University of Southern Queensland

Faculty of Engineering and Surveying

Object Avoidance System Using Millimeter-Wave Technology and Machine Vision in Crop-Dusting Aircraft

A dissertation submitted by

Hayden Lennox

[redacted]

In fulfilment of the requirements of

ENP4111 Research Project

Towards the degree of

Bachelor of Engineering (Honours)

(Mechatronic)

Submitted 03/11/2024

ABSTRACT

Keywords: *Crop-Dusting Aircraft, Bird Strike, Machine Vision, Millimeter-Wave Radar, Object Avoidance System*

For decades, the agricultural industry has relied heavily on crop-dusting aircraft to assist with crop protection. Crop-dusters are used due to their efficient land area coverage and the ability to not damage crops like a generic agricultural wheeled vehicle might. These aircraft can protect crops by spraying fertilizers, insecticides, herbicides and/or fungicides over a crop field.

As can be expected, there are many risks associated with using a crop-dusting aircraft. The pilots are subjected to large g-forces due to quick manoeuvres from descending, ascending, and turning at rapid speeds which can greatly accelerate fatigue. One lesser-known risk among the public is that of bird strikes. Bird strikes occur when a bird makes contact with an aircraft. This usually causes damage to the plane and in extreme cases, can cause death due to the aircraft crashing. Crop-dusting aircraft are required to fly as low as 10-15 feet and often birds or bush turkeys fly out of the field. Fatigue and foreign objects emerging from crop fields are leading causes of crop-dusting crashes. To date, there are no object avoidance systems fitted to the aircraft, and it is the pilot's responsibility to try and identify and avoid a possible bird strike.

This research proposal will aim to investigate whether an object avoidance system will be a feasible alternative to identify and avoid bird strikes. There are three experiments that will be completed to accomplish this. This includes:

1. Object Avoidance System using Millimeter-Wave Radar Technology
2. Object Avoidance System using Machine Vision
3. Object Avoidance System using a combination of Both Machine Vision and Millimeter-Wave Radar Technology

These experiments have been proposed due to current Unmanned Aerial Vehicle (UAV) systems utilising these technologies. They have been broken into three separate tests as it is not conclusive as to whether the object avoidance system will be able to detect an object in time to avoid it. By proposing three different experiment techniques, it is expected that at least one of them will be successful. Benefits of this success may include this system being

fitted to aircraft such that crop-dusting aircraft are safer to operate and will greatly reduce injury or death from bird strikes.

University of Southern Queensland

School of Engineering

ENP4111 Research Project

LIMITATIONS OF USE

The Council of the University of Southern Queensland, its School of Engineering, and the staff of the University of Southern Queensland, do not accept any responsibility for the truth, accuracy or completeness of material contained within or associated with this dissertation.

Persons using all or any part of this material do so at their own risk, and not at the risk of the Council of the University of Southern Queensland, its School of Engineering or the staff of the University of Southern Queensland.

This dissertation reports an educational exercise and has no purpose or validity beyond this exercise. The sole purpose of the course pair entitled "Research Project" is to contribute to the overall education within the student's chosen degree program. This document, the associated hardware, software, drawings, and other material set out in the associated appendices should not be used for any other purpose: if they are so used, it is entirely at the risk of the user.

ACKNOWLEDGEMENTS

I would like to thank everyone who has supported me throughout this entire project.

I would like to offer a special thank you to Hedley Watt, the owner of Aero Professional Services, for allowing me to conduct my research and data collection on his property. His experience of over 40 years being a crop-dusting pilot has presented me with invaluable assistance in the completion of this project.

I would like to thank Oliver Watt for his assistance with the field testing and data collection for this project.

I would also like to thank my supervisor, Dr Tobias Low, for his continuous support and feedback throughout the entirety of this dissertation.

DECLARATION

I declare that:

- No part of this assignment has been copied from any other person's work or from any other source except where due acknowledgment is made in the assignment, and
- No part of this assignment has been written for me by any other person except where such collaboration has been authorized in writing by the course examiner concerned.

Hayden Lennox

Student Number: [redacted]

ABBREVIATIONS

MMW: Millimeter-Wave

UAV: Unmanned Aerial Vehicle

OAS: Object Avoidance System

3D: Three Dimensional

Table of Contents

ABSTRACT....ii

LIMITATIONS OF USE....iv

ACKNOWLEDGEMENTS....v

DECLARATION....vi

ABBREVIATIONS....vii

1 INTRODUCTION....14

1.1 Aim....14

1.2 Background....14

1.3 Problems....14

1.4 Objectives....15

1.5 Outcomes & Benefits....15

2 LITERATURE REVIEW....17

2.1 Established Knowledge....17

2.1.1 Evidence Supporting Need for Project Research....17

2.1.2 OAS Using Machine Vision....17

2.1.3 OAS Using MMW Radar....21

2.2 Knowledge Gap....22

2.2.1 Study Justification....22

2.2.2 Project Feasibility Analysis....23

3 METHODOLOGY....24

3.1 Project Parameters....24

3.1.1 Scope....24

3.1.2 Limitations....24

3.2 Project Experiments....25

3.3 Quality Assurance....27

3.4 Research Questions....27

4 SYSTEM DESIGN 28
4.1 Overview 28
4.2 Component Selection 28
4.2.1 Pulley System 28
4.2.2 IWR1642BOOST Millimeter-Wave Radar 32
4.2.3 GoPro Hero 4 Black 42
4.3 Additional Notes 44
5 TESTING AND CONCLUSIONS 46
5.1 Millimeter-Wave Radar 46
5.2 Object Detection Algorithms 50
5.2.1 Test 1 – Testing Comparisons Between Two Frames 51
5.2.2 Test 2 – Altering Frame Number from Video Footage 56
5.2.3 Test 3 – Cropping Pulley Plate from Earlier Image and Altering Frame Number from Video Footage 58
5.2.4 Test 4 – Segmented Image Frames and Altering Frame Number from Video Footage 63
5.2.5 Test 5 – Importing a Clear Pulley Plate Image and Altering Frame Number from Video Footage 66
5.2.6 Test 6 – Edge Detection in Image Segmentation 69
5.3 Combination of MMW Radar Technology and Machine Vision 81
5.4 Additional Notes 88
5.5 Conclusions 91
REFERENCES 93
Appendix A: Project Schedule 95
Appendix B: Risk Assessment 96
Appendix C: Budget and Resources Required 101
Appendix D: Test 1 – ORB Matlab Code 102

Appendix E: Test 1 – SURF Matlab Code............104
Appendix F: Test 1 – SIFT Matlab Code............106
Appendix G: Test 3 – ORB Matlab Code............108
Appendix H: Test 3 – SURF Matlab Code............110
Appendix I: Test 3 – SIFT Matlab Code............112
Appendix J: Test 4 – ORB Matlab Code............114
Appendix K: Test 4 – SURF Matlab Code............118
Appendix L: Test 4 – SIFT Matlab Code............122
Appendix M: Test 5 – ORB Matlab Code............126
Appendix N: Test 5 – SURF Matlab Code............128
Appendix O: Test 5 – SIFT Matlab Code............130
Appendix P: Test 6 – Edge Detection in Image Segmentation Matlab Code............132

List of Figures

Figure 1: Plane Impact From Bird Strike (Loftus 2023) 17
Figure 2: Expected Flight Paths Using an OAS (Al-Kaff et al. 2017) 18
Figure 3: SURF Algorithm (Tyagi 2019) 19
Figure 4: ORB Feature Matching with Change in Orientation (Rublee et al. 2011) 20
Figure 5: YOLOv4 Model to Identify Birds (Mpouziotas et al. 2023) 21
Figure 6: Different Radar Signal Representations (Abdu et al. 2021) 22
Figure 7: 3D Pulley Model (1) 29
Figure 8: 3D Pulley Model (2) 29
Figure 9: Solenoid Lock (Element14 2024) 30
Figure 10: Table 4E4B of BS 7671 31
Figure 11: Pulley Assembly 32
Figure 12: IWR1642BOOST Hardware Configuration (Instruments 2020) 33
Figure 13: Block Diagram of IWR1642BOOST (Instruments 2020) 33
Figure 14: mmWave Demo Visualizer Console (1) 35
Figure 15: mmWave Demo Visualizer Console (2) 35
Figure 16: Perspex Protection for IWR1642BOOST 36
Figure 17: Polaris Ranger 37
Figure 18: mmWave Demo Visualizer Configuration Modifications 38
Figure 19: mmWave Demo Visualizer Real-Time Tuning Adjustment 38
Figure 20: Photograph of Plane Used for Experiment 39
Figure 21: Radar Positioning 40
Figure 22: IWR1642BOOST Cable Management 41
Figure 23: GoPro Secured to Plane 43
Figure 24: Test Footage to Verify Correct Position of GoPro 44
Figure 25: Plane Damage (1) 44
Figure 26: Plane Damage (2) 45
Figure 27: Plane Damage (3) 45
Figure 28: Flyover 1 (Top Left), Flyover 2 (Top Right), Flyover 3 (Bottom Left), Flyover 4 (Bottom Right) 47
Figure 29: Flyover 5 (Top Left), Flyover 6 (Top Right), Flyover 7 (Bottom Left), Flyover 8 (Bottom Right) 48
Figure 30: Flyover 9 49

Figure 31: Cropped Pulley Plate Test Image 51
Figure 32: Test Environment Image 51
Figure 33: Test 1 - ORB Pulley Plate Feature Points 52
Figure 34: Test 1 - ORB Plane Footage Feature Points..52
Figure 35: Test 1 - ORB Matched Points..52
Figure 36: Test 1 - SURF Pulley Plate Feature Points...53
Figure 37: Test 1 - SURF Plane Footage Feature Points..53
Figure 38: Test 1 - SURF Matched Points..53
Figure 39: Test 1 - SIFT Pulley Plate Feature Points 54
Figure 40: Test 1 - SIFT Plane Footage Feature Points..54
Figure 41: Test 1 - SIFT Matched Points..54
Figure 42: Test 2 - SURF Matched Points..56
Figure 43: Test 2 - SURF Matched Points..57
Figure 44: Test 3 - ORB Pulley Plate Feature Points (Small Cropped Frame) 58
Figure 45: Test 3 - ORB Pulley Plate Feature Points (Large Cropped Frame) 58
Figure 46: Test 3 - SURF Pulley Plate Feature Points...59
Figure 47: Test 3 - SURF Matched Points Frame 20534..59
Figure 48: Test 3 - SURF Matched Points Frame 20540..60
Figure 49: Test 3 - SURF Matched Points Frame 20543..60
Figure 50: Test 3 - SURF Matched Points Frame 20544..60
Figure 51: Test 3 - SURF Matched Points Frame 20546..61
Figure 52: Test 3 - SIFT Pulley Plate Feature Points (Small Cropped Frame) 61
Figure 53: Test 3 - SIFT Pulley Plate Feature Points (Large Cropped Frame) 61
Figure 54: Test 3 - SIFT Matched Points Frame 20541 62
Figure 55: Test 4 Original Frame Example 63
Figure 56: Test 4 Segmented Frame Example..64
Figure 57: Test 4 - SURF Matched Points Frame 20544..65
Figure 58: Clear Imported Image...66
Figure 59: Test 5 - SURF Matched Points Frame 20534..67
Figure 60: Test 5 - SURF Matched Points Frame 20534..67
Figure 61: Test 5 - SIFT Matched Points Frame 20538 67
Figure 62: Sobel Method 69
Figure 63: Canny Method...70

Figure 64: Original Image……………………………………………………………………71
Figure 65: Canny Edge Detection Method 71
Figure 66: Pulley Plate with Closed Edges..71
Figure 67: Pulley Plate with Filled Shapes 72
Figure 68: Pulley Plate with Small Blobs Removed..72
Figure 69: Final Pulley Plate Image with Joins Removed..72
Figure 70: Video Footage Frame with Closed Edges (1) 73
Figure 71: Video Footage Frame with Filled Shapes (1)..74
Figure 72: Video Footage Frame with Small Blobs Removed (2) 75
Figure 73: Video Footage Frame with Closed Edges (2) 76
Figure 74: Video Footage Frame with Filled Shapes (2)..77
Figure 75: Video Footage Frame with Small Blobs Removed (2) 78
Figure 76: Final Video Footage Frame with Joins Removed 79
Figure 77: Flyover 1 Radar Scatter Plot and Plane Footage...82
Figure 78: Flyover 2 Radar Scatter Plot and Plane Footage...82
Figure 79: Flyover 3 Radar Scatter Plot and Plane Footage...83
Figure 80: Flyover 4 Radar Scatter Plot and Plane Footage...83
Figure 81: Flyover 5 Radar Scatter Plot and Plane Footage...84
Figure 82: Flyover 6 Radar Scatter Plot and Plane Footage...84
Figure 83: Flyover 7 Radar Scatter Plot and Plane Footage...85
Figure 84: Flyover 8 Radar Scatter Plot and Plane Footage...85
Figure 85: Flyover 9 Radar Scatter Plot and Plane Footage...86
Figure 86: Flyover 10 Radar Scatter Plot and Plane Footage...86
Figure 87: Foreign Debris on Perspex After Tests 87
Figure 88: Ground View of Test (1) 89
Figure 89: Ground View of Test (2) 90

1 INTRODUCTION

1.1 Aim

This research proposal aims to investigate if there is an OAS capable of being incorporated into a crop-dusting aircraft to minimise aircraft crashes from bird strikes. MMW radar technology and machine vision will be assessed to investigate if either, or a combination of both, technologies will be a feasible system to incorporate into crop dusting aircraft. This will remove the human element of identifying and trying to avoid a bird in a significantly short period of time by incorporating an automated OAS.

1.2 Background

Many experiments have been conducted using MMW radars or machine vision on UAVs, however; none have been completed in a manner that will require extremely rapid responses such as that required in this research proposal.

This project will utilise strategies from other experiments as a foundation to build on and will attempt to develop an OAS that can be incorporated into a crop-dusting plane.

1.3 Problems

There are many problems that will arise from this research proposal. The most obvious is an automated system that will either notify the pilot or override the pilot's control. This is clearly a large safety concern, and this project will only be successful if the false object identification rate is extremely low as the plane cannot falsely respond to an object that does not exist. The accuracy of the system will therefore need to be tested thoroughly. Due to the extremely rapid response of this system, the pilot may not be prepared for the automated manoeuvre of the aircraft and as such, the plane should have an inbuilt safety feature that will stabilise the aircraft after the rapid manoeuvre along with an override option if required. Otherwise, a simple alert system can be used.

There is also an option to incorporate this research project topic of an OAS into a UAV which will raise several more problems. These include developing a UAV capable of crop-dusting and the algorithm it will use to navigate an airspace. Due to the complexity of this proposed system, these problems will be neglected; it will be assumed that only the OAS system will be researched and that it can be incorporated into either a manned aircraft or UAV.

1.4 Objectives

The objectives will be broken down into the following:

Objective 1: Research

Data, such as news articles and other evidence will be gathered to support the statement that bird strikes are responsible for many plane crashes. This will reinforce the need for this research proposal.

Previous experiments will also be investigated as they are likely to build a foundation for this research project.

Objective 2: Experimental Data Collection

After the experiments have been clearly planned out, they must be conducted, and data must be collected. This will be completed by using a drone with an inbuilt camera and attaching a MMW radar to the drone.

Objective 3: Data Analysis

The resulting data from the experiments will then be analysed and modelled using the gathered video footage and MMW data. It is expected that MATLAB and/or OpenCV may be required to complete this.

Objective 4: Data Comparison / Feasibility Check

The data will then be compared, and it will be determined as to whether it is feasible to incorporate one or both methods into an OAS for a crop-dusting aircraft.

1.5 Outcomes & Benefits

The outcomes that are expected for this research project are as follows:

1. An improved understanding of whether MMW radar technology and/or machine vision can be used as an OAS in crop-dusting aircraft.
2. An improved understanding of the reliability of MMW radar technology and machine vision in an OAS.
3. An improved understanding of a rapid response OAS.

The benefits of these outcomes are capable of including:

1. Increased job safety for crop-duster pilots if incorporated into manned aircraft.

2. Increased safety systems fitted to crop-dusting aircraft.
3. Increased investment into aircraft OAS's.
4. Removing risk for crop-duster pilots if incorporated into a UAV.
5. Reducing environmental damage caused by aircraft crashes.
6. Minimising aircraft damage.

2 LITERATURE REVIEW

2.1 Established Knowledge

2.1.1 Evidence Supporting Need for Project Research

Bird strikes are all too common in the aviation industry and are increasingly dangerous in the profession of crop-dusting. In 2022, there were 1300 bird strikes reported to the Australian Transport Safety Bureau (Barton 2023). The most recent news article relating to a fatal bird strike from a crop-dusting aircraft was published on 29/6/2023 – less than a year ago at the time of writing. The below image was extracted from this article and has been included to demonstrate how serious bird strikes are and why this research project is so important.

Figure 1: Plane Impact From Bird Strike (Loftus 2023)

It is clear that there is a need for an OAS to be incorporated into crop-dusting aircraft to avoid fatalities like this. The mentioned reports stated that the pilot was flying 200kmh^{-1} at an elevation of just 2.5m while crop-dusting and that it struck a bird which commonly weighs between 3 and 12kg. It is emphasised that for this project to be successful, the OAS must be able to rapidly avoid objects or notify the pilot the moment they are detected.

2.1.2 OAS Using Machine Vision

There have been numerous projects which have investigated the incorporation of machine vision into aircraft systems. The following research papers have been published on topics closely relating to this:

1. Obstacle Avoidance System for UAVs Using Computer Vision (Richards et al. 2015)
2. A Survey on Unmanned Aerial Vehicle Collision Avoidance Systems (Pham et al. 2015)
3. Colanet: A UAV Collision Avoidance Dataset (Pedro et al. 2020)

4. UAV Obstacle Avoidance Algorithm Based on Ellipsoid Geometry (Sasongko, Rawikara & Tampubolon 2017)
5. Obstacle Detection and Avoidance System Based on Monocular Camera and Size Expansion Algorithm for UAVs (Al-Kaff et al. 2017)
6. UAV Environmental Perception and Autonomous Obstacle Avoidance: A Deep Learning and Depth Camera Combined Solution (Wang et al. 2020)

These six papers all present entirely unique solutions to incorporating machine vision into an aircraft. The most interesting research paper is that of the ellipsoid geometry object avoidance algorithm, this focuses on creating an elliptical path to avoid a detected object. The experiments conducted in this paper may be utilised in developing the OAS using machine vision as it would provide a smooth trajectory if it was a manned aircraft. A problem that could arise from using this method, however; is that there may not be enough time to smoothly avoid an object if it was to emerge from crops at close range.

The A* algorithm has been most commonly used in machine vision of OAS's but it presents its own obstacles such as restricted heading angles which makes it difficult to overcome 3D path planning manoeuvres (Choi, Jimenez & Mavris 2017). The mentioned paper has been published which introduces the use of a Beta* algorithm that overcomes this issue. It is likely that the OAS using machine vision will use one of the mentioned algorithms, however; the Beta* algorithm is mostly developed for urban environments and may not be useful in the project.

The OAS would be designed such that would avoid obstacles in a similar manner to that presented below with the exception that it should also fly above the object if required. Due to the altitude of the plane, it would be deemed impractical to fly under the object as the aircraft would collide with the aircraft.

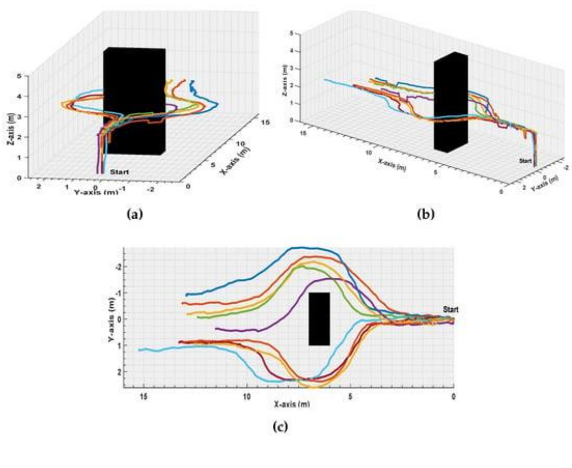


Figure 2: Expected Flight Paths Using an OAS (Al-Kaff et al. 2017)

To simplify this problem, further articles have been researched that present algorithms without the incorporation of a UAV. There are numerous algorithms that could be incorporated into this research task. Such algorithms can include optical flow, SURF, SIFT and ORB.

Optical Flow

Optical flow is typically used as an algorithm for a video or images with small time steps. This technique describes motion in an image by calculating velocity from points in the images (*Computer Vision Demonstration Website* 2005). The main advantages of optical flow is that there is less error due to corner detection across larger flows and has good noise tolerance typically; disadvantages include that it requires high computational requirements and is quite slow (Raju & Joseph 2014). These issues can be addressed by using a different algorithm method. There are many different methods that can be utilized for this algorithm that have been developed by programmers. Such methods can include Lucas-Kanade, Horn-Schunck, FlowNet, Buxton-Buxton and Sparse. If optical flow is a chosen algorithm, the Sparse method will likely be chosen due to it being most effective for high frame rate which will be the case due to the high velocity of the aircraft.

SURF

SURF is an acronym for 'Speeded Up Robust Features' which looks for similarities in multiple images. It uses a box filter such as that shown below to track and recognize objects.

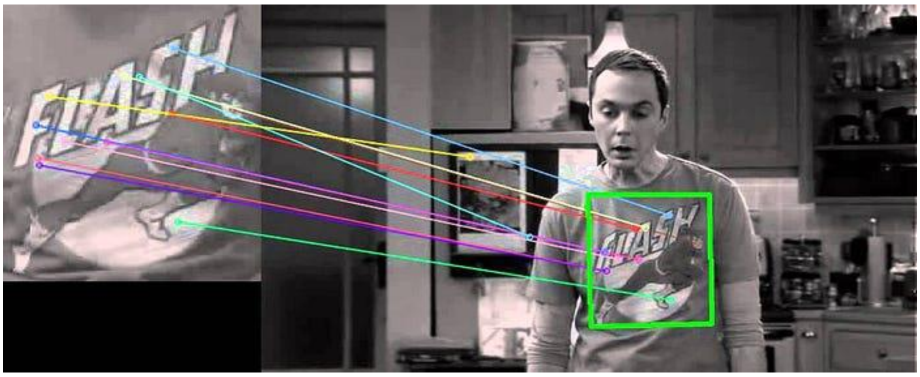


Figure 3: SURF Algorithm (Tyagi 2019)

SURF is composed of feature extraction and feature description. Feature extraction uses a simple 'Hessian matrix' approximation to determine unique parts in an image. Feature description works by initially using information from a circular region around the reference point to develop a reproducible orientation of the image. It then creates a square which surrounds the selected orientation to extract the feature of the image (Tyagi 2019). The SURF algorithm can be computed and compared at an incredibly fast speed. One research paper found that SURF outperformed most algorithms in both accuracy and speed (Bay, Tuytelaars & Van Gool 2006).

SIFT

SIFT is an acronym for 'Scale Invariant Feature Transform' which is invariance to rotation and scale of images. There have been many applications for SIFT which include machine vision, image retrieval and image stitching. It can identify images with blur change, illumination change, scale/rotation change and affine change. There are numerous algorithm methods that can be used which all have their own advantages and disadvantages. This includes SIFT, CSIFT, GSIFT, ASIFT and PCA-SIFT. Out of all the tests conducted, PCS_SIFT performed second best in every benchmark test including identification of images with the following changes: blur, illumination, scale/rotation and affine (Wu et al. 2013). Due to this, PCA-SIFT would be chosen if the SIFT algorithm was used.

ORB

ORB is an acronym for 'Oriented FAST and Rotated BRIEF' which is a rapid speed binary descriptor that is resistant to noise and is rotation invariant. This algorithm was developed to be an alternative to SURF and SIFT due to SURF and SIFT being patented algorithms. It has been proven to be almost two times faster than SIFT. ORB required low computing power and can be incorporated into video processing in real time (Rublee et al. 2011). It is efficient in identifying similarities in images with different orientations as shown below making it a viable option for an object identification system.

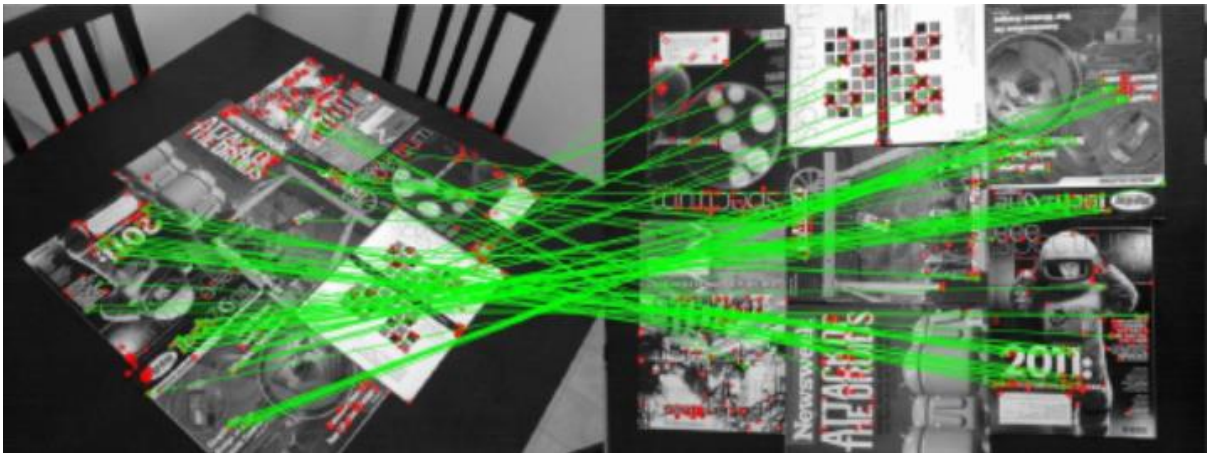


Figure 4: ORB Feature Matching with Change in Orientation (Rublee et al. 2011)

Additional Research

Due to the fact that bird strike avoidance is the primary objective of this paper, current research on bird detection should be reviewed. One such paper utilised the YOLOv4 model to identify bird species. This paper utilised a drone to fly above birds and was able to identify the bird species with an accuracy of 91.28%. The authors utilised the following methodology: Darknet framework, DarkHelp C++ API/CLI Tool, DarkMark GUI Application, Image Tiling, Image Enhancement and finally the YOLOv4 detection algorithm to give such accurate results. An extract from this paper identifying the birds is presented below.

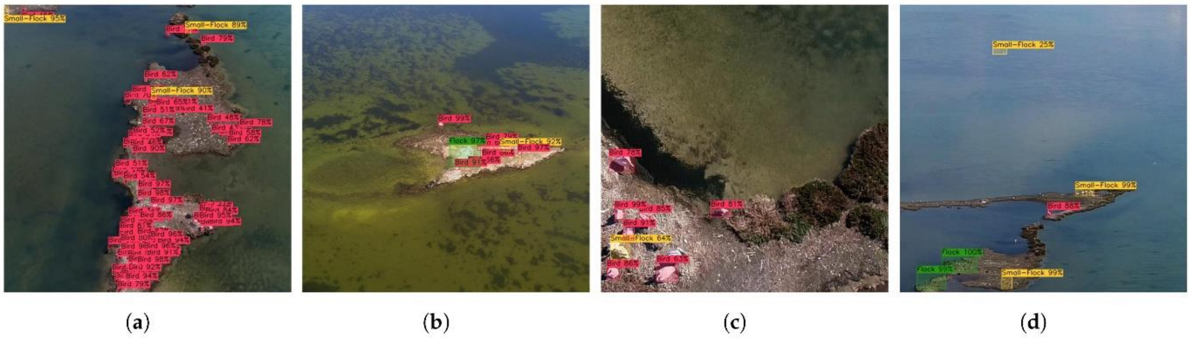


Figure 5: YOLOv4 Model to Identify Birds (Mpouziotas et al. 2023)

2.1.3 OAS Using MMW Radar

There are numerous applications where MMW radars are used. They are extremely practical for OAS design due to their ability to measure small movements at a rapid pace. The below papers have been published which includes research that is closely related to this application.

1. Application of Deep Learning on Millimeter-Wave Radar Signals: A Review (Abdu et al. 2021)
2. A Gaussian Process Model for UAV Localization Using Millimetre Wave Radar (Paredes et al. 2021)
3. The improved A* Obstacle Avoidance Algorithm for the Plant Protection UAV with Millimeter Wave Radar and Monocular Camera Data Fusion (Huang et al. 2021)
4. Unified Calibration Method for Millimeter-Wave Radar and Machine Vision (Junlong & Feng 2018)

There has been little research completed in singular MMW radar systems as most research has been completed along with that of machine vision. This benefits the proposed research project as it shows there is successful papers written on OAS using a combination of both MMW radars and machine vision.

The MMW radar is capable of mapping out terrain at rapid speeds and has been implemented into a number of UAVs which is fundamental for the inbuilt collision avoidance system (Connolly 2007). To conduct the experiment using MMW radars of this project, the environment will be mapped, and it will be determined whether the MMW radar is able to detect an object protruding from a crop field.

These articles identify that there are numerous radar data representations; this can be seen in the figure below. These different radar representations will need to be investigated during testing to determine which would be most efficient to incorporate in the proposed OAS.

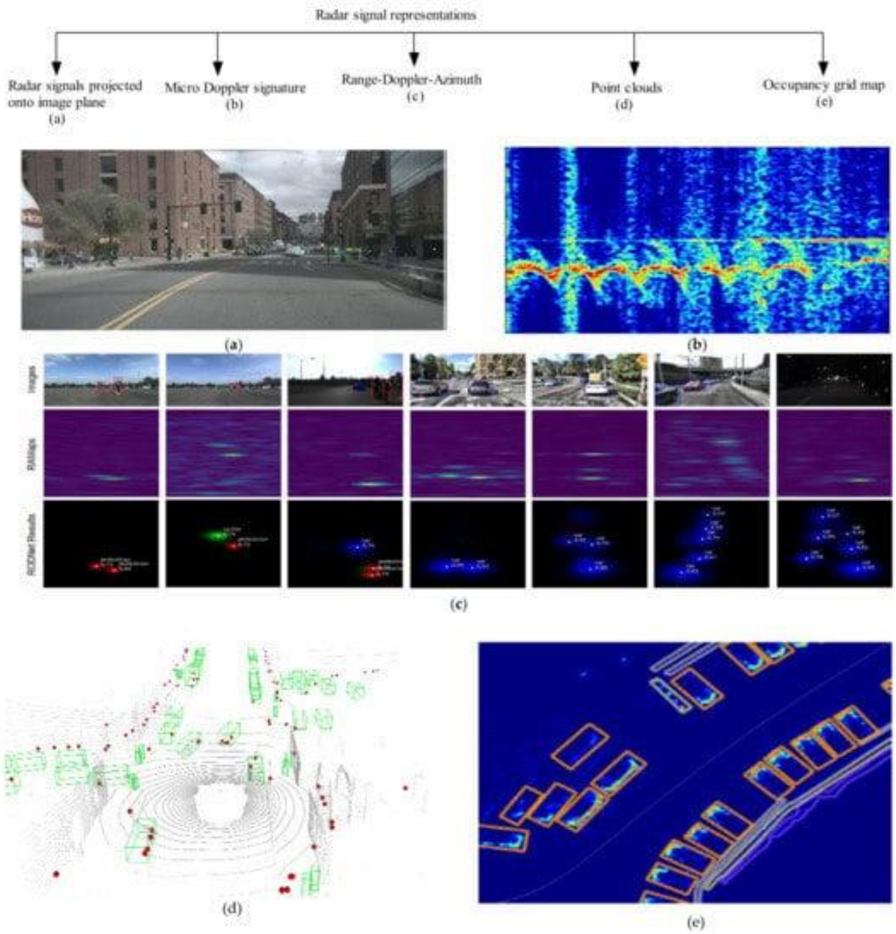


Figure 6: Different Radar Signal Representations (Abdu et al. 2021)

2.2 Knowledge Gap

2.2.1 Study Justification

There are some studies that have been undertaken in developing a general OAS for UAVs but there is a clear lack of study committed to this unique field of developing an OAS for a crop-dusting application. This system will be unique in the sense that it requires a rapid response system due to the high speeds of the aircraft and extremely low altitude the aircraft will be flown at. It is possible that not much research has been dedicated to a system like this due to the technologies being relatively new. This research topic will determine if an OAS is capable of being incorporated into a crop-dusting aircraft and if it is feasible.

2.2.2 Project Feasibility Analysis

The initial goal of the project was to develop a crop-dusting UAV with an inbuilt OAS. After conducting research, it was determined that this was not feasible to complete in one academic university year due to the complexity of the system. This project has then been broken down such that it only focuses on developing an OAS with the assumption that it can easily be incorporated into a UAV or manned aircraft. After conducting the literature review, it appears feasible to conduct research of this topic in the allocated timeframe. The cost estimates shown later in this proposal will also show that it is cost effective to conduct this research due to the MMW radar being made available from the engineering facility. It is expected that the data analysis will take the most time to complete due to different programming requirements for machine vision and MMW radar data.

3 METHODOLOGY

3.1 Project Parameters

3.1.1 Scope

This research project will only require one experiment to gather the data. The data will be collected by recreating a typical situation in which a bird would fly out of the crops. Instead of a bird, a metal object will rise from the crop field utilising an automated pulley system. A camera and MMW radar will be mounted to a crop-duster and data will be collected during a live crop-dusting application. The exact process in which the data will be collected will be outlined in the 'Project Experiments' section of this paper. Once the data has been collected, it will be analysed using computed programs. The conclusions reached from this data analysis will be vital in arriving at a definitive conclusion to this research proposal.

3.1.2 Limitations

The most prominent limitation is the timeframe in which this research is to be completed. Due to the fact that less than one academic year has been strictly allocated to complete this project, it will prove that efficient planning and time management is absolutely frugal. The additional pressure of having to both work full time and complete other subjects alongside this project will reduce the time allocated for this project significantly. It will be required that the project schedule is always adhered to, or it may result in failing to reach a conclusion of the project. The strict timeframe also limits the ability to incorporate other sensors. Research papers that have had extremely high success rates have been able to incorporate an abundance of sensors including, but not limited to, SONAR, infrared devices, ultrasonic devices and doppler radars (Viquerat et al. 2008). Due to the time restrictions and only one person conducting the research, only two sensors have been investigated in this paper.

The costs associated with this project will also be a significant limitation. Fuel costs associated with flying an actual plane are incredibly high and as such, the data collection must be thoroughly planned out and executed as there will only be one attempt made to collect the required data. It has been organised to conduct the testing of the experiments at Aero Professional Services based in Emerald due to positive past work relations. The cost of travelling to Emerald and staying there will also need to be considered. The faculty of engineering at UniSQ has a MMW radar which will be used for the experiment. This is already obtained and as such will not need to be included in the budget. It is likely that the student will be outlaying a significant cost on this project and as such, budgeting practices will need to be implemented to ensure all equipment is ready for the research.

A further limitation will be that of the data collected. Due to the given timeframe and the distance from the student to the test field (approx. 900km) the data collected will be limited. The student will gather data at the time proposed in the project schedule. It is currently not clear what crops will be grown and how mature the crops are. This will provide only a narrow representation of the different crops and crop sizes that crop-dusters regularly spray. This will result in a general data collection to work with. It is also not yet known what the weather will be like. Due to this experiment requiring dry conditions, it may result in the project schedule needing to be adjusted. A review of the typical wet months will be conducted prior to scheduling when the data collection will occur.

3.2 Project Experiments

The experiments will be conducted in a realistic environment during live operations of a crop-dusting aircraft. The data will be collected in a typical crop field with the hope that the crop will be close to maturing so that the pulley system is not visible. To accurately represent the different conditions crop-dusters are subjected to, it would be ideal to have this data collected at different times of the day to impose environmental factors such as shadows, glare from sunlight etc. This would make the experiment as realistic as possible which improves the accuracy of the data; however, due to the significant costs in flying a plane, the tests will be conducted at during a single flight.

The experiment will be completed by using a metal plate, which will be mounted to a pulley system located in a crop field. The pulley will be designed to meet safety ratings outlined in the risk assessment. The pulley will also be designed with an electronic latch such that when a switch is toggled, the latch will be released, and the metal plate will raise out of the crops as if to represent a bird flying upwards. The crop-duster will have a GoPro and the MMW sensor mounted to it. Both the GoPro and the MMW radar will be mounted on the underside of the plane. The GoPro will be standalone running off its own battery, whereas the MMW radar will need a 9V power bank to power it and a laptop to communicate with. The power bank and laptop will be in the cockpit of the aircraft with cables run securely from the sensor to the cockpit. When the plane flies, the direction of travel will be in line with where the pulley is located. As the plane approaches the metal plate, the pulley will be activated, and the plate will raise out of the crop. If permitted, numerous attempts will be made to gather as much data as possible.

The data will then be analysed using computer programs to provide results on the OAS. The expected computer programs that may be used include MATLAB, OpenCV and a radar toolbox from Texas instruments, but this will be reviewed when the data has been collected.

The MMW radar data will be the first dataset to be analysed. Due to the MMW radar being manufactured by Texas Instruments, it is obvious that the radar toolbox published by Texas Instruments will be used. MMW visualiser is a Texas Instruments program used to give live data feedback. The data from this application will be screen recorded during the testing phase so that the results can be analysed significantly easier.

Object identification using machine vision will then be analysed. To accurately analyse the data that will be collected from the camera footage in the experimental phase, a minimum of two algorithms researched in the previous chapter will be implemented. This footage will already be captured at the time that the algorithms are implemented and as such, real time processing will not be advantageous. From the previously mentioned algorithms, SURF, SIFT and ORB will be utilized to analyse the data. It is possible that edge detection may also be used given that the plate will have distinct edges.

This will likely be completed with either MATLAB or OpenCV. OpenCV has been used in many of the studies in the literature review and is one of the most common programs used for machine vision (Pulli et al. 2012). According to the official OpenCV site, it also supports machine learning (OpenCV 2023) which may prove useful in automated object identification. A main reason why OpenCV may not be chosen is because it is most used for real time video processing, and it requires a high-level understanding of C++. MATLAB is a more viable option as it has all packages required for the machine vision algorithms.

If the analysed data shows that both the machine vision and MMW radar have been successful at identifying the object, both sensors will be combined to make a more reliable OAS. The results of both sensors will be analysed to develop a more stable system.

The system will only be successful if there is a small margin of error, and it can successfully identify an object from all tests with a low false alarm rate.

It is important to note that the student has not had experience programming in OpenCV or through the Texas Instruments radar toolbox. To ensure that this can be completed, the student will allocate time before the commencement of semester 1 in 2024 to become familiar with these programs. The project schedule will also provide an abundance of time for data analysis as this will likely take the most time when the project is being undertaken.

3.3 Quality Assurance

There are numerous measures that will be completed to ensure the experimental data is as accurate as possible. These are listed below:

1. Conduct testing and calibration of MMW radar with pulley system using a ground vehicle initially to increase accuracy when mounted to the plane.
2. Activate Pulley when plane is located at different distances to give more realistic data.
3. Only operate plane when it is not raining, or it may damage the hardware.
4. Battery levels of power bank and laptop will be closely monitored to ensure data is recorded accurately.
5. Perspex lens (if mounted to MMW radar) will be wiped prior to tests being conducted.
6. A minimum of 10 tests will be conducted.
7. Ensure the plane is manoeuvring as it typically would in a crop-dusting situation.

3.4 Research Questions

This paper will aim to answer the following questions.

1. Will the MMW data accurately detect the required object and if so, will it provide a high accuracy?
2. What machine vision algorithm will provide the best results?
3. Is it feasible to incorporate MMW radar and machine vision into a crop-dusting aircraft?

4 SYSTEM DESIGN

4.1 Overview

This section will detail the steps taken to design the OAS. The component selection, initial designs, calibration and tests will be discussed. This was then used to assemble and manufacture the final products that were used for the field testing. When the pulley system was designed, it allowed for realistic tests to be conducted and more accurate calibrations of the MMW radar.

4.2 Component Selection

As detailed in Table 2: Cost Estimate & Resource Requirements, there were numerous parts that were required for the successful completion of this project; these will now be discussed.

4.2.1 Pulley System

The pulley system was required to be situated in a field of crops. There were numerous designs that were investigated for this system, ranging from a vertically actuated plate, manually pivoting a test piece out of the crops and a catapult style system. Due to the nature of the low flying plane, it was required that the pulley had to be autonomous to some extent to mitigate the risk of a person being harmed. A catapult style design was ultimately chosen to be investigated as it was the simplest system to design.

The catapult style pulley was designed in such a way that a large bar could pivot on a circular rod. On one end of the bar, a large metal plate was mounted which would pop up when activated and simulate a foreign object raising out of the crops. To stop the plate from overextending, a stopper bar was added so that the plate would stop at a vertical point.

It was decided that the easiest was to assemble this pulley would be to weld and fabricate the frame. Aluminium was the initial material chosen because of the lightweight attributes but due to the difficulty of welding aluminium, steel was chosen instead. To increase the chance of collecting data from the MMW radar, a reflective metal plate made of stainless steel was used.

The height of the pulley was entirely determined by the height of the crops at the time of field testing. It was not practical to design a pulley that stood out of the crops before the plate rose otherwise the pulley itself could be detected. Likewise, it was not practical to design a pulley that was not high enough to protrude from the crops when the plate rose. The crops were measured at the start of the month of testing and were approximately 1.2m high. The top of the pulley was therefore designed to be 1.2m in height. A 3D model of this pulley was generated and can be seen below.

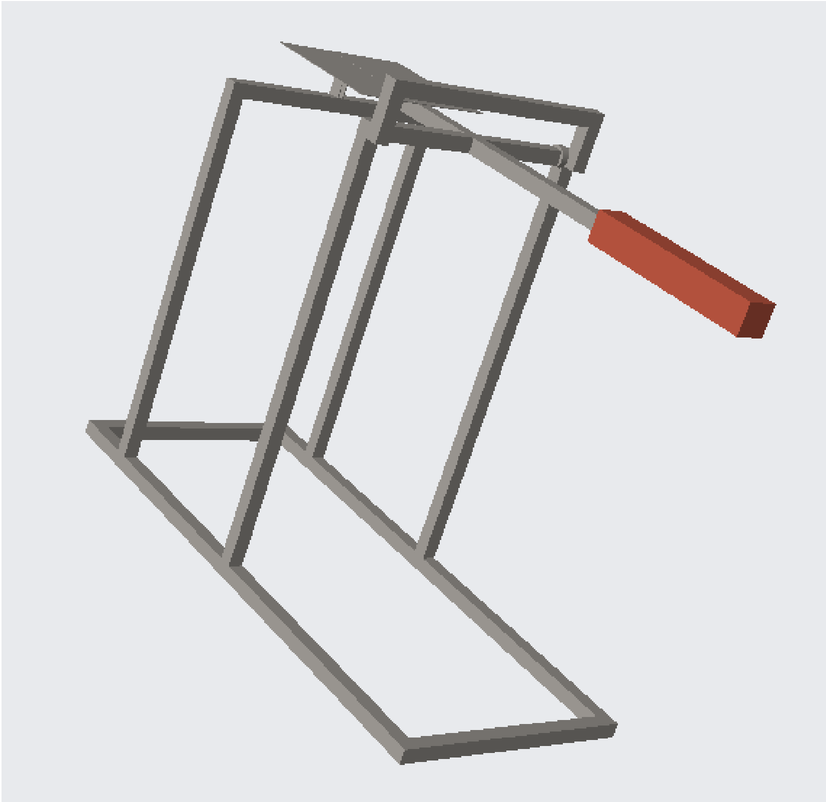


Figure 7: 3D Pulley Model (1)

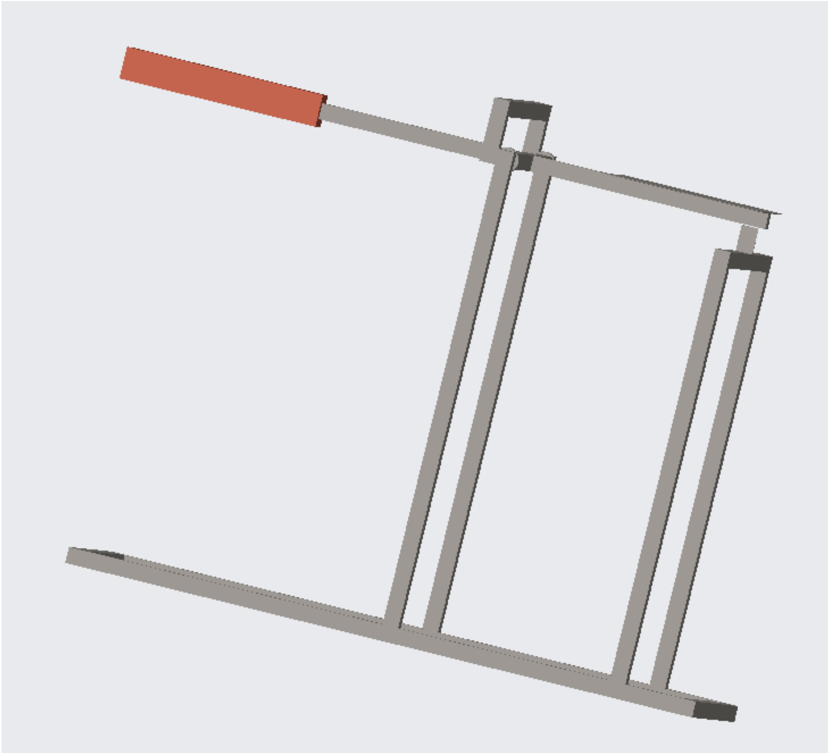


Figure 8: 3D Pulley Model (2)

To make the pulley somewhat autonomous, an electro-mechanical element was needed to be incorporated into the design. The initial design utilised a motor and a draw-wire which latched onto on eyelet on the bar so that when the motor was activated, the plate would be raised into the air. It became apparent that there was a simpler design that could be incorporated, which ultimately became the final design. It involved using a counterweight on the end of the bar opposing the metal plate and a latching switch.

The latching switch was chosen so that when it was activated, the counterweight would drop and raise the metal plate. There were numerous latching switches to choose from and it was decided that a 12V DC switch would be used so that it could easily operate from a 12V DC car battery. The chosen switch can be seen below.

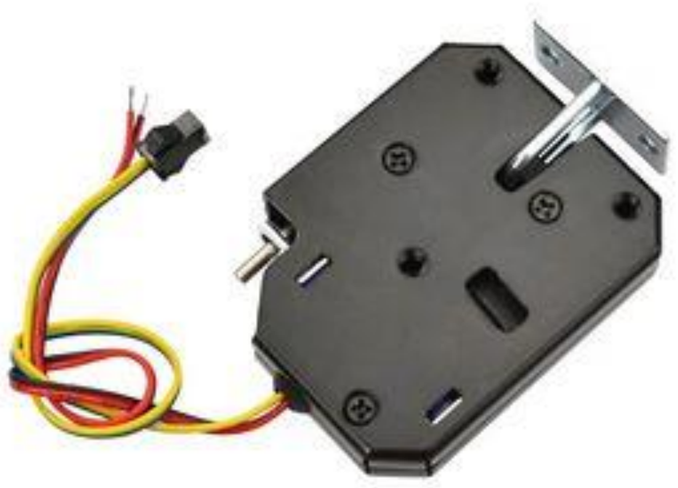


Figure 9: Solenoid Lock (Element14 2024)

Specifications:

- Operation Voltage: 9V ~ 12V DC
- Operating Current: 1.5A
- Stress Range: ≤735N (75kg)
- Lock Body Dimension: 73×58×13.3mm

The latch was capable of being triggered using an Arduino code but to simplify the system, it was decided that it would be triggered by a manually operated switch. To minimise risks, a 30m twin cable was used which joined to the positive and negative cable of the latch and ran through the crops to a 12V battery. The positive cable joined into a switch which was located near the battery so that when it was activated, the plate would raise with the operator being 30m away. The voltage drop had to be calculated to select the right cable size due to the distance being 30m. This was completed using the below table and formula which is extracted from table 4E4B of BS 7671.

Conductor cross-sectional area	Two-core cable d.c.	Two-core cable, single-phase a.c.			Three- or four-core cable, three-phase a.c.		
1	2	3			4		
(mm^2)	(mV/A/m)	(mV/A/m)			(mV/A/m)		
1.5	31	31			27		
2.5	19	19			16		
4	12	12			10		
6	7.9	7.9			6.8		
10	4.7	4.7			4.0		
16	2.9	2.9			2.5		
		r	x	z	r	x	z
25	1.85	1.85	0.160	1.90	1.60	0.140	1.65
35	1.35	1.35	0.155	1.35	1.15	0.135	1.15
50	0.98	0.99	0.155	1.00	0.86	0.135	0.87
70	0.67	0.67	0.150	0.69	0.59	0.130	0.60
95	0.49	0.50	0.150	0.52	0.43	0.130	0.45
120	0.39	0.40	0.145	0.42	0.34	0.130	0.37
150	0.31	0.32	0.145	0.35	0.28	0.125	0.30
185	0.25	0.26	0.145	0.29	0.22	0.125	0.26
240	0.195	0.20	0.140	0.24	0.175	0.125	0.21
300	0.155	0.16	0.140	0.21	0.140	0.120	0.185
400	0.120	0.13	0.140	0.190	0.115	0.120	0.165

Figure 10: Table 4E4B of BS 7671

$$Voltage\ Drop = \frac{(mV/A/m)z \times L \times Ib}{1000}$$

Using this formula, it was determined that a cable with a CSA of $4mm^2$ was sufficient to power the latch successfully.

The pulley was then built and tested; the final build can be seen below.

Figure 11: Pulley Assembly

The pulley was then tested extensively by connecting the cable to a 12V lead acid battery and activating the switch. The pulley would be reset by manually rotating the free moving bar so that the eyelet would click back into the latch.

4.2.2 IWR1642BOOST Millimeter-Wave Radar

The MMW radar was required to be mounted on the aircraft. Due to the crop-duster reaching speeds of up to 320kmh^{-1} it was critical that the radar was calibrated correctly and firmly secured to the aircraft. Extensive testing was conducted on the MMW radar to increase the prospect that it would collect the data required.

Upon receiving the MMW radar from UniSQ, the radar setup commenced. The first step was to understand the product specifications of the IWR1642BOOST. Below are some images extracted from the IWR1642BOOST user guide which present the hardware configuration.

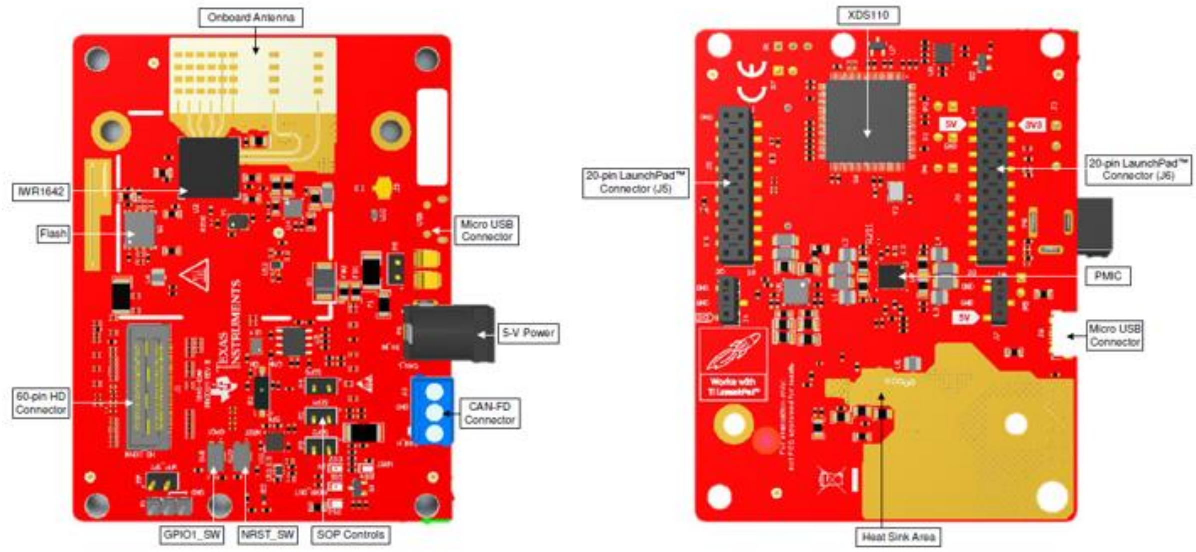


Figure 12: IWR1642BOOST Hardware Configuration (Instruments 2020)

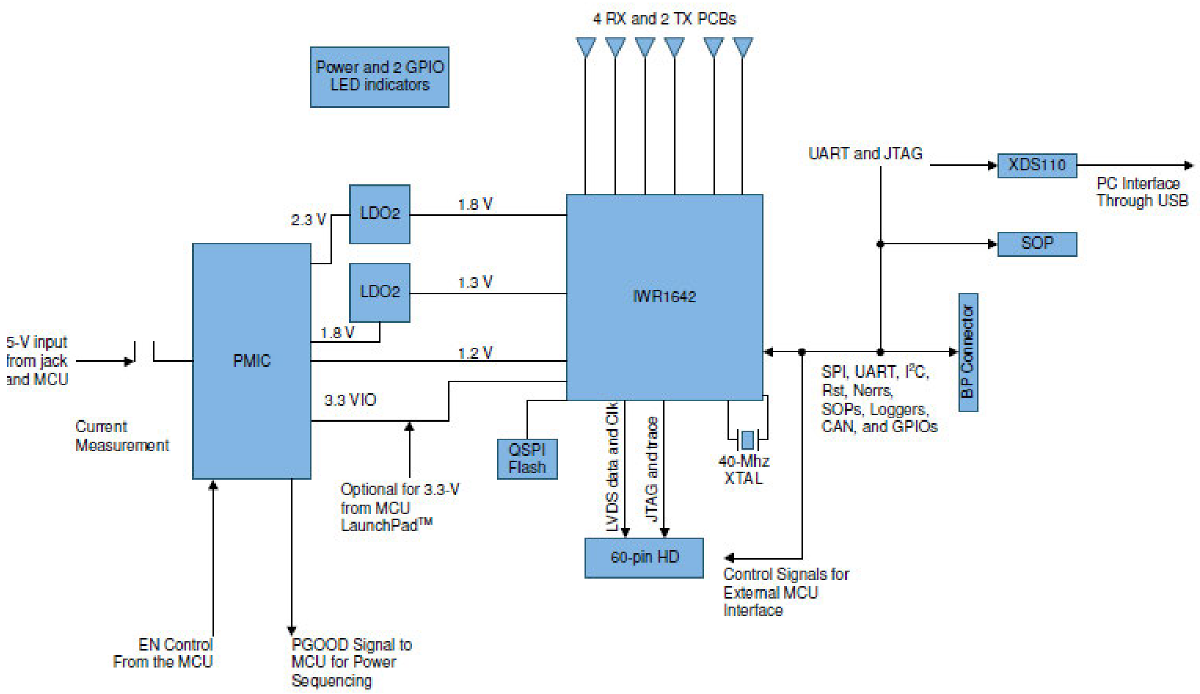


Figure 13: Block Diagram of IWR1642BOOST (Instruments 2020)

Specifications:

- Number of transceivers: 4
- Number of transmitters: 2
- Frequency range: 76 – 81 GHz
- Power requirements: 5V, >2.5A
- Total weight: 246g
- Dimensions: 84 x 66 x 17mm

To operate the MMW radar, a 5V, 3A power supply and micro-USB cable was required.

The next step was to download the required software to operate the radar. The software was downloaded from the Texas Instruments website and included:

- **UniFlash**: A software tool to program on-chip flash on the IWR1642BOOST mm-wave radar. It was used to allow the microcontroller to be reset and allow programs to be uploaded to the radar.
- **Silicon Labs CP210x USB to UART Bridge VCP Drivers**: A SiLabs driver which allowed the radar to communicate with the laptop.
- **TI Radar Toolbox**: A toolbox which allowed access to binaries and source code.
- **Code Compressor Studio**: An IDE for the IWR1642BOOST which was used to develop and debug applications.
- **MMWave Demo Visualizer 3.6.0**: A gallery app which allowed for real time configuring of the radar and a visualizer of the data being collected from the radar.
- **MMWAVE – SDK**: A collection of software packages which allow for development and application development of the IWR1642BOOST.

The IWR1642BOOST was connected to a laptop via a micro-USB cable and was powered by the 5V power supply. The Silicon Labs driver was first initialised to allow the laptop to communicate with the radar. The ‘Out of Box’ demo was then utilized from the TI radar toolbox, which included a sample code that was modified slightly and ultimately utilized for data collection. The radar was initially configured for flashing mode which required a jumper to be connected to SPO0 and SPO2. The out of box demo binary was then flashed onto the radar using UniFlash. The radar was then configured for functional mode which required the SPO2 jumper to be removed from the radar. The mmWave demo visualizer was then opened and the correct device and SDK version was selected. The SDK version of the radar would be displayed in the script box on the visualizer when the SW2 button was pressed on the radar. The serial ports were then configured such that the CFG_port and DATA_port was connected to the appropriate serial port. The config could then be sent to the mm-wave radar and the live data was then visualised in the ‘plots tab’. A preview of what the mmWave Demo Visualizer looks like is displayed below.

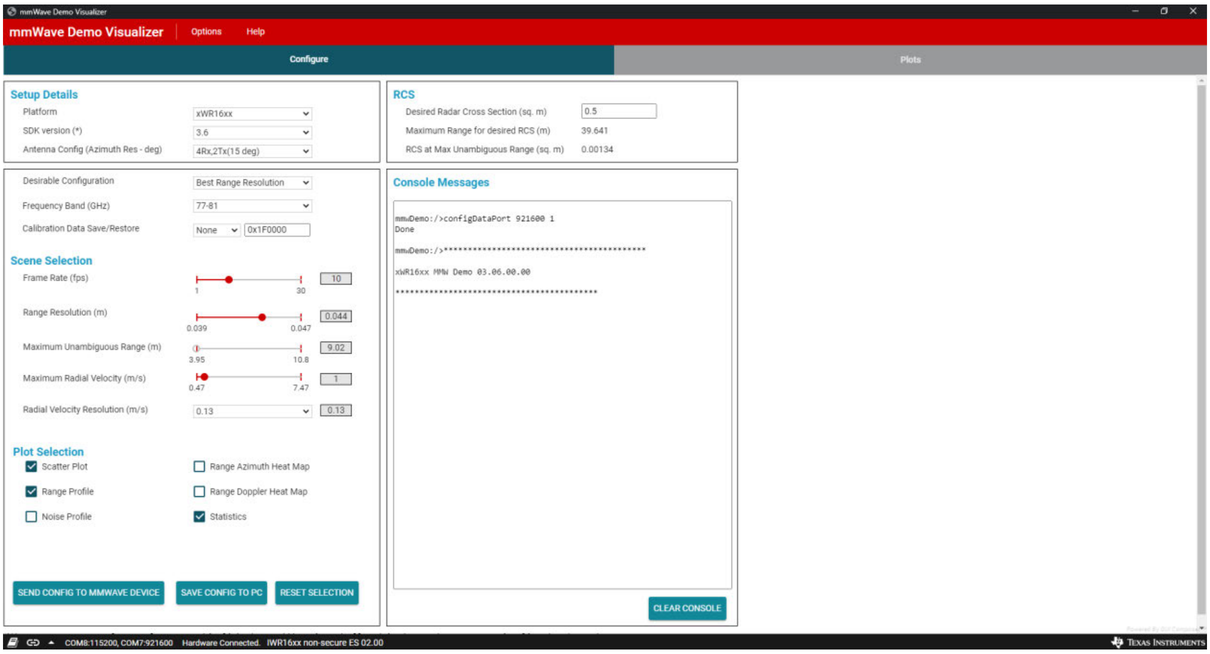


Figure 14: mmWave Demo Visualizer Console (1)

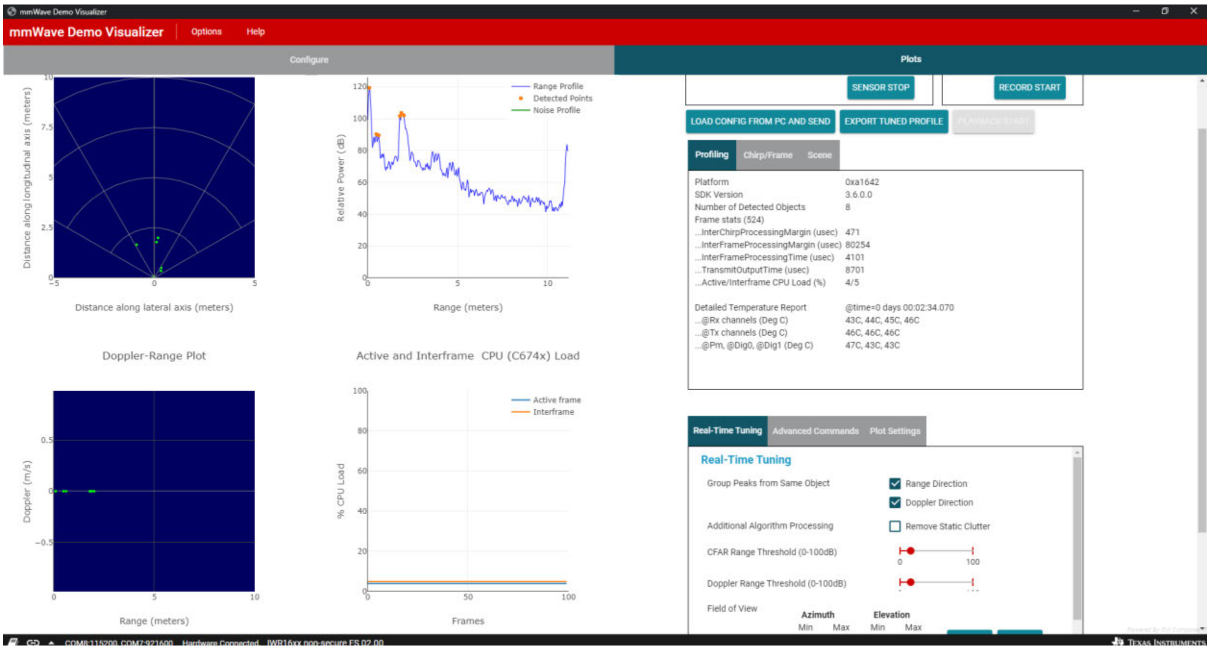


Figure 15: mmWave Demo Visualizer Console (2)

Upon successful connection to the radar, tests were conducted. The first test was to simply position the radar upright and move around the room to confirm that it was detecting and mapping an object. The next test was to take the radar outside and move away from the radar until it stopped detecting the object. This distance was determined to be roughly 12m. After conducting research on the radar, it was identified that reflective objects were detected more easily and accurately. Due to this information, a large reflective plate made of stainless steel with dimensions of 600 x 600mm was used for testing. In close range, the radar plot would be very inaccurate as it would show green dots flickering across the plot sporadically. It would do this for distances of up to 5m. It did, however, reach distances of up to 25m which is roughly double that of a non-reflective object. It was decided that a reflective object would be utilized to maximize the potential of obtaining data to be analysed.

It was determined that the radar might get damaged by foreign objects such as bugs whilst the plane was in mid-flight. To navigate this issue, a sheet of Perspex was cut and mounted onto the device. The below image illustrates how this was done.

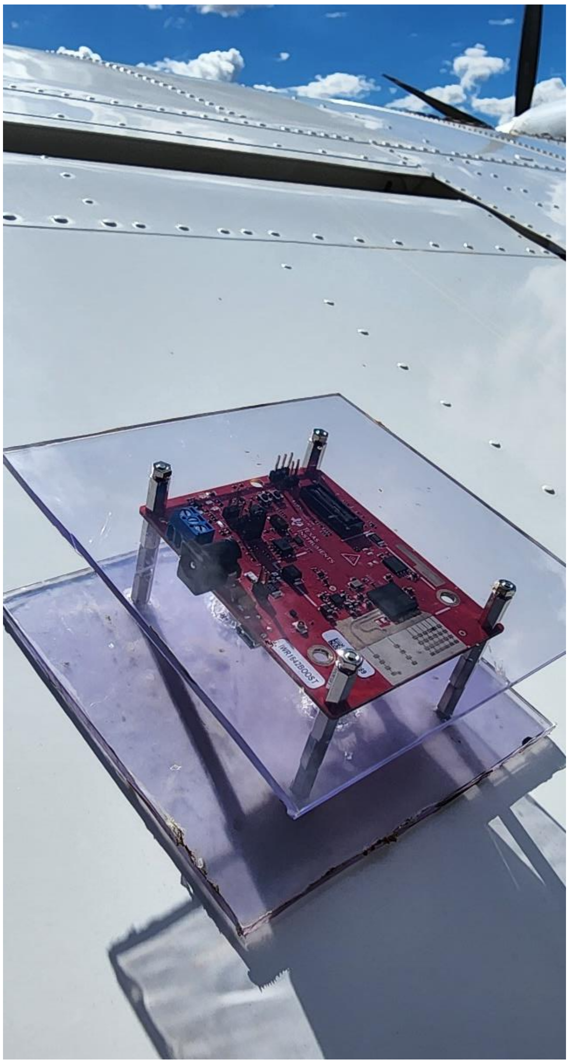


Figure 16: Perspex Protection for IWR1642BOOST

The Perspex was mounted to the radar using a PCB spacer kit. The number of spacers used will be discussed later in this section. As a result of the added Perspex, more tests were required to be conducted. Using the same method as above, it was found that with the Perspex, the range of the radar accurately reached 20m; 5m less than the total range without the Perspex. This range was more than sufficient for the purpose of this experiment.

At this point, all the testing was done with a stationary radar and a slow-moving object. To test the radar accurately, data needed to be collected so that the radar was moving rapidly with the assembled pulley being the test piece. To accomplish this, a Polaris Ranger was used which reached speeds of 80kmh^{-1}. An additionally person was utilized as the driver whilst the researcher sat in the passenger seat holding the radar and the laptop as the Polaris Ranger drove past the pulley. The vehicle used can be seen below.

Figure 17: Polaris Ranger

The testing was conducted such that the pulley system was already activated and in the upright position to reduce timing errors and to ensure the calibration results were accurate. The buggy passed the pulley system at maximum speed twelve times. During these tests there were many modifications to both the configuration and ‘Real-Time Tuning’ of the mmWave Demo Visualizer which were conducted.

The first modification was to allow for the velocity change, as the plane would be flying at speeds of up to 320kmh^{-1}. After the twelve tests, the following modifications were made:

- The ‘Maximum Unambiguous Range (m)’ was set to 10.8,
- The ‘Maximum Radial Velocity (m/s)’ was set to 7.47, and
- The ‘Radial Velocity Resolution (m/s)’ was set to 0.94.

The ‘Frame Rate (fps)’ and ‘Range Resolution (m)’ were also adjusted during the tests to find the best results. It was determined that the frame rate would stay the same, as a higher frame rate would draw too much power, and the radar temperature increased considerably; this caused the radar to crash. Too low a frame rate resulted in the radar not detecting the pulley. The range resolution resulted in an inverse impact on the minimum and maximum levels of

'Maximum Unambiguous Range (m)' and 'Maximum Radial Velocity (m/s)', respectively. A higher resolution resulted in the maximum range to decrease and the velocity to increase. A maximum range of 10.8m was chosen as the maximum range for the device when detecting from the aircraft.

These changes are depicted below.

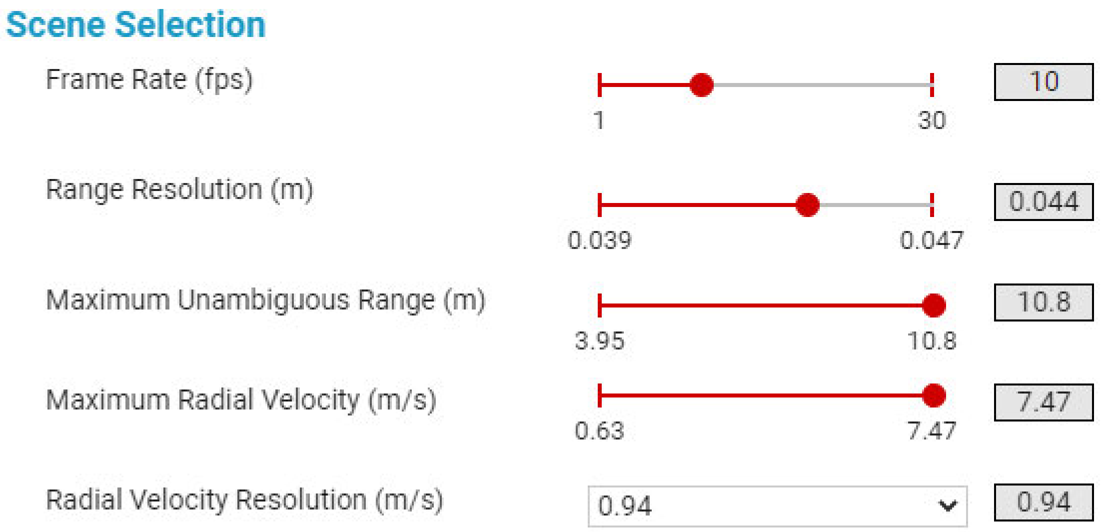


Figure 18: mmWave Demo Visualizer Configuration Modifications

Tests were performed and some adjustments were made in the 'Real-Time Tuning' section of the visualizer. The modifications were made to reduce background interference on the scatter plot. This included increasing the 'Doppler Range Threshold (0-100dB)' from 14 to 21 as seen below.

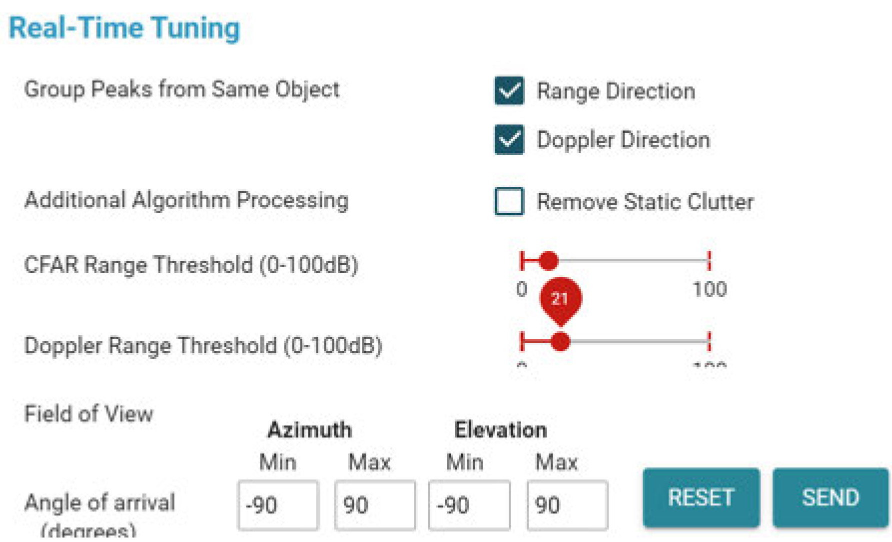


Figure 19: mmWave Demo Visualizer Real-Time Tuning Adjustment

After these modifications were made, an additional ten tests were conducted. The results demonstrated a 100% success rate in detecting the pulley, and therefore, the above configuration and tuning was utilized for the final data collection phase.

The final step in the setup was to investigate where to mount the radar on the plane without modifying the structural integrity of the aircraft (i.e., drilling). A photograph of the plane can be seen below.

Figure 20: Photograph of Plane Used for Experiment

After careful consideration, it was decided that the spray rail would be the best place to mount the radar. To accomplish this, a large threaded U bolt was used. The rail was 60 x 30 mm in diameter so a U bolt with dimensions 95 x 34 mm and M8 thread was purchased. After consulting with the pilot, the recommendation was to mount the radar on a slight angle such that the radar was parallel with the underside of the wing. This is depicted below.

Figure 21: Radar Positioning

The U bolt was fitted with anti-vibration M8 ‘Nord-Lock’ washers and were secured with two nuts on either side to ensure that it would be resilient to the extreme wind forces and vibration placed on the device. It is important to note that one PCB spacer was adequate for the top section of the radar, however; the bottom section of the radar required three PCB spacers so that the U bolt was not touching the radar.

The power and data cables were then required to be run from the radar to the cockpit of the plane. This required USB cable extenders. The length of the cable run was minimised as best as practically possible, however; the total cable length resulted in being 5m. The user guide for the IWR1642BOOST recommends that ‘the length of the power cable should be < 3m’ (Instruments 2020) but this was not possible unless the power supply was mounted to the spray rail. Due to the low power consumption of the radar with the fine-tuned configuration file, it was decided that a 5m USB 3.0 cable was used for the power supply. A standard 5m USB 2.0 cable was then selected for the data transfer. Both cables were wrapped around the spray rail and taped up the side of the aircraft using high velocity duct tape. This is depicted below.

Figure 22: IWR1642BOOST Cable Management

Due to there being no AC power outlets on the plane, a USB portable charger was selected. The chosen portable charger had a storage capacity of 20,000mAh. The laptop chosen to be stationed in the cockpit was a surface pro-3. To ensure that both the portable charger and laptop had enough battery storage to collect the data, a battery test was conducted. The battery test was setup so that the radar was connected to the laptop, with live data being recorded using a screen recorder app. Both the cables used for the data and power transfer were connected with the USB cable extender to replicate the conditions of the experiment. The tuned configuration file was also uploaded to the radar for accurate results. The test concluded with the laptop being the first device to run out of battery after 2 hours. The portable charger was still indicating full charge when the laptop was out of battery. This provided a good indication of how long the plane could be collecting data before it needed to be grounded.

4.2.3 GoPro Hero 4 Black

The camera chosen for the experiment was required to be small and easy to mount to the plane. It was determined that a GoPro Hero 4 Black met these requirements sufficiently. This was due to the camera already being acquired and the numerous attachments that are readily available for GoPro's. The chosen GoPro has the following specifications as listed on GoPro's official website (GoPro 2014).

- Up to 4K at 30 fps
- Ultra Wide Angle Glass Lens
- 3.8 VDC 1160 mAh

A battery test was conducted which resulted in the GoPro having just over 3 hours of charge. The resolution selected was 1080p and the frame rate selected was 30 frames per second. A 64 Gb SD card was selected which was capable of storing up to 8 hours of footage.

To protect the GoPro from foreign objects such as bugs, a waterproof case was used. This fully enclosed the GoPro and allowed a mounting bracket attachment. The GoPro needed to be mounted near the radar to ensure results are accurate and to minimise errors. After consulting with the pilot, it was determined that the camera would be best situated on the wing flap bracket, which was within 10cm from the radar. Securing the camera in this location required it to be mounted sideways. The camera was secured using a GoPro helmet mount. The mount was then cable tied and finally secured with high velocity duct tape as shown below.

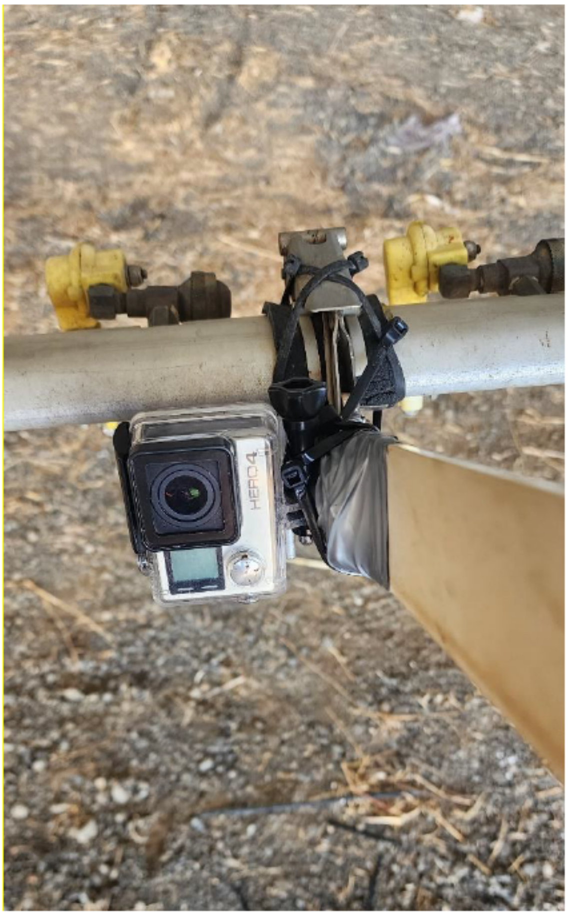


Figure 23: GoPro Secured to Plane

Due to the camera having such a wide field of view, the angle it was mounted was not critical, but it was decided to be positioned roughly parallel to the wing to correspond with the radar. A test was conducted which involved a 10 second video being recorded to verify that the camera was positioned correctly. This test resulted in the below frame which was sufficient and did not require modification of the GoPro positioning.

Figure 24: Test Footage to Verify Correct Position of GoPro

4.3 Additional Notes

Whilst conducting testing and calibrations, it was discovered that the plane had been damaged from bird strikes previously. It has been listed as an additional note in this section as it reinforces the need for an OAS for crop-dusting aircraft. The above images depict the damage caused by birds on the plane.

Figure 25: Plane Damage (1)

Figure 26: Plane Damage (2)

Figure 27: Plane Damage (3)

5 TESTING AND CONCLUSIONS

5.1 Millimeter-Wave Radar

To detect the pulley plate, the MMW radar would need to successfully display an object on the 2D radar map. This section will present the results of the MMW radar data. A total of ten flyovers were conducted with the exact time the radar passed the pulley noted for record keeping and ease of analysing the data. This data is presented in the below table.

Table 1: Time of Each Flyover

Flyover Number	Time (hour:minute:second)
1	1:49:55
2	1:51:00
3	1:52:09
4	1:53:10
5	1:54:07
6	1:55:07
7	1:55:53
8	1:56:46
9	1:57:39
10	1:58:53

Referring to the data, it can be seen that each flyover was approximately one minute apart. There are slight discrepancies in the time due to slight changes in the flight path after each flyover. The data was collected in under ten minutes which was a lot quicker than anticipated, this was because the flight run distance was kept to a minimum.

The mmWave Demo Visualizer was screen recorded during the data collection. It was discovered that the application froze at 1:57:45 which resulted in the last test being inconclusive. This could have been due to several reasons, such as a loss of connection to the radar or the CPU load of the radar being too high (at approximately 42%). When the laptop was removed from the plane, it was very hot which was not expected, and it is likely that the laptop overheated which resulted in the application freezing. The following figures are extracts from the mmWave Demo Visualizer at the above-mentioned times.

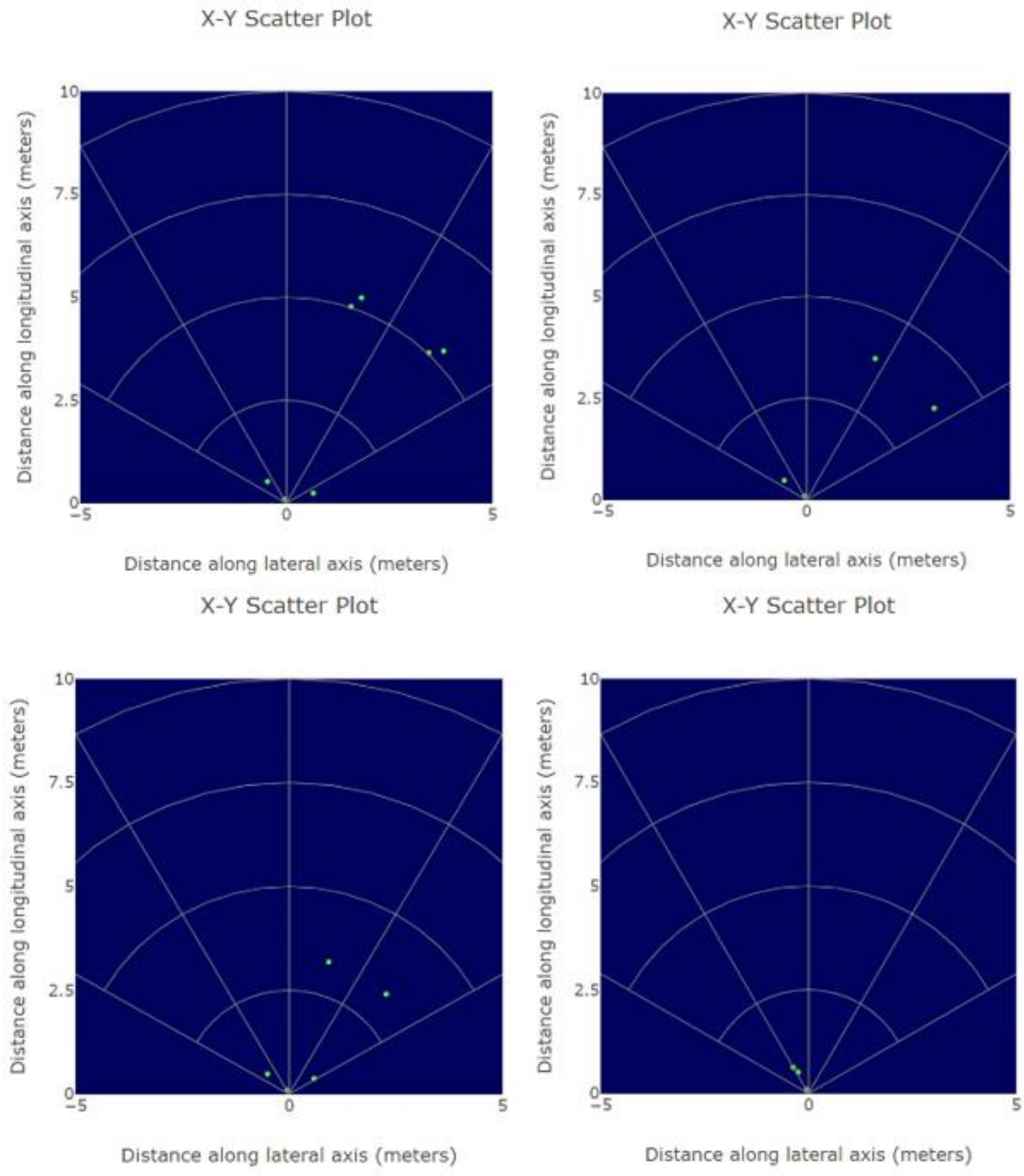


Figure 28: Flyover 1 (Top Left), Flyover 2 (Top Right), Flyover 3 (Bottom Left), Flyover 4 (Bottom Right)

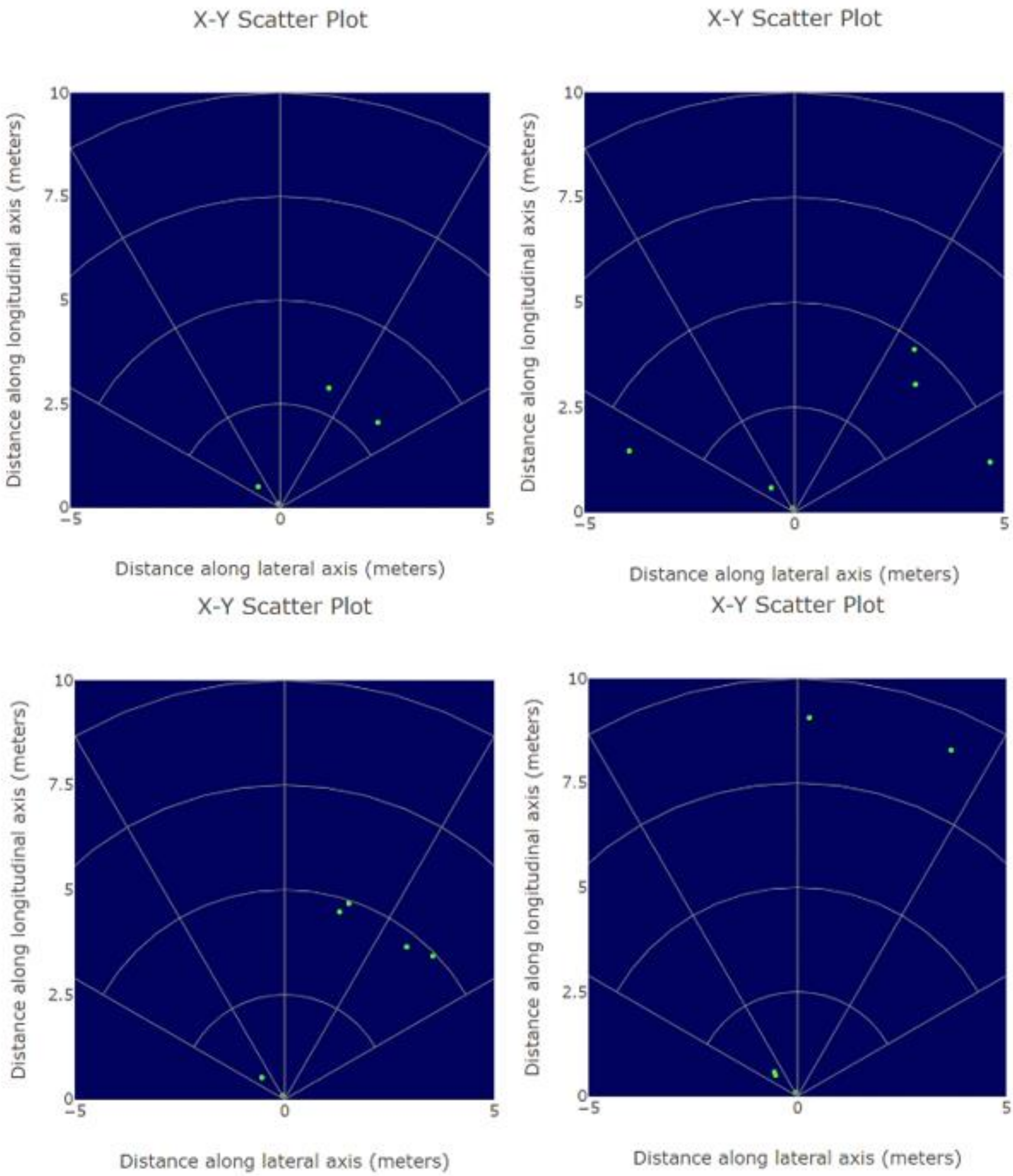


Figure 29: Flyover 5 (Top Left), Flyover 6 (Top Right), Flyover 7 (Bottom Left), Flyover 8 (Bottom Right)

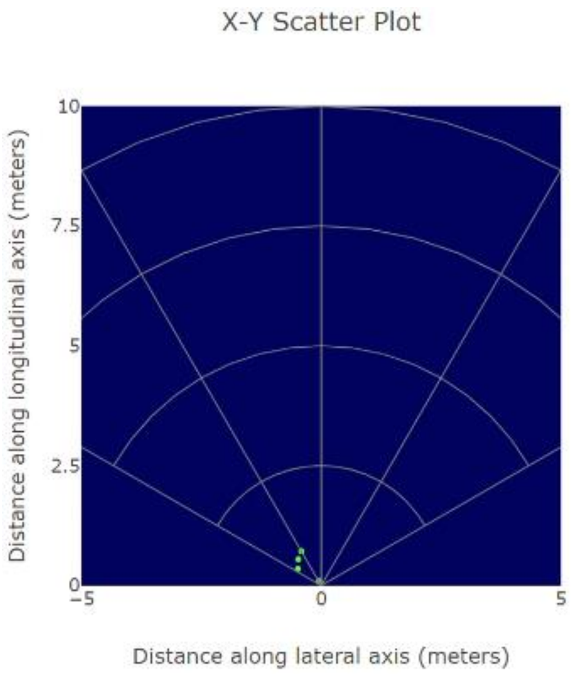


Figure 30: Flyover 9

Given that the data for the last flyover was inconclusive, the total test number was reduced to nine. Out of the nine tests, only seven accurately detected the pulley plate, which is equivalent to a 77.8% success rate. Flyover number 4 and 9 did not detect anything. Flyover number 9 may have been because the application may have been in the process of crashing as it crashed six seconds after this data extract. Further reasoning for this may be found in later results where a combination of the video footage and MMW data are discussed. The data presented shows multiple detected points when the plate is detected. This is due to the plate being reflective, if the plate was not reflective, it may have resulted in a single point of detection or possibly no points of detection. Most of the flyovers identified the plate in similar areas at roughly 2.5-5 meters but flyover number 8 was able to identify the plate at a further distance of approximately 9 meters. To understand the reasoning for this, the video footage needed to be compared, this can be found in section 5.3: Combination of MMW Radar Technology and Machine Vision.

5.2 Object Detection Algorithms

To detect the pulley plate with an object detection algorithm, several tests have been conducted. This section will investigate multiple algorithms as well as manipulation of the code to find if these algorithms are practical and if they can be adjusted to detect the pulley plate. Each section will present an initial test of an algorithm and if unsuccessful in identifying the plate, further manipulations of the code will be undertaken with additional tests and results. All code constructed and manipulated for the tests can be found in the relevant appendix sections.

5.2.1 Test 1 – Testing Comparisons Between Two Frames

The initial test conducted for the algorithm was to look at two individual frames of the video to match features of the pulley plate between the frames. In this test, the 20549th frame was selected, and the pulley plate was cropped out to be the test image. The 20550th frame was then loaded as the comparator image which included the entire frame. These images were then converted to grayscale and displayed in separate figures. The features are then detected, and the points are displayed on separate figures with green circles. The matched points (if any) are then displayed on the final figure with a yellow line connecting all feature points between the two images.

The first two images are the same for all algorithms, as such they are displayed below.

Figure 31: Cropped Pulley Plate Test Image

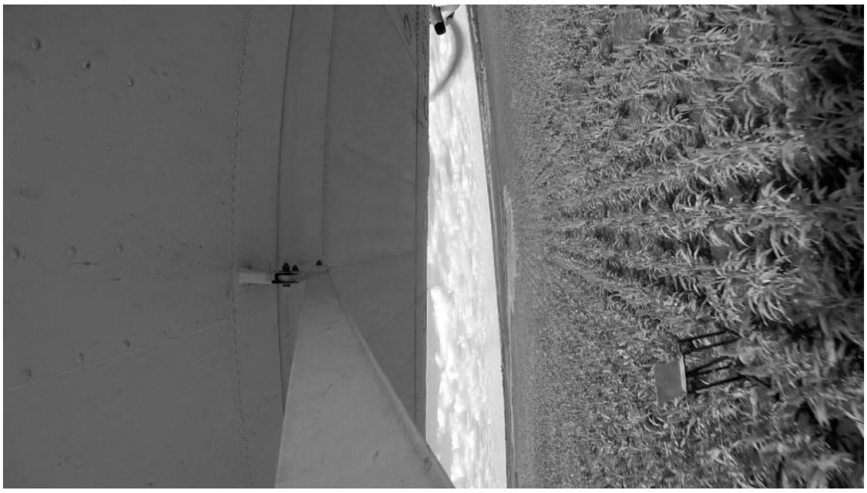


Figure 32: Test Environment Image

ORB Detection Algorithm

The ORB Detection algorithm can be found in Appendix D.

The outputted images are displayed below.

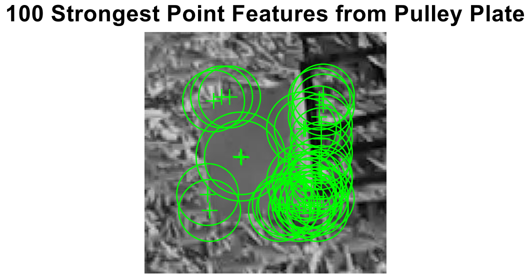


Figure 33: Test 1 - ORB Pulley Plate Feature Points

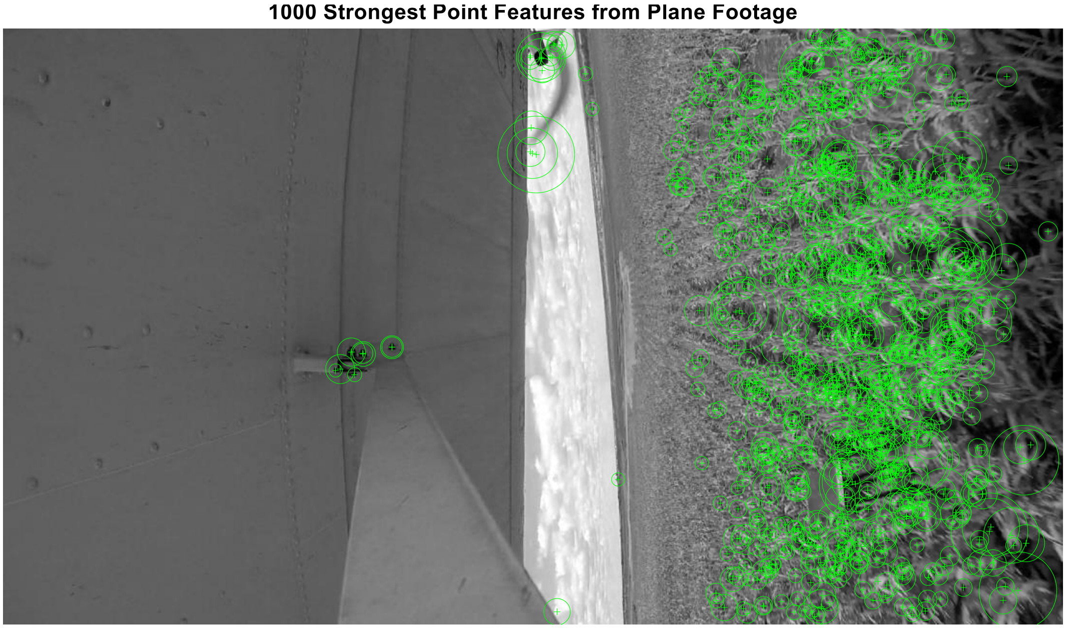


Figure 34: Test 1 - ORB Plane Footage Feature Points

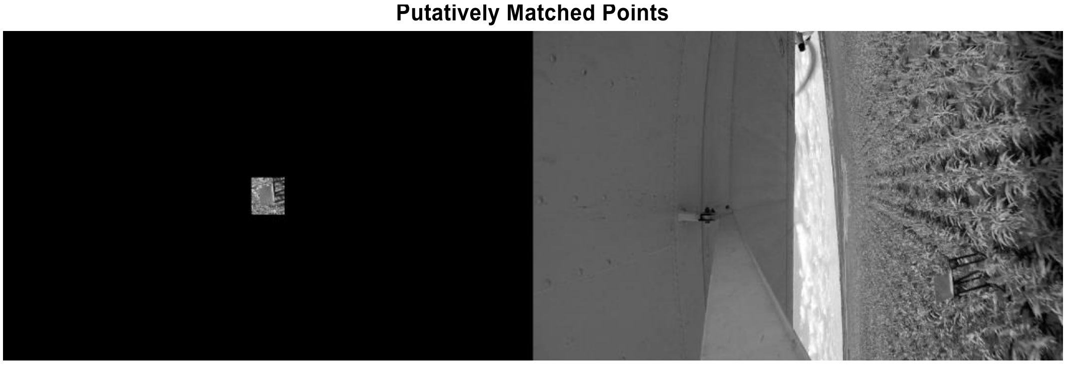


Figure 35: Test 1 - ORB Matched Points

SURF Detection Algorithm

The SURF Detection algorithm can be found in appendix E. Note that a metric threshold was added to the ‘detectSURFFeatures’ function. This altered the number of blobs that were detected within the images.

The outputted images are displayed below.

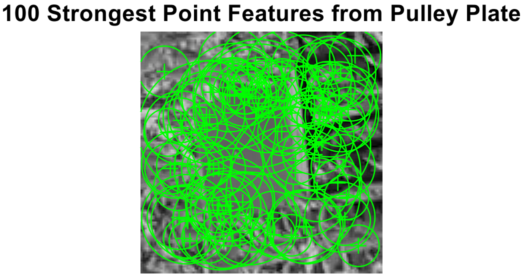


Figure 36: Test 1 - SURF Pulley Plate Feature Points

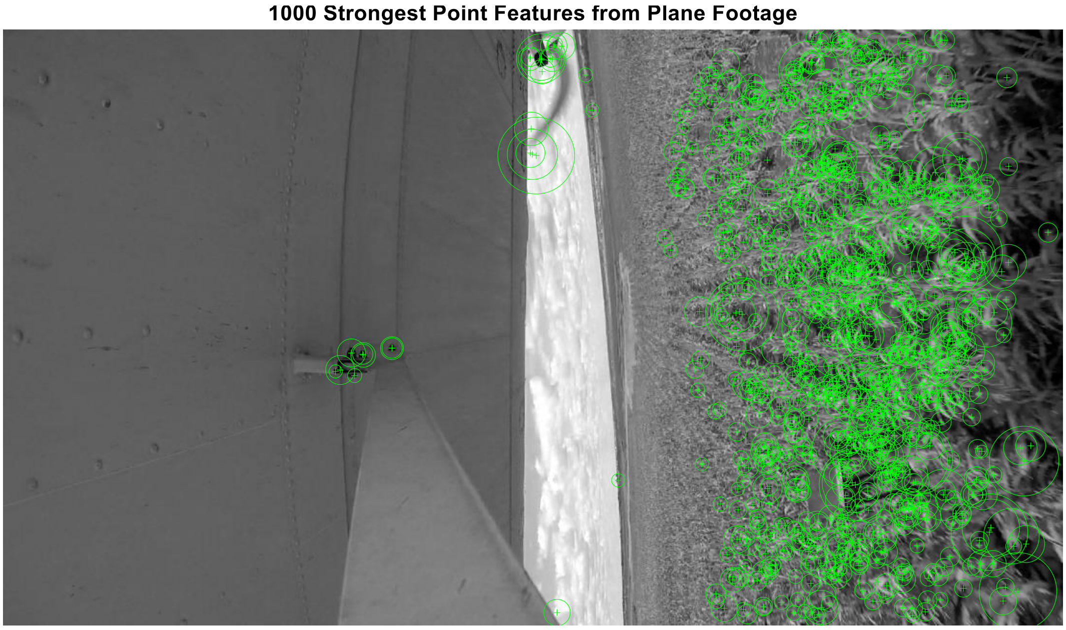


Figure 37: Test 1 - SURF Plane Footage Feature Points

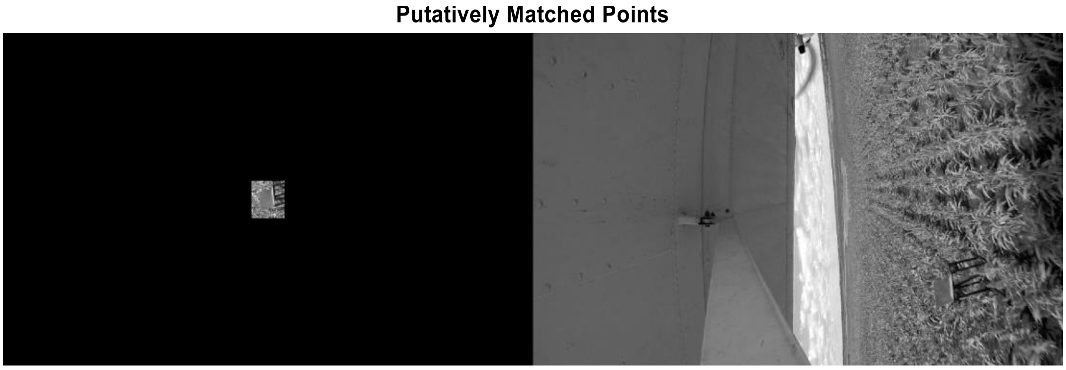


Figure 38: Test 1 - SURF Matched Points

SIFT Detection Algorithm

The SIFT Detection algorithm can be found in appendix F.

The outputted images are displayed below.

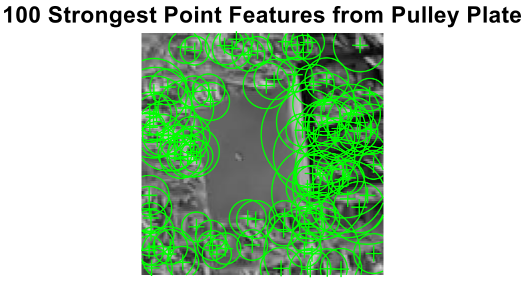


Figure 39: Test 1 - SIFT Pulley Plate Feature Points

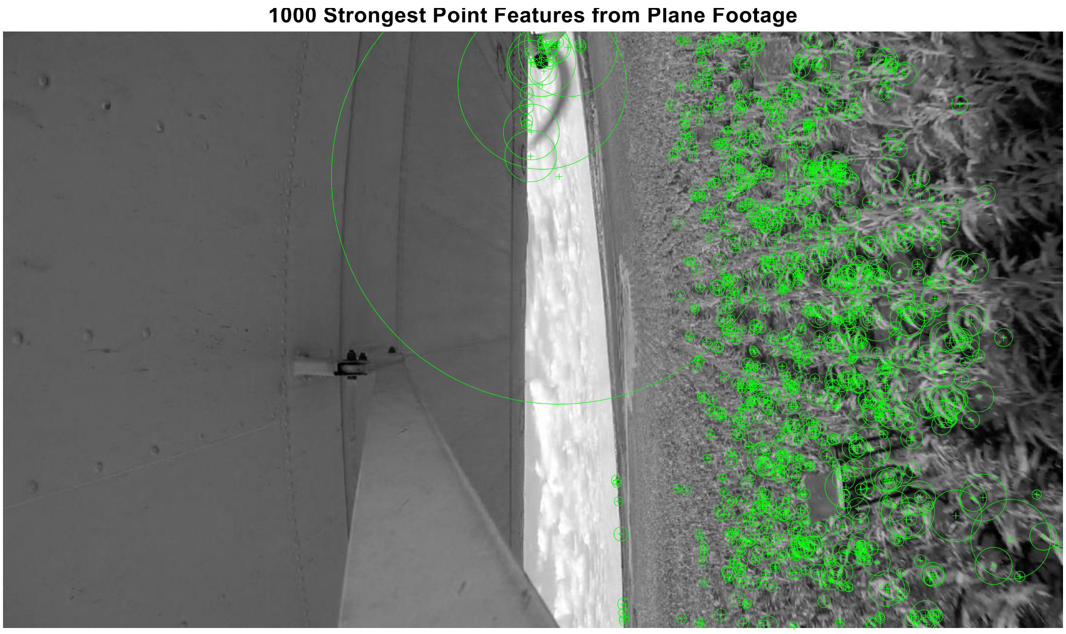


Figure 40: Test 1 - SIFT Plane Footage Feature Points

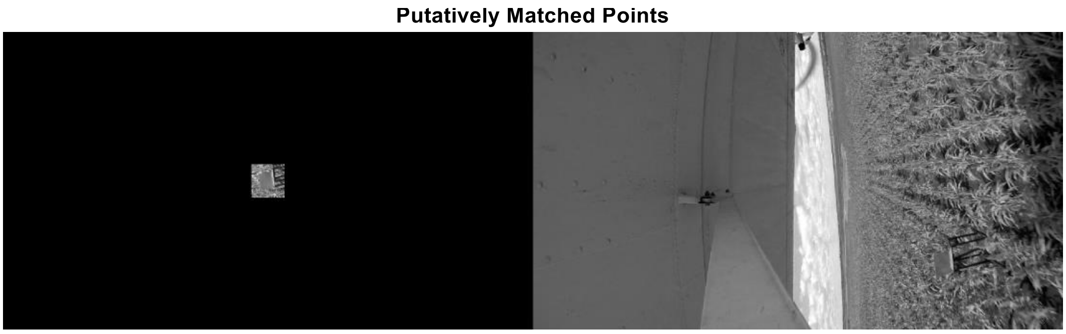


Figure 41: Test 1 - SIFT Matched Points

Results

Comparing the ORB, SURF and SIFT algorithms, it is clear that none of the initial tests worked. Each algorithm was able to present different areas of interest when comparing the first two figures in each test set. ORB and SIFT focused more on the outer edge of the plate whereas SURF focused on the entirety of the plate. The next test will be to alter the frame number from the plane footage to identify if no comparisons are found simply due to a blurry frame.

5.2.2 Test 2 – Altering Frame Number from Video Footage

To ensure that the selected plane footage frame is not blurry, multiple frames from the footage will be compared. This test will require the adjustment of the 'videoFrame' frame number. It has been determined that the pulley plate is identifiable between frame 20534-20551 giving a total of 17 frames to adjust for this test. Note that the limitations of this test are that data from only one of the ten flyovers will be used due to the amount of data that has been collected. The code has not been included in the appendix as the relevant appendices mentioned above are able to be used with the adjustment of the frame number in line 21.

ORB Detection Algorithm

The ORB algorithm did not identify any comparisons between the initial cropped image and the range of test frames that were adjusted.

SURF Detection Algorithm

The SURF algorithm was able to find the first correlation between the pulley plate and the plane footage on frame 20547. Unfortunately, the correlation was between the frame and the plane wing as presented below.

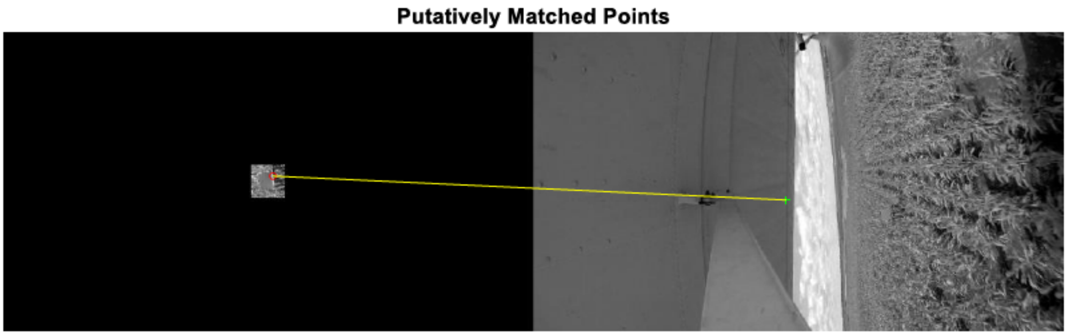


Figure 42: Test 2 - SURF Matched Points

It was also able to identify a correlation on frame 20544. This also did not make a correct connection as it identified something in the crops as seen below.

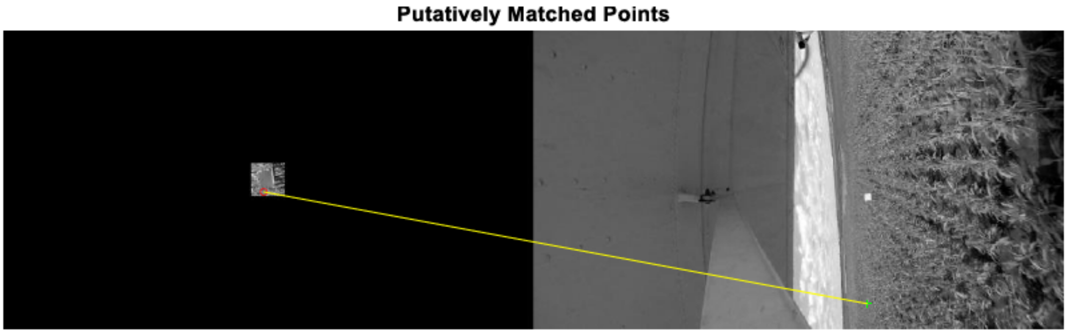


Figure 43: Test 2 - SURF Matched Points

SIFT Detection Algorithm

The SIFT algorithm did not identify any comparisons between the initial cropped image and the range of test frames that were adjusted.

Results

This test was unsuccessful in finding any common feature points between the pulley plate and the 17 frames that were being analysed. Both ORB and SIFT were not able to find a single similarity within the 17 frames analysed. The SURF algorithm was able to find similarities in two separate frames but unfortunately, these were incorrect comparisons and were made by points in the environment rather than between the plate. It was noted that the frame for the cropped pulley plate (20549) was towards the end of the frame range (20551) where it was shown. This was initially done deliberately to have a bigger and clearer test image. After completing this test, it was found that motion blur was more dominant with objects closer to the plane. The next test will follow the same process as this one but will also choose an earlier frame to crop the pulley plate from to get a clearer image.

5.2.3 Test 3 – Cropping Pulley Plate from Earlier Image and Altering Frame Number from Video Footage

This test will aim to select a clearer pulley plate image for comparison from an earlier frame. With a clearer pulley plate image, it is hopeful that more correlation will be found between it and the frames from the plane footage. Once completed, test 2 will be repeated in the same manner. Again, the limitations of this test are that data from only one of the ten flyovers will be used due to the amount of data that has been collected. Code will be included in the appendix for this test as the first section of the code will be modified significantly more than in the previous test.

ORB Detection Algorithm

Whilst conducting the test, it was discovered that the ORB algorithm only would error if the height and width of the cropped pulley plate was less than 69. To rectify this, a larger cropped area was used than what was required which included a large portion of the crops. It was also discovered that when only a 69x69 cropped frame was used, the ORB algorithm only identified 1 point of interest as shown below.

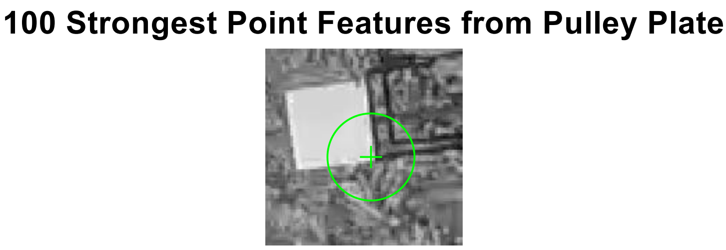


Figure 44: Test 3 - ORB Pulley Plate Feature Points (Small Cropped Frame)

A greater area of 90x90 resulted in several more points of interest. Due to this, the cropped image area was set at 90x90 which gave the below results.

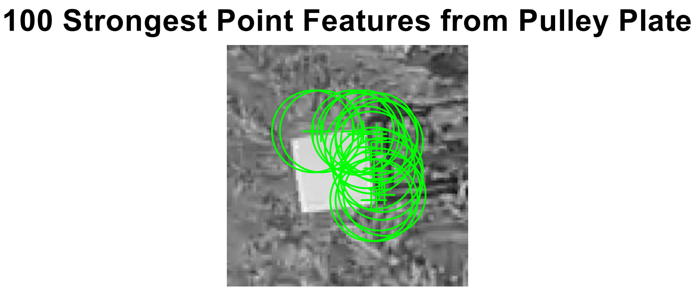


Figure 45: Test 3 - ORB Pulley Plate Feature Points (Large Cropped Frame)

The code used for the ORB detection algorithm can be found in appendix G. Unfortunately, there was no correlation between images found between videoframe images 20534-20551.

SURF Detection Algorithm

It was noted that SURF responded similarly to ORB in terms of points of interest. By having a small, cropped area, only one detection point was identified; increasing this area to 90x90 resulted in several more points of interest as displayed below.

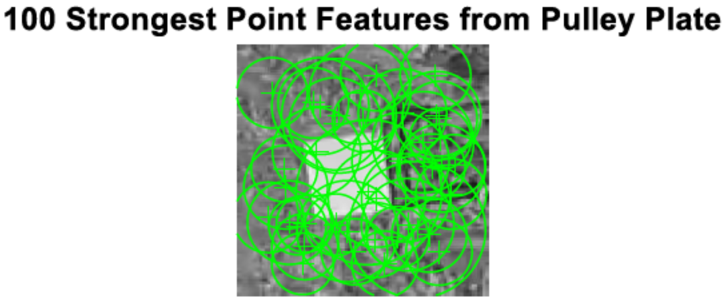


Figure 46: Test 3 - SURF Pulley Plate Feature Points

There were some correlations found between the cropped image and chosen frame, but the points were matched in incorrect areas. At frame 20534, the below correlation was found.

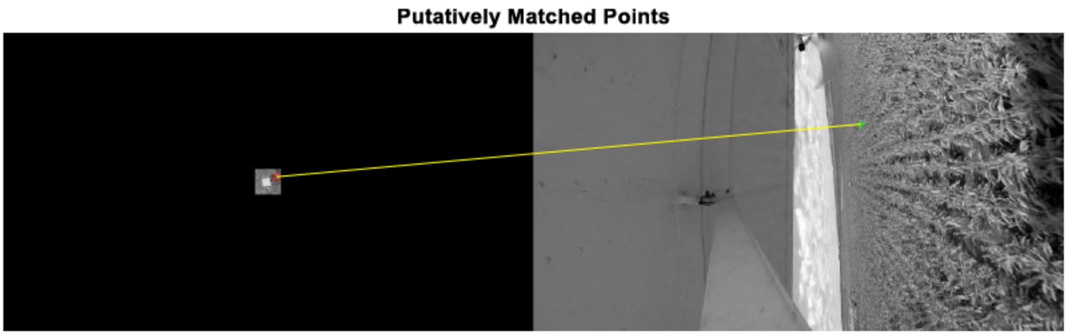


Figure 47: Test 3 - SURF Matched Points Frame 20534

Frame 20540 came extremely close to correctly identifying the plate, but when the image was zoomed, it showed a correlation with the crops as presented below.

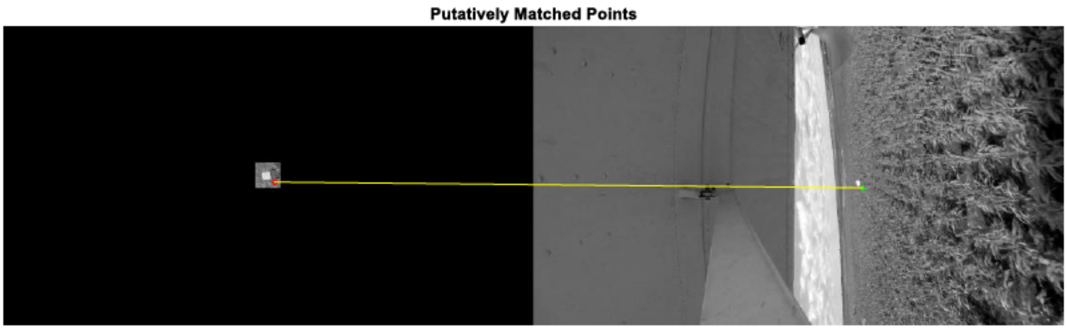


Figure 48: Test 3 - SURF Matched Points Frame 20540

Frame 20543 correctly showed a correlation between the plate. A total of three points were identified. One point was of the lower edge of the plate, another was of the frame, and one was an outlier. This is presented below.

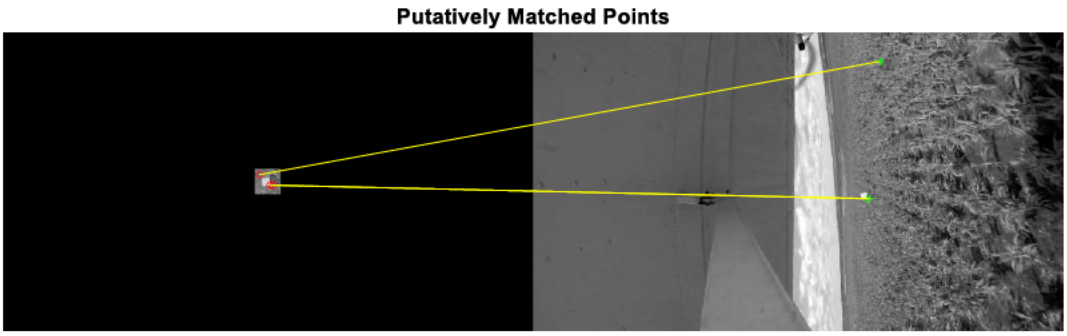


Figure 49: Test 3 - SURF Matched Points Frame 20543

Frame 20544 had extremely similar results as shown below.

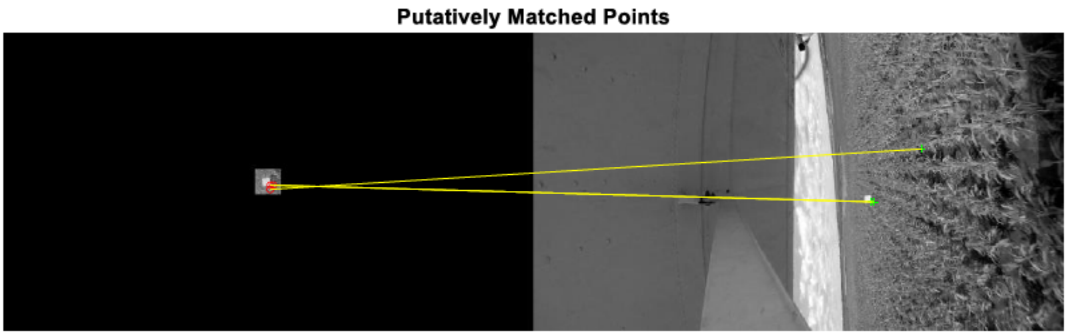


Figure 50: Test 3 - SURF Matched Points Frame 20544

Frame 20546 is shown below.

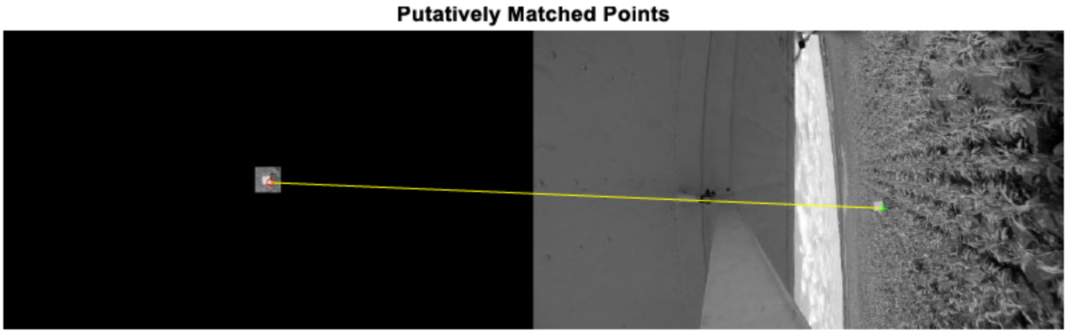


Figure 51: Test 3 - SURF Matched Points Frame 20546

The code used for the SURF detection algorithm can be found in appendix H.

SIFT Detection Algorithm

It was noted that SIFT did not respond like ORB or SURF regarding the cropped image size. By having a small, cropped area, several points of interest were identified instead of just one as shown below.

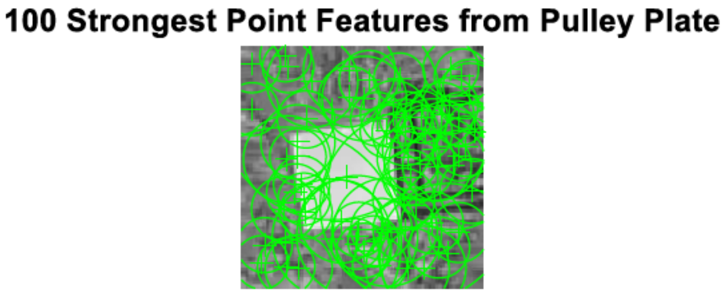


Figure 52: Test 3 - SIFT Pulley Plate Feature Points (Small Cropped Frame)

Increasing this area to 90x90 resulted in several more points of interest which made the image points a lot more intense as displayed below.

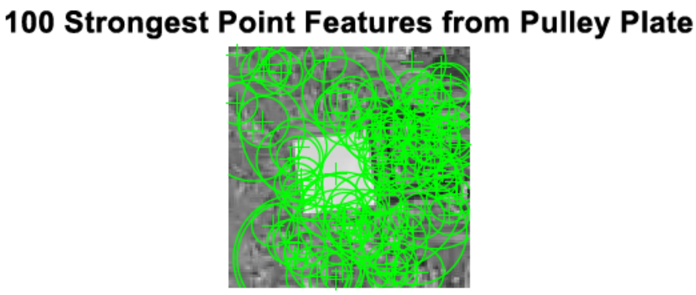


Figure 53: Test 3 - SIFT Pulley Plate Feature Points (Large Cropped Frame)

Comparing the two different images, there is not much benefit in having more points of interest as most are not shown on the plate. Given this, the smaller cropped area of 69x69 will be used.

Frame 20541 resulted in the first and only correlation; however, it was incorrectly matched as shown below.

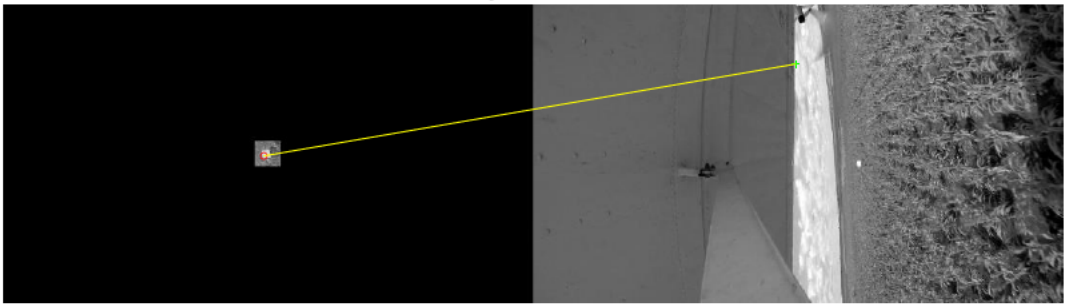


Figure 54: Test 3 - SIFT Matched Points Frame 20541

The code used for the SIFT detection algorithm can be found in appendix I.

Results

The SURF detection algorithm presented the most promising results by being able to find correlations in 6 out of 17 frames (inclusive of the frame the cropped image was extracted from) presenting an accuracy of 35%. It should be noted that the correlations were found in frame numbers near that of the cropped plate, meaning that there is a chance that correlations will only be found in similar/surrounding frames. ORB and SIFT did not present any correct correlations at all. The next test will look at segmenting the video footage to enhance points of interest.

5.2.4 Test 4 – Segmented Image Frames and Altering Frame Number from Video Footage

To aid in identifying the plate, the video frame could be cropped to exclude the wing and the resulting image be segmented into four sections to be independently tested. This would be efficient because the camera is fixed to the wing meaning that half of the image will be stable and not move at all when frame numbers are adjusted. The same process will be used as in other tests where the frame number will be incremented. Again, the limitations of this test are that data from only one of the ten flyovers will be used due to the amount of data that has been collected. Further limitations include that the plate may get divided over the segmented area resulting in inconclusive results. Code will also be included in the appendix for this test.

Cropping the image by [1000 0 920 1080] gives the below result noting that the frame size is 1080x1920.

Figure 55: Test 4 Original Frame Example

Dividing this image into four sections results in the following.

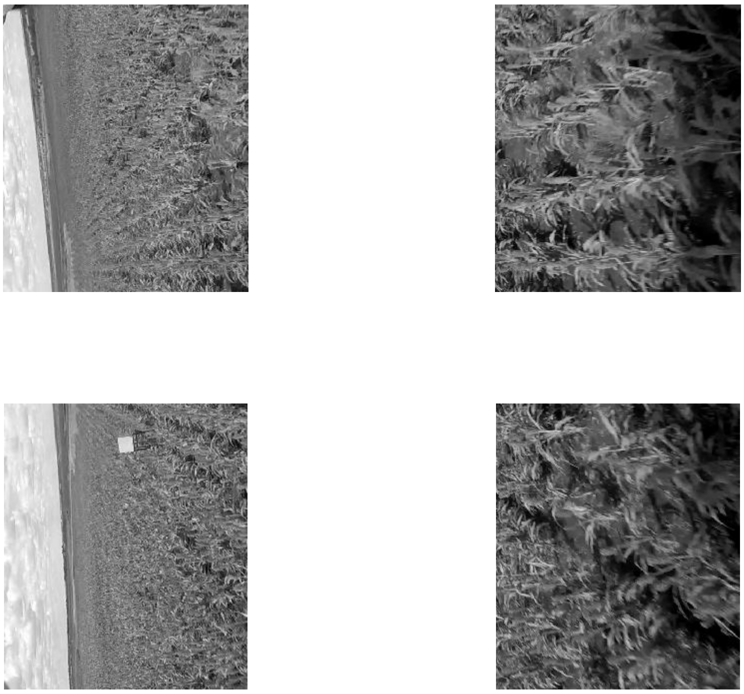


Figure 56: Test 4 Segmented Frame Example

For the frame range selected, the plate appeared in segment 2 and 4 only, therefore, for this test only segments 2 and 4 were analysed to reduce processing time.

ORB Detection Algorithm

The ORB detection algorithm was unsuccessful in matching any points between the cropped plate and the frame range selected. The code used can be found in appendix J.

SURF Detection Algorithm

The SURF detection algorithm was successfully able to detect a single correlation between the cropped image and frame 20544 as shown below.

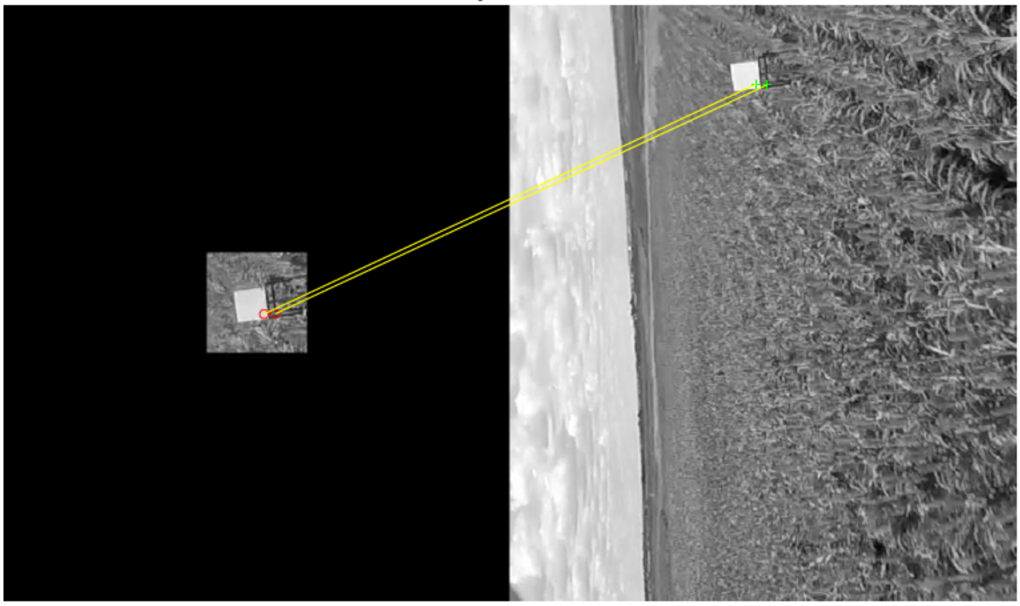


Figure 57: Test 4 - SURF Matched Points Frame 20544

The code used can be found in appendix K.

SIFT Detection Algorithm

Like the ORB detection algorithm, the SIFT detection algorithm did not identify any correlation between the two images. The code used can be found in appendix L.

Results

Surprisingly, segmenting the video frames resulted in worse results than leaving the video frame as is. Both ORB and SIFT were unsuccessful in identifying any correlations. SURF was able to find a correlation between one frame only. This is a huge decrease in accuracy compared to the last test. A further test that will be conducted is to use a clear image of the pulley plate and repeat the process of going through a video frame range.

5.2.5 Test 5 – Importing a Clear Pulley Plate Image and Altering Frame Number from Video Footage

To reduce resolution limitations for the image being used for correlations, a clear photograph of the plate could be used. This would provide clear feature points that could assist in finding correlations in the plane footage. The same process will be used as in other tests where the frame number will be incremented. Again, the limitations of this test are that data from only one of the ten flyovers will be used due to the amount of data that has been collected. The test image used for this test is depicted below.

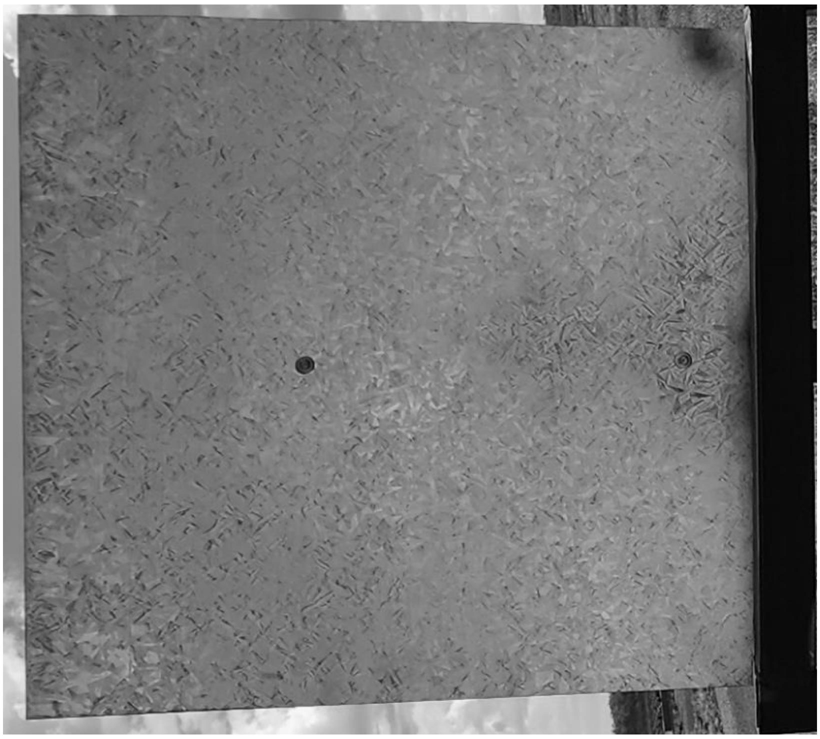


Figure 58: Clear Imported Image

ORB Detection Algorithm

Unfortunately, the ORB detection algorithm was unsuccessful in identifying any correlation between the test image and any of the frames. The code can be found in appendix M.

SURF Detection Algorithm

The SURF algorithm provided unexpected results with a range of correlations found in every frame. Frame 20534 resulted in the below correlations, unfortunately, none of which correctly correlated with the correct position of the frame as seen below.

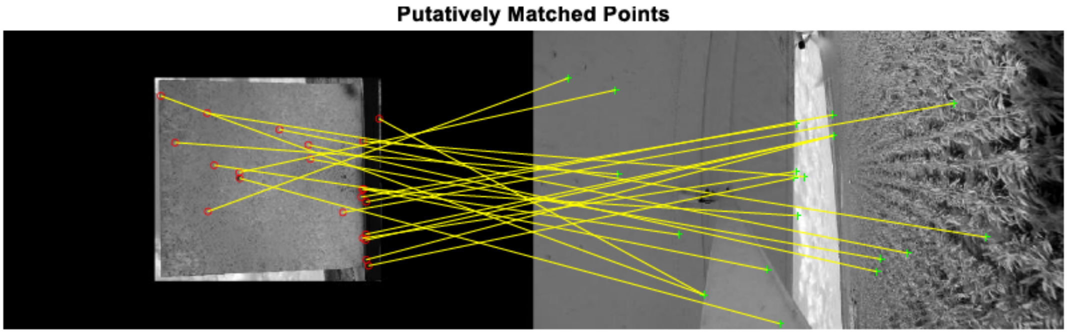


Figure 59: Test 5 - SURF Matched Points Frame 20534

All frames until frame 20544 were nearly identical to the image displayed above. Frame 20544; however, resulted in the first point being matched correctly as shown below.

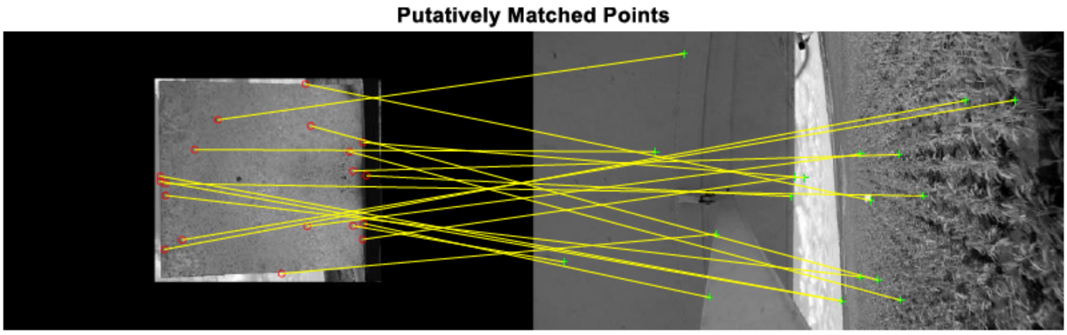


Figure 60: Test 5 - SURF Matched Points Frame 20534

Unfortunately, this only occurred at this particular frame and the rest of the images were the same as that shown in Figure 59: Test 5 - SURF Matched Points Frame 20534. The code used can be found in appendix N.

SIFT Detection Algorithm

Frame 20538 provided the first correlations identified between the two images, unfortunately the correlations were not accurate as represented below.

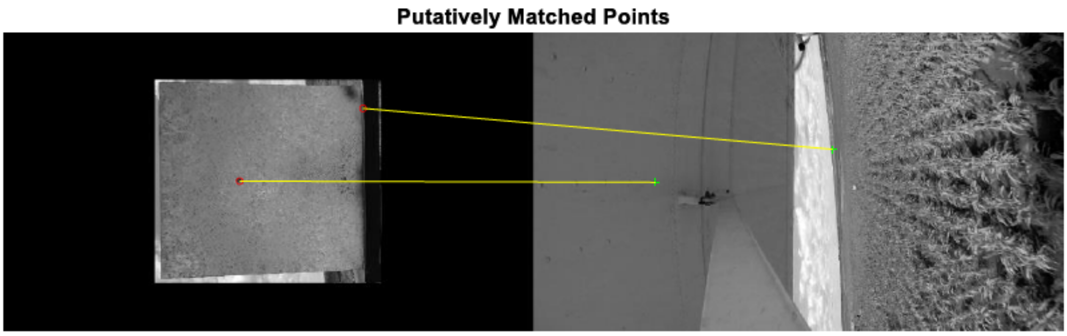


Figure 61: Test 5 - SIFT Matched Points Frame 20538

After conducting the test of the frame range identified, it was determined that the following frames incorrectly found correlations between the images: 20539, 20543, 20546, 20547 and 20549. Unfortunately, none of the frames correctly correlated with the plate. The code used can be found in appendix O.

Results

Unfortunately, this test did not result in any accurate results. Out of all three tests, the SURF algorithm provided the most correlations but only correctly matched the pulley with one frame. The SIFT algorithm was able to match points incorrectly within several frames. The ORB algorithm was inconclusive with no frames providing any matched points. A final test will be completed using edge detection.

5.2.6 Test 6 – Edge Detection in Image Segmentation

After conducting the previous tests, it was clear that the plane footage provides a very busy frame in terms of features. In an attempt to reduce the background features, edge detection in image segmentation was applied to the video footage. This required edge detection to be applied to a frame followed by image feature alteration so that the pulley plate is the only object displayed in the image.

Two edge detection methods were initially trialled and compared which consisted of the sobel and the canny method. After running a test to compare the methods between each other, it was decided that the canny method would be used as it had more distinct lines. The below shows the difference between the two methods.

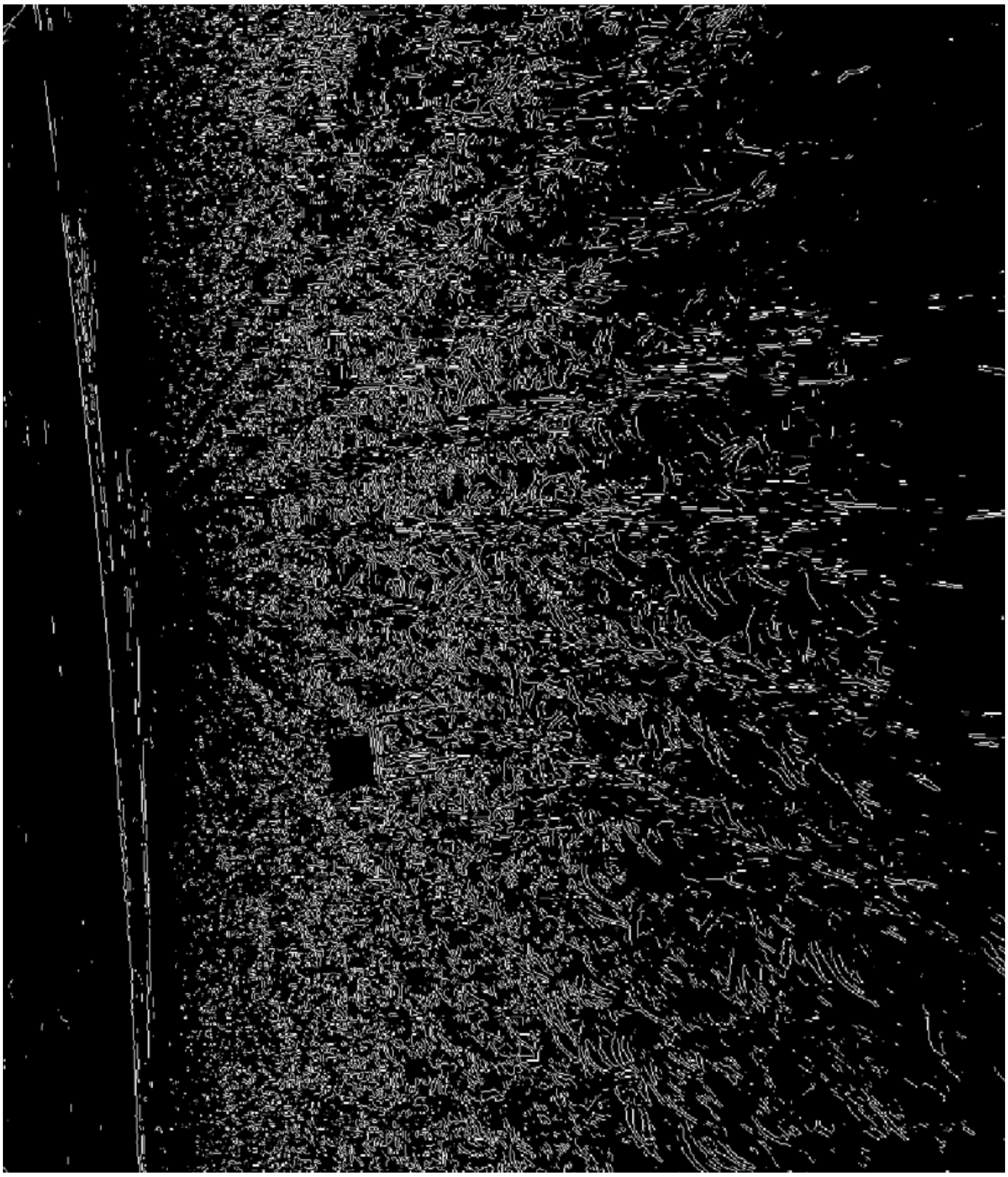


Figure 62: Sobel Method

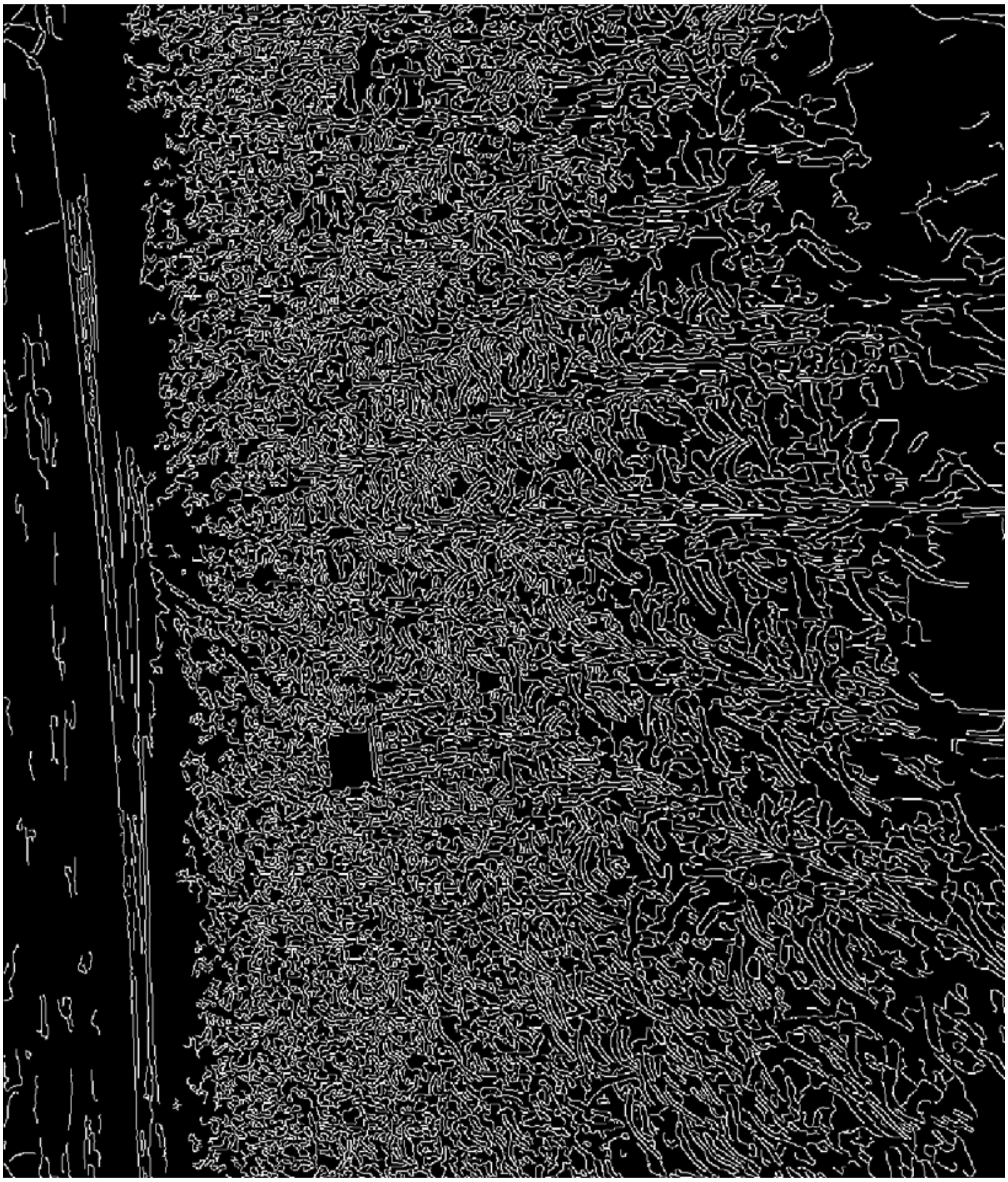


Figure 63: Canny Method

The plate outline can be easily seen in Figure 63: Canny Method, but it does not have any distinct fully enclosed outlines. The next two sub-sections attempt to highlight only the pulley plate. The first subsection was created to analyse a smaller test image such as that in test 3 to verify that the plate can be extracted. The second subsection then applied this method to the video footage to see if the results are successful in only identifying the pulley plate.

The code used in this test can be found in Appendix P.

Extracting Pulley Plate from a Small Cropped Image

The same cropped image used in test 3 was applied to this test. The first step was to apply the canny edge detection method to the image as shown below. The original image is also depicted below for ease of reference and comparison purposes.

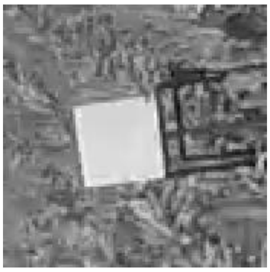


Figure 64: Original Image

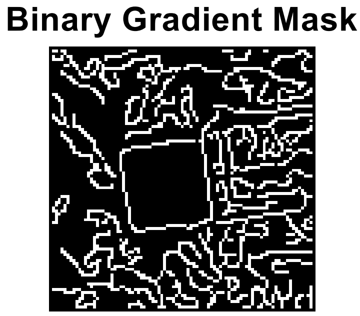


Figure 65: Canny Edge Detection Method

It can be seen that the pulley plate is not fully enclosed such as in the top right of the plate outline. To resolve this, the MATLAB function 'imclose' was used which resulted in the below.

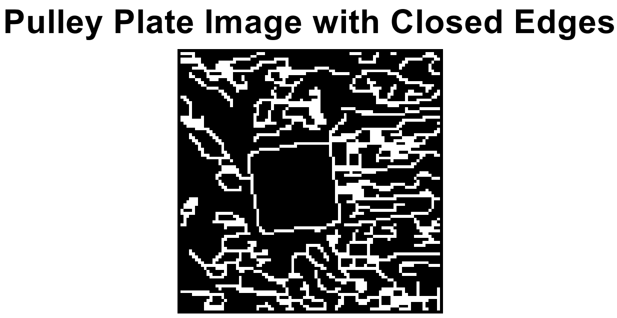


Figure 66: Pulley Plate with Closed Edges

All enclosed features of the image were then filled which resulted in the below.

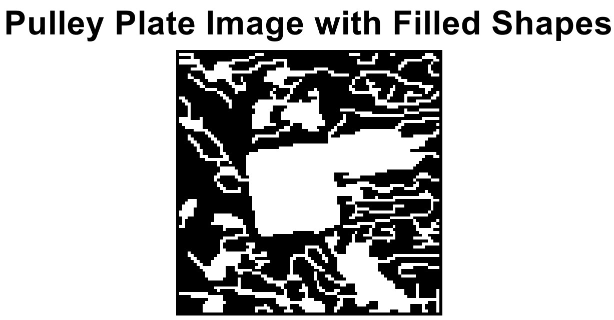


Figure 67: Pulley Plate with Filled Shapes

The image was then trimmed such that any blobs within the image that had an area of less than 1500 pixels were removed. This resulted in the below.

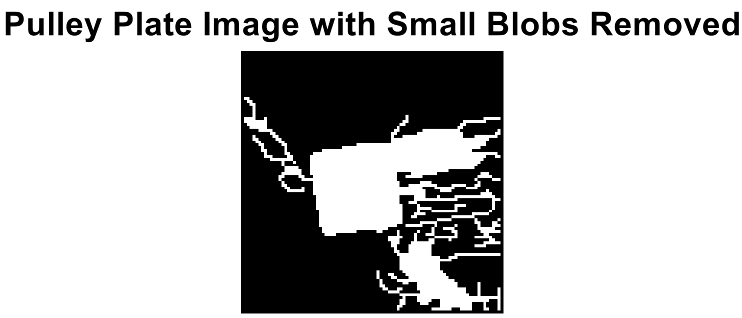


Figure 68: Pulley Plate with Small Blobs Removed

Finally, after trial and error, any joined sections that were less than 15x15 pixels in area were removed from the image which resulted in the below image.

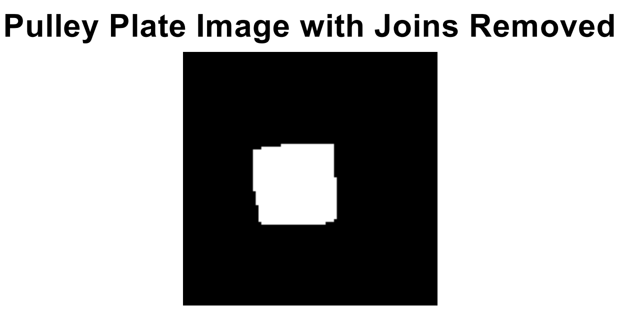


Figure 69: Final Pulley Plate Image with Joins Removed

This test was successful in removing all background features of the pulley plate and only displaying the pulley plate which verified that this method could be applied to the video footage.

Extracting Pulley Plate from Video Footage

Due to the positive edge detection results in the previous sub-section, the code created was then applied to the video footage. This test focuses on frame 20548 initially. Implications include that the video footage had a significant increase in features than that of the previous test. Figure 63: Canny Method represents the canny method used in this test. The image of the pulley plate was not fully enclosed, to mitigate this issue, the 'imclose' function was utilised which resulted in the below.

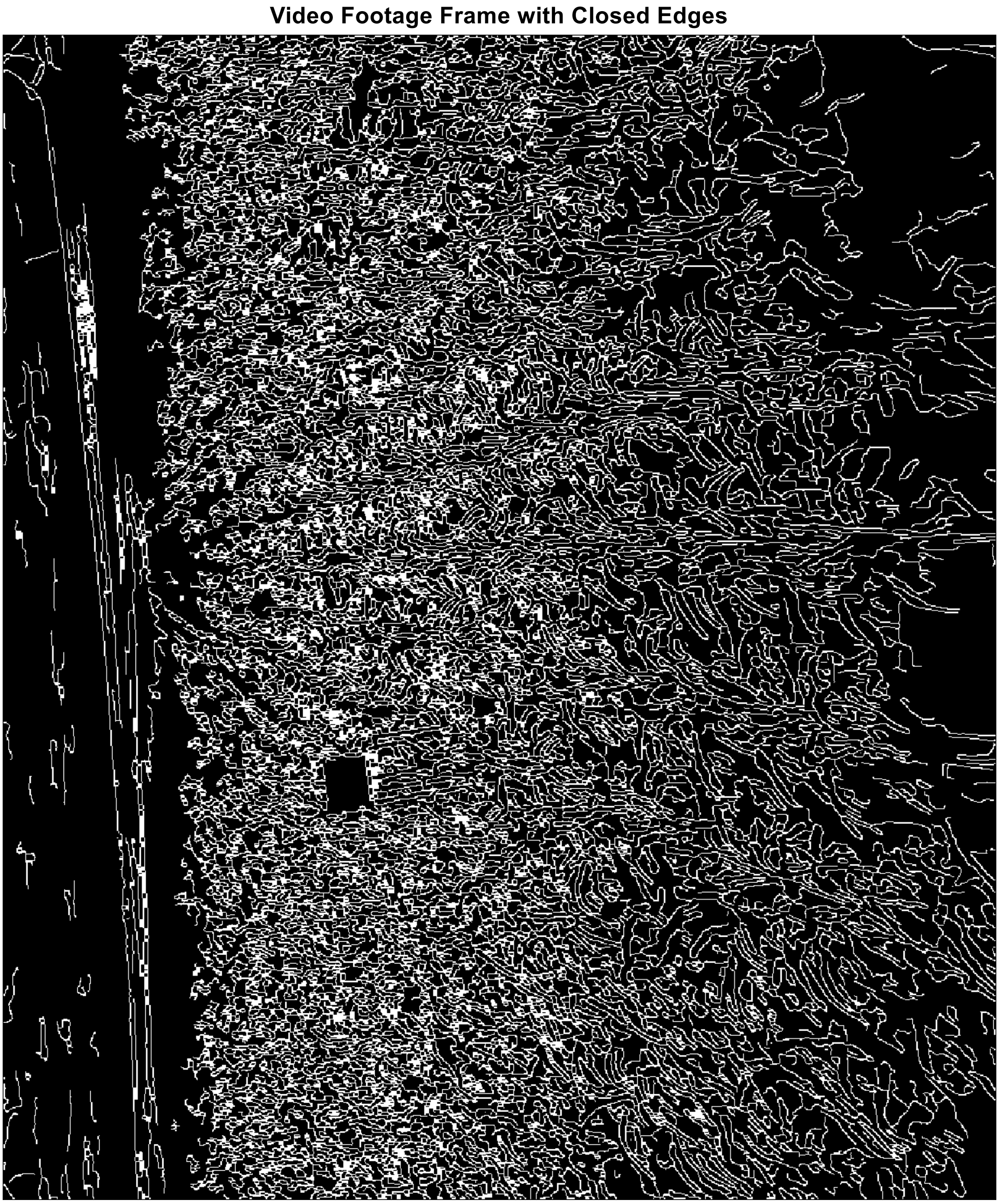


Figure 70: Video Footage Frame with Closed Edges (1)

The ’imclose’ function was then used to fill all connected figures which resulted in the below image.

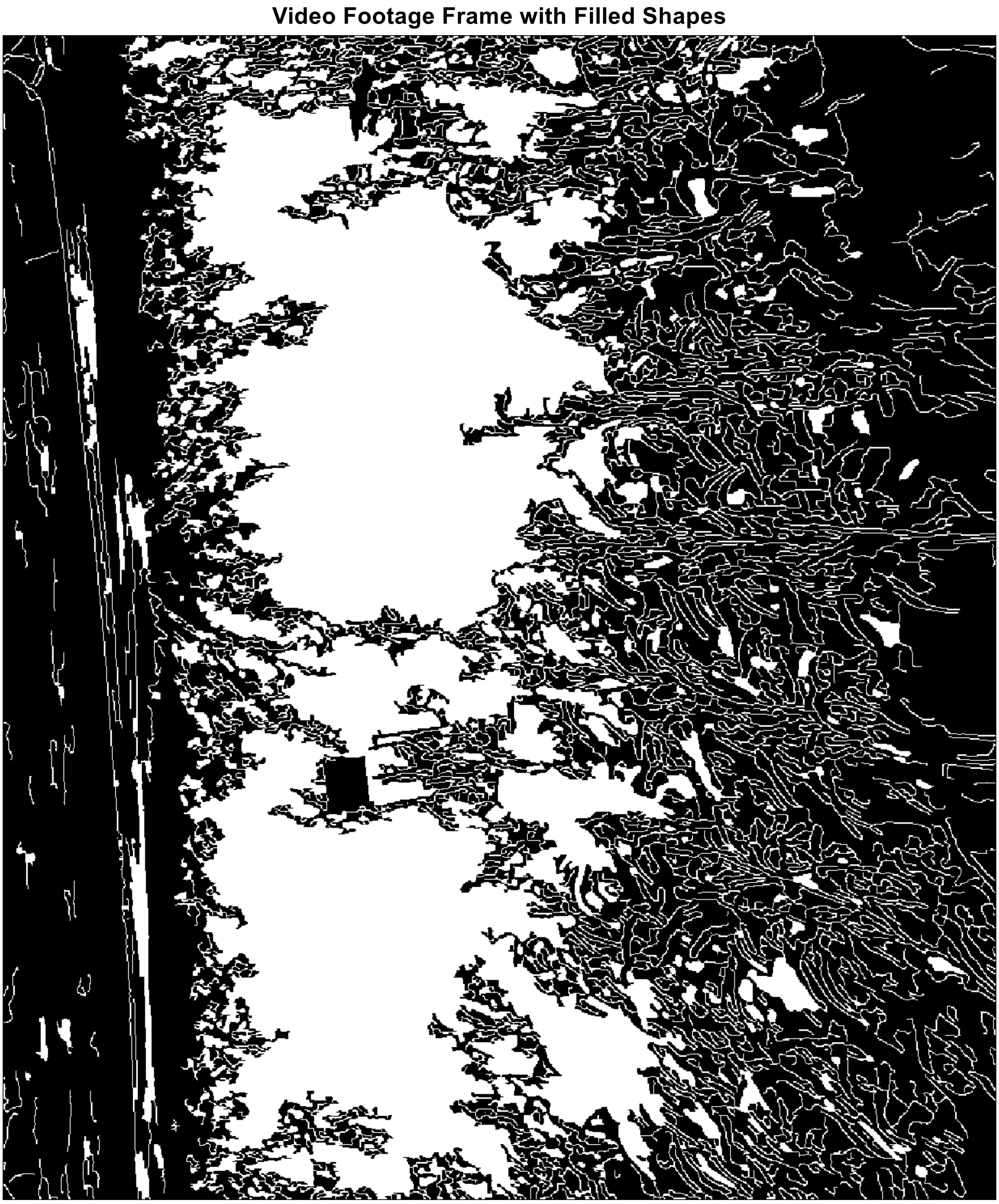


Figure 71: Video Footage Frame with Filled Shapes (1)

The image was then trimmed by removing small blobs as shown below.

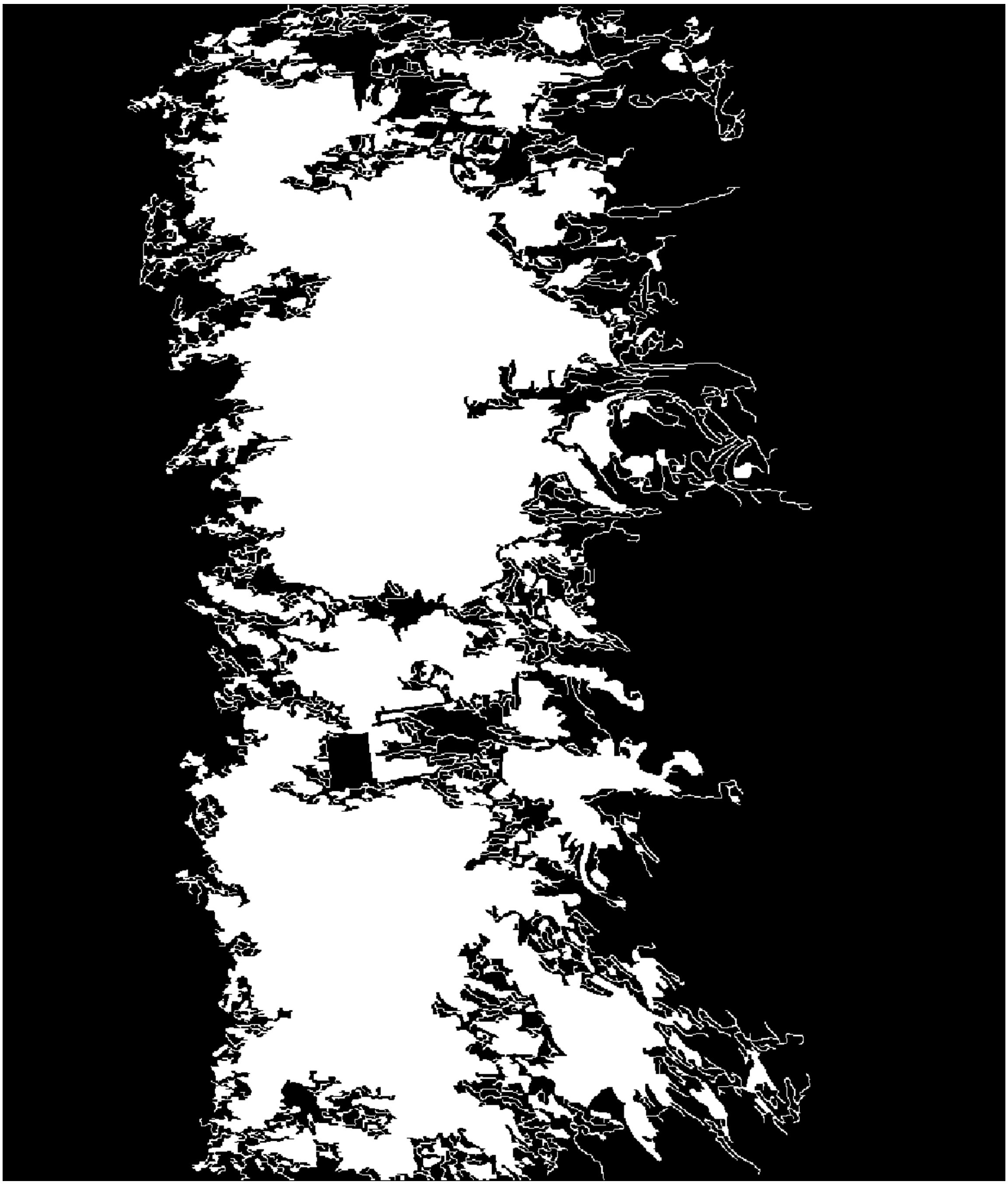


Figure 72: Video Footage Frame with Small Blobs Removed (2)

At this point, it was clear that the pulley plate had not been filled. It was decided to remove the already filled blobs and create another outline of the image with the remaining data using the same process as above. This resulted in the following.

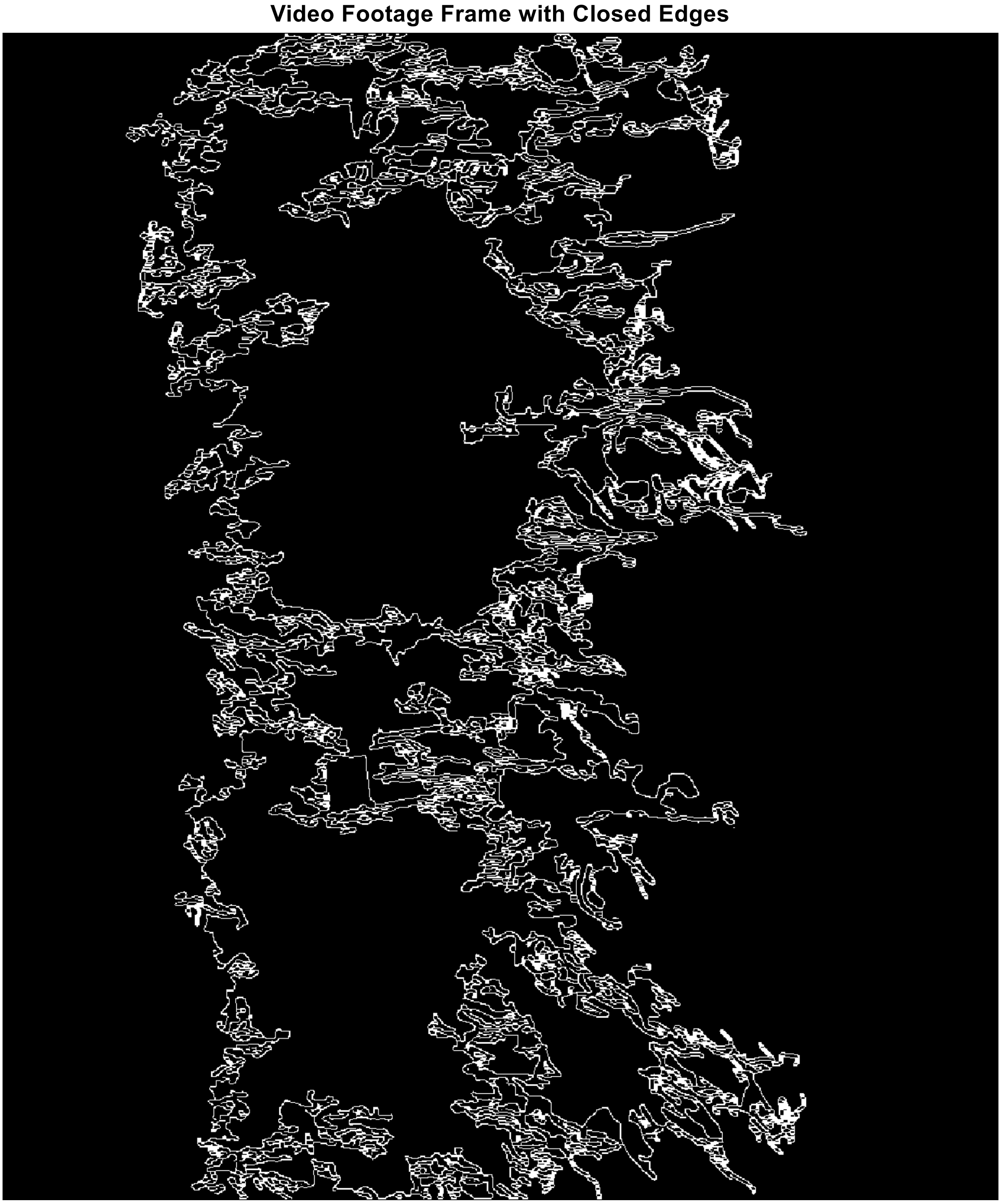


Figure 73: Video Footage Frame with Closed Edges (2)

This image was then filled which resulted in the below.

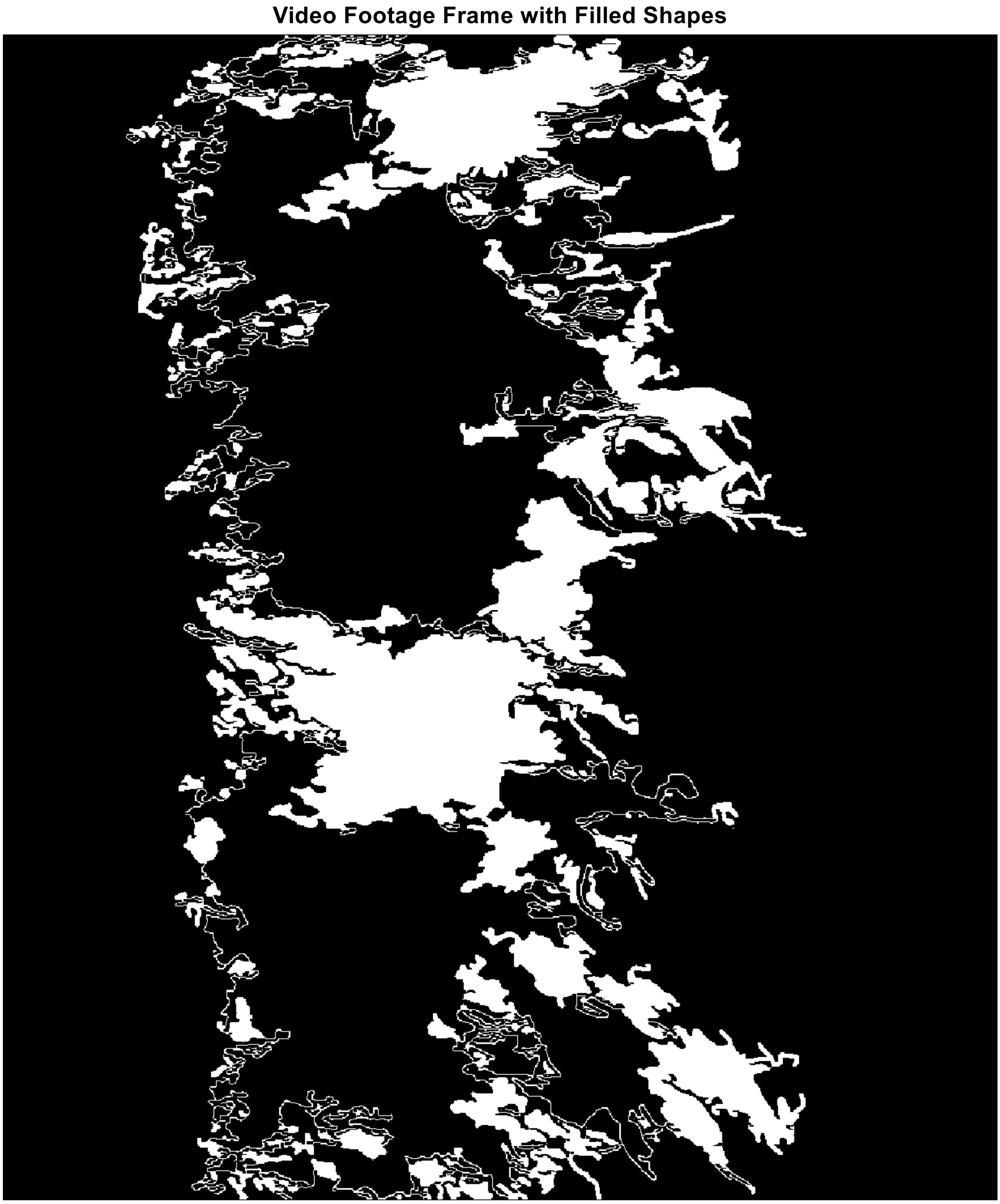


Figure 74: Video Footage Frame with Filled Shapes (2)

Small blobs were then removed as depicted below.

Figure 75: Video Footage Frame with Small Blobs Removed (2)

Finally, all joined sections with an area greater than 110x110 pixels were removed from the image and resulted in the following.

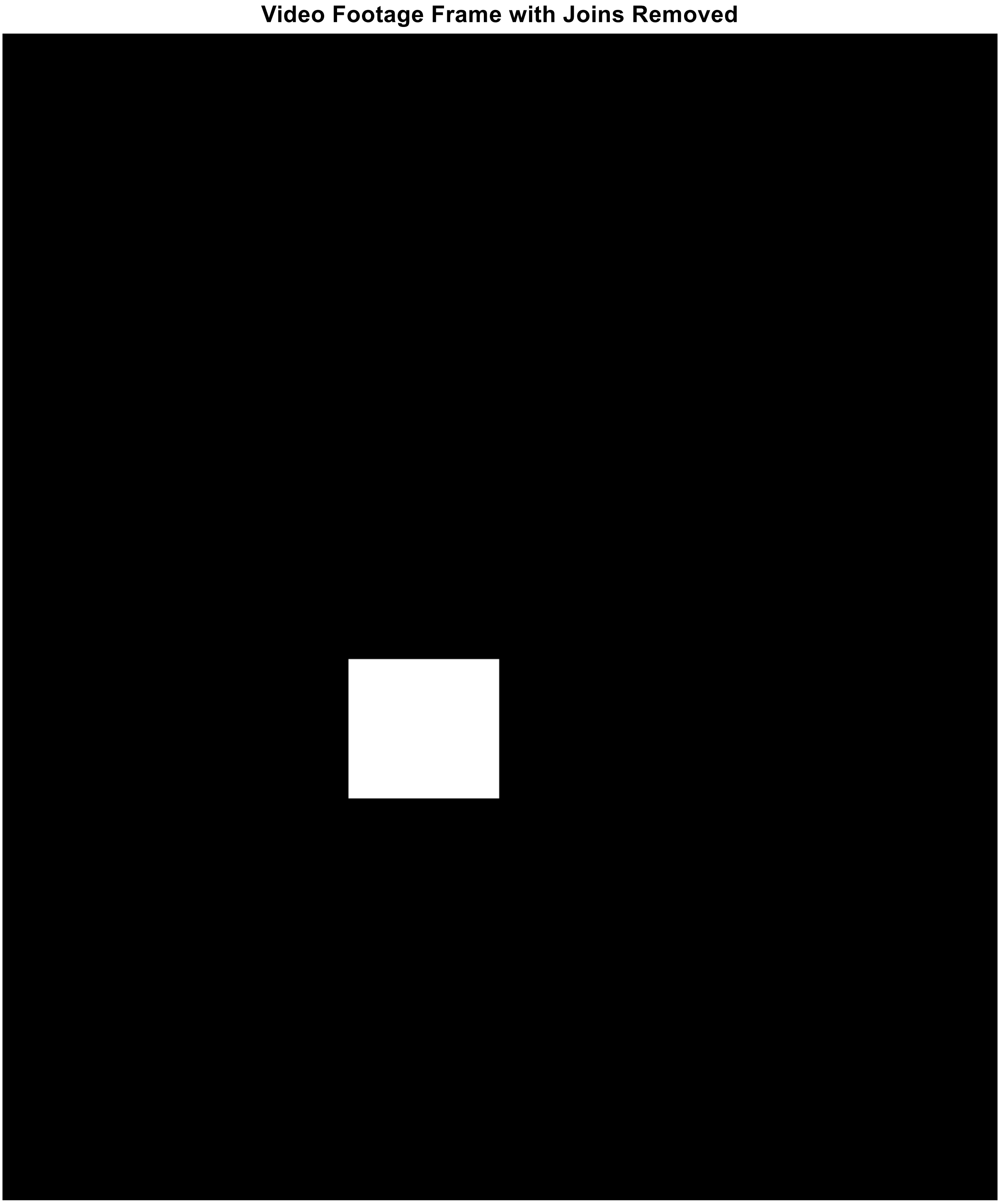


Figure 76: Final Video Footage Frame with Joins Removed

Although this image clearly represents a blob significantly larger than the pulley plate, it does highlight the area which the pulley plate is positioned in resulting in a successful result.

The frame number then needed to be altered to test and verify the accuracy of the results. After applying this code to several frames, it was discovered that it was not successful in identifying the plate. This is due to the code used being customised for this particular frame. A common issue was that the first 'imclose' function resulted in the pulley plate outline being closed, filled, and ultimately removed from the image in the first section of the code.

Results

The initial test using a smaller cropped image was successful and clearly highlighted the exact area of the pulley plate. After this, the code was applied to the video footage of frame 20548. This resulted in a positive outcome where the main area of the pulley plate was highlighted. Unfortunately, after applying this code to several of the remaining frames, it was discovered that the pulley plate could not be identified. To resolve this issue, a more generalised code could have been developed.

5.3 Combination of MMW Radar Technology and Machine Vision

This section will investigate an OAS using a combination of MMW radar technology and machine vision. Unfortunately, the algorithms in the previous section did not result in a desired outcome and as such, the video footage alone will be combined with the MMW radar results. In section 5.1 it was discovered that only nine out of the ten results were recorded due to technical issues and that the radar did not detect the plate during two of the flyovers. The data from the MMW radar will now be compared with the video footage in an attempt to understand why these two flyovers did not provide the expected data. A further investigation will then be conducted to determine how the data corresponds to the video footage from the other flyovers. It should be noted that the GoPro did not display the time as presented in Table 1: Time of Each Flyover resulted in an approximate frame being selected (within one second) for this combination of data. Below is the data from each of the flyovers with the MMW radar and approximate frame from the plane footage.

Figure 77: Flyover 1 Radar Scatter Plot and Plane Footage

Figure 78: Flyover 2 Radar Scatter Plot and Plane Footage

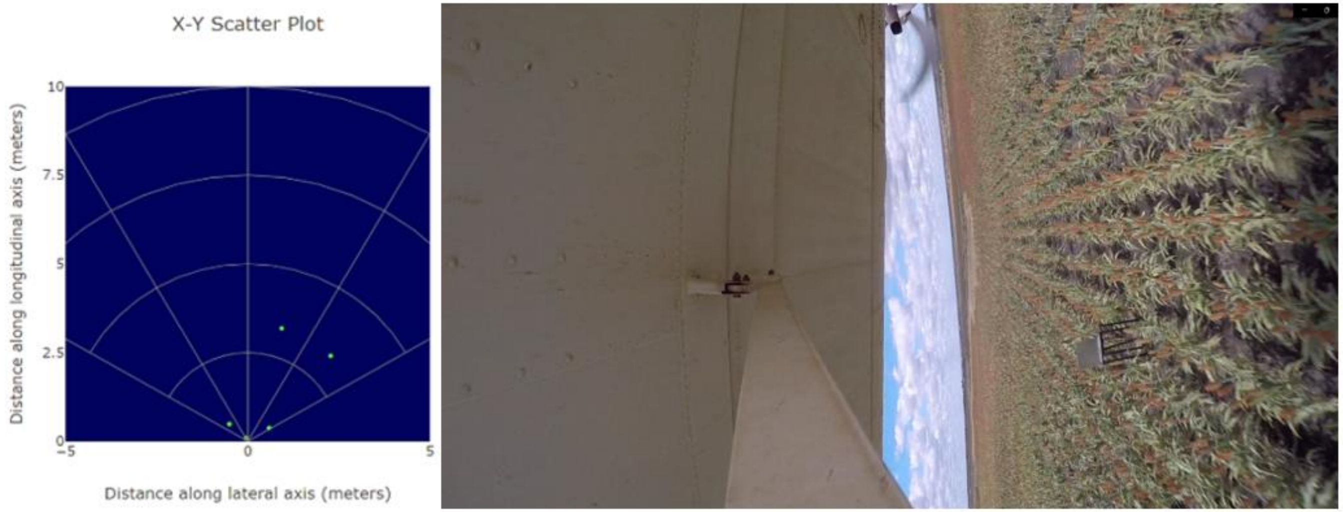


Figure 79: Flyover 3 Radar Scatter Plot and Plane Footage

Figure 80: Flyover 4 Radar Scatter Plot and Plane Footage

Figure 81: Flyover 5 Radar Scatter Plot and Plane Footage

Figure 82: Flyover 6 Radar Scatter Plot and Plane Footage

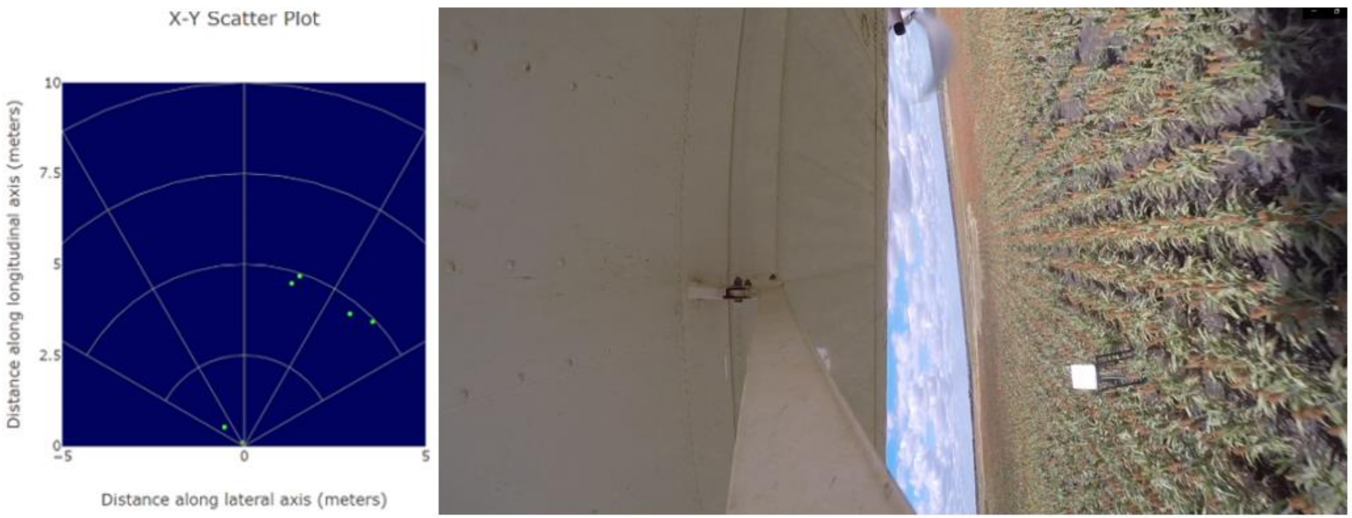


Figure 83: Flyover 7 Radar Scatter Plot and Plane Footage

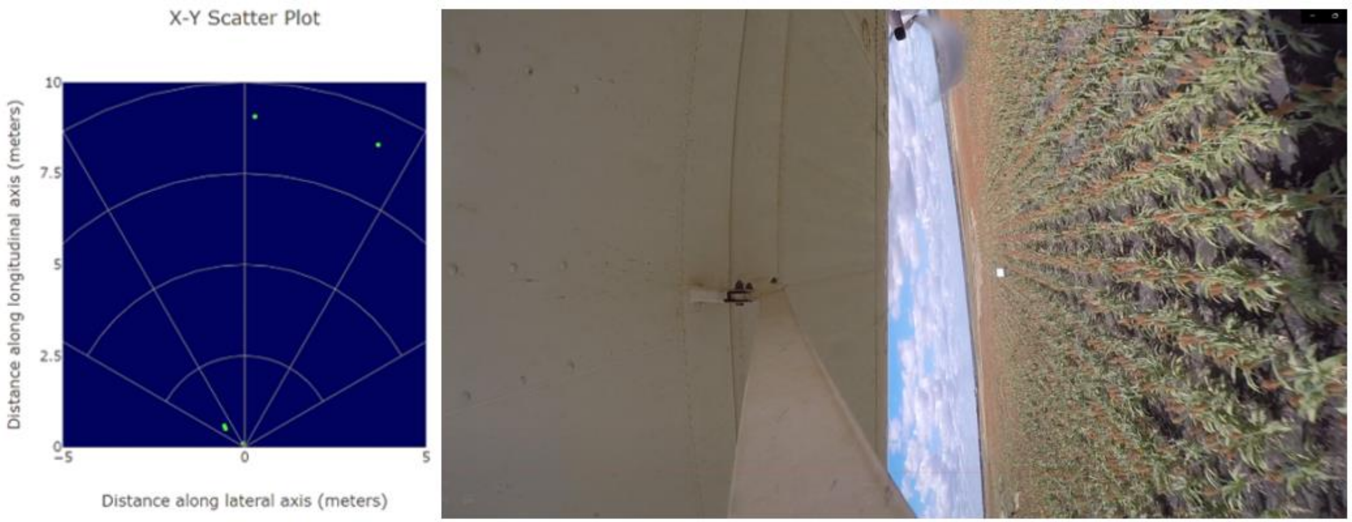


Figure 84: Flyover 8 Radar Scatter Plot and Plane Footage

Figure 85: Flyover 9 Radar Scatter Plot and Plane Footage

Figure 86: Flyover 10 Radar Scatter Plot and Plane Footage

The presented data shows a clear comparison between the MMW radar and the video footage. As previously noted, there are slight inaccuracies due to the GoPro footage not displaying the time which was recorded on the MMW radar during the flyovers. When reviewing the plane footage, it could be seen that the pulley plate was released at different times as expected to simulate different times an object may emerge from the crops. It was discovered that the pulley would frequently pop up but then hit the bar making it rebound and angle backwards. This did not provide a direct upright plate result, as can be seen in flyover 2 and 6. The data from flyover 2 was similar to the other results, but flyover 6 presented more scattered points of detection, this could have been due to the angle that the pulley plate was positioned when the plane flew past. The data from flyover 4 shows that the plate was not detected. When reviewing the plane footage, it was clear that the pulley was not released which resulted in the pulley frame not being detected at all. This was a human error and as such, this test should be removed from the total number of tests to keep the results accurate. Flyover 8 showed that the plate emerged significantly earlier than the other tests and the MMW radar was still able to detect it meaning that the results are still accurate with a greater distance. Flyover 9 did not result in any MMW radar data. As previously discussed, this could be because the application crashed six seconds after this test. Another possibility is that the plate was not very reflective at the time of the flyover as it blended into the background. It is also possible that small bug splatter as seen in the below image may have interfered with the radar sensor. Flyover 10 did not provide any results to compare.

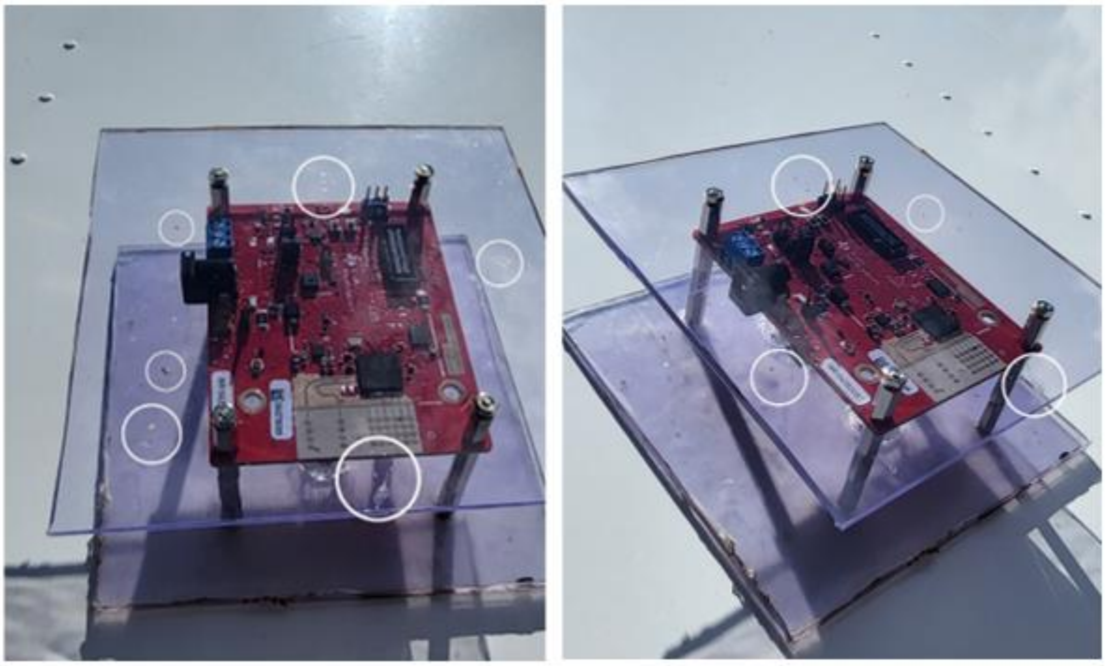


Figure 87: Foreign Debris on Perspex After Tests

Given that a total of two tests were unsuccessful due to technical and human errors, the test total will be reduced to eight. Out of the eight tests, seven successfully identified the object using the MMW radar which increased the accuracy from 77.8% as calculated in section 5.1 to 87.5%.

5.4 Additional Notes

The following pages contain images taken from the Polaris Ranger during testing for a ground perspective of the data collection and to depict how low the plane was flying. The pulley plate can be seen lowered in the crops as the plan is approaching in the first image. The second image presents the pulley plate being activated as the plane flies over.

Figure 88: Ground View of Test (1)

Figure 89: Ground View of Test (2)

5.5 Conclusions

This research paper resulted in several conclusions regarding the incorporation of an OAS into a crop-dusting aircraft. Ground testing results of the MMW radar were exceptional and provided a success rate of 100% proving that the radar is an excellent choice for object detection in certain applications. When the radar was mounted to the plane; however, the results were not as consistent. A total of ten flyovers were conducted by the crop-dusting aircraft, of which two tests were deemed inconclusive. During one of the flyovers, it was revealed that the pulley plate was not activated in time for when the plane flew over due to human error. The final flyover was also inconclusive due to the mmWave radar toolbox crashing, likely due to the laptop overheating. Of the remaining eight flyovers, seven were successful which is an equivalent of an 87.5% success rate. This is a very high success rate and given the number of total flyovers, it is a positive outcome. Although if this system were to be incorporated into an aircraft, a higher success rate would be required to ensure the system does not risk environmental damage and the pilot's life. In an attempt to increase this success rate and have more accurate data, a larger number of tests should be conducted, preferably more than 100. Due to resource restrictions; however, this was not feasible for this project.

The machine vision aspect of this paper provided results that were not expected. There were several attempts made to identify the pulley plate from the plane footage using a combination of ORB, SIFT, SURF and edge detection. A total of six main tests were conducted, five of these main tests included three sub-tests using ORB, SIFT and SURF algorithms. It was determined that the environment background in the plane footage had an incredible number of features that these algorithms focussed on instead of the plate. Most of the papers referenced would have had a definitive result due to their tests being in a plain, featureless field. A better way of trying to detect this pulley plate would be to use a featureless detection algorithm which is the opposite of the algorithms used in this paper. Despite this, the SURF algorithm responded the best in almost all tests and gave the most accurate matched feature points, especially in test 3. Test 3 required the main feature comparison image of the pulley frame to be cropped out of a different frame from the plane footage. It was able to identify several frames that found similar comparisons within the footage and frame. It was expected that these algorithms would detect features in every frame that the pulley plate was present in; however, this was not the case. The SURF algorithm in test 3 successfully identified the plate in 6 out of the 17 frames. Although this is not expected, it is still a positive outcome as a trigger signal could be integrated that alerts the pilot the moment the object is detected as opposed to a machine vision tracker highlighting the object during the flyover. The edge detection in image segmentation was successful in one frame of the video footage but this was due to the code being finely customised to give the desired outcome. It was determined that the code used on that particular frame was not useful for any of the remaining frames and as such, a more generalised code would need to be developed. It should be noted that no algorithm was developed to analyse the entirety of the plane footage.

A combination of MMW radar technology and machine vision to develop this OAS was ultimately not successful. The machine vision aspect of this paper did not provide the expected results to accurately correlate data between the MMW radar and the machine vision. This is mainly due to the crops having more features than the pulley plate in the plane footage. The MMW radar results did become clearer; however, when compared with that of the plane footage, it resulted in one of the tests being removed from the test set due to human error.

Ultimately, the MMW radar provided a successful outcome; however, further tests would need to be conducted before implementing this into an OAS for crop-dusting aircraft. The machine vision algorithms identified that the SURF algorithm was the most accurate and could be used to send a trigger signal to indicate that it has identified an object. It is expected that a featureless algorithm would be better for this application so that the features of the crops do not get detected. Due to the machine vision algorithm not working as expected, a combination of both algorithms was not achievable in this paper.

REFERENCES

Abdu, FJ, Zhang, Y, Fu, M, Li, Y & Deng, Z 2021, 'Application of deep learning on millimeter-wave radar signals: A review', *Sensors*, vol. 21, no. 6, p. 1951.

Al-Kaff, A, García, F, Martín, D, De La Escalera, A & Armingol, JM 2017, 'Obstacle detection and avoidance system based on monocular camera and size expansion algorithm for UAVs', *Sensors*, vol. 17, no. 5, p. 1061.

Barton, F 2023, 'Bird strike behind fatal crop duster crash, ATSB finds', viewed 2/3/2024,<https://www.canberratimes.com.au/story/8252385/bird-strike-behind-fatal-crop-duster-crash-atsb-finds/>.

Bay, H, Tuytelaars, T & Van Gool, L 2006, 'Surf: Speeded up robust features', *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*, Springer, pp. 404-17.

Choi, Y, Jimenez, H & Mavris, DN 2017, 'Two-layer obstacle collision avoidance with machine learning for more energy-efficient unmanned aircraft trajectories', *Robotics and Autonomous Systems*, vol. 98, pp. 158-73.

Computer Vision Demonstration Website, 2005, University of Southampton, viewed 23/6, <https://www.southampton.ac.uk/~msn/book/new_demo/opticalFlow/#:~:text=Optical%20flow%20is%20a%20technique,in%20the%20next%20image%20sequence.>.

Connolly, C 2007, 'Collision avoidance technology: from parking sensors to unmanned aircraft', *Sensor Review*, vol. 27, no. 3, pp. 182-8.

Element14 2024, *DFROBOT FIT0620*, viewed 11/5/2024, <https://au.element14.com/dfrobot/fit0620/electric-solenoid-lock-75kg-12vdc/dp/3769927>.

GoPro 2014, *GoPro Introduces HERO4: The Most Powerful GoPro Lineup, Ever*, GoPro, https://gopro.com/en/au/news/gopro-introduces-hero4-the-most-powerful-gopro-lineup-ever>.

Huang, X, Dong, X, Ma, J, Liu, K, Ahmed, S, Lin, J & Qiu, B 2021, 'The improved A* Obstacle avoidance algorithm for the plant protection UAV with millimeter wave radar and monocular camera data fusion', *Remote Sensing*, vol. 13, no. 17, p. 3364.

Instruments, T 2020, ' IWR1642 Evaluation Module (IWR1642BOOST) Single-Chip mmWave Sensing Solution', viewed 18/04/2024,<https://www.ti.com/lit/ug/swru521c/swru521c.pdf>.

Junlong, H & Feng, D 2018, 'Unified Calibration Method for Millimeter-Wave Radar and Machine Vision', *Dimensions*, vol. 50, no. 39mm, p. 83mm.

Loftus, T 2023, 'Bird strike caused Chinchilla plane crash that killed pilot Tom Grey, ATSB finds', viewed 3/3/2024,<https://www.abc.net.au/news/2023-06-29/atsb-finds-bird-strike-caused-chinchilla-plane-crash/102539612>.

Mpouziotas, D, Karvelis, P, Tsoulos, I & Stylios, C 2023, 'Automated wildlife bird detection from drone footage using computer vision techniques', *Applied Sciences*, vol. 13, no. 13, p. 7787.

OpenCV 2023, viewed 3/3/2024, <https://opencv.org/>.

Paredes, JA, Álvarez, FJ, Hansard, M & Rajab, KZ 2021, 'A Gaussian Process model for UAV localization using millimetre wave radar', *Expert Systems with Applications*, vol. 185, p. 115563.

Pedro, D, Mora, A, Carvalho, J, Azevedo, F & Fonseca, J 2020, 'Colanet: A uav collision avoidance dataset', *Technological Innovation for Life Improvement: 11th IFIP WG 5.5/SOCOLNET Advanced Doctoral Conference on Computing, Electrical and Industrial Systems, DoCEIS 2020, Costa de Caparica, Portugal, July 1–3, 2020, Proceedings 11*, Springer, pp. 53-62, viewed 9/3/2024.

Pham, H, Smolka, SA, Stoller, SD, Phan, D & Yang, J 2015, 'A survey on unmanned aerial vehicle collision avoidance systems', *arXiv preprint arXiv:1508.07723*.

Pulli, K, Baksheev, A, Kornyakov, K & Eruhimov, V 2012, 'Real-time computer vision with OpenCV', *Communications of the ACM*, vol. 55, no. 6, pp. 61-9.

Raju, D & Joseph, P 2014, 'Motion Detection and Optical Flow', vol. 5, pp. 5716-9.

Richards, B, Dayton, J, Liu, J, Enriquez, M, Gan, M & Quintana, J 2015, 'Obstacle avoidance system for UAVs using computer vision'.

Rublee, E, Rabaud, V, Konolige, K & Bradski, G 2011, 'ORB: An efficient alternative to SIFT or SURF', *2011 International conference on computer vision*, Ieee, pp. 2564-71.

Sasongko, RA, Rawikara, S & Tampubolon, HJ 2017, 'UAV obstacle avoidance algorithm based on ellipsoid geometry', *Journal of Intelligent & Robotic Systems*, vol. 88, pp. 567-81.

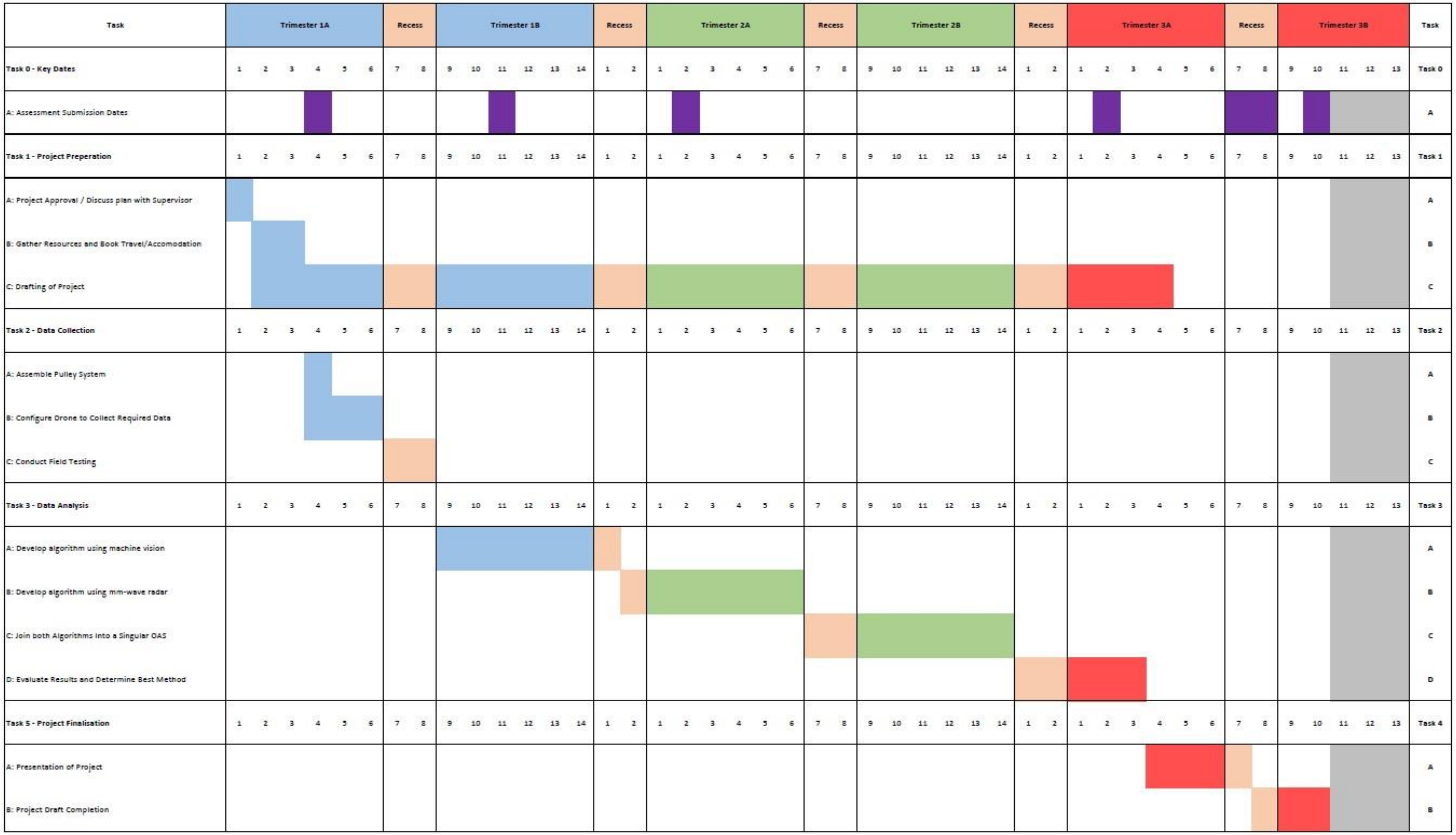
Tyagi, D 2019, *Introduction to SURF (Speeded-Up Robust Features)*, Medium, viewed 23/06, <https://medium.com/@deepanshut041/introduction-to-surf-speeded-up-robust-features-c7396d6e7c4e#:~:text=Surf%20first%20calculate%20the%20Haar,at%20that%20current%20scale%20s.>.

Viquerat, A, Blackhall, L, Reid, A, Sukkarieh, S & Brooker, G 2008, 'Reactive collision avoidance for unmanned aerial vehicles using doppler radar', *Field and service robotics: results of the 6th international conference*, Springer, pp. 245-54.

Wang, D, Li, W, Liu, X, Li, N & Zhang, C 2020, 'UAV environmental perception and autonomous obstacle avoidance: A deep learning and depth camera combined solution', *Computers and Electronics in Agriculture*, vol. 175, p. 105523.

Wu, J, Cui, Z, Sheng, VS, Zhao, P, Su, D & Gong, S 2013, 'A Comparative Study of SIFT and its Variants', *Measurement science review*, vol. 13, no. 3, pp. 122-31.

Appendix A: Project Schedule



Appendix B: Risk Assessment

UNIVERSITY OF SOUTHERN QUEENSLAND

University of Southern Queensland **Offline Version**

USQ Safety Risk Management System

Note: This is the offline version of the Safety Risk Management System (SRMS) Risk Management Plan (RMP) and is only to be used for planning and drafting sessions, and when working in remote areas or on field activities. It must be transferred to the online SRMS at the first opportunity.

Safety Risk Management Plan – Offline Version			
Assessment Title:	Journal Paper - Research Project A & B	Assessment Date:	23/03/2024
Workplace (Division/Faculty/Section):	Faculty of Engineering and Surveying	Review Date:(5 Years Max)	1/04/2024
Context			
Description:			
What is the task/event/purchase/project/procedure?	Field Data Collection of an OAS Using MM-Wave Radar Technology and Machine Vision		
Why is it being conducted?	Experiment for my Research Project		
Where is it being conducted?	Aero Professional Services		
Course code (if applicable)	ENP4111	Chemical name (if applicable)	N/A
What other nominal conditions?			
Personnel involved	Hayden Lennox		
Equipment	Crop-Dusting Plane, Pulley System, Software Applications, GoPro Camera, MM-Wave Radar, Welder		
Environment	Crop Field		
Other	N/A		
Briefly explain the procedure/process	Fly plane over Crop Field, Use Pulley to Represent Bird Flying Out of Field		
Assessment Team - who is conducting the assessment?			
Assessor(s)	Belal Yousif, Tobias Low		
Others consulted:			

Eg 1. Enter Consequence

Eg 2. Enter Probability

Eg 3. Find Action

	Consequence				
Probability	**Insignificant** No Injury 0-$5K	**Minor** First Aid $5K-$50K	**Moderate** Med Treatment $50K-$100K	**Major** Serious Injuries $100K-$250K	**Catastrophic** Death More than $250K
Almost Certain 1 in 2	M	H	E	E	E
Likely 1 in 100	M	H	H	E	E
Possible 1 in 1000	L	M	H	H	H
Unlikely 1 in 10 000	L	L	M	M	M
Rare 1 in 1 000 000	L	L	L	L	L

Recommended Action Guide

- **E**=Extreme Risk – Task ***MUST NOT*** proceed
- **H**=High Risk – Special Procedures Required (See USQSafe)
- **M**=Moderate Risk – Risk Management Plan/Work Method Statement Required
- **L**=Low Risk – Use Routine Procedures

Step 1 (cont)	**Step 2**	**Step 2a**	**Step 2b**	**Step 3**			**Step 4**				
Hazards: From step 1 or more if identified	***The Risk:*** What can happen if exposed to the hazard without existing controls in place?	***Consequence:*** What is the harm that can be caused by the hazard without existing controls in place?	***Existing Controls:*** What are the existing controls that are already in place?	***Risk Assessment:*** Consequence x Probability = Risk Level			***Additional controls:*** Enter additional controls if required to reduce the risk level	***Risk assessment with additional controls:***			
				Probability	**Risk Level**	**ALARP? Yes/no**		**Consequence**	**Probability**	**Risk Level**	**ALARP? Yes/no**
Example											
Working in temperatures over 35^0 C	Heat stress/heat stroke/exhaustion leading to serious personal injury/death	catastrophic	Regular breaks, chilled water available, loose clothing, fatigue management policy.	possible	high	No	temporary shade shelters, essential tasks only, close supervision, buddy system	catastrophic	unlikely	mod	Yes
Environmental Conditions	Damaging Hardware, personal injuries	Minor	Monitor weather conditions, do not conduct test in extreme wind or any rain	Likely	High	No	Schedule testing for month that has least severe weather conditions	Minor	Possible	Moderate	Yes
Plane colliding with person	Serious personal injury	Major	Build pulley system to raise object out of field	Possible	High	No	Maintain exclusion zone for flying area	Minor	Unlikely	Low	Yes
Snakes / Vermin	Snake and vermin bites - poison	Moderate	Maintain personal constant vigilance	Unlikely	Moderate	No	Conduct testing with other person present in case of bite	Moderate	Rare	Low	Yes
Working at desk	Ergonomic risks	Minor	Take regular breaks	Unlikely	Low	Yes		Select a consequence	Select a probability	Select a Risk Level	Yes or No
Slips, Trips and Falls	Falling over and causing personal injuries	Moderate	Maintain personal constant vigilance	Possible	High	No	Do not run and ensure other person is present	Moderate	Unlikely	Moderate	Yes
MM-wave radar and GoPro falling from plane	Damage to hardware	Minor	Ensure mm-wave radar is secured correctly	Unlikely	Low	Yes		Select a consequence	Select a probability	Select a Risk Level	Yes or No
Hot work for building pulley system	Personal injuries including burns and blindness	Moderate	Wear fire retardant long sleeve clothes, gloves and steel cap boots. Wear welding helmet	Unlikely	Low	Yes		Select a consequence	Select a probability	Select a Risk Level	Yes or No
		Select a consequence		Select a probability	Select a Risk Level	Yes or No		Select a consequence	Select a probability	Select a Risk Level	Yes or No
		Select a consequence		Select a probability	Select a Risk Level	Yes or No		Select a consequence	Select a probability	Select a Risk Level	Yes or No

Step 5 - Action Plan (for controls not already in place)			
Additional controls:	*Resources:*	*Persons responsible:*	*Proposed implementation date:*
Weather will be constantly monitored when field testing	Mobile Weather App	Hayden Lennox	1/04/2024
			Click here to enter a date.
			Click here to enter a date.
			Click here to enter a date.
			Click here to enter a date.
			Click here to enter a date.
			Click here to enter a date.
			Click here to enter a date.
			Click here to enter a date.
			Click here to enter a date.
			Click here to enter a date.
			Click here to enter a date.

Step 6 - Approval			
Drafter's name:	Hayden Lennox	Draft date:	23/03/2024
Drafter's comments:	N/A		
Approver's name:	Tobias Low	Approver's title/position:	Research Project Supervisor
Approver's comments:	Note procedures at end of Risk assessment to be followed. Please find the process that will be followed below to safely collect the data from the crop duster. Attached is also a photo of the field the crop duster will be spraying for reference. 1. Mount radar and go-pro to crop-duster with anti-vibration nord lock washers to ensure the fasteners don't come undone. 2. Discuss flight routes with pilot. 3. Load pulley system into tray of ute with another person. 4. Drive ute to field that is being sprayed.		

	5. Setup pulley system in field. 6. Run cable for locking mechanism to a safe place to activate pulley system. 7. Crop duster takes off and contact is maintained through 2-way radio system. 8. Once crop-duster has passed through line of the field, the pilot will radio through to advise when it is safe to load pulley system back onto ute with another person. 9. Move pulley system several rows up the crop field to allow enough time to relocate setup safely. 10. Radio pilot to advise system is setup. 11. Repeat steps 6-8 several times until crop field has been sprayed. 12. Pilot will advise when they are finished spraying the field and landing – pack everything up and analyse data.

I am satisfied that the risks are as low as reasonably practicable and that the resources required will be provided.			
Approver's signature:	[redacted]	Approval date:	2/04/2024

Appendix C: Budget and Resources Required

Table 2: Cost Estimate & Resource Requirements

Item	Quantity	Source	Cost	Comment
PC with Microsoft Windows	1	Student	Nil	
OpenCV, MATLAB & Radar Toolbox Software	1	UniSQ	Nil	To be installed using UniSQ student licence
Texas Instruments MMW radar	1	UniSQ	Nil	Dr Tobias Low has put aside for research project
Microsoft Word	1	UniSQ	Nil	
Ball Pulley System	1	Student	$100	
Accommodation / Travel Expenses	1	Student	$1500	
Miscellaneous / extras	1	Student	$300	Extras that may be required (i.e., to secure MMW radar to drone, cables, Perspex etc)

The total budget of project is expected to be a maximum of $1900.

Appendix D: Test 1 – ORB Matlab Code

```
% HAYDEN LENNOX
% PROFESSIONAL ENGINEERING RESEARCH PROJECT
% OBJECT DETECTION ALGORITHM
% ORB
% TEST 1- Testing Comparisons Between Two Frames
% 2024

clc
clear
close all

%% Load Frames
objectVideo = VideoReader("PLANE_FOOTAGE1.MP4");    % Load video footage from
plane
pulleyPlate = read(objectVideo,20549);    % Display 20549th frame in video
rect = [1330 690 120 120];         % Create rectangle section of image at [xmin
ymin width height]. NOTE: top left of image is 0.
pulleyPlateCropped = imcrop(pulleyPlate,rect);    % Fit rectangle around image to
only show that section of the image
pulleyPlateGS = im2gray(pulleyPlateCropped);       % Convert to gray scale
figure;
imshow(pulleyPlateGS);        % Display pulley plate

videoFrame = read(objectVideo,20550);    % Display 20550th frame in video
videoFrameGS = im2gray(videoFrame);     % Convert to gray scale
figure;
imshow(videoFrameGS);

%% Detect Point Features
framePoints = detectORBFeatures(pulleyPlateGS);      % Detect ORB features of
pulley plate
videoPoints = detectORBFeatures(videoFrameGS);    % Detect ORB features of video

figure;
imshow(pulleyPlateGS);       % Display pulley plate
title('100 Strongest Point Features from Pulley Plate');
hold on;
plot(selectStrongest(framePoints,100));     % Show 100 strongest points from the
pulley plate
hold off

figure;
imshow(videoFrameGS);       % Display video frame
title('1000 Strongest Point Features from Plane Footage');
hold on;
plot(selectStrongest(videoPoints,1000));    % Show 1000 strongest points from the
plane footage
hold off

%% Extract Feature Descriptors
[pulleyPlateFeatures, framePoints] = extractFeatures(pulleyPlateGS, framePoints);
% Extract features of interest from pulley plate
[videoFrameFeatures, videoPoints] = extractFeatures(videoFrameGS, videoPoints);
% Extract features of interest from plane footage

%% Find Putative Point Matches
```

```
pulleyPlatePairs = matchFeatures(pulleyPlateFeatures,videoFrameFeatures);   %
Match features using their descriptors

matchedPulleyPlatePoints = framePoints(pulleyPlatePairs(:,1),:);
matchedVideoFramePoints = videoPoints(pulleyPlatePairs(:,2),:);
figure;
showMatchedFeatures(pulleyPlateGS,videoFrameGS,matchedPulleyPlatePoints,matchedVid
eoFramePoints,'montage');
title('Putatively Matched Points');
```

Appendix E: Test 1 – SURF Matlab Code

```
% HAYDEN LENNOX
% PROFESSIONAL ENGINEERING RESEARCH PROJECT
% OBJECT DETECTION ALGORITHM
% SURF
% TEST 1- Testing Comparisons Between Two Frames
% 2024

clc
clear
close all

%% Load Frames
objectVideo = VideoReader("PLANE_FOOTAGE1.MP4");    % Load video footage from
plane
pulleyPlate = read(objectVideo,20549);    % Display 20549th frame in video
rect = [1330 690 120 120];         % Create rectangle section of image at [xmin
ymin width height]. NOTE: top left of image is 0.
pulleyPlateCropped = imcrop(pulleyPlate,rect);    % Fit rectangle around image to
only show that section of the image
pulleyPlateGS = im2gray(pulleyPlateCropped);       % Convert to gray scale
figure;
imshow(pulleyPlateGS);        % Display pulley plate

videoFrame = read(objectVideo,20550);    % Display 20550th frame in video
videoFrameGS = im2gray(videoFrame);     % Convert to gray scale
figure;
imshow(videoFrameGS);

%% Detect Point Features
framePoints = detectSURFFeatures(pulleyPlateGS,'MetricThreshold',100);      %
Detect SURF features of pulley plate NOTE: Metric threshold detects more blobs,
decreasing number (1-100) increases blob count
videoPoints = detectSURFFeatures(videoFrameGS,'MetricThreshold',1);    % Detect
SURF features of video

figure;
imshow(pulleyPlateGS);       % Display pulley plate
title('100 Strongest Point Features from Pulley Plate');
hold on;
plot(selectStrongest(framePoints,100));     % Show 100 strongest points from the
pulley plate
hold off

figure;
imshow(videoFrameGS);       % Display video frame
title('1000 Strongest Point Features from Plane Footage',fontSize = 20);
hold on;
plot(selectStrongest(videoPoints,1000));    % Show 1000 strongest points from the
plane footage
hold off

%% Extract Feature Descriptors
[pulleyPlateFeatures, framePoints] = extractFeatures(pulleyPlateGS, framePoints);
% Extract features of interest from pulley plate
[videoFrameFeatures, videoPoints] = extractFeatures(videoFrameGS, videoPoints);
% Extract features of interest from plane footage
```

```
%% Find Putative Point Matches
pulleyPlatePairs = matchFeatures(pulleyPlateFeatures,videoFrameFeatures);   %
Match features using their descriptors

matchedPulleyPlatePoints = framePoints(pulleyPlatePairs(:,1),:);
matchedVideoFramePoints = videoPoints(pulleyPlatePairs(:,2),:);
figure;
showMatchedFeatures(pulleyPlateGS,videoFrameGS,matchedPulleyPlatePoints,matchedVid
eoFramePoints,'montage');
title('Putatively Matched Points',fontSize = 20);
```

Appendix F: Test 1 – SIFT Matlab Code

```
% HAYDEN LENNOX
% PROFESSIONAL ENGINEERING RESEARCH PROJECT
% OBJECT DETECTION ALGORITHM
% SIFT
% TEST 1- Testing Comparisons Between Two Frames
% 2024

clc
clear
close all

%% Load Frames
objectVideo = VideoReader("PLANE_FOOTAGE1.MP4");    % Load video footage from
plane
pulleyPlate = read(objectVideo,20549);    % Display 20549th frame in video
rect = [1330 690 120 120];         % Create rectangle section of image at [xmin
ymin width height]. NOTE: top left of image is 0.
pulleyPlateCropped = imcrop(pulleyPlate,rect);    % Fit rectangle around image to
only show that section of the image
pulleyPlateGS = im2gray(pulleyPlateCropped);       % Convert to gray scale
figure;
imshow(pulleyPlateGS);        % Display pulley plate

videoFrame = read(objectVideo,20550);    % Display 20550th frame in video
videoFrameGS = im2gray(videoFrame);     % Convert to gray scale
figure;
imshow(videoFrameGS);

%% Detect Point Features
framePoints = detectSIFTFeatures(pulleyPlateGS);      % Detect SIFT features of
pulley plate
videoPoints = detectSIFTFeatures(videoFrameGS);    % Detect SIFT features of video

figure;
imshow(pulleyPlateGS);       % Display pulley plate
title('100 Strongest Point Features from Pulley Plate');
hold on;
plot(selectStrongest(framePoints,100));     % Show 100 strongest points from the
pulley plate
hold off

figure;
imshow(videoFrameGS);       % Display video frame
title('1000 Strongest Point Features from Plane Footage',fontSize = 20);
hold on;
plot(selectStrongest(videoPoints,1000));    % Show 1000 strongest points from the
plane footage
hold off

%% Extract Feature Descriptors
[pulleyPlateFeatures, framePoints] = extractFeatures(pulleyPlateGS, framePoints);
% Extract features of interest from pulley plate
[videoFrameFeatures, videoPoints] = extractFeatures(videoFrameGS, videoPoints);
% Extract features of interest from plane footage

%% Find Putative Point Matches
```

```
pulleyPlatePairs = matchFeatures(pulleyPlateFeatures,videoFrameFeatures);   %
Match features using their descriptors

matchedPulleyPlatePoints = framePoints(pulleyPlatePairs(:,1),:);
matchedVideoFramePoints = videoPoints(pulleyPlatePairs(:,2),:);
figure;
showMatchedFeatures(pulleyPlateGS,videoFrameGS,matchedPulleyPlatePoints,matchedVid
eoFramePoints,'montage');
title('Putatively Matched Points',fontSize = 20);
```

Appendix G: Test 3 – ORB Matlab Code

```
% HAYDEN LENNOX
% PROFESSIONAL ENGINEERING RESEARCH PROJECT
% OBJECT DETECTION ALGORITHM
% ORB
% TEST 3
% 2024

clc
clear
close all

%% Load Frames
objectVideo = VideoReader("PLANE_FOOTAGE1.MP4");    % Load video footage from
plane
pulleyPlate = read(objectVideo,20545);    % Display 20549th frame in video
% rect = [1207 590 69 69];       % Smaller test area only shows 1 point of
interest using ORB
rect = [1190 570 90 90];         % Create rectangle section of image at [xmin ymin
width height]. NOTE: top left of image is 0.
pulleyPlateCropped = imcrop(pulleyPlate,rect);    % Fit rectangle around image to
only show that section of the image
pulleyPlateGS = im2gray(pulleyPlateCropped);       % Convert to gray scale
figure;
imshow(pulleyPlateGS);        % Display pulley plate

videoFrame = read(objectVideo,20551);    % Display 20550th frame in video
videoFrameGS = im2gray(videoFrame);     % Convert to gray scale
figure;
imshow(videoFrameGS);

%% Detect Point Features
framePoints = detectORBFeatures(pulleyPlateGS);      % Detect ORB features of
pulley plate
videoPoints = detectORBFeatures(videoFrameGS);    % Detect ORB features of video

figure;
imshow(pulleyPlateGS);       % Display pulley plate
title('100 Strongest Point Features from Pulley Plate');
hold on;
plot(selectStrongest(framePoints,100));     % Show 100 strongest points from the
pulley plate
hold off

figure;
imshow(videoFrameGS);       % Display video frame
title('1000 Strongest Point Features from Plane Footage');
hold on;
plot(selectStrongest(videoPoints,1000));    % Show 1000 strongest points from the
plane footage
hold off

%% Extract Feature Descriptors
[pulleyPlateFeatures, framePoints] = extractFeatures(pulleyPlateGS, framePoints);
% Extract features of interest from pulley plate
[videoFrameFeatures, videoPoints] = extractFeatures(videoFrameGS, videoPoints);
% Extract features of interest from plane footage
```

```
%% Find Putative Point Matches
pulleyPlatePairs = matchFeatures(pulleyPlateFeatures,videoFrameFeatures);   %
Match features using their descriptors

matchedPulleyPlatePoints = framePoints(pulleyPlatePairs(:,1),:);
matchedVideoFramePoints = videoPoints(pulleyPlatePairs(:,2),:);
figure;
showMatchedFeatures(pulleyPlateGS,videoFrameGS,matchedPulleyPlatePoints,matchedVid
eoFramePoints,'montage');
title('Putatively Matched Points');
```

Appendix H: Test 3 – SURF Matlab Code

```
% HAYDEN LENNOX
% PROFESSIONAL ENGINEERING RESEARCH PROJECT
% OBJECT DETECTION ALGORITHM
% SURF
% TEST 3 – Altering Frame Number from Video Footage
% 2024

clc
clear
close all

%% Load Frames
objectVideo = VideoReader("PLANE_FOOTAGE1.MP4");    % Load video footage from
plane
pulleyPlate = read(objectVideo,20545);    % Display 20549th frame in video
% rect = [1207 590 69 69];       % Smaller test area only shows 1 point of
interest using SURF
rect = [1190 570 90 90];         % Create rectangle section of image at [xmin ymin
width height]. NOTE: top left of image is 0.
pulleyPlateCropped = imcrop(pulleyPlate,rect);    % Fit rectangle around image to
only show that section of the image
pulleyPlateGS = im2gray(pulleyPlateCropped);       % Convert to gray scale
figure;
imshow(pulleyPlateGS);        % Display pulley plate

videoFrame = read(objectVideo,20544);    % Variable frame adjustment between
20534-20551
videoFrameGS = im2gray(videoFrame);     % Convert to gray scale
figure;
imshow(videoFrameGS);

%% Detect Point Features
framePoints = detectSURFFeatures(pulleyPlateGS,'MetricThreshold',100);      %
Detect SURF features of pulley plate NOTE: Metric threshold detects more blobs,
decreasing number (1-100) increases blob count
videoPoints = detectSURFFeatures(videoFrameGS,'MetricThreshold',1);    % Detect
SURF features of video

figure;
imshow(pulleyPlateGS);       % Display pulley plate
title('100 Strongest Point Features from Pulley Plate');
hold on;
plot(selectStrongest(framePoints,100));     % Show 100 strongest points from the
pulley plate
hold off

figure;
imshow(videoFrameGS);       % Display video frame
title('1000 Strongest Point Features from Plane Footage',fontSize = 20);
hold on;
plot(selectStrongest(videoPoints,1000));    % Show 1000 strongest points from the
plane footage
hold off

%% Extract Feature Descriptors
```

```
[pulleyPlateFeatures, framePoints] = extractFeatures(pulleyPlateGS, framePoints);
% Extract features of interest from pulley plate
[videoFrameFeatures, videoPoints] = extractFeatures(videoFrameGS, videoPoints);
% Extract features of interest from plane footage

%% Find Putative Point Matches
pulleyPlatePairs = matchFeatures(pulleyPlateFeatures,videoFrameFeatures);   %
Match features using their descriptors

matchedPulleyPlatePoints = framePoints(pulleyPlatePairs(:,1),:);
matchedVideoFramePoints = videoPoints(pulleyPlatePairs(:,2),:);
figure;
showMatchedFeatures(pulleyPlateGS,videoFrameGS,matchedPulleyPlatePoints,matchedVid
eoFramePoints,'montage');
title('Putatively Matched Points',fontSize = 20);
```

Appendix I: Test 3 – SIFT Matlab Code

```
% HAYDEN LENNOX
% PROFESSIONAL ENGINEERING RESEARCH PROJECT
% OBJECT DETECTION ALGORITHM
% SIFT
% TEST 3 - Altering Frame Number from Video Footage
% 2024

clc
clear
close all

%% Load Frames
objectVideo = VideoReader("PLANE_FOOTAGE1.MP4");    % Load video footage from
plane
pulleyPlate = read(objectVideo,20545);    % Display 20549th frame in video
rect = [1200 580 69 69];       % Smaller test area only shows plenty points of
interest using SIFT
% rect = [1190 570 90 90];         % Create rectangle section of image at [xmin
ymin width height]. NOTE: top left of image is 0.
pulleyPlateCropped = imcrop(pulleyPlate,rect);    % Fit rectangle around image to
only show that section of the image
pulleyPlateGS = im2gray(pulleyPlateCropped);       % Convert to gray scale
figure;
imshow(pulleyPlateGS);        % Display pulley plate

videoFrame = read(objectVideo,20551);    % Variable frame adjustment between
20534-20551
videoFrameGS = im2gray(videoFrame);     % Convert to gray scale
figure;
imshow(videoFrameGS);

%% Detect Point Features
framePoints = detectSIFTFeatures(pulleyPlateGS);      % Detect SIFT features of
pulley plate
videoPoints = detectSIFTFeatures(videoFrameGS);    % Detect SIFT features of video

figure;
imshow(pulleyPlateGS);       % Display pulley plate
title('100 Strongest Point Features from Pulley Plate');
hold on;
plot(selectStrongest(framePoints,100));     % Show 100 strongest points from the
pulley plate
hold off

figure;
imshow(videoFrameGS);       % Display video frame
title('1000 Strongest Point Features from Plane Footage',fontSize = 20);
hold on;
plot(selectStrongest(videoPoints,1000));    % Show 1000 strongest points from the
plane footage
hold off

%% Extract Feature Descriptors
[pulleyPlateFeatures, framePoints] = extractFeatures(pulleyPlateGS, framePoints);
% Extract features of interest from pulley plate
```

```
[videoFrameFeatures, videoPoints] = extractFeatures(videoFrameGS, videoPoints);
% Extract features of interest from plane footage

%% Find Putative Point Matches
pulleyPlatePairs = matchFeatures(pulleyPlateFeatures,videoFrameFeatures);   %
Match features using their descriptors

matchedPulleyPlatePoints = framePoints(pulleyPlatePairs(:,1),:);
matchedVideoFramePoints = videoPoints(pulleyPlatePairs(:,2),:);
figure;
showMatchedFeatures(pulleyPlateGS,videoFrameGS,matchedPulleyPlatePoints,matchedVid
eoFramePoints,'montage');
title('Putatively Matched Points',fontSize = 20);
```

Appendix J: Test 4 – ORB Matlab Code

```
% HAYDEN LENNOX
% PROFESSIONAL ENGINEERING RESEARCH PROJECT
% OBJECT DETECTION ALGORITHM
% ORB
% TEST 4 – Segmented Image Frames and Altering Frame Number from Video Footage
% 2024

clc
clear
close all

%% Load Frames
objectVideo = VideoReader("PLANE_FOOTAGE1.MP4");    % Load video footage from
plane
pulleyPlate = read(objectVideo,20545);    % Display 20549th frame in video
% rect = [1207 590 69 69];       % Smaller test area only shows 1 point of
interest using ORB
rect = [1190 570 90 90];         % Create rectangle section of image at [xmin ymin
width height]. NOTE: top left of image is 0.
pulleyPlateCropped = imcrop(pulleyPlate,rect);    % Fit rectangle around image to
only show that section of the image
pulleyPlateGS = im2gray(pulleyPlateCropped);       % Convert to gray scale
figure;
imshow(pulleyPlateGS);        % Display pulley plate

videoFrame = read(objectVideo,20549);    % Variable frame adjustment between
20534-20551
size(videoFrame)
rect2 = [1000 0 920 1080];      % Note that frame size is 1080x1920
halfFrame = imcrop(videoFrame,rect2);    % Crop out plane wing in video frame
halfFrameGS = im2gray(halfFrame);     % Convert to gray scale
% Create segments from image
rect3 = [0 0 460 540];
rect4 = [0 540 460 540];
rect5 = [460 0 460 540];
rect6 = [460 540 460 540];
% Apply segments to image
segment1 = imcrop(halfFrameGS,rect3);
segment2 = imcrop(halfFrameGS,rect4);
segment3 = imcrop(halfFrameGS,rect5);
segment4 = imcrop(halfFrameGS,rect6);
figure;
imshow(halfFrameGS);
% Display segments in 1 figure to visualise segments
figure;
subplot(2,2,1); imshow(segment1);
subplot(2,2,3); imshow(segment2);
subplot(2,2,2); imshow(segment3);
subplot(2,2,4); imshow(segment4);

%% Detect Point Features Segment 1
framePoints = detectORBFeatures(pulleyPlateGS);      % Detect ORB features of
pulley plate
videoPoints1 = detectORBFeatures(segment1);     % Detect ORB features of video

figure;
```

```
imshow(pulleyPlateGS);        % Display pulley plate
title('100 Strongest Point Features from Pulley Plate');
hold on;
plot(selectStrongest(framePoints,100));     % Show 100 strongest points from the
pulley plate
hold off

figure;
imshow(segment1);       % Display video frame
title('1000 Strongest Point Features from Plane Footage');
hold on;
plot(selectStrongest(videoPoints1,1000));    % Show 1000 strongest points from the
plane footage
hold off

%% Extract Feature Descriptors Segment 1
[pulleyPlateFeatures, framePoints] = extractFeatures(pulleyPlateGS, framePoints);
% Extract features of interest from pulley plate
[videoFrameFeatures, videoPoints1] = extractFeatures(segment1, videoPoints1);
% Extract features of interest from plane footage

%% Find Putative Point Matches Segment 1
pulleyPlatePairs = matchFeatures(pulleyPlateFeatures,videoFrameFeatures);   %
Match features using their descriptors

matchedPulleyPlatePoints = framePoints(pulleyPlatePairs(:,1),:);
matchedVideoFramePoints = videoPoints1(pulleyPlatePairs(:,2),:);
figure;
showMatchedFeatures(pulleyPlateGS,segment1,matchedPulleyPlatePoints,matchedVideoFr
amePoints,'montage');
title('Putatively Matched Points');

%% Detect Point Features Segment 2
framePoints = detectORBFeatures(pulleyPlateGS);      % Detect ORB features of
pulley plate
videoPoints2 = detectORBFeatures(segment2);    % Detect ORB features of video

figure;
imshow(pulleyPlateGS);        % Display pulley plate
title('100 Strongest Point Features from Pulley Plate');
hold on;
plot(selectStrongest(framePoints,100));     % Show 100 strongest points from the
pulley plate
hold off

figure;
imshow(segment2);       % Display video frame
title('1000 Strongest Point Features from Plane Footage');
hold on;
plot(selectStrongest(videoPoints2,1000));    % Show 1000 strongest points from the
plane footage
hold off

%% Extract Feature Descriptors Segment 2
[pulleyPlateFeatures, framePoints] = extractFeatures(pulleyPlateGS, framePoints);
% Extract features of interest from pulley plate
[videoFrameFeatures, videoPoints2] = extractFeatures(segment2, videoPoints2);
% Extract features of interest from plane footage
```

```
%% Find Putative Point Matches Segment 2
pulleyPlatePairs = matchFeatures(pulleyPlateFeatures,videoFrameFeatures);   %
Match features using their descriptors

matchedPulleyPlatePoints = framePoints(pulleyPlatePairs(:,1),:);
matchedVideoFramePoints = videoPoints2(pulleyPlatePairs(:,2),:);
figure;
showMatchedFeatures(pulleyPlateGS,segment2,matchedPulleyPlatePoints,matchedVideoFr
amePoints,'montage');
title('Putatively Matched Points');

%% Detect Point Features Segment 3
framePoints = detectORBFeatures(pulleyPlateGS);      % Detect ORB features of
pulley plate
videoPoints3 = detectORBFeatures(segment3);    % Detect ORB features of video

figure;
imshow(pulleyPlateGS);       % Display pulley plate
title('100 Strongest Point Features from Pulley Plate');
hold on;
plot(selectStrongest(framePoints,100));     % Show 100 strongest points from the
pulley plate
hold off

figure;
imshow(segment3);       % Display video frame
title('1000 Strongest Point Features from Plane Footage');
hold on;
plot(selectStrongest(videoPoints3,1000));    % Show 1000 strongest points from the
plane footage
hold off

%% Extract Feature Descriptors Segment 3
[pulleyPlateFeatures, framePoints] = extractFeatures(pulleyPlateGS, framePoints);
% Extract features of interest from pulley plate
[videoFrameFeatures, videoPoints3] = extractFeatures(segment3, videoPoints3);
% Extract features of interest from plane footage

%% Find Putative Point Matches Segment 3
pulleyPlatePairs = matchFeatures(pulleyPlateFeatures,videoFrameFeatures);   %
Match features using their descriptors

matchedPulleyPlatePoints = framePoints(pulleyPlatePairs(:,1),:);
matchedVideoFramePoints = videoPoints3(pulleyPlatePairs(:,2),:);
figure;
showMatchedFeatures(pulleyPlateGS,segment3,matchedPulleyPlatePoints,matchedVideoFr
amePoints,'montage');
title('Putatively Matched Points');

%% Detect Point Features Segment 4
framePoints = detectORBFeatures(pulleyPlateGS);      % Detect ORB features of
pulley plate
videoPoints4 = detectORBFeatures(segment4);    % Detect ORB features of video

figure;
imshow(pulleyPlateGS);       % Display pulley plate
title('100 Strongest Point Features from Pulley Plate');
hold on;
```

```
plot(selectStrongest(framePoints,100));     % Show 100 strongest points from the
pulley plate
hold off

figure;
imshow(segment4);       % Display video frame
title('1000 Strongest Point Features from Plane Footage');
hold on;
plot(selectStrongest(videoPoints4,1000));    % Show 1000 strongest points from the
plane footage
hold off

%% Extract Feature Descriptors Segment 4
[pulleyPlateFeatures, framePoints] = extractFeatures(pulleyPlateGS, framePoints);
% Extract features of interest from pulley plate
[videoFrameFeatures, videoPoints4] = extractFeatures(segment4, videoPoints4);
% Extract features of interest from plane footage

%% Find Putative Point Matches Segment 4
pulleyPlatePairs = matchFeatures(pulleyPlateFeatures,videoFrameFeatures);   %
Match features using their descriptors

matchedPulleyPlatePoints = framePoints(pulleyPlatePairs(:,1),:);
matchedVideoFramePoints = videoPoints4(pulleyPlatePairs(:,2),:);
figure;
showMatchedFeatures(pulleyPlateGS,segment4,matchedPulleyPlatePoints,matchedVideoFr
amePoints,'montage');
title('Putatively Matched Points');
```

Appendix K: Test 4 – SURF Matlab Code

```
% HAYDEN LENNOX
% PROFESSIONAL ENGINEERING RESEARCH PROJECT
% OBJECT DETECTION ALGORITHM
% SURF
% TEST 4 - Segmented Image Frames and Altering Frame Number from Video Footage
% 2024

clc
clear
close all

%% Load Frames
objectVideo = VideoReader("PLANE_FOOTAGE1.MP4");    % Load video footage from
plane
pulleyPlate = read(objectVideo,20545);    % Display 20549th frame in video
% rect = [1207 590 69 69];       % Smaller test area only shows 1 point of
interest using SURF
rect = [1190 570 90 90];         % Create rectangle section of image at [xmin ymin
width height]. NOTE: top left of image is 0.
pulleyPlateCropped = imcrop(pulleyPlate,rect);    % Fit rectangle around image to
only show that section of the image
pulleyPlateGS = im2gray(pulleyPlateCropped);       % Convert to gray scale
figure;
imshow(pulleyPlateGS);        % Display pulley plate

videoFrame = read(objectVideo,20551);    % Variable frame adjustment between
20534-20551
size(videoFrame)
rect2 = [1000 0 920 1080];      % Note that frame size is 1080x1920
halfFrame = imcrop(videoFrame,rect2);   % Crop out plane wing in video frame
halfFrameGS = im2gray(halfFrame);     % Convert to gray scale
% Create segments from image
rect3 = [0 0 460 540];
rect4 = [0 540 460 540];
rect5 = [460 0 460 540];
rect6 = [460 540 460 540];
% Apply segments to image
segment1 = imcrop(halfFrameGS,rect3);
segment2 = imcrop(halfFrameGS,rect4);
segment3 = imcrop(halfFrameGS,rect5);
segment4 = imcrop(halfFrameGS,rect6);
figure;
imshow(halfFrameGS);
% Display segments in 1 figure to visualise segments
figure;
subplot(2,2,1); imshow(segment1);
subplot(2,2,3); imshow(segment2);
subplot(2,2,2); imshow(segment3);
subplot(2,2,4); imshow(segment4);

%% Detect Point Features Segment 1
framePoints = detectSURFFeatures(pulleyPlateGS);      % Detect SURF features of
pulley plate
videoPoints1 = detectSURFFeatures(segment1);    % Detect SURF features of video

figure;
```

```
imshow(pulleyPlateGS);       % Display pulley plate
title('100 Strongest Point Features from Pulley Plate');
hold on;
plot(selectStrongest(framePoints,100));     % Show 100 strongest points from the
pulley plate
hold off

figure;
imshow(segment1);       % Display video frame
title('1000 Strongest Point Features from Plane Footage');
hold on;
plot(selectStrongest(videoPoints1,1000));    % Show 1000 strongest points from the
plane footage
hold off

%% Extract Feature Descriptors Segment 1
[pulleyPlateFeatures, framePoints] = extractFeatures(pulleyPlateGS, framePoints);
% Extract features of interest from pulley plate
[videoFrameFeatures, videoPoints1] = extractFeatures(segment1, videoPoints1);
% Extract features of interest from plane footage

%% Find Putative Point Matches Segment 1
pulleyPlatePairs = matchFeatures(pulleyPlateFeatures,videoFrameFeatures);   %
Match features using their descriptors

matchedPulleyPlatePoints = framePoints(pulleyPlatePairs(:,1),:);
matchedVideoFramePoints = videoPoints1(pulleyPlatePairs(:,2),:);
figure;
showMatchedFeatures(pulleyPlateGS,segment1,matchedPulleyPlatePoints,matchedVideoFr
amePoints,'montage');
title('Putatively Matched Points');

%% Detect Point Features Segment 2
framePoints = detectSURFFeatures(pulleyPlateGS);      % Detect SURF features of
pulley plate
videoPoints2 = detectSURFFeatures(segment2);    % Detect SURF features of video

figure;
imshow(pulleyPlateGS);       % Display pulley plate
title('100 Strongest Point Features from Pulley Plate');
hold on;
plot(selectStrongest(framePoints,100));     % Show 100 strongest points from the
pulley plate
hold off

figure;
imshow(segment2);       % Display video frame
title('1000 Strongest Point Features from Plane Footage');
hold on;
plot(selectStrongest(videoPoints2,1000));    % Show 1000 strongest points from the
plane footage
hold off

%% Extract Feature Descriptors Segment 2
[pulleyPlateFeatures, framePoints] = extractFeatures(pulleyPlateGS, framePoints);
% Extract features of interest from pulley plate
[videoFrameFeatures, videoPoints2] = extractFeatures(segment2, videoPoints2);
% Extract features of interest from plane footage
```

```
%% Find Putative Point Matches Segment 2
pulleyPlatePairs = matchFeatures(pulleyPlateFeatures,videoFrameFeatures);   %
Match features using their descriptors

matchedPulleyPlatePoints = framePoints(pulleyPlatePairs(:,1),:);
matchedVideoFramePoints = videoPoints2(pulleyPlatePairs(:,2),:);
figure;
showMatchedFeatures(pulleyPlateGS,segment2,matchedPulleyPlatePoints,matchedVideoFr
amePoints,'montage');
title('Putatively Matched Points');

%% Detect Point Features Segment 3
framePoints = detectSURFFeatures(pulleyPlateGS);      % Detect SURF features of
pulley plate
videoPoints3 = detectSURFFeatures(segment3);    % Detect SURF features of video

figure;
imshow(pulleyPlateGS);       % Display pulley plate
title('100 Strongest Point Features from Pulley Plate');
hold on;
plot(selectStrongest(framePoints,100));     % Show 100 strongest points from the
pulley plate
hold off

figure;
imshow(segment3);       % Display video frame
title('1000 Strongest Point Features from Plane Footage');
hold on;
plot(selectStrongest(videoPoints3,1000));    % Show 1000 strongest points from the
plane footage
hold off

%% Extract Feature Descriptors Segment 3
[pulleyPlateFeatures, framePoints] = extractFeatures(pulleyPlateGS, framePoints);
% Extract features of interest from pulley plate
[videoFrameFeatures, videoPoints3] = extractFeatures(segment3, videoPoints3);
% Extract features of interest from plane footage

%% Find Putative Point Matches Segment 3
pulleyPlatePairs = matchFeatures(pulleyPlateFeatures,videoFrameFeatures);   %
Match features using their descriptors

matchedPulleyPlatePoints = framePoints(pulleyPlatePairs(:,1),:);
matchedVideoFramePoints = videoPoints3(pulleyPlatePairs(:,2),:);
figure;
showMatchedFeatures(pulleyPlateGS,segment3,matchedPulleyPlatePoints,matchedVideoFr
amePoints,'montage');
title('Putatively Matched Points');

%% Detect Point Features Segment 4
framePoints = detectSURFFeatures(pulleyPlateGS);      % Detect SURF features of
pulley plate
videoPoints4 = detectSURFFeatures(segment4);    % Detect SURF features of video

figure;
imshow(pulleyPlateGS);       % Display pulley plate
title('100 Strongest Point Features from Pulley Plate');
hold on;
```

```
plot(selectStrongest(framePoints,100));     % Show 100 strongest points from the
pulley plate
hold off

figure;
imshow(segment4);       % Display video frame
title('1000 Strongest Point Features from Plane Footage');
hold on;
plot(selectStrongest(videoPoints4,1000));    % Show 1000 strongest points from the
plane footage
hold off

%% Extract Feature Descriptors Segment 4
[pulleyPlateFeatures, framePoints] = extractFeatures(pulleyPlateGS, framePoints);
% Extract features of interest from pulley plate
[videoFrameFeatures, videoPoints4] = extractFeatures(segment4, videoPoints4);
% Extract features of interest from plane footage

%% Find Putative Point Matches Segment 4
pulleyPlatePairs = matchFeatures(pulleyPlateFeatures,videoFrameFeatures);   %
Match features using their descriptors

matchedPulleyPlatePoints = framePoints(pulleyPlatePairs(:,1),:);
matchedVideoFramePoints = videoPoints4(pulleyPlatePairs(:,2),:);
figure;
showMatchedFeatures(pulleyPlateGS,segment4,matchedPulleyPlatePoints,matchedVideoFr
amePoints,'montage');
title('Putatively Matched Points');
```

Appendix L: Test 4 – SIFT Matlab Code

```
% HAYDEN LENNOX
% PROFESSIONAL ENGINEERING RESEARCH PROJECT
% OBJECT DETECTION ALGORITHM
% SIFT
% TEST 4 – Segmented Image Frames and Altering Frame Number from Video Footage
% 2024

clc
clear
close all

%% Load Frames
objectVideo = VideoReader("PLANE_FOOTAGE1.MP4");    % Load video footage from
plane
pulleyPlate = read(objectVideo,20545);    % Display 20549th frame in video
rect = [1200 580 69 69];       % Smaller test area only shows plenty points of
interest using SIFT
% rect = [1190 570 90 90];         % Create rectangle section of image at [xmin
ymin width height]. NOTE: top left of image is 0.
pulleyPlateCropped = imcrop(pulleyPlate,rect);    % Fit rectangle around image to
only show that section of the image
pulleyPlateGS = im2gray(pulleyPlateCropped);       % Convert to gray scale
figure;
imshow(pulleyPlateGS);        % Display pulley plate

videoFrame = read(objectVideo,20551);    % Variable frame adjustment between
20534-20551
size(videoFrame)
rect2 = [1000 0 920 1080];      % Note that frame size is 1080x1920
halfFrame = imcrop(videoFrame,rect2);    % Crop out plane wing in video frame
halfFrameGS = im2gray(halfFrame);     % Convert to gray scale
% Create segments from image
rect3 = [0 0 460 540];
rect4 = [0 540 460 540];
rect5 = [460 0 460 540];
rect6 = [460 540 460 540];
% Apply segments to image
segment1 = imcrop(halfFrameGS,rect3);
segment2 = imcrop(halfFrameGS,rect4);
segment3 = imcrop(halfFrameGS,rect5);
segment4 = imcrop(halfFrameGS,rect6);
figure;
imshow(halfFrameGS);
% Display segments in 1 figure to visualise segments
figure;
subplot(2,2,1); imshow(segment1);
subplot(2,2,3); imshow(segment2);
subplot(2,2,2); imshow(segment3);
subplot(2,2,4); imshow(segment4);

%% Detect Point Features Segment 1
framePoints = detectSIFTFeatures(pulleyPlateGS);      % Detect SIFT features of
pulley plate
videoPoints1 = detectSIFTFeatures(segment1);     % Detect SIFT features of video

figure;
```

```
imshow(pulleyPlateGS);       % Display pulley plate
title('100 Strongest Point Features from Pulley Plate');
hold on;
plot(selectStrongest(framePoints,100));     % Show 100 strongest points from the
pulley plate
hold off

figure;
imshow(segment1);       % Display video frame
title('1000 Strongest Point Features from Plane Footage');
hold on;
plot(selectStrongest(videoPoints1,1000));    % Show 1000 strongest points from the
plane footage
hold off

%% Extract Feature Descriptors Segment 1
[pulleyPlateFeatures, framePoints] = extractFeatures(pulleyPlateGS, framePoints);
% Extract features of interest from pulley plate
[videoFrameFeatures, videoPoints1] = extractFeatures(segment1, videoPoints1);
% Extract features of interest from plane footage

%% Find Putative Point Matches Segment 1
pulleyPlatePairs = matchFeatures(pulleyPlateFeatures,videoFrameFeatures);   %
Match features using their descriptors

matchedPulleyPlatePoints = framePoints(pulleyPlatePairs(:,1),:);
matchedVideoFramePoints = videoPoints1(pulleyPlatePairs(:,2),:);
figure;
showMatchedFeatures(pulleyPlateGS,segment1,matchedPulleyPlatePoints,matchedVideoFr
amePoints,'montage');
title('Putatively Matched Points');

%% Detect Point Features Segment 2
framePoints = detectSIFTFeatures(pulleyPlateGS);      % Detect SIFT features of
pulley plate
videoPoints2 = detectSIFTFeatures(segment2);    % Detect SIFT features of video

figure;
imshow(pulleyPlateGS);       % Display pulley plate
title('100 Strongest Point Features from Pulley Plate');
hold on;
plot(selectStrongest(framePoints,100));     % Show 100 strongest points from the
pulley plate
hold off

figure;
imshow(segment2);       % Display video frame
title('1000 Strongest Point Features from Plane Footage');
hold on;
plot(selectStrongest(videoPoints2,1000));    % Show 1000 strongest points from the
plane footage
hold off

%% Extract Feature Descriptors Segment 2
[pulleyPlateFeatures, framePoints] = extractFeatures(pulleyPlateGS, framePoints);
% Extract features of interest from pulley plate
[videoFrameFeatures, videoPoints2] = extractFeatures(segment2, videoPoints2);
% Extract features of interest from plane footage
```

```
%% Find Putative Point Matches Segment 2
pulleyPlatePairs = matchFeatures(pulleyPlateFeatures,videoFrameFeatures);   %
Match features using their descriptors

matchedPulleyPlatePoints = framePoints(pulleyPlatePairs(:,1),:);
matchedVideoFramePoints = videoPoints2(pulleyPlatePairs(:,2),:);
figure;
showMatchedFeatures(pulleyPlateGS,segment2,matchedPulleyPlatePoints,matchedVideoFr
amePoints,'montage');
title('Putatively Matched Points');

%% Detect Point Features Segment 3
framePoints = detectSIFTFeatures(pulleyPlateGS);      % Detect SIFT features of
pulley plate
videoPoints3 = detectSIFTFeatures(segment3);    % Detect SIFT features of video

figure;
imshow(pulleyPlateGS);       % Display pulley plate
title('100 Strongest Point Features from Pulley Plate');
hold on;
plot(selectStrongest(framePoints,100));     % Show 100 strongest points from the
pulley plate
hold off

figure;
imshow(segment3);       % Display video frame
title('1000 Strongest Point Features from Plane Footage');
hold on;
plot(selectStrongest(videoPoints3,1000));    % Show 1000 strongest points from the
plane footage
hold off

%% Extract Feature Descriptors Segment 3
[pulleyPlateFeatures, framePoints] = extractFeatures(pulleyPlateGS, framePoints);
% Extract features of interest from pulley plate
[videoFrameFeatures, videoPoints3] = extractFeatures(segment3, videoPoints3);
% Extract features of interest from plane footage

%% Find Putative Point Matches Segment 3
pulleyPlatePairs = matchFeatures(pulleyPlateFeatures,videoFrameFeatures);   %
Match features using their descriptors

matchedPulleyPlatePoints = framePoints(pulleyPlatePairs(:,1),:);
matchedVideoFramePoints = videoPoints3(pulleyPlatePairs(:,2),:);
figure;
showMatchedFeatures(pulleyPlateGS,segment3,matchedPulleyPlatePoints,matchedVideoFr
amePoints,'montage');
title('Putatively Matched Points');

%% Detect Point Features Segment 4
framePoints = detectSIFTFeatures(pulleyPlateGS);      % Detect SIFT features of
pulley plate
videoPoints4 = detectSIFTFeatures(segment4);    % Detect SIFT features of video

figure;
imshow(pulleyPlateGS);       % Display pulley plate
title('100 Strongest Point Features from Pulley Plate');
hold on;
```

```
plot(selectStrongest(framePoints,100));     % Show 100 strongest points from the
pulley plate
hold off

figure;
imshow(segment4);       % Display video frame
title('1000 Strongest Point Features from Plane Footage');
hold on;
plot(selectStrongest(videoPoints4,1000));    % Show 1000 strongest points from the
plane footage
hold off

%% Extract Feature Descriptors Segment 4
[pulleyPlateFeatures, framePoints] = extractFeatures(pulleyPlateGS, framePoints);
% Extract features of interest from pulley plate
[videoFrameFeatures, videoPoints4] = extractFeatures(segment4, videoPoints4);
% Extract features of interest from plane footage

%% Find Putative Point Matches Segment 4
pulleyPlatePairs = matchFeatures(pulleyPlateFeatures,videoFrameFeatures);   %
Match features using their descriptors

matchedPulleyPlatePoints = framePoints(pulleyPlatePairs(:,1),:);
matchedVideoFramePoints = videoPoints4(pulleyPlatePairs(:,2),:);
figure;
showMatchedFeatures(pulleyPlateGS,segment4,matchedPulleyPlatePoints,matchedVideoFr
amePoints,'montage');
title('Putatively Matched Points');
```

Appendix M: Test 5 – ORB Matlab Code

```
% HAYDEN LENNOX
% PROFESSIONAL ENGINEERING RESEARCH PROJECT
% OBJECT DETECTION ALGORITHM
% ORB
% TEST 5 - Importing a Clear Pulley Plate Image with Segmented Image Frames and
Altering Frame Number from Video Footage with Edge Detection Applied
% 2024

clc
clear
close all

%% Load Frames
objectVideo = VideoReader("PLANE_FOOTAGE1.MP4");    % Load video footage from
plane
pulleyPlate = imread('Frame_Image.jpg');    % Load close up of pulley plate with
photo of pulley only
rect = [620 620 820 735];         % Create rectangle section of image at [xmin
ymin width height]. NOTE: top left of image is 0.
pulleyPlateCropped = imcrop(pulleyPlate,rect);    % Fit rectangle around image to
only show that section of the image
pulleyPlateGS = im2gray(pulleyPlateCropped);       % Convert to gray scale
figure;
imshow(pulleyPlateGS);        % Display pulley plate

videoFrame = read(objectVideo,20551);    % Test between frames 20534 and 20551
videoFrameGS = im2gray(videoFrame);     % Convert to gray scale
figure;
imshow(videoFrameGS);

%% Detect Point Features
framePoints = detectORBFeatures(pulleyPlateGS);      % Detect ORB features of
pulley plate
videoPoints = detectORBFeatures(videoFrameGS);    % Detect ORB features of video

figure;
imshow(pulleyPlateGS);       % Display pulley plate
title('100 Strongest Point Features from Pulley Plate');
hold on;
plot(selectStrongest(framePoints,100));     % Show 100 strongest points from the
pulley plate
hold off

figure;
imshow(videoFrameGS);       % Display video frame
title('1000 Strongest Point Features from Plane Footage');
hold on;
plot(selectStrongest(videoPoints,1000));    % Show 1000 strongest points from the
plane footage
hold off

%% Extract Feature Descriptors
[pulleyPlateFeatures, framePoints] = extractFeatures(pulleyPlateGS, framePoints);
% Extract features of interest from pulley plate
[videoFrameFeatures, videoPoints] = extractFeatures(videoFrameGS, videoPoints);
% Extract features of interest from plane footage
```

```
%% Find Putative Point Matches
pulleyPlatePairs = matchFeatures(pulleyPlateFeatures,videoFrameFeatures);    %
Match features using their descriptors

matchedPulleyPlatePoints = framePoints(pulleyPlatePairs(:,1),:);
matchedVideoFramePoints = videoPoints(pulleyPlatePairs(:,2),:);
figure;
showMatchedFeatures(pulleyPlateGS,videoFrameGS,matchedPulleyPlatePoints,matchedVid
eoFramePoints,'montage');
title('Putatively Matched Points');
```

Appendix N: Test 5 – SURF Matlab Code

```
% HAYDEN LENNOX
% PROFESSIONAL ENGINEERING RESEARCH PROJECT
% OBJECT DETECTION ALGORITHM
% SURF
% TEST 5 – Importing a Clear Pulley Plate Image with Segmented Image Frames and
Altering Frame Number from Video Footage with Edge Detection Applied
% 2024

clc
clear
close all

%% Load Frames
objectVideo = VideoReader("PLANE_FOOTAGE1.MP4");    % Load video footage from
plane
pulleyPlate = imread('Frame_Image.jpg');    % Load close up of pulley plate with
photo of pulley only
rect = [620 620 820 735];         % Create rectangle section of image at [xmin
ymin width height]. NOTE: top left of image is 0.
pulleyPlateCropped = imcrop(pulleyPlate,rect);    % Fit rectangle around image to
only show that section of the image
pulleyPlateGS = im2gray(pulleyPlateCropped);       % Convert to gray scale
figure;
imshow(pulleyPlateGS);        % Display pulley plate

videoFrame = read(objectVideo,20551);    % Test between frames 20534 and 20551
videoFrameGS = im2gray(videoFrame);     % Convert to gray scale
figure;
imshow(videoFrameGS);

%% Detect Point Features
framePoints = detectSURFFeatures(pulleyPlateGS,'MetricThreshold',100);      %
Detect SURF features of pulley plate NOTE: Metric threshold detects more blobs,
decreasing number (1-100) increases blob count
videoPoints = detectSURFFeatures(videoFrameGS,'MetricThreshold',1);    % Detect
SURF features of video

figure;
imshow(pulleyPlateGS);       % Display pulley plate
title('100 Strongest Point Features from Pulley Plate');
hold on;
plot(selectStrongest(framePoints,100));     % Show 100 strongest points from the
pulley plate
hold off

figure;
imshow(videoFrameGS);       % Display video frame
title('1000 Strongest Point Features from Plane Footage',fontSize = 20);
hold on;
plot(selectStrongest(videoPoints,1000));    % Show 1000 strongest points from the
plane footage
hold off

%% Extract Feature Descriptors
[pulleyPlateFeatures, framePoints] = extractFeatures(pulleyPlateGS, framePoints);
% Extract features of interest from pulley plate
```

```
[videoFrameFeatures, videoPoints] = extractFeatures(videoFrameGS, videoPoints);
% Extract features of interest from plane footage

%% Find Putative Point Matches
pulleyPlatePairs = matchFeatures(pulleyPlateFeatures,videoFrameFeatures);   %
Match features using their descriptors

matchedPulleyPlatePoints = framePoints(pulleyPlatePairs(:,1),:);
matchedVideoFramePoints = videoPoints(pulleyPlatePairs(:,2),:);
figure;
showMatchedFeatures(pulleyPlateGS,videoFrameGS,matchedPulleyPlatePoints,matchedVid
eoFramePoints,'montage');
title('Putatively Matched Points',fontSize = 20);
```

Appendix O: Test 5 – SIFT Matlab Code

```
% HAYDEN LENNOX
% PROFESSIONAL ENGINEERING RESEARCH PROJECT
% OBJECT DETECTION ALGORITHM
% SIFT
% TEST 5 - Importing a Clear Pulley Plate Image with Segmented Image Frames and
Altering Frame Number from Video Footage with Edge Detection Applied
% 2024

clc
clear
close all

%% Load Frames
objectVideo = VideoReader("PLANE_FOOTAGE1.MP4");    % Load video footage from
plane
pulleyPlate = imread('Frame_Image.jpg');    % Load close up of pulley plate with
photo of pulley only
rect = [620 620 820 735];         % Create rectangle section of image at [xmin
ymin width height]. NOTE: top left of image is 0.
pulleyPlateCropped = imcrop(pulleyPlate,rect);    % Fit rectangle around image to
only show that section of the image
pulleyPlateGS = im2gray(pulleyPlateCropped);       % Convert to gray scale
figure;
imshow(pulleyPlateGS);        % Display pulley plate

videoFrame = read(objectVideo,20551);    % Test between frames 20534 and 20551
videoFrameGS = im2gray(videoFrame);     % Convert to gray scale
figure;
imshow(videoFrameGS);

%% Detect Point Features
framePoints = detectSIFTFeatures(pulleyPlateGS);      % Detect SIFT features of
pulley plate
videoPoints = detectSIFTFeatures(videoFrameGS);    % Detect SIFT features of video

figure;
imshow(pulleyPlateGS);       % Display pulley plate
title('100 Strongest Point Features from Pulley Plate');
hold on;
plot(selectStrongest(framePoints,100));     % Show 100 strongest points from the
pulley plate
hold off

figure;
imshow(videoFrameGS);       % Display video frame
title('1000 Strongest Point Features from Plane Footage',fontSize = 20);
hold on;
plot(selectStrongest(videoPoints,1000));    % Show 1000 strongest points from the
plane footage
hold off

%% Extract Feature Descriptors
[pulleyPlateFeatures, framePoints] = extractFeatures(pulleyPlateGS, framePoints);
% Extract features of interest from pulley plate
[videoFrameFeatures, videoPoints] = extractFeatures(videoFrameGS, videoPoints);
% Extract features of interest from plane footage
```

```
%% Find Putative Point Matches
pulleyPlatePairs = matchFeatures(pulleyPlateFeatures,videoFrameFeatures);   %
Match features using their descriptors

matchedPulleyPlatePoints = framePoints(pulleyPlatePairs(:,1),:);
matchedVideoFramePoints = videoPoints(pulleyPlatePairs(:,2),:);
figure;
showMatchedFeatures(pulleyPlateGS,videoFrameGS,matchedPulleyPlatePoints,matchedVid
eoFramePoints,'montage');
title('Putatively Matched Points',fontSize = 20);
```

Appendix P: Test 6 – Edge Detection in Image Segmentation Matlab Code

```
% HAYDEN LENNOX
% PROFESSIONAL ENGINEERING RESEARCH PROJECT
% OBJECT DETECTION ALGORITHM
% ORB
% TEST 6 - Edge Detection in Image Segmentation
% 2024

clc
clear
close all

%% Load Frames
objectVideo = VideoReader("PLANE_FOOTAGE1.MP4");    % Load video footage from
plane
pulleyPlate = read(objectVideo,20545);    % Display 20549th frame in video
rect = [1190 570 90 90];         % Create rectangle section of image at [xmin ymin
width height]. NOTE: top left of image is 0.
pulleyPlateCropped = imcrop(pulleyPlate,rect);    % Fit rectangle around image to
only show that section of the image
pulleyPlateGS = im2gray(pulleyPlateCropped);       % Convert to gray scale
figure;
imshow(pulleyPlateGS);        % Display pulley plate

videoFrame = read(objectVideo,20548);    % Test between frames 20534 and 20551
size(videoFrame)
rect2 = [1000 0 920 1080];      % Note that frame size is 1080x1920
halfFrame = imcrop(videoFrame,rect2);   % Crop out plane wing in video frame
halfFrameGS = im2gray(halfFrame);     % Convert to gray scale
figure;
imshow(halfFrameGS);        % Display video frame

%% Convert to Edges
% sobel method trial (commented out as it was only added for testing)
% [~,threshold] = edge(pulleyPlateGS,'sobel');    % Calculate threshold value for
edge and Sobel
% fudgeFactor = 0.5;      % Manually adjust threshold value
% pulleyPlateEdge = edge(pulleyPlateGS,'sobel',threshold * fudgeFactor);  % Tune
threshold value again
% figure;
% imshow(pulleyPlateEdge);        % Display pulley plate with edge detection

% Canny method
pulleyPlateEdge = edge(pulleyPlateGS,'Canny');    % Calculate threshold value for
edge and Sobel
figure;
imshow(pulleyPlateEdge)
title('Binary Gradient Mask')

% sobel method trial (commented out as it was only added for testing)
% [~,threshold] = edge(halfFrameGS,'sobel');    % Calculate threshold value for
edge and Sobel
% fudgeFactor = 0.7;      % Manually adjust threshold value
% halfFrameEdge = edge(halfFrameGS,'sobel',threshold * fudgeFactor);  % Tune
threshold value again
% figure;
```

```
% imshow(halfFrameEdge);        % Display pulley plate with edge detection

% Canny method
halfFrameEdge = edge(halfFrameGS,'Canny');    % Calculate threshold value for edge 
and Sobel
figure;
imshow(halfFrameEdge)
title('Binary Gradient Mask')

%% Extracting Pulley Plate from a Small Cropped Image
% Pulley Plate Modifications
closedPulleyPlateEdge = imclose(edge(pulleyPlateGS,'Canny'),strel('line',2,0));  % 
Connect lines of plate that may have slight gaps
figure;
imshow(closedPulleyPlateEdge)
title('Pulley Plate Image with Closed Edges')

fill1 = imfill(closedPulleyPlateEdge,'holes');    % Fill shapes that are fully 
connected
figure;
imshow(fill1)
title('Pulley Plate Image with Filled Shapes')

trim1 = bwareaopen(fill1,1500);  % Remove any remaining blobs that have an area 
less than 1500
figure;
imshow(trim1)
title('Pulley Plate Image with Small Blobs Removed')

open1 = imopen(trim1,strel(ones(15,15)));   % Remove joined sections that are less 
than 15x15 (discovered by trial and error)
figure;
imshow(open1)
title('Pulley Plate Image with Joins Removed')

%% Extracting Pulley Plate from Video Footage
closedHalfFrameEdge = imclose(edge(halfFrameGS,'Canny'),strel('line',4,4));  % 
Connect lines of plate that may have slight gaps
figure;
imshow(closedHalfFrameEdge)
title('Video Footage Frame with Closed Edges')

fill2 = imfill(closedHalfFrameEdge,'holes');    % Fill shapes that are fully 
connected
figure;
imshow(fill2)
title('Video Footage Frame with Filled Shapes')

trim2 = bwareaopen(fill2,10000);  % Remove any remaining blobs that have an area 
less than 300000
figure;
imshow(trim2)
title('Video Footage Frame with Small Blobs Removed')

closedHalfFrameEdge2 = imclose(edge(trim2,'Canny'),strel('line',2,0));  % Connect 
lines of plate that may have slight gaps
figure;
imshow(closedHalfFrameEdge2)
title('Video Footage Frame with Closed Edges')
```

```
fill3 = imfill(closedHalfFrameEdge2,'holes');    % Fill shapes that are fully
connected
figure;
imshow(fill3)
title('Video Footage Frame with Filled Shapes')

trim3 = bwareaopen(fill3,50000);  % Remove any remaining blobs that have an area
less than 50000
figure;
imshow(trim3)
title('Video Footage Frame with Small Blobs Removed')

open3 = imopen(trim3,strel(ones(110,110)));   % Remove joined sections that are
less than 110x110 (discovered by trial and error)
figure;
imshow(open3)
title('Video Footage Frame with Joins Removed')
```