 3.21. Disparity Map of Mono+Stereo model results with KITTI dataset

The Monodepth2 model shared by Niantic labs was used in the studies for depth analysis. It includes 9 different models trained on the KITTI dataset. There are monocular, stereo, mono+stereo and 640x192 models with pretrained 640x192p and 1024x320p resolutions in Imagenet.

Pretrained 640x320 and 1024x320p resolution mono, stereo and mono + stereo models were used in our experiments. According to the observation in the results, the stereo model at 640p and the mono+stereo model at 1024p seem to produce more usable results.

In the tests performed, firstly, the metric equivalents of the distance inference were tested. Also, as the authors stated, models trained with stereo supervision were trained with a nominal baseline of 0.1 units. The baseline of the KITTI rig is 54 cm. Therefore, to convert our stereo predictions to real-world scale, depth multiplied with 5.4. It can be seen the image and disparity map in Fig.3.21.

The KITTI Dataset's metric distance measurements have been converted from npy to excel format and reformatted as color scale for convenience in our thesis work, and are

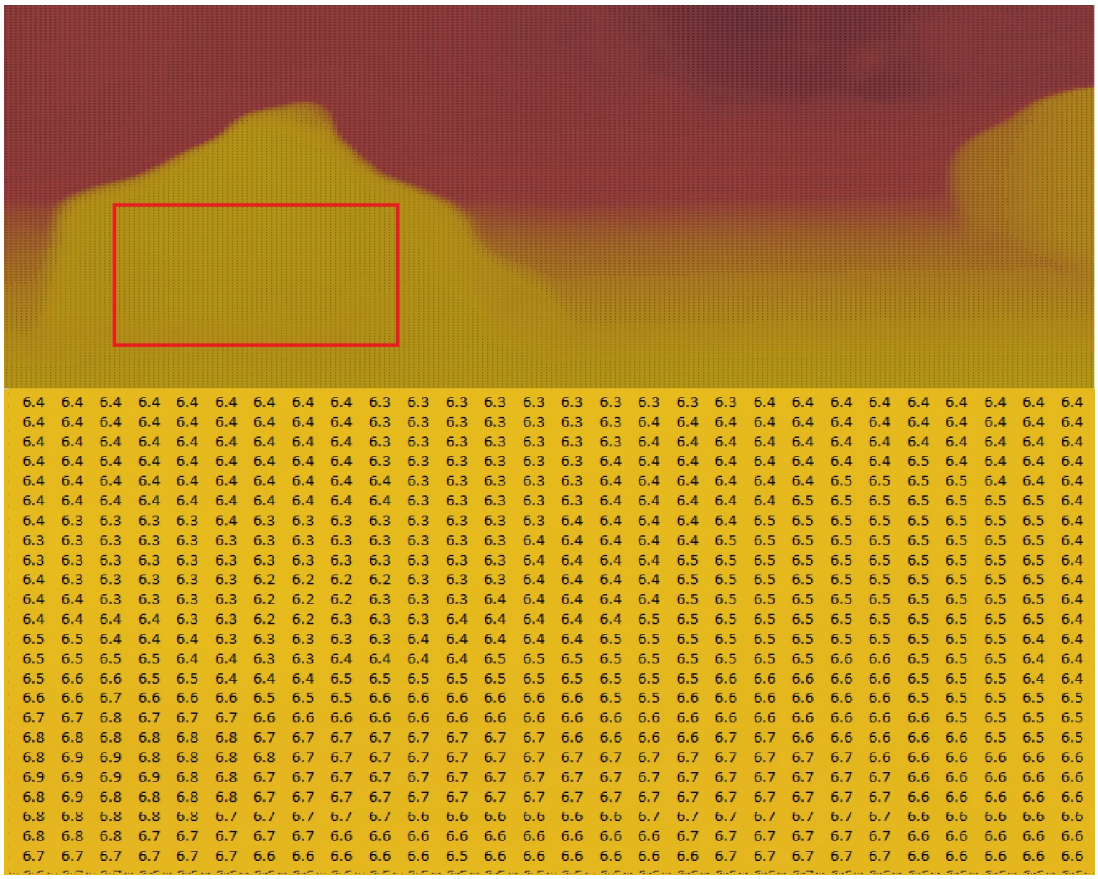

Figure 3.22. Metrics comparison of figure

compatible with disparity maps. Distance measurements increase from 4 meters due to the metric scale factor in the code. The authors report high consistency between metric estimates and actual measurements. Fig.3.22. shows that the framed area is 6 meters away.

In the next step, although it was stated that the metric outputs were especially makes sense for stereo-trained KITTI models, field images were also tested in the pre-trained models.

Fig.3.23. shows 640 x 192 Monocular, Mono+Stereo, Stereo Models and 1024 x 320 Monocular, Mono+Stereo, Stereo Trained Models outputs, respectively. Despite the visually realistic output of the disparity map in the 1024x320 Stereo Map, it was deemed necessary to compare the results by testing other models across the entire visual dataset for the study. This decision was made in order to ensure the robustness and validity of the findings, and to avoid any potential biases or limitations in the data analysis process.

1024x360 Mono+Stereo metric predictions for the middle region of the same image are also shown in Fig.3.24. In the results obtained, parallel to the KITTI dataset results, the near plane diverges starting from about 5 meters.

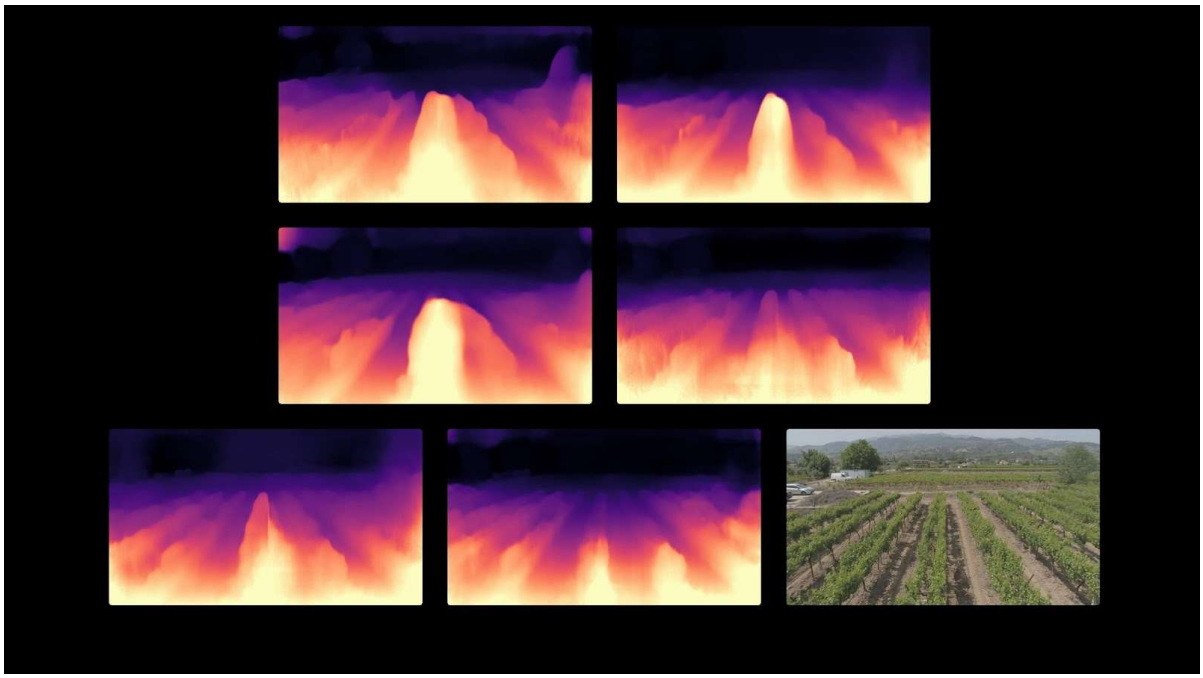

Figure 3.23. Vineyard metric predictions

It seems possible to keep the altitude within certain ranges by specifying threshold values, even in this state, which has not been retrained for agricultural field work.

According to the results we have received, it is seen that an acceptable limit is formed as the threshold value of the 6-meter prediction results. A range where it will not go below this value and will not exceed 10 meters offers a testable infrastructure for the flight of the vehicle. Apart from this, an approach to act as a percentage rate such as 20% and 40% of the minimum metric value can be presented.

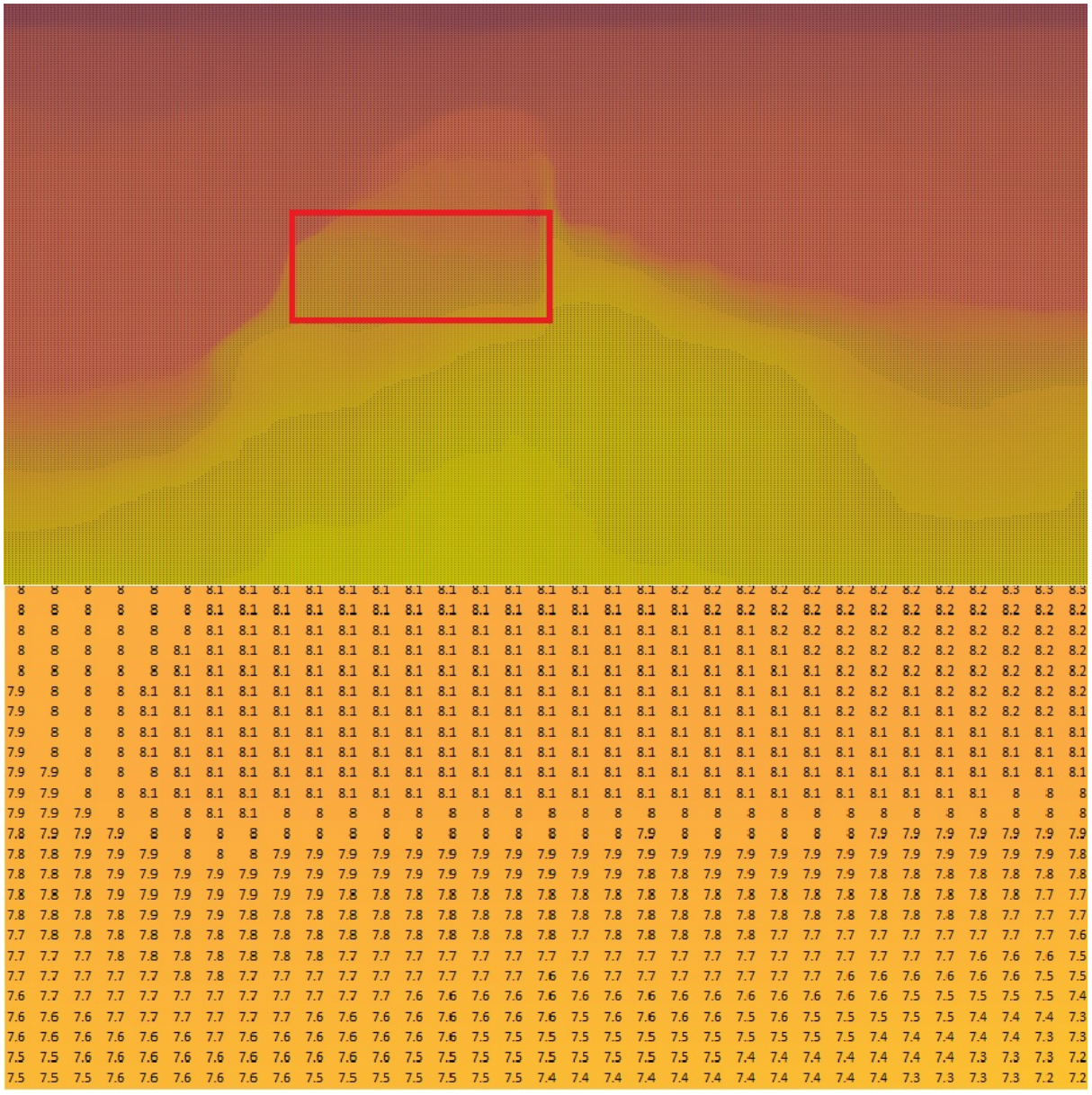

Figure 3.24. Vineyard disparity map comparison

3.6. Navigating UAV with Row Following

Line tracking algorithms can be used to autonomously navigate a UAV along a predefined path. This can be used for a variety of applications, such as inspection, mapping, and delivery.

The successful navigation of a PX4-based UAV requires a camera, a computer with ROS installed, and a reliable row following and altitude estimation algorithm. Recommended UAV system design can be found online. Once all necessary components are obtained, the following steps can be taken to navigate the UAV with ease.

1. Install the PX4 firmware on the UAV autopilot system.
2. Connect and test the camera in the companion computer.
3. Launch the row following algorithm in the ROS installed companion computer.
4. Point the camera at the desired line, and the algorithm will calculate its position and send commands to the UAV with MavROS to keep it centered.
5. The UAV will continue to follow the line until the line tracking algorithm is stopped.

These steps can significantly improve the efficiency and accuracy of aerial spraying in agricultural areas. The camera should be securely mounted on the UAV to ensure stabilization. It is important to use a high-quality camera with a wide FOV. Testing the line tracking algorithm in a safe environment before using it in a real-world application is crucial. It is also important to consider limitations of the line tracking algorithm, as it may not be able to track the line if it is too dark or if there are obstacles in the way. Real-time image transfer during drone flights through the first two stages has been achieved using an external computer, although integration has not yet been implemented in current work. Detection has been carried out on the incoming images, and feedback and guidance based on the detections made are required. This work is ongoing as preparation for the thesis presentation progresses.

CHAPTER 4

EXPERIMENTS AND RESULTS

In order to establish a reliable ground truth for the height information in this study, the flight on May 2022 was tracked with Flight Record in the drone's mobile application for a duration of seven minutes. Height updates were made during the flight at 3 meters, 4 meters, and 5 meters, and the flight record was subsequently imported into a computer environment. Below is the information shared about the photo and video sets recorded during the flight Tab.4.1.. During the flight in the dataset, the photos and videos taken are numbered in the first column. The duration of each shot, the number of frames used from them, the average flight altitude during that time interval according to the flight logs, and the angle of the camera relative to the horizon are shown. The gimbal angle and visual height data from IMU and ultrasonic sensor based VPS information were then extracted and analyzed to determine the accuracy of the data obtained from the existing sensors. The accuracy of the altitude data obtained from the existing IMU and VPS sensors was then examined by comparing them for the entire flight, as shown in Fig.4.1.

Table 4.1. Dataset information

Dataset	Number of Frames	Shooting Time (s)	IMU Ave (m)	VPS Ave (m)	Gimbal Pitch Angle
Video-1	32	4m5s	3.38	2.69	-12.4
Photo-1	44	55s	4	3	-12.4
Video-2	24	2m42s	4.27	4.1	-13.8
Photo-2	21	26s	4	3.83	-13.8
Video-3	8	57s	4.78	4.58	-13.8
Photo-3	21	37s	4.16	4.87	-13.8

The result for flight showed that VPS data is more susceptible to reflections and changes in altitude, leading to more uncertain results compared to IMU data Fig.4.1. As a result, the IMU data was prioritized for the study. The graph with only IMU can be seen below Fig.4.2. Similarly, it can be observed in the same graph that flight altitude varies gradually for 3, 4, and 5 meters.

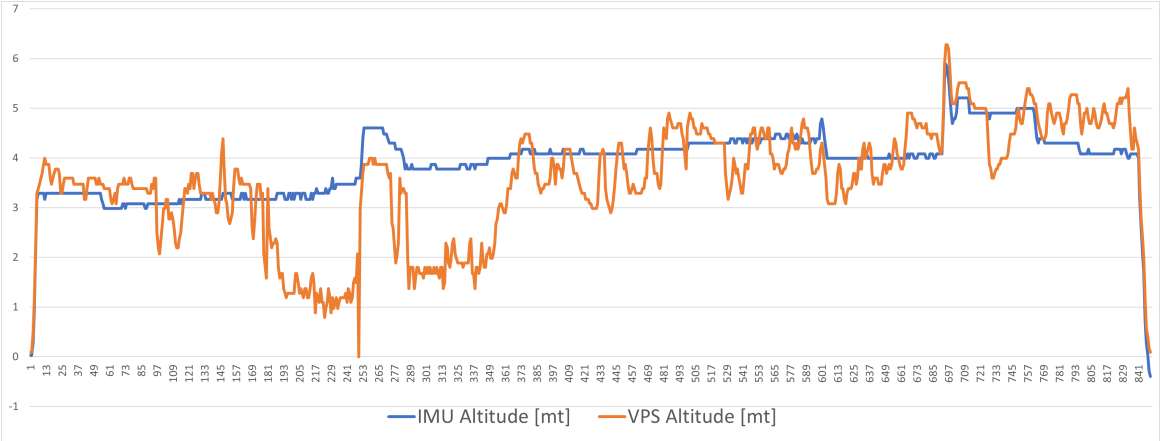

Figure 4.1. IMU vs VPS altitude values

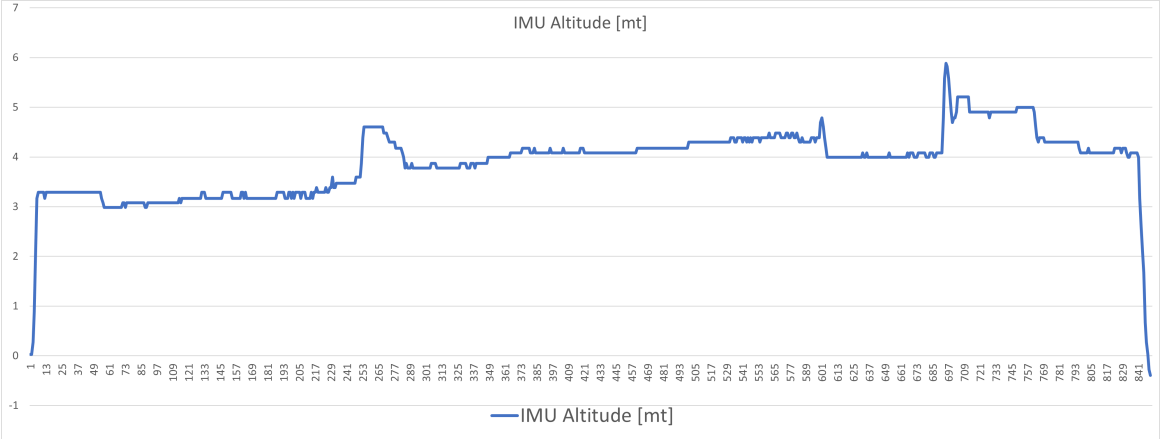

Figure 4.2. IMU altitude values

4.1. Altitude Estimation Results via Camera Calibration

At this point, it is necessary to recalculate the obtained data using the altitude calculation method suggested in this study and make a comparison. With the constituted method, the altitude calculation can be derived if the pixel length in the image, its metric equivalent, gimbal pitch angle, and focal length values are known. In the previous section, the altitude data was obtained by providing the images one by one as input to the program through automatic pixel distance for blob width calculation mentioned earlier. For this, dataset images obtained from different altitudes were organized and used as input for the program. The outputs are given Fig.4.3..

IMU and Estimated Altitute values are shown in Fig.4.3., Fig.4.4. and Fig.4.5., with deviations of up to 1 meter on average at different heights. The algorithm of the automatic pixel calculation program to find the half-height line where the middle row is located can find the correct values in 30 of 80 photos for 4m height. Due to the accuracy

rate of 37%, it does not yet work at the desired efficiency. In order to obtain more accurate and reliable results, it is imperative that the algorithm be improved and then retested.

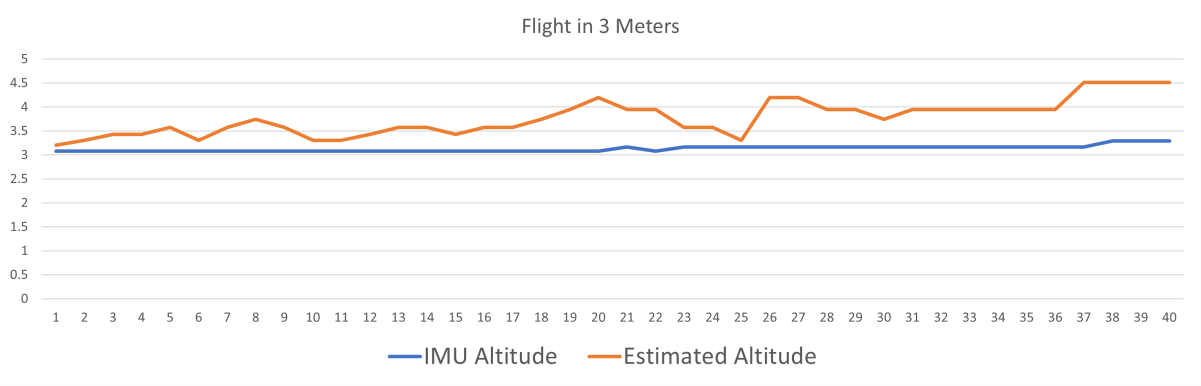

Figure 4.3. 3m altitude values

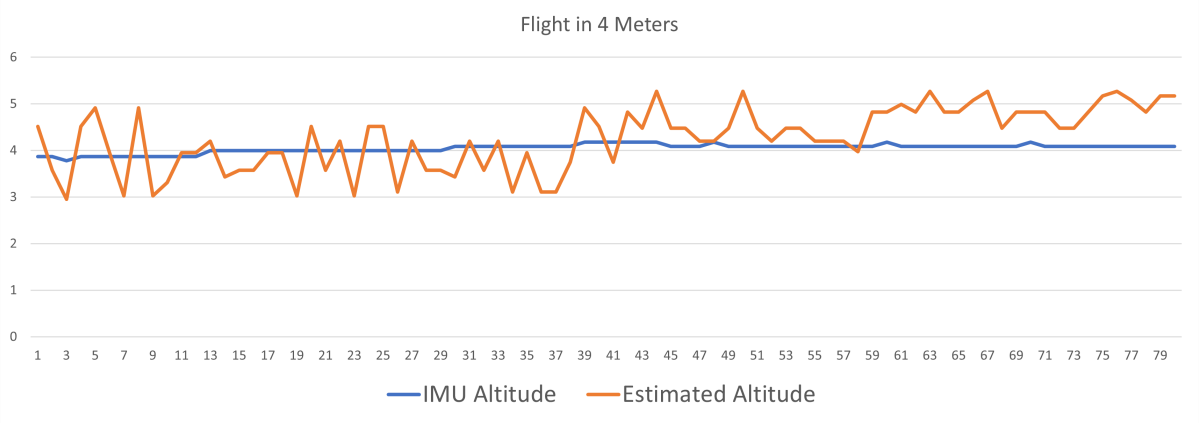

Figure 4.4. 4m altitude values

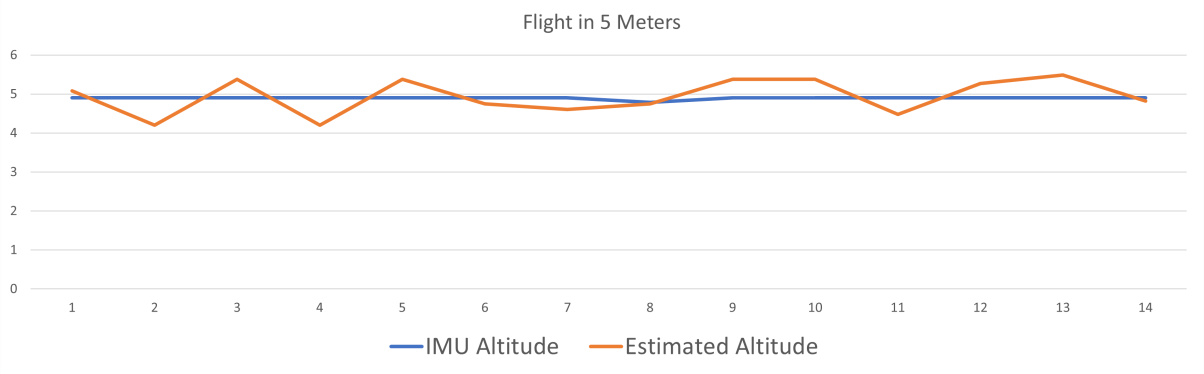

Figure 4.5. 5m altitude values

The graph was created by calculating the pixel distances manually. Linear outputs close to IMU data were observed in Fig.4.3. and Fig.4.5., while results were obtained with frequent and variable movements of values in Fig.4.4., 4m flight chart. Average Estimated altitude values, standard deviation and error rates are shown in Tab.4.2..

Table 4.2. Altitude evaluation results from camera calibration

Altitude	Average Altitude	Average Error	Standard Deviation
3m	3.775	-0.644	0.728
4m	4.259	-0.217	0.939
5m	4.375	0.523	0.659
Total	4.127	-0.112	0.681

4.2. Altitude Estimation Results via Depth Maps

To generate reliable altitude estimations, depth maps of the vineyard images were extracted using the Monodepth2 depth prediction model. The Monodepth2 model is trained for autonomous vision road tracking with the KITTI dataset, using images with heights and widths of 640x192 and 1024x320, respectively.

The network produces an output in the form of a disparity map $(disp)$, sigmoid, between [0,1]. The formula for converting the disparity map to a depth prediction is given in the 'additional considerations' section of the paper.[21] First, the algorithm converts the sigmoid value to values between 0.1 and 100 as $scaled\ disp$ in Eq.4.1, which also generates unitless depth values between 0.1 and 100 as Eq.4.2 inside the function Eq.4.1.

$$scaled\ disp, depth = disp\ to\ depth(disp, 0.1, 100) \tag{4.1}$$

$$depth = 1/scaled\ disp \tag{4.2}$$

The metric depth information was determined through the stereo-featured dataset. The actual KITTI stereo rig baseline value is 0.54 m, and an effective baseline value of 0.1 was used during monodepth2 training.

Therefore, when obtaining metric values, the unitless depth value is multiplied by the 5.4 Stereo Scale factor to obtain metric measurements Eq.4.3.

$$metric\ depth = Stereo\ Scale\ Factor * depth \tag{4.3}$$

The metric predictions for these images were obtained from the outputs of the 1024x320 stereo model, with respect to the 512 x 160 principal point. The resulting metric prediction values were subsequently stored, analyzed, and compared across different meter heights.

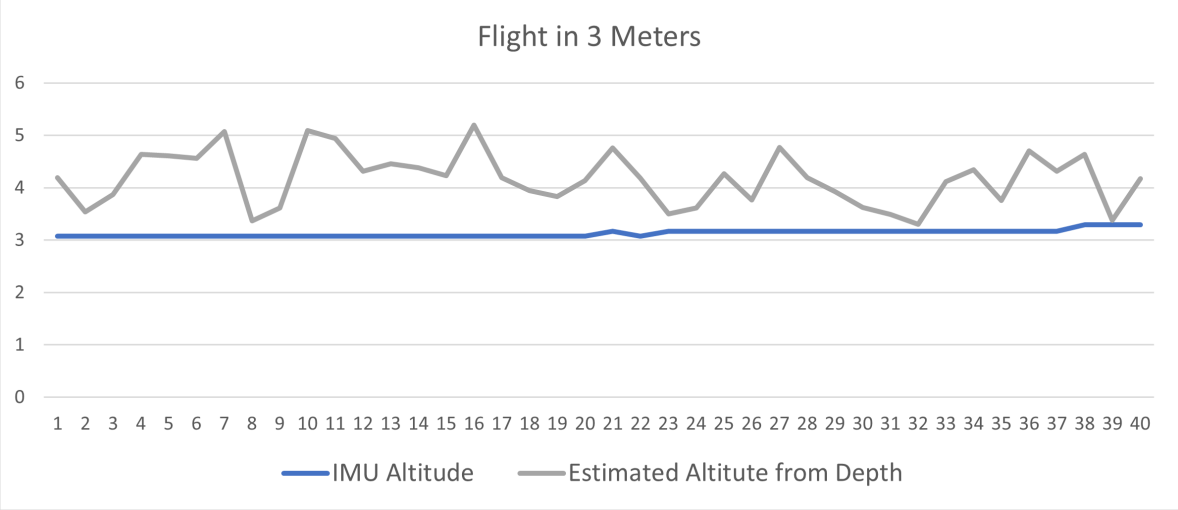

Figure 4.6. 3m altitude values from depth

The obtained metric predictions yielded diagonal L values. Height calculation was performed relative to the ground based on Eq.3.8. The resulting values were then added to the figures with height and sequence axes. As shown in Fig.4.6., a comparison was made for a height of 3 meters and an average error of 1 meter was observed.

Similarly, altitude estimation for 4 meters, as seen in Fig.4.4., resulted in an average error rate of 1 meter. However, altitude estimation for 5 meters, as shown in Fig.4.5., calculated an error rate of only 0.15 meters. The error and standard deviation rates for the figures are shown in the accompanying Tab.4.3.

When comparing the two tables Tab.4.2. and Tab.4.3., it can be seen that there is a deviation of more than 0.5 in the errors and standard deviations of the depth measurements.

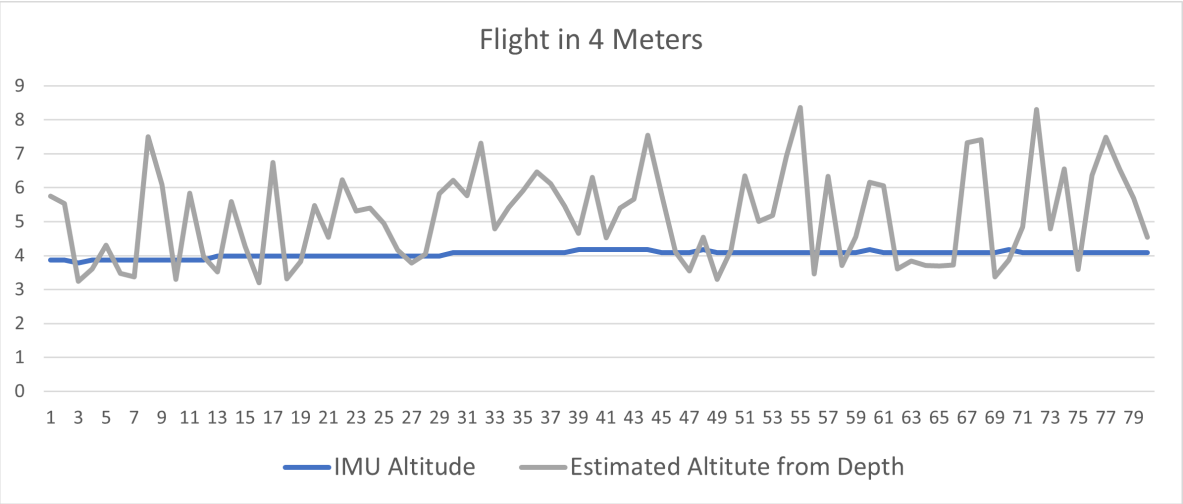

Figure 4.7. 4m altitude values from depth

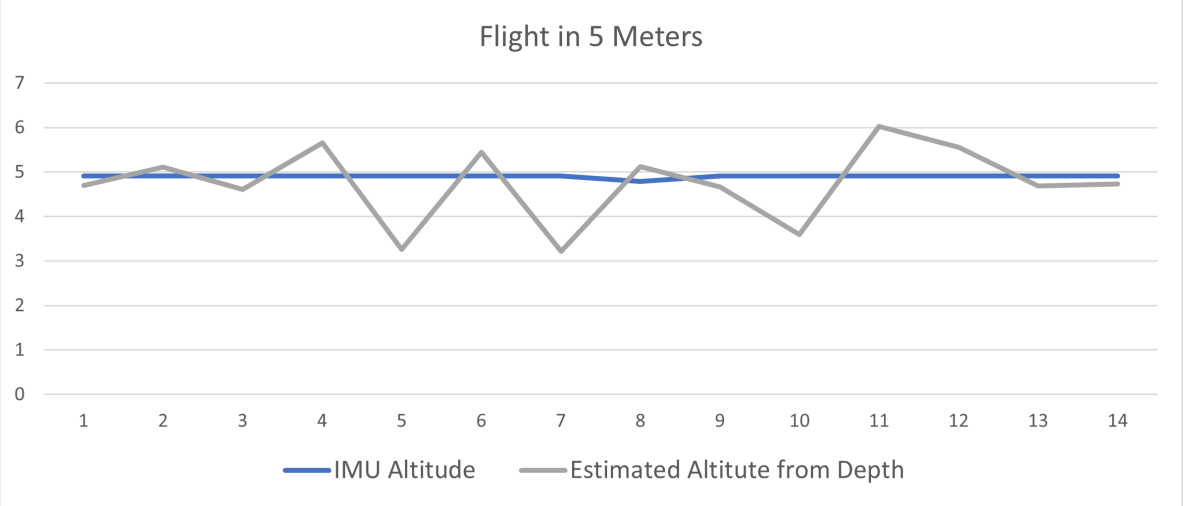

Figure 4.8. 5m altitude values from depth

While the data obtained from camera calibration produced more consistent results, it is necessary to evaluate the factors that led to the current results.

Firstly, it should be noted that the images taken at different heights were not equally distributed in the dataset. While there were 85 outputs recorded at 4 meters, only 55 and 14 images were taken at 3 and 5 meters respectively. This imbalance in the dataset may have had an impact on the results. Additionally, the results provided were only compared

Table 4.3. Altitude evaluation results from depth map

Altitude	Average Altitude	Average Error	Standard Deviation
3m	4.174	1.043	1.182
4m	5.131	1.089	1.730
5m	4.739	-0.160	0.888
Total	4.805	0.658	1.267

to those of the stereo model, with no comparison made to the results of the monocular + stereo model or the 640x192 image. Furthermore, the depth method used in the study relied on a pre-trained model from the KITTI dataset, which produces outputs based on a baseline distance of 54 cm. However, considering the minimum distance in our images with a 3-meter altitude, a crop of 1.3 meters is required. When taking into account the error margins and angles, a minimum distance of 1-1.5 meters can be observed. Moreover, the ground truth inputs provided were the same as those used in the model. Therefore, it is necessary to record the ground truth distances that can be used and to train the model with them to compare the depth outputs once again.

Currently, different experimental outputs have been obtained by reducing the image size to 320x180, changing the scale factor, and testing vertical points outside the principal point. However, consistent results were not obtained with these outputs. For example, the reduced version of the 4-meter images produced an average of 4.45, an error rate of 0.4, and a standard deviation of 0.6 at [512,200] point. The reduced 3-meter images produced an average of 3.81 with an error rate of 0.68 and a standard deviation of 0.72 at [512,200] point. However, the reduced 5-meter images produced outputs that were further from the current model, with an average of 4.08, an error rate of -0.81, and a standard deviation of 0.87 at the same point. In the original images, [512,200] point was consistent for 4-meter altitude, but it produced worse results for 3-meter altitude. For 5-meter altitude, [512,180] point was found to be more consistent. While these outputs were mentioned in the here, they were not shared within this thesis.

4.3. Row Following Results

The outputs were obtained by applying a row following algorithm to all the data in the dataset. Initially, the images were reduced to a size of 320x180 in order to reduce the processing power required in the real application. The distance between the center point in the images and the center of the row was manually recorded as the pixel distance. For different heights, the pixel equivalent of half the length of the vineyard row was manually calculated and noted. It was observed that the revised images varied between 10 and 20 pixels for different heights. The deviation distance was recorded according to the ratio of the real 60 cm length in the field. The recorded manual calculations were compared with the outputs from the algorithm, and graphic analyses were generated and presented.

In Figure 4.9., the row following output of the 3-meter flight is presented, indicating an average row deviation is 11 cm, a peak difference of comparison 45 cm. As reported in Table 4.4. , the mean error was calculated to be 9 cm with a standard deviation of 12 cm.

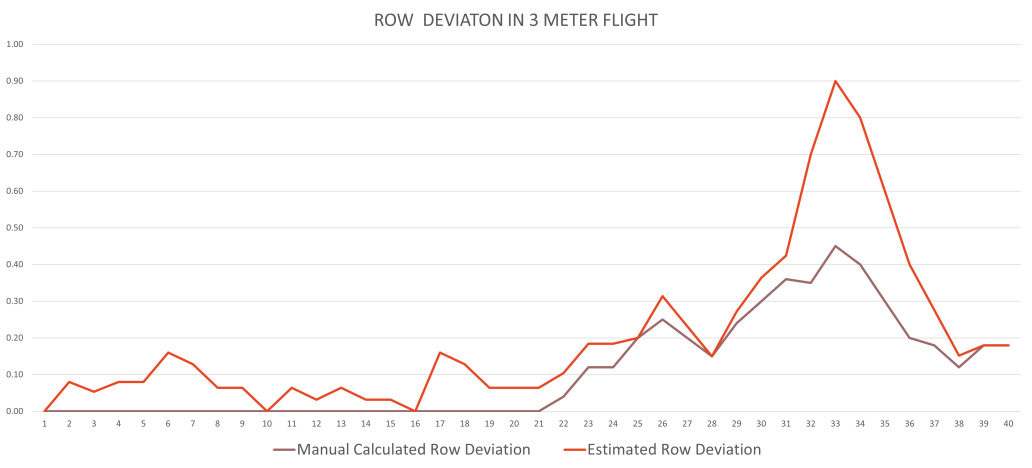

Figure 4.9. 3m row following values

In Figure 4.10., the row following output of the 4-meter flight is presented, indicating an average row deviation is 38 cm, a peak difference about 45 cm again. As reported in Table 4.4. , the mean error was calculated to be 32 cm with a standard deviation of 34 cm.

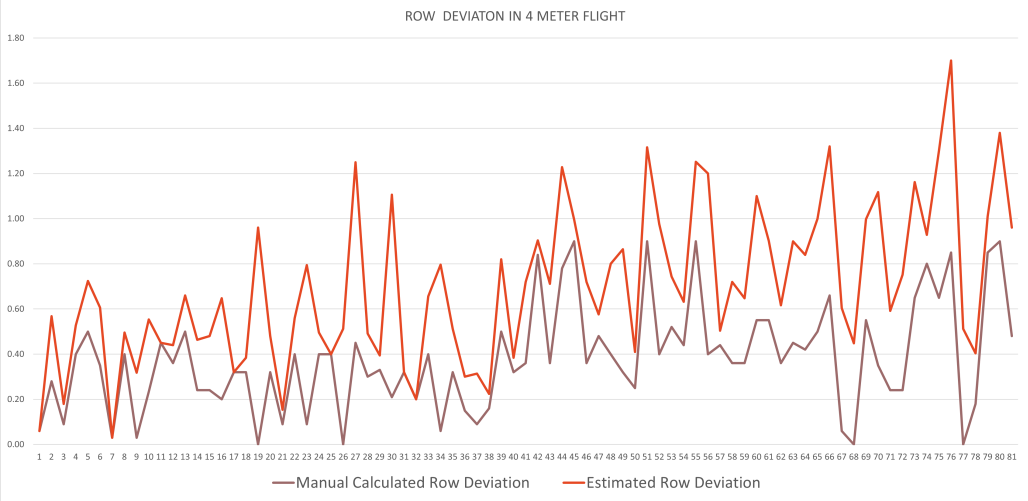

Figure 4.10. 4m row following values

In Figure 4.11., the row following output of the 5-meter flight is presented, indicating an average row deviation is 54 cm, a peak difference of about 10 cm again. As reported in Table 4.4. , the mean error was calculated to be 4 cm with a standard deviation of 59 cm.

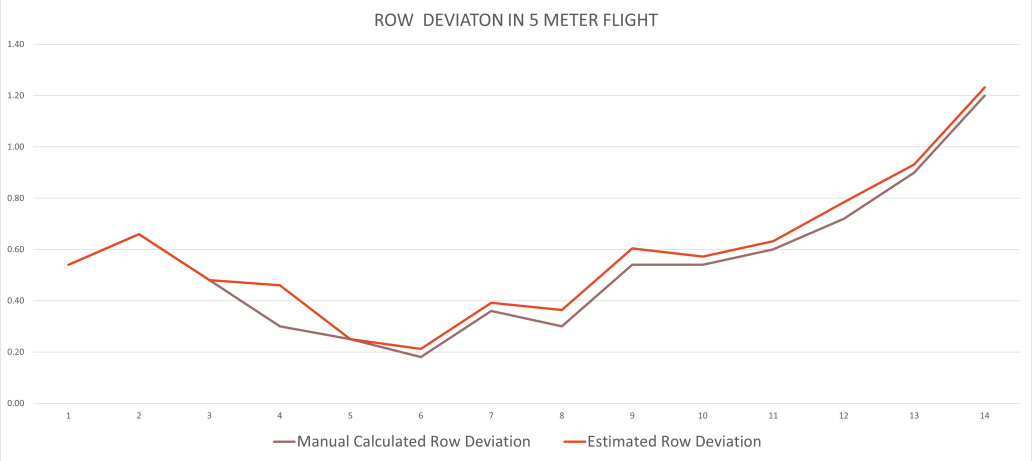

Figure 4.11. 5m row following values

The results of the study revealed deviations from the mean order of 34 cm at all altitudes Tab.4.4., with an average of 15 pixel difference between the manually recorded and estimated deviation values, and a standard deviation of 35 cm. These values exceeded expectations and suggest promising results in terms of field applicability. Moreover, during the algorithmic calculation, the average vineyard row half length pixel value in the code was given as 15 from the default value. It can be suggested that defining flight altitude and a dynamic value optimized for each data, instead of the default value, would lead to more accurate results.

Table 4.4. Row following results

Altitude	Average Row Deviation	Average Error	Standard Deviation
3m	0.11	0.09	0.12
4m	0.38	0.32	0.34
5m	0.54	0.04	0.59
Total	0.34	0.15	0.35

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

This thesis proposes alternative and improved methods for UAV navigation in agricultural fields, specifically for altitude estimation and row following. It contributes to the existing literature by exploring the application of unmanned aerial vehicles and vision-based stable navigation in vineyard agriculture.

The problem addressed in this thesis was experienced first-hand as a result of issues encountered in the Meshine Swarm Technologies start-up company, of which I am also a founding partner. During UAV flights for pesticide spraying in vineyards, altitude stabilization could not be achieved, and row tracking could not be performed with the desired precision. As a result, unwanted pesticide contamination occurred in adjacent rows during low pesticide spraying, and sufficient flights could not be carried out due to GPS losses. It was decided to address these field problems and it is anticipated that the results of this thesis will provide additional benefits in the field for our company, which has been conducting industrial drone services for approximately five years.

In the initial phase, existing methods were investigated, and computer vision studies according to plant health in agricultural fields were observed. Navigation with image processing has mostly been developed for autonomous movements of ground vehicles such as tractors. Although autonomous navigation in UAV models has many sensors based on Radar, Lidar, Ultrasonic, IMU, and GPS for high precision, these sensors are quite expensive, and deficiencies were detected in autonomous vision. Therefore, it can be said that this is a niche and pioneering study specific to viticulture.

Consequently, we collected our own images from vineyards to create a visual dataset. Although not large scale, we managed to create a dataset by taking flights at different times. The captured images were processed using image processing methods for row detecting, and we successfully identified the rows. We designed and tested a new algorithm for row following, which provided a metric output of the deviation from the middle row in the image for a flight initially assumed to be centered, making it ready for field application. For altitude estimation, we attempted two methods using camera calibration and depth maps. We obtained our camera calibration matrix for the captured

images and found a deviation of 0.2mm. Altitude estimation was achieved using geometric calculations by first calculating the diagonal distance of the middle point seen by the camera in the image, and then using the angle value to calculate the height of the row. We added plant height to obtain the altitude value relative to the ground.

In the depth measurement experiments, metric outputs were obtained from the Monodepth2 model. The tests were conducted on the vineyard dataset with different image size scales of the stereo and monocular images trained in the model. It was observed that the stereo and monostereo models obtained more consistent results. The metric distances of these outputs were observed by converting them from npy file to Excel and creating a color map.

The subsequent objective is to transmit the data that will fix the height position to the motors at the conclusion of this stage and to construct the necessary PID control mechanism.

As drone tests were not possible in the field, altitude data outputs were attempted to be determined through two different tests in computer-based experiments during the study. An additional image processing program was developed to automatically detect the half-distance of the vineyard row in the images with respect to the principal point. However, this program was found to be inefficient. Using pixel values calculated manually, we calculated the altitude at which the drone was flying during operation with an average error of 0.64 meters and a standard deviation of approximately 0.9 meters. However, the same level of precision could not be achieved for depth measurement. This was attributed to not training a new model, and development in this direction is ongoing. Therefore, it has been demonstrated that the outputs obtained through camera calibration values can be used with an approximate margin of error for this thesis output. It can be used as a backup system when GPS is unavailable, and IMU data is insufficient.

During the experiments, multiple methods were attempted, and a feasible calculation was devised by simplifying the process as much as possible. It is believed that an artificial intelligence model can be trained by utilizing more pixel distance data. For instance, when the average human height is trained by collecting pixel data at various camera resolutions and heights, the model can infer pixel distance for people it has not seen before. In this way, it can be ensured that an airborne camera is at a fixed altitude against the object by only receiving the gimbal angle data. This study represents a crucial

backup method for UAVs, particularly in height measurement.

The outputs of this thesis have significant potential for further development. By implementing an automatic pixel width algorithm, more precise results can be obtained, and manual calculations can be avoided entirely. Moreover, by training the model with our own Ground Truth data for depth, more consistent results can be obtained over depth. If the two outputs demonstrate promising results, they can be integrated using sensor fusion methods and synchronized with other sensor outputs. Once the findings have been validated through field tests, patent applications can be pursued.

The findings of this study have the potential to extend beyond the agricultural sector, benefiting other industries that require precise row following and altitude estimation. Mathematical calculations can be used for distance stabilization with lower processing power for a given object height, making it an efficient solution for human or vehicle following.

The anticipation is that this research will serve as a catalyst for subsequent explorations into the expansive capabilities of unmanned aerial vehicles and precise positioning techniques, facilitating the resolution of intricate issues across a multitude of disciplines.

REFERENCES

(1) *AGROFOOD INDUSTRY REPORT*; Standard; Ankara, TR: Presidency Of The Republic Of Turkey, Investment Office, 2023.

(2) Gomes, N. M. SUAS NEWS, Chinese Agri-tech Giant XAG Defended 20-million-hectare Farmlands with Crop Spraying Drones.

(3) Herrero, M.; Thornton, P. K.; Mason-D'Croz, D.; Palmer, J.; Benton, T. G.; Bodirsky, B. L.; Bogard, J. R.; Hall, A.; Lee, B.; Nyborg, K., et al. *Nature Food* **2020**, *1*, 266–272.

(4) Moe, A. Passive aircraft altitude estimation using computer vision, Ph.D. Thesis, Linköping University Electronic Press, 2000.

(5) Eynard, D.; Vasseur, P.; Demonceaux, C.; Frémont, V. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp 646–651.

(6) Sanger, T. D. *Biological cybernetics* **1988**, *59*, 405–418.

(7) Szafranski, G.; Czyba, R.; Janusz, W.; Blotnicki, W. In *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2013, pp 508–515.

(8) Patricio, D. I.; Rieder, R. *Computers and electronics in agriculture* **2018**, *153*, 69–81.

(9) Gomes, J. F. S.; Leta, F. R. *European Food Research and Technology* **2012**, *235*, 989–1000.

(10) Liu, Y.; Noguchi, N.; Liang, L. *Computers and Electronics in Agriculture* **2019**, *162*, 126–133.

(11) Choi, H.; Geeves, M.; Alsalam, B.; Gonzalez, F. In *2016 IEEE Aerospace Conference*, 2016, pp 1–5.

(12) Vidovic, I.; Cupec, R.; Hocenski, Ž. *Pattern Recognit.* **2016**, *55*, 68–86.

(13) Beyeler, A.; Mattiussi, C.; Zufferey, J.-C.; Floreano, D. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* 2006, pp 2836–2841.

(14) Cherian, A.; Andersh, J.; Morellas, V.; Papanikolopoulos, N.; Mettler, B. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp 3900–3905.

(15) Mo, J.; Sattar, J. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

(16) Sumikura, S.; Sakurada, K.; Kawaguchi, N.; Nakamura, R. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14*, 2019, pp 281–297.

(17) Tanaka, M. In *Proceedings of the ISCIE International Symposium on Stochastic Systems Theory and its Applications*, 2017; Vol. 2017, pp 48–55.

(18) Kim, I. S.; Kim, H.; Lee, S.; Jung, S. K. *Electronics* **2023**, *12*, 350.

(19) Perera, D. M.; Arachchige, D. D.; Mallikarachchi, S.; Ghafoor, T.; Kanj, I.; Chen, Y.; Godage, I. S. In *2023 IEEE International Conference on Soft Robotics (RoboSoft)*, 2023, pp 01–07.

(20) Tekir, S.; Bastanlar, Y. In *Data Mining Methods, Applications and Systems, IntechOpen*, 2020.

(21) Godard, C.; Mac Aodha, O.; Firman, M.; Brostow, G. J. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp 3828–3838.

(22) Zhang, Y.; Qi, Y.; Qu, X.; Fan, W.; Liu, C. In *Optical Metrology and Inspection for Industrial Applications IX*, 2022; Vol. 12319, pp 316–320.

(23) Salazar, A. A. D. *LatinX in AI at Computer Vision and Pattern Recognition Conference 2021* **2021**.

(24) Kim, W.-S.; Lee, D.-H.; Kim, Y.-J.; Kim, Y.-S.; Kim, T.; Park, S.-U.; Kim, S.-S.; Hong, D.-H. *Agronomy* **2020**, *10*, 1670.

(25) Li, M.; Lin, R.; Wang, H.; Xu, H. In *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2013, pp 1653–1658.

(26) Ibraheem, N. A.; Hasan, M. M.; Khan, R. Z.; Mishra, P. K. In 2012.

(27) Pascale, D. In 2003.

(28) Joblove, G.; Greenberg, D. P. *Proceedings of the 5th annual conference on Computer graphics and interactive techniques* **1978**.

(29) Johnson, S. In 2006.

(30) Image Filtering Using Convolution in OpenCV — learnopencv.com, https://learnopencv.com/image-filtering-using-convolution-in-opencv/, [Accessed 14-Jul-2023].

(31) Shapiro, L.; Stockman, G. Computer Vision, 2001.

(32) Wikipedia Binary image — Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/w/index.php?title=Binary\image&oldid=1157382988, [Online; accessed 14-July-2023], 2023.

(33) Product Documentation - NI — ni.com, https://www.ni.com/docs/en-US/bundle/ni-vision-assistant-help/page/invertbinaryimg_howto.html, [Accessed 14-Jul-2023].

(34) Sezgin, M.; Sankur, B. *J. Electronic Imaging* **2004**, *13*, 146–168.

(35) Otsu, N. *IEEE Transactions on Systems, Man, and Cybernetics* **1979**, *9*, 62–66.

(36) Wikipedia Blob detection — Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/w/index.php?title=Blob\detection&oldid=1136288125, [Online; accessed 14-July-2023], 2023.

(37) Gong, X.; Su, H.; Xu, D.; Zhang, Z.; Shen, F.; Yang, H.-B. *International Journal of Automation and Computing* **2018**, *15*, 656–672.

(38) Ballard, D. H.; Brown, C. M. In 1982.

(39) Stockman, G. C.; Shapiro, L. G. In 2001.

(40) González, R. C.; Woods, R. E.; Masters, B. R. *Journal of Biomedical Optics* **2009**, *14*, 029901.

(41) Richard Green, P. N. Detecting Tramways in Crops for Robot Navigation.

(42) Alexander Mordvintsev, A. K. R. Camera Calibration.

(43) Bastanlar, Y. In 2009.

(44) Newman, P.; Ho, K. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005.

(45) Rettenmund, D.; Fehr, M.; Cavegn, S.; Nebiker, S. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* **2018**.

(46) Obeida ElJundi, T. Camera Calibration toolbox using Python, openCV.

(47) Daniela Aguirre Salazar, S. S. MRT Camera Calibration Toolbox.

(48) Inc., T. M. Computer Vision Toolbox version: 9.4 (R2022b), Natick, Massachusetts, United States, 2023.