

University of Southern Queensland
Faculty of Health, Engineering and Sciences

Identifying Abnormalities in Indoor Environments using Obstacle Detection Programming

A dissertation submitted by
Nicholas Jennings

In fulfilment of the requirements of
ENG4111 and 4112 Research Project
Towards the degree of
Bachelor of Engineering (Honours)(Mechatronic)
Submitted October 2021

**Abstract:**

In autonomous mechatronic systems, object detection is a vital functionality which allows the system to process information about its surrounding environment and interpret it in order to act without the need for human interference. In order to procure information regarding its environment, a system must be equipped with sensors appropriate for the conditions in which it is operating, which are also capable of perceiving the required types of object for the current application. Despite the numerous existing systems tailored to the environment, in applications requiring autonomous navigation in an indoor environment, little research has been conducted observing the accuracy of object detection when applied to small-scale low-lying obstacles. Of the common forms of sensors employed in object detection applications, being ultrasonic, LiDAR and both monocular and stereo vision camera configurations, the sensors system most appropriate for the detection of small-scale low-lying obstacles is a stereo vision camera. A sample stereo vision system was created and subsequently evaluated in a series of tests in order to determine the accuracy of its depth estimation when directed at applicable obstacles likely to be present within an indoor environment. The sample system was prepared with hardware consisting of a pair of identical webcams fixed a set distance apart, and software developed utilising MATLAB, governing the calibration of the stereo camera, the image segmentation and the depth estimation of identified obstacles. First tested within a controlled environment, the system was determined to have a baseline error of less than 4% when detecting more traditional obstacles. When applied to the analysis of small-scale low-lying obstacles, factors such as the size of flat obstacles, the angle at which they were orientated and their positioning within the camera's visible area were determined to impact the accuracy of the depth estimation.



University of Southern Queensland

Faculty of Health, Engineering and Sciences

ENG4111 & ENG4112 Research Project

Limitations of Use

The Council of the University of Southern Queensland, its Faculty of Health, Engineering and Sciences, 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 Faculty of Health, Engineering and Sciences 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 any 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.



Certification

I certify that the ideas, designs and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.

I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

Nicholas Jennings

Student Number: 0061107128



Acknowledgements:

I would like to acknowledge the various people who have provided me assistance which has allowed for the completion of this project. This includes the family and friends that have assisted me in tasks which required multiple people to complete and supported me throughout the process of completing my project work. I would like to acknowledge the supervisors of previous courses I have completed which have afforded me the knowledge necessary to operate programs such as MATLAB and understand the concepts involved in the completion of this project. Foremost I would like to thank my project supervisor Dr Tobias Low for the assistance and guidance he has provided throughout the undertaking of this project which has greatly assisted me in improving the quality of this dissertation.



Table of Contents:

Abstract	Page 1
Disclaimer	Page 2
Certification	Page 3
Acknowledgements	Page 4
List of Figures	Page 7
List of Tables	Page 9
1. Introduction	Page 10
1.1 Introduction	Page 10
1.2 Problem Identification and Aim	Page 10
1.3 Objectives	Page 11
1.4 Research Questions	Page 11
1.5 Background	Page 12
2. Literature Review:	Page 15
2.1 Ultrasonic Sensor:	Page 15
2.1.1 Ultrasonic Sensor Example 1: Radar Based Design	Page 15
2.1.2 Ultrasonic Sensor Example 2: 3D Mapping and Position Estimation	Page 16
2.2 LIDAR Sensor	Page 16
2.2.1 LIDAR Example 1: Adjusted Orientation	Page 16
2.2.2 LIDAR Example 2: LOAM Method	Page 17
2.3 Camera Sensor:	Page 18
2.3.1 Image Processing	Page 18
2.3.2 Monocular Vision	Page 23
2.3.3 Stereo Vision	Page 28
2.4 Evaluation	Page 31
2.5 Potential Outcomes and Consequential Effects	Page 34
3. Methodology	Page 35
3.1 Methodology Preparation	Page 35
3.1.1 Hardware	Page 35
3.1.2 Software	Page 36
3.2 Research Methodology Overview	Page 42
3.2.1 Preliminary Testing	Page 42
3.2.2 Controlled Environment Testing	Page 42
3.2.3 Alternate Environment Testing	Page 43
3.2.4 Evaluation	Page 43
3.3 Preliminary Testing	Page 44
3.4 Sample Obstacle Testing	Page 46
3.4.1 Controlled Environment Testing	Page 46
3.4.2 Alternate Environment Testing	Page 47
3.4.3 Testing Process	Page 47
4. Resources, Risk Management and Timelines	Page 48
4.1 Resources	Page 48
4.2 Risk Assessment	Page 48
4.3 Project Plan	Page 49



5. Results and Analysis	Page 52
2.1. Preliminary Testing	Page 52
3.1. Preliminary Test 1	Page 53
3.2. Preliminary Test 2	Page 56
3.3. Preliminary Test 3	Page 57
3.4. Other Changes	Page 61
2.2. Controlled Environment Testing	Page 62
2.3. Alternate Environment Testing	Page 73
6. Conclusion and Further Work	Page 74
2.1. Conclusion	Page 74
2.2. Further Work	Page 74
7. References	Page 77
8. Appendix A	Page 79
9. Appendix B	Page 80



List of Figures:

Figure 1.1: Stereo Vision Principles (Sharma, Sahoo, Manivannan 2018)

Figure 2.1: Radar Based Ultrasonic Object Detection System (Kulkarni, Potdar, Hegde, Baligar 2019)

Figure 2.2: LIDAR Example 1 Obstacle classification and path planning.
(Peng, Qu, Zhong, Xie, Luo, Gu 2015)

Figure 2.3: LOAM method 3D Maps. (Zhang & Singh 2014)

Figure 2.4: Basic forms of Image Segmentation. (MathWorks 2021)

Figure 2.5: YOLO Process (Redmon, Divvala, Girshick, Farhadi 2016)

Figure 2.6: Block Based Motion Estimation (Kim & Do 2012)

Figure 2.7: Trajectory Optimisation (Zhang, Cao, Ding, Zhuang, Tao 2020)

Figure 2.8: Monocular Example 3 Setup (Young & Simic 2015)

Figure 2.9: Dynamic Feature Tracking and Depth Map (Sharma, Sahoo, Manivannan 2018)

Figure 2.10: Stereo Vision Example 2 Image Segmentation Process (SOLAK & BOLAT 2018)

Figure 3.1: Sample System Hardware Setup

Figure 3.2: Stereo Camera Calibration

Figure 3.3: Stereo Anaglyph and Disparity Map

Figure 3.4: Input Image

Figure 3.5: Rectified Image

Figure 3.6: Undistorted Image

Figure 3.7: Binary Mask (Overlaid on Undistorted Image)

Figure 3.8: Completed Image Processing

Figure 3.9: System Output

Figure 3.10: Testing Environment

Figure 3.11: Testing Coordinate System

Figure 5.1: Preliminary Testing Sample Image Pair (40, 380)

Figure 5.2: Error Image Segmentation

Figure 5.3: Sample Obstacle Testing SD Orientation 1 Sample Image Pair (0, 400)

Figure 5.4: Sample Obstacle Testing SD Orientation 2 Sample Image Pair (0, 400)

Figure 5.5: Sample Obstacle Testing Mouse Orientation 1 Sample Image Pair (400)

Figure 5.6: Sample Obstacle Testing Mouse Orientation 2 Sample Image Pair (400)

Figure 5.7: Sample Obstacle Testing Mouse Orientation 3 Sample Image Pair (400)

Figure 5.8: Sample Obstacle Testing Mouse Orientation 4 Sample Image Pair (400)



Figure 5.9: Sample Obstacle Testing Mouse Orientation 5 Sample Image Pair (400)

Figure 5.10: Sample Obstacle Testing Paper 0 Degrees Sample Image Pair (400)

Figure 5.11: Sample Obstacle Testing Paper 45 Degrees Sample Image Pair (400)

Figure 5.12: Sample Obstacle Testing Paper 90 Degrees Sample Image Pair (400)

Figure 5.13: Sample Obstacle Testing Paper 135 Degrees Sample Image Pair (400)

Figure 5.14: Sample Obstacle Testing Barrier 3mm Degrees Sample Image Pair (400)

Figure 5.15: Sample Obstacle Testing Barrier 6mm Degrees Sample Image Pair (400)

Figure 5.16: Sample Obstacle Testing Barrier 9mm Degrees Sample Image Pair (400)

Figure 5.17: Sample Obstacle Testing Barrier 12mm Degrees Sample Image Pair (400)

Figure 5.18: Sample Obstacle Testing Barrier 15mm Degrees Sample Image Pair (400)

Figure 5.19: 12mm Barrier at 400mm Segmentation

Figure 5.20: Alternate Environment segmentation sample (Raw – Left, Segmented – Right)



List of Tables:

Table 4.1: Resource Requirements

Table 4.2: Risk Assessment Key

Table 4.3: Safety Risks and Precautions

Table 4.4: Original Project Plan

Table 4.5: Altered Project Plan

Table 5.1: Preliminary Test 1 Raw Data

Table 5.2: Preliminary Test 1 Error

Table 5.3: Preliminary Test 2 Raw Results

Table 5.4: Preliminary Test 2 Error

Table 5.5: Preliminary Test 3 Raw Results

Table 5.6: Preliminary Test 3 Error

Table 5.7: SD Sample Obstacle Orientation 1 Readings

Table 5.8: SD Sample Obstacle Orientation 1 Error

Table 5.9: SD Sample Obstacle Orientation 2 Readings

Table 5.10: SD Sample Obstacle Orientation 2 Error

Table 5.11: Mouse Sample Obstacle Readings

Table 5.12: Mouse Sample Obstacle Error

Table 5.13: A4 Paper Sample Obstacle Readings

Table 5.14: A4 Paper Sample Obstacle Error

Table 5.15: Barrier Sample Obstacle Readings

Table 5.16: Barrier Sample Obstacle Error



Chapter 1: Introduction

1.1 - Introduction:

As mechatronic and other automated system technologies have advanced, there has been an increased focus on creating systems with the ability to respond to their environments automatically, without the need for human assistance. In mobile robotics and automation applications, obstacle detection programming is a vital means of providing environmental data for a system to respond to. Such systems vary in complexity and capability, with numerous different means of performing object detection available, with unique methods being continuously developed. The majority of these methods tend to focus on the detection of obstacles which would obstruct a system's movement and otherwise require the alteration of its travel path. This project will focus on the analysis of a sample obstacle detection system, to determine its accuracy in identifying the relative position of certain types of obstacles. To do this, research will be conducted on existing obstacle detection systems in order to determine the most appropriate aspects to utilise in the sample system. Once the sample obstacle detection system has been prepared, it will be tested in a series of scenarios and evaluated on its performance. This will be conducted in three phases, first the sample system will be tested through the use of a more distinct obstacle in a prepared test environment, then a series of sample obstacles will be analysed within the same test environment, and then in a series of alternate scenarios indicative of indoor environments, in which systems utilising object detection may be employed. The results obtained through this testing will then be analysed in order to determine the baseline behaviour of the sample system, its behaviour under ideal conditions and finally its behaviour in practical scenarios.

1.2 - Problem Identification and Aim:

Despite the numerous examples of obstacle detection code testing, there has been little conducted that focusses on obstacles which do not directly obstruct a robot's path, obstacles which the system could navigate through but may cause complications. In an indoor environment, these obstacles could include sensitive documents and equipment which could be damaged if navigated through and other small obstructions which could be navigated over at the cost of time, or risk of damage to the system. In order to determine the ability for simplistic obstacle detection systems to detect such obstacles, this project will analyse the effectiveness of such an obstacle detection program in detecting low-lying, small-scale obstacles in an indoor environment. Despite there being exceptions in a practical setting, for the purposes of this project 'small-scale' refers to objects with a height less than 25% of a system's wheel diameter which the system has enough clearance to navigate over if required.



1.3 - Objectives:

In order to conduct such research, key objectives were identified which, when completed, should provide data detailing the effectiveness of a sample obstacle detection system in identifying specific types of test obstacles.

1. Research examples of obstacle detection programming and hardware.
2. Determine which programming and hardware elements can be utilised for the project.
3. Develop obstacle detection programming using the MATLAB environment.
4. Test the program's capabilities in controlled scenarios.
5. Perform additional tests in other indoor environments, outside the testing scenarios.
6. Evaluate the program's ability to analyse different types of obstacles.

1.4 – Research Questions:

During the completion of this project, several key research questions will be answered as progress is made on the project objectives. At the conclusion of this project, the research and experimentation conducted should provide most if not all of these questions with a suitable answer.

1. What are the capabilities and limitations of object detection systems currently in use?
2. What hardware and software components would be best suited for use in an object detection system capable of detecting small-scale low-lying obstacles in an indoor environment?
3. What is the baseline accuracy of the chosen object detection system?
4. How does the accuracy of the system change when analysing sample small-scale low-lying obstacles?
5. How accurate is the system when utilised outside a controlled test environment?



1.5 – Background:

Amongst the various example object detection systems which will be discussed during the literature review, similar types of sensors tend to be utilised in order to collect data for analysis. This section will detail the broad categories under which the majority of object detection sensors can be categorised and provide a brief outline as to how they operate, and the general benefits and detriments of their use.

Ultrasonic sensors are one of the most commonly utilised sensors in object detection applications, being primarily utilised as proximity sensors. Typical ultrasonic sensors are capable of determining the distance and location of objects within their effective range. They do so by bouncing ultrasonic waves off objects within their range and measuring the time it takes the waves to return, allowing for the calculation of the object's distance. Frequently, ultrasonic sensors are utilised in the generation of 3D maps of environments, from which the locations of objects within it can be determined. As such, a large portion of ultrasonic based obstacle avoidance systems will extract obstacle location data from full or partial 3D maps of their environment. One of the key advantages of ultrasonic sensors over Cameras is the fact that their means of object detection is unaffected by visual obstructions, such as, airborne particles which can affect the results obtained by other sensors.

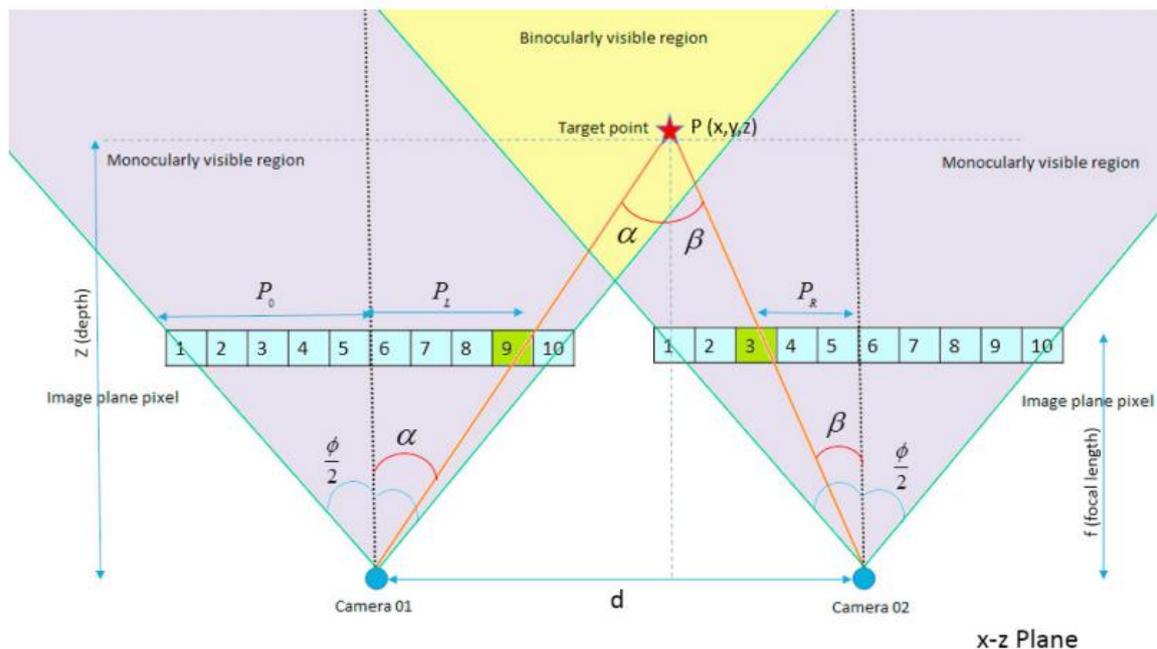
Light Detection and Ranging (LIDAR) sensors are a type of sensor primarily utilised in the surveying of large environments, and as such some object avoidance methods utilising them have been developed. LIDAR sensors detect objects by reflecting a laser off them and come in either 2D or 3D variants, which vary in their perception capabilities. 2D LIDARs measure along a single plane, either by rotating a single point laser or using a continuous laser covering the effective axis, resulting in two-dimensional data on the environment. 3D LIDARs build upon this basic system, translating the two-dimensional laser system along a perpendicular axis, thereby generating three-dimensional data. Like with ultrasonic sensors, a large portion of LIDAR based object detection systems utilise 3D maps in some capacity.



Camera sensors are the most widely known form of sensor which sees use in object detection applications, being a common fixture in everyday life. Although not specialised for the measurement of distance as with the previously discussed sensors, means have been devised in order to interpret image data to obtain such information. This however is not the primary benefit for the use of cameras in object detection. Cameras are primarily used in order to detect the presence of objects through the detection of shifts in light levels or colour within an image. As such cameras are often used in tandem with other sensors and typically require more intensive processing to retrieve useful data. When incorporated into systems, camera sensors are typically arranged into monocular or stereo vision configurations.

Monocular vision systems refer to obstacle detection systems which primarily make use of a single camera for sensory input. In general, the obstacle detection performed by monocular systems is limited to the detection of an objects presence, with other supplementary sensors determining it's depth if necessary. Even though this functionality is the norm, various methods have been explored in the attempts to derive useable depth information from monocular systems. Despite these inherent limitations with their use, the environmental data available to systems utilising a monocular camera can be instrumental to a system's functionality.

Stereo vision-based obstacle detection systems consist of a pair of cameras fixed a set distance apart, generating a pair of images used in analysis. The vast majority of stereo vision systems tend to operate under the same general principles. One of the fundamental steps of this process is the rectification of the input images. This process projects the input images from each camera in the stereo system onto a common image plane, simplifying the process of locating matched points between the images in the pair. Using the rectified images produced by this process, a stereo vision system can determine the depth of any of the pixels shared between the input images. This is accomplished using the stereo vision principle as outlined by Figure 1.1.



For left camera image plane : $\frac{x}{z} = \frac{P_L}{f}$ (1)

Similarly for right camera image plane : $\frac{d-x}{z} = \frac{P_R}{f}$ (2)

After eliminating the x terms from (1) and (2), the depth point can be given by : $z = \frac{f \times d}{(P_R + P_L)}$ (3)

Figure 1.1: Stereo Vision Principles (Sharma, Sahoo, Manivannan 2018)

These calculations are typically either performed on individual pixels to determine their depth, or on entire images in order to create 3D depth maps of the visible region. The range at which these calculations can be performed is dependent on the positions and fields of view of the cameras which constitute the stereo vision system. As shown by Figure 1.1, the area in which these calculations can be performed are points where each camera’s fields of vision overlap, meaning that the smaller the distance between the two camera lenses, the closer the system can perceive obstacles.

In cases where resources do not allow for the depth analysis of entire environments simultaneously, systems employ methods of determining which point in an area to focus the relative depth calculations on. In systems which utilise a camera input, this typically involves the analysis of the image(s) to determine points of interest. This can be done in a variety of methods ranging from simple segmentation methods to the training of deep learning neural networks to perform image segmentation specialised for the task at hand.

Chapter 2: Literature Review

This chapter details the literature review process undertaken over the course of this project. This includes the identification of existing concepts and systems related to obstacle detection, or those which present the potential to be adapted for such an application. This research will focus on the most common forms of sensor utilised in obstacle detection applications, being ultrasonic sensors, LIDAR and cameras. From the collected sources, the benefits and drawbacks of each sensor will be discussed, and the most optimal configuration for this project will be found. Following this, the potential outcomes of the project will be discussed along with the consequential effects their occurrence may entail.

2.1 - Ultrasonic Sensor:

2.1.1 - Ultrasonic Example 1: Radar Based Design

Radar systems can determine the range, angle and even speed of objects via the transmission and subsequent detection of electromagnetic signals bouncing off objects. Due to the similarities in their operation, some ultrasonic object detection systems have been used in a manner which mimics radars (Kulkarni, Potdar, Hegde, Baligar 2019). This typically involves the use of the ultrasonic sensor attached to a servo motor which will oscillate it along the desired range. By recording the angle at which the ultrasonic sensor is orientated at each distance measurement, the approximate location of objects relative to the sensor can be determined. One particular example created a GUI to display this data in a manner which resembles a radar's interface which also provided more exact details on an object's range and angle via SMS. Testing of a simple HC-SR04 ultrasonic sensor rotating across 180 degrees was conducted using this form of presentation. It proved capable of detecting objects between 2cm and 400cm away from the sensor in a 150-degree area. (Kulkarni, Potdar, Hegde, Baligar 2019)

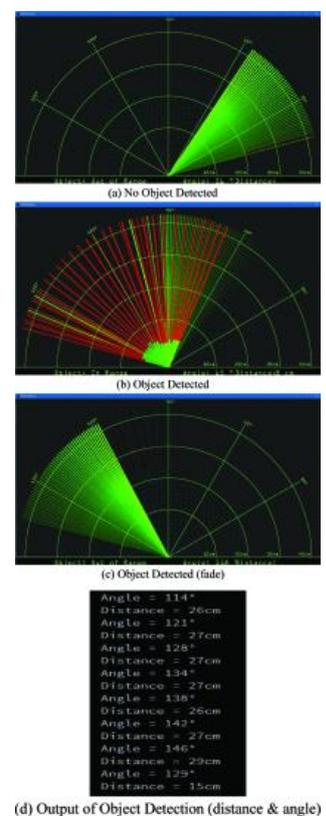


Figure 2.1: Radar Based Ultrasonic Object Detection System (Kulkarni, Potdar, Hegde, Baligar 2019)

2.1.2 – Ultrasonic Example 2: 3D Mapping and Position Estimation

Ultrasonic based 3D environmental mapping techniques are commonly utilised in obstacle avoidance methods as a means of organising environmental data. Such methods typically employ either multiple ultrasonic sensors at different orientations, or a primary sensor capable of reorientation. These systems utilise similar concepts to the previous ultrasonic radar example, recording both the depth data and the sensor's orientation to determine the distance and relative angle of an obstacle with respect to the main system. Some methods have proved capable of utilising such data in the position estimation of a system, a concept which could be adapted to suit obstacle avoidance applications (Nakajima, Premachandra, Kato 2017). This position estimation was accomplished by comparing a simplistic ultrasonic map of a system's immediate surroundings to a world map created from previous maps. When tested using a simple quad rotor drone, the main system was further provided with additional elevation data obtained from an additional ultrasonic sensor, allowing for the generation of simplistic 3D maps. (Nakajima, Premachandra, Kato 2017)

2.2 - LiDAR Sensor:

2.2.1 – LiDAR Example 1: Adjusted Orientation

Two-dimensional LIDARs see an abundance of use as a sensor in obstacle detection systems, due to their viability in a vast majority of practical applications. As previously stated, the primary constraint of 2D LIDARs is the limited area in which they can perceive obstacles, being a two-dimensional plane. In order to help compensate for this, some obstacle detection methods have adjusted the angle at which the sensor is orientated in order to allow it to perceive obstacles below its elevation (Peng, Qu, Zhong, Xie, Luo, Gu 2015). A proposed real-time detection method was based around a sensor in such an orientation, with the intention of implementing it in the control of a small-scale mobile

robot. The detection algorithms for this method are capable of classifying the point data obtained from the LIDAR, to determine the general shape of object. These determined shapes are then utilised in the obstacle avoidance algorithms, generating a simplistic map of the robot's immediate environment. Tested via MATLAB simulation, the obstacle avoidance algorithm proved capable of producing a travel path to navigate the robot around the obstacle shapes in order to reach its destination. (Peng, Qu, Zhong, Xie, Luo, Gu 2015)

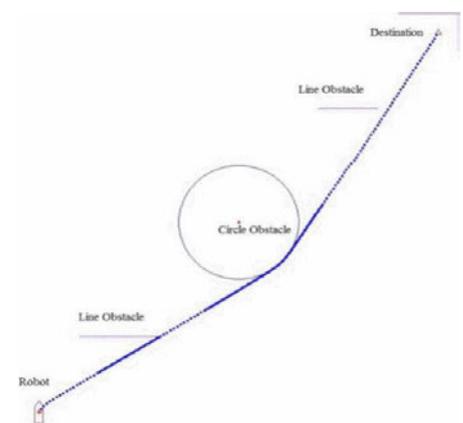


Figure 2.2: LIDAR Example 1 Obstacle classification and path planning. (Peng, Qu, Zhong, Xie, Luo, Gu 2015)



2.2.2 – LIDAR Example 2: LOAM method

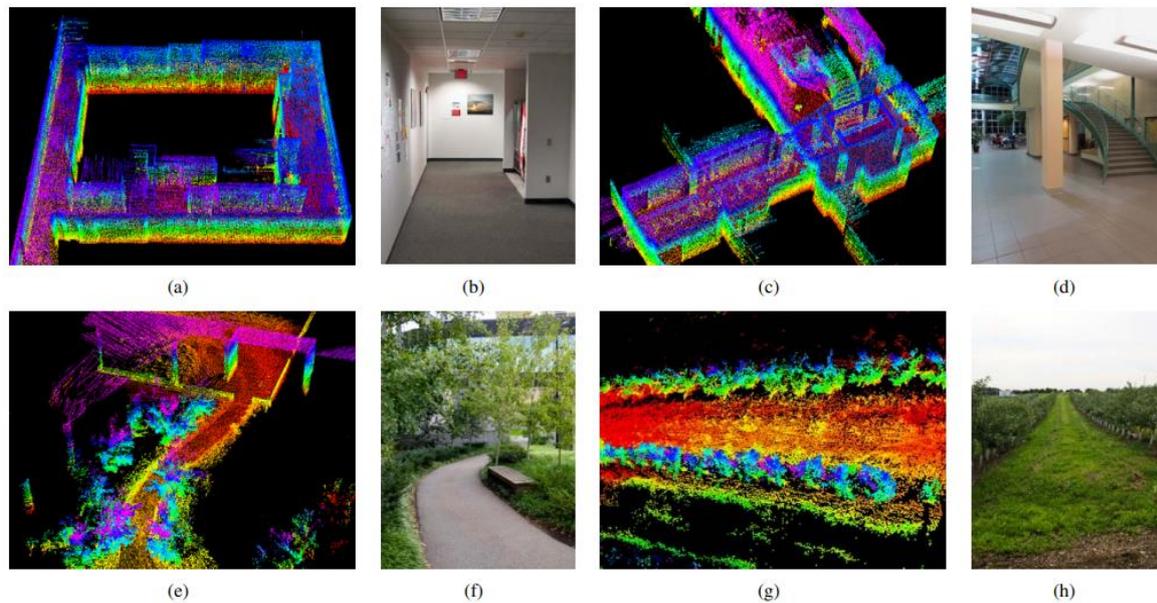


Figure 2.3: LOAM method 3D Maps. (Zhang & Singh 2014)

One of the most common uses for LIDAR technology is in the generation of 3D Maps and, although not explicitly intended for obstacle detection purposes, the concepts present the capability to be adapted for such applications. One LIDAR based 3D mapping method used in a variety of different applications is the Lidar Odometry and Mapping (LOAM) real-time method (Zhang & Singh 2014). As the name implies, the base version of this method utilises a 2-axis LIDAR capable of moving in 6 degrees of freedom, and its software concurrently operates both odometry and mapping algorithms in order to generate 3D point cloud maps of a given environment. Typically, when the LOAM method is applied to an application, a new variant is created in order to better suit the task at hand, one such variant was designed for use in mobile robot trajectory planning (Pan, Li, Liu, Xu, Ji, Kang 2020). The improved method aimed to decrease the possibility of point cloud matching errors which can occur as a result of the robot's movement, a factor which was not considered in the original LOAM's design. This altered version of the LOAM method was tested as part of a trajectory planning system controlling a crawler mobile robot and demonstrated its applicability in such an application. (Pan, Li, Liu, Xu, Ji, Kang 2020)



2.3 - Camera Sensor:

In order to utilise solely camera sensors in the determination of an objects depth, mathematical principles regarding the position and size of features in image(s) must be applied. These calculations are typically either performed on the entirety of an image input over the course of the analysis or are performed on determined points of interest identified within them. Which of these methods a system will employ depends on the application and processing power available, with full image analysis typically being more taxing. This section will first discuss some generic image segmentation techniques that can be used to highlight points of interest within images for analysis purposes. Following this, some examples of monocular and stereo vision systems will be outlined, some of which include their own dedicated segmentation systems.

2.3.1 – Image Processing

2.3.1.1 - Traditional Segmentation:

Image segmentation is a broad term which covers a large variety of methods which segment an image for analysis purposes, the majority of which generate binary image masks which highlight the desired portions of an image. One of the simplest forms of this is the thresholding process which creates binary images from either grayscale or colour images, by specifying a threshold value which denotes the transition point between 0 and 1 on a binary image. Another prominent form of image segmentation is the semiautomatic technique known as graph cut. This method requires the identification of a region of interest (ROI) and background of a given image, from which it generates a binary mask. (MathWorks 2021)

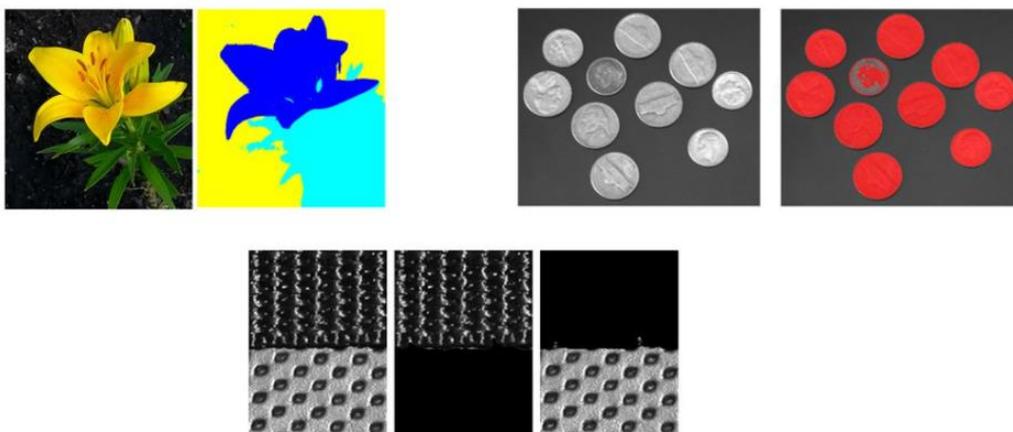


Figure 2.4: Basic forms of Image Segmentation. (MathWorks 2021)



The background subtraction and image differencing obstacle detection techniques are commonly utilised in monocular systems as they allow for the isolation of foreground elements and the determination of changes between subsequent images respectively (Garcia-Garcia, Bouwmans, Silva 2020). General background segmentation methods can be divided into three steps. This process begins with the computation of training image of the background in question, from which a model is created to represent it. Once this model has been established, the active portions of the programming take place. These consist of the classification of pixels within an image as either foreground or background and the subsequent update of the model using the new background data. In a practical application, the training data used to create the background model would be provided by a static camera of a designated area. The primary downfall of such systems is their reliance on a stationary camera in order to operate effectively, with a constantly changing image background being difficult for a system to comprehend. (Kim & Do 2012) (Garcia-Garcia, Bouwmans, Silva 2020)

2.3.1.2 - Neural Networks and YOLO Method:

More powerful obstacle detection systems typically make use of neural networks for the identification of obstacles within images. Such methods are not only highly accurate but can also possess the capability to classify obstacles by type, which can be useful data for more complex systems. The fundamental concept which allows humans to recognise objects within images is the human mind's ability to recognise patterns. Traditional computers are incapable of reliably recognising patterns within data and images, this issue is something which deep learning neural networks (DLNN) are designed to rectify. As can be inferred from the name, the algorithms which comprise DLNN systems are modelled off of the functionality of the human brain. The specifications and details of these systems vary between designs, but the vast majority of these systems can be trained to process specific types of data or image input to produce a desired result. Typically, in image processing applications, this training requires the preparation of training data which takes the form of a series of example images and accompanying label data. Generally, the more specialised and comprehensive the training data provided to the system is, the more accurate the system's image analysis will be. The form of analysis performed on images using a DLNN are classified as either Image Classification, Semantic Segmentation or Instance Segmentation. (Wang, Chen, Hu, Li 2020) (Ye, Liu, Wang 2017)



Image Classification is the most straightforward of the aforementioned forms of image analysis possible using a DLNN. Image classification systems will provide an applicable label to an input image, classifying it accordance to the training data it was provided. These types of DLNN systems see use in applications which do not require intricate analysis of an image, merely a surface level observation such as detecting an object's presence but not it's exact position. Common examples of systems which utilise this process include the identification of single handwritten characters, the detection of tumours within x-rays and some forms of facial recognition systems. (Wang, Chen, Hu, Li 2020)

Semantic segmentation takes this process one step further, not only identifying the presence of a specified feature within an image, but also the exact pixels within the image the feature occupies. Such a system will label the pixels within a provided image according to the information the network was trained with in the same manner as other DLNN systems. The data required to train a DLNN to perform semantic segmentation is more complex than that required for an image classification system, requiring labels for individual pixels as opposed to entire images. As such, both the creation of the training data and the system's image processing are more complex than that required to create an image classification system. Semantic segmentation sees applications in the tracking of motion in a series of images as well as vision-based object detection applications. (Wang, Chen, Hu, Li 2020)

One of the primary downsides of semantic segmentation DLNN systems are their inability to distinguish between multiple instances of the same pixel class. Instance segmentation DLNN networks expand upon the capabilities of semantic segmentation networks by including the ability to distinguish between such instances, categorising each feature separately. Such functionality allows systems to expand upon the usefulness of the previously mentioned methods allowing for more detailed analysis of images. In particular this added capability is utilised to allow a system to count the number of like features within an image, as well as record position and location data for multiple objects of the same type simultaneously. (Ye, Liu, Wang 2017)

One example of a neural network obstacle detection system is the You Only Look Once (YOLO): Unified, Real-Time Object Detection system (Redmon, Divvala, Girshick, Farhadi 2016). Like other neural network-based systems, the YOLO detection system can be tailored to the task at hand, trained to detect and classify the obstacles needed for a given application. The process the code undertakes in order to perform this detection first requires the resizing of the provided image input to a 448 x 448 resolution. Following this, the image is divided into a grid, each cell of which is individually analysed to generate bounding boxes and assigned a confidence score. Each bounding box encapsulates all of the pixels that could constitute an object, with multiple boxes capable of existing in each cell. The confidence scores on the other hand denote the possibility that an object is visible within the respective cell, classifying the type of object if applicable. These two processes run concurrently and are both taken into consideration in determining the final detection, as can be seen in Figure 2.5. (Redmon, Divvala, Girshick, Farhadi 2016)

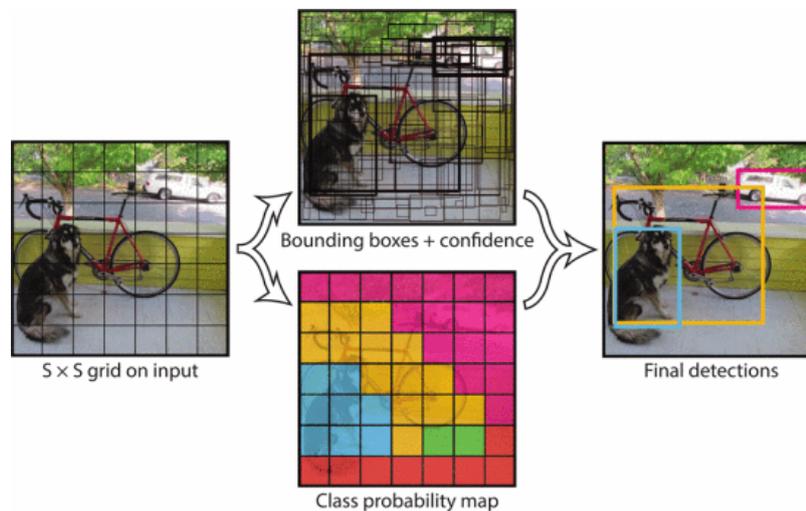


Figure 2.5: YOLO Process (Redmon, Divvala, Girshick, Farhadi 2016)



The initial iteration of the YOLO system possessed capabilities which placed it in competition with other obstacle detection systems at the time of its release. The YOLO's analysis systems process images at a rate of up to 45 frames per second on the minimum specifications, with more powerful variants being capable of analysing at upwards of 150 frames per second. One function which sets the YOLO system apart from its contemporaries is the method employed in making predictions. Unlike other obstacle detection systems, the YOLO system does not use a search window when performing analysis, instead performing the analysis globally. Doing so allows the system to somewhat understand the context of objects, improving the detection accuracy of the system as a whole. (Redmon, Divvala, Girshick, Farhadi 2016)

The YOLO system's versatility has allowed it to be used in a number of different applications including autonomous vehicles (Sarda, Dixit, Bhan 2021), facial recognition (Ghenescu, Mihaescu, Carata, Ghenescu, Barnoviciu, Chindea 2018) and other miscellaneous detection problems. This utility is further demonstrated by the fact that, despite the YOLO system typically being used with a monocular camera setup, there have been instances where it has been utilised in conjunction with a stereo vision system. One such instance accomplished this by simply using a depth camera and calculating an average depth for all of the pixels present within the bounding box, producing an approximate depth for the obstacle in question (Inoue, Kaizu, Igarashi, Imou 2019). This data was then utilised by a path planning algorithm, namely the Teb local planner, to generate an obstacle free path for the system to follow. (Inoue, Kaizu, Igarashi, Imou 2019)



2.3.2 - Monocular Vision:

2.3.2.1 – Monocular Example 1: Dynamic Humanoid Obstacle

Research has been conducted analysing the effectiveness of monocular obstacle avoidance programming in the control a mobile robot. Focussed on the detection of dynamic humanoid obstacles in an indoor environment, this research makes use of the obstacle detection technique known as Block-based motion estimation (Kim & Do 2012). As the name implies, this method divides an image into a series of 'blocks', the motion vector of which is determined through the 'Sum of Absolute Differences' criterion comparing two subsequent image frames. The blocks in each image which have the highest degree of similarity identified by the search window, through their change in position within the image frame, demonstrate the motion vector of their block and the features within it. In order to utilise this method in the detection of dynamic (moving) obstacles within the environment, information regarding the positions and dimensions of the environments static obstacles can be provided in order to focus the code's detection capabilities on the identification of moving obstacles. (Kim & Do 2012)

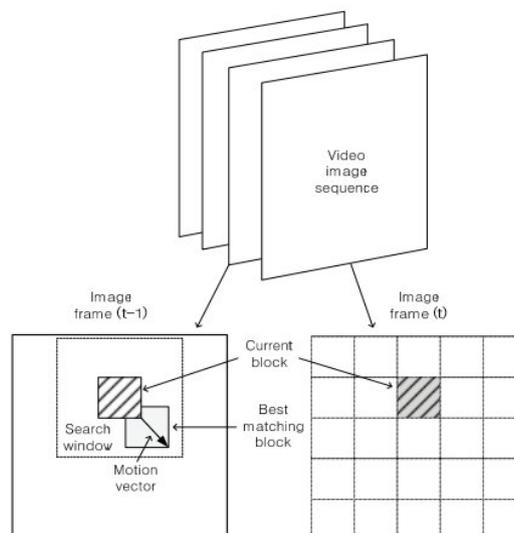


Figure 2.6: Block Based Motion Estimation (Kim & Do 2012)



Testing of this form of Block-based motion estimation when used on a video feed provided from a 640x480 30 fps mobile camera provided mixed results. The testing in question initially halved the image size prior to processing and then proceeded to divide the image in to blocks of 21 square pixels and begin searching using a window 32x21 pixels in size. The detection failures in the system were determined to be caused by a series of factors that can vary in different situations. The most prominent of these factors was the distance between the moving obstacle and the robot. Due to the fact that obstacles further away from the camera appear to be moving at a slower speed, the algorithm experienced difficulties identifying said obstacles until they became closer to the camera. Other factors which caused detection errors include an obstacle's colour which may result in the algorithm's inability to detect it at all, and the identification of reflected light which appeared to be in motion as a result of the camera's movement. (Kim & Do 2012)



2.3.2.2 – Monocular Example 2: UAV Trajectory Planning

Monocular based obstacle avoidance has also seen use outside of land-bound mobile robotics, one such area is in unmanned aerial vehicle applications (Zhang, Cao, Ding, Zhuang, Tao 2020). Traditionally, a UAV's ability to detect and avoid obstacles is referred to as a Sense and Avoid (SAA) system, which utilise onboard sensors to determine an obstacle's relative position and the potential risks it imposes. Despite not being the primary sensor used in UAV applications, research has been conducted into the viability of monocular systems as an alternative sensory system. One particular study proposed the avoidance of obstacles through the planning of a trajectory through an environment as opposed to traditional, reactionary obstacle avoidance methods. The method this research proposed combined two forms of obstacle avoidance programming, which performed relative range based and relative angle-based calculations respectively. Simulations of such methods demonstrated the viability of such a system which could not only determine a trajectory through a series of obstacles, but also optimise the time and energy required for the system to follow through. (Zhang, Cao, Ding, Zhuang, Tao 2020)

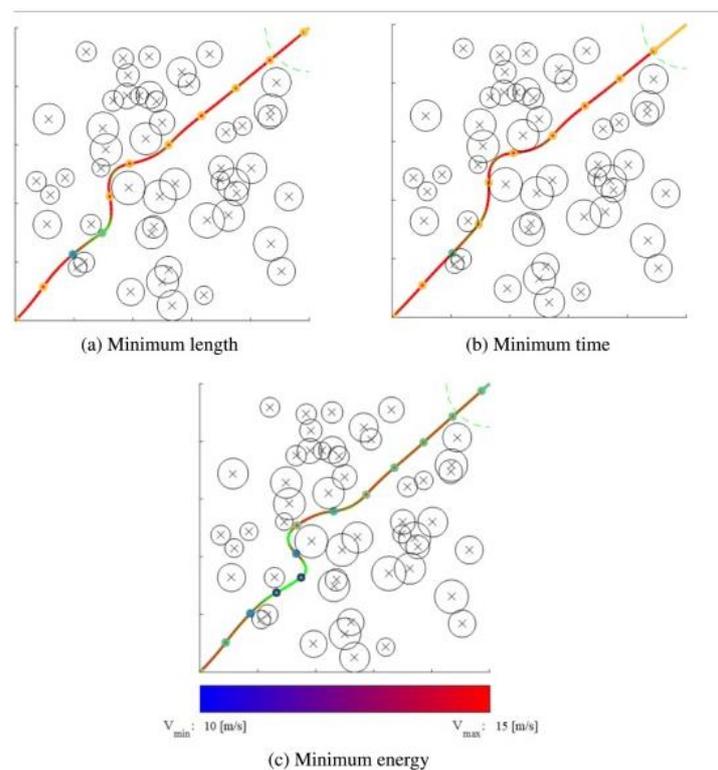


Figure 2.7: Trajectory Optimisation (Zhang, Cao, Ding, Zhuang, Tao 2020)



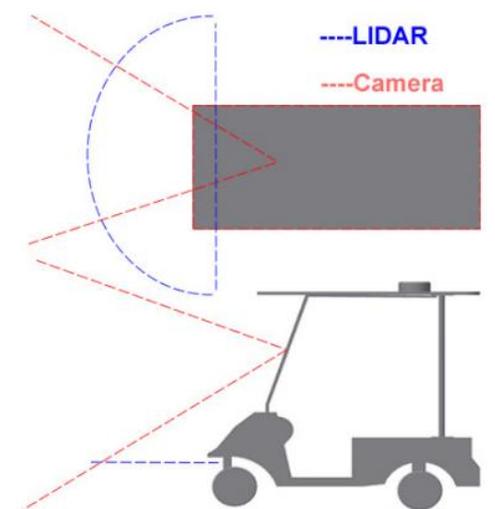
This system utilised the Perspective-n-Point (PnP) problem which is a core aspect of geometry-based methods of monocular obstacle detection. These methods estimate the relative orientation of a camera in terms of its rotation and translation relative to the 3D feature points of an obstacle after it has been projected onto a 2D image plane. The solving of this PnP problem is typically approached in an either iterative or non-iterative manner, with more iterative solutions requiring more computational power to run efficiently, and as a result being harder to implement. (Zhang, Cao, Ding, Zhuang, Tao 2020)

2.3.2.3 – Monocular Example 3: LIDAR Hybrid – Overhang Obstacles

In some applications, a monocular camera system is supplemented with a traditional distance sensor. This is typically done in order to reduce the computational load on a system which typically accompanies purely monocular obstacle detection and help to circumvent the inherent detection weaknesses of each sensor (Young,& Simic 2015). Generally, monocular camera systems do not provide as reliable distance information as other better suited sensors, instead providing more reliable information on the environment in a 2D manner. In contrast, light detection and ranging sensors (LIDAR) are capable of providing reliable distance information but are limited by their orientation in the points at which they can conduct this measurement. Research which has sought to combine these two technologies in a way which compensates for each respective sensor's weakness have been conducted, with particular focus given to autonomous driving applications. (Young,& Simic 2015)

Such research has specifically cited the inability for LIDARs to identify overhangs outside their field of view, which may impede the systems movement, as one of the key issues which the use of a camera could remedy. As such, the camera portion of the system was

focussed entirely on the detection of overhang obstacles, with the LIDAR detecting traditional, positive obstacles. The obstacle detection system the research designed to perform such identification uses the LIDAR as the primary sensory system. Acting under the assumption that all overhangs the system would encounter would be supported by traditional obstacles within the LIDAR's radius, the overhang detection programming is only activated once the LIDAR has detected such obstacles. When this occurs, the monocular camera runs detection algorithms to determine whether an overhang is present, and if one is discovered, the system will adapt its pathing accordingly. (Young,& Simic 2015)



(Young & Simic 2015)



Experimentation into the effectiveness of such a system was conducted using a golf cart as a base in a configuration shown in Figure 2.8, making use of a relatively low-resolution camera and a Raspberry Pi board as a controller. This configuration was tested on its ability to identify a boom gate, a robotic arm and block tape as examples of the types of overhang obstacles the system was designed for. The final iteration of the identification code resulted in obstacle detection at an average of approximately half a second across the 3 types of obstacles tested. Overall, the experimentation demonstrated the navigational capabilities of the system, with the system capable of detecting the relatively small overhanging obstacles reliably and adjusting its course accordingly. The only downfall identified of the proposed system was the speed at which it conducted its detection, with the current iteration not processing fast enough for autonomous vehicle applications. (Young,& Simic 2015)



2.3.3 - Stereo Vision:

2.3.3.1 – Stereo Example 1: Dynamic Feature Tracking:

Some research regarding stereo vision have approached object detection in a way which creates 3D depth maps of specific features within the input images. One particular example of this form of object detection focused on dynamic obstacles passing by a stationary stereo camera system (Sharma, Sahoo, Manivannan 2018). Making use of a simplistic camera setup consisting of a pair of USB webcams, the research focussed on the utilisation of the Kanade-Lucas-Tomasi (KLT) feature tracker in conjunction with the stereo camera system as a means of performing dynamic obstacle detection. The system's stereo vision depth calculations were performed by making use of the equations outlined in Figure 1.1, with MATLAB being used to determine the relevant camera intrinsic values. The KLT feature tracker is capable of following the movement of specified feature points between subsequent images, this even includes the reacquisition of feature points lost due to factors such as illumination. This however is reliant on the assumption that the dynamic objects have a consistent shape and that the environment's texture remains unchanged. The interaction of the KLT algorithms with the stereo vision is relatively simplistic. As the KLT tracked the selected feature points as they progressed through the video's frames, the depth of each of those points was calculated, creating a depth map for each frame. (Sharma, Sahoo, Manivannan 2018)

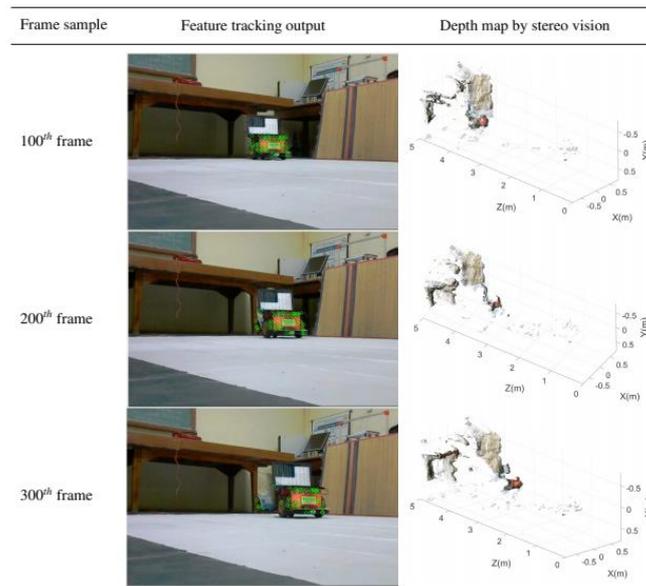


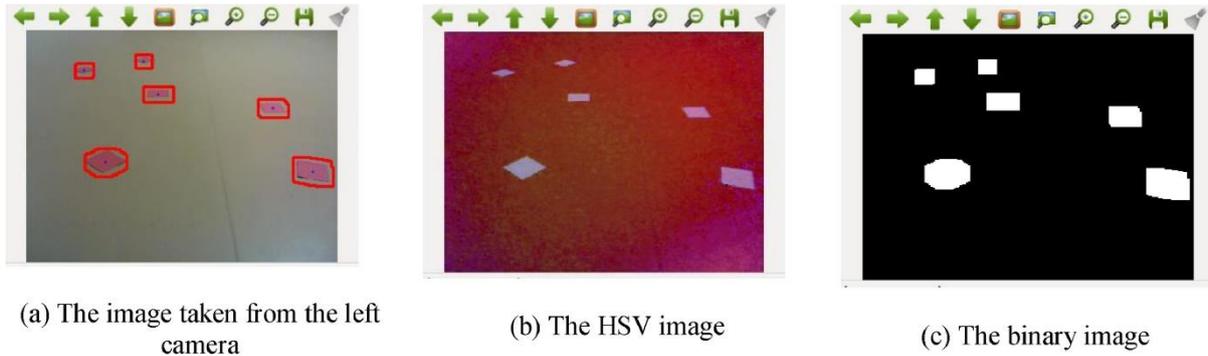
Figure 2.9: Dynamic Feature Tracking and Depth Map (Sharma, Sahoo, Manivannan 2018)



The testing of this system both demonstrated some of its capabilities and highlighted some downfalls. Foremost, the system proved capable of determining an object's path, plotting relatively accurate approximations of the path's the test obstacles followed. The primary downfall identified was that the speed of the dynamic obstacle affected the system's ability to maintain its feature points, with faster speeds increasing the amount lost during analysis. As previously stated regarding stereo vision, the system's stereo camera setup, being two cameras 128.28mm apart, with fields of vision up to 60° limited its ability to perceive objects within a metre of the system. (Sharma, Sahoo, Manivannan 2018)

2.3.3.2 – Stereo Vision Example 2: Mobile Robot Platforms – Hybrid

Despite the inherent advantages stereo-vision systems possess, as with monocular sensors, some methods choose to augment their stereo camera systems with additional functionality, expanding upon the capabilities of the system. One particular example of such a hybrid system focussed on providing a stereo vision system with the capability to detect objects while the camera angled downwards (SOLAK & BOLAT 2018). The proposed system made use of the standard stereo triangulation concepts for the applicable regions, forming a standard obstacle avoidance system. The detection of objects within the images is completed using an adaptable thresholding process which determines the properties of visible objects prior to the thresholding process. This is supplemented by the processing of the image to remove noise and blurring as well as the conversion of the input image from a RGB to Hue-Saturation-Value (HSV) image which displays changes in light levels more clearly. Once the thresholding process has created a binary image, additional functions are utilised in order to improve its clarity by filling of gaps within the identified regions and making each individual object more distinct. This corrected version of the input image's binary mask is then utilised in the determination of object's dimensions and central coordinates for use in the depth estimation process. Supplementary to the standard stereo vision triangulation, the system performs Look-up Table and Curve-fitting operations. The Look-up Table consists of pre-programmed data detailing the relationship between disparities and depth values taken from examples, with multiple distance measurements for each disparity value used to create an average value for the system to use. The Curve-Fitting process then makes use of the table's data in the construction of a function which describes the relationship between disparity and depth, which serves as an additional means of determining a point's depth. (SOLAK & BOLAT 2018)



Testing of the system was conducted, comparing the distance values the system returned with both a

Figure 2.10: Stereo Vision Example 2 Image Segmentation Process (SOLAK & BOLAT 2018)

traditional ruler and a laser distance meter. The testing conditions included three main studies, one in which the cameras were level, another in which the cameras were angled downwards and a third which the horizontal and vertical distances of the obstacles were varied. The system demonstrated a high degree of accuracy with all the test scenarios resulting in average accuracies above 95%. Notably, the second scenario, which was engineered as to employ the Curve-fitting depth estimation, only demonstrated an approximately 0.5% reduction in average accuracy, demonstrating that, while less accurate than the stereo vision, the Curve-fitting retained a high degree of accuracy. (SOLAK & BOLAT 2018)



2.4 - Evaluation:

In order to complete the second objective of this project, the examples outlined in the literature review will be evaluated in order to determine the general capabilities of their sensory systems and control algorithms which constitute their obstacle detection systems. This will involve the identification of the strengths and downfalls of using each system, taking into account the methods some examples have used in order to circumvent the inherent shortcomings of their primary sensory system.

The research conducted into obstacle detection systems indicates that the generation of 3D maps is possible with each of the sensors the literature review discussed. However, for obstacle detection and avoidance purposes, this appears largely unnecessary. Several sources have indicated that comprehensive environmental data such as 3D reconstruction is not vital in obstacle avoidance, with rough information regarding the position of obstacles in the local area being sufficient for mobile robot applications. Some sources directly state this (Monocular Example 1) whilst others simply demonstrate the possibility of performing obstacle detection without such processing. This can be attributed to the resource requirements associated with the generation of 3D maps which can be impractical when considering the controllers used in the operation of mobile robots.

The advantages of ultrasonic based obstacle detection systems primarily lie in the functionality of the sensor itself. The fact that an ultrasonic sensor is unaffected by visual obstructions can prove advantageous in applications in which this is a common occurrence, such as outdoors, but in environments where this is not an issue, other sensors can provide more comprehensive and accurate environmental data. The primary reason ultrasonic sensors are used as often as they are in obstacle detection systems is their cost effectiveness, which results in them being more viable for the design of low-cost obstacle detection systems.

LIDAR sensors outclass ultrasonic sensors in terms of both effective range and accuracy but are much more expensive as a result. Despite this cost being somewhat mitigated if a 2D LIDAR is used, to the detriment of reducing the effective area the sensor can perceive. Despite methods existing to circumvent the reduced effective area of the 2D iteration, the proposed method outlined in LIDAR Example 1 would only be effective when used in a system in motion.



Both the monocular and stereo vision camera systems are capable of providing more than just depth data, which can be helpful in the obstacle detection process. This extra data can allow for more accurate classification of obstacles, considering more than just their geometric shape. Monocular systems appear to be largely inferior to stereo vision systems, requiring additional sensors or extensive interpretation to generate any depth data.

Therefore, taking these factors into account, the most optimal sensory system to base the project's sample system around would be stereo vision. Based upon the research conducted, a stereo vision system's detection area is best suited for detecting obstacles at a lower elevation than the sensor. For this project, the ability to process colour data is vital in the detection of paper documentation and other flat objects in the assessed area, a process which is only possible when utilising either of the camera-based systems. Research however demonstrated that stereo vision systems involve less complex interpretation, mitigating the cost of a second camera. It was also determined that, as opposed to creating 3D depth maps of entire images, image segmentation techniques should be employed in order to isolate the obstacles within an image for depth analysis.

Following the determination that a stereo vision system will form the base of this project's object detection system, the means by which the input images will be processed in order for depth analysis to be conducted must be considered. Due to the nature of a stereo vision system, image segmentation would need to be employed in the sample system, and as such an image classification DLNN would not be applicable for this project. Amongst the segmentation methods discussed, the thresholding method would be the easiest to create, merely requiring the establishment of a series of colour thresholds, to which an image's pixels will be compared to. Use of this method comes with two primary issues, the first of which being that the thresholding method is susceptible to noise, and as such additional processing may be needed beyond the initial segmentation in order to create useable results. The secondary issue comes from the limitations of thresholding as a segmentation method, restricting the types of obstacles that can be identified to those of a specific colour. The semantic and instance segmentation methods possible using a DLNN do not have these issues, with their customisable nature affording them a high degree of versatility. This is counterbalanced by the fact that they are significantly more difficult to create and train, requiring extensive coding to create a relatively simple system, and in some cases specialised hardware.



Considering the benefits and limitations of the discussed image segmentation methods, the following conclusions were reached regarding the project's sample system. Despite the inherent advantages of DLNN based image segmentation, the time it would take to prepare such a system is impractical with the project's time constraints. As such the initial system will utilise a thresholding method to segment the input images. Due to the fact that the project is largely focussed on determining the system's ability to detect different object shapes as opposed to colours, the testing objects can be prepared in a manner compensate for this limitation. In order to correct the noise a thresholding system is likely to perceive, additional functions will need to be utilised to refine the data for the stereo vision system to process. Should this system prove to be unreliable in the preliminary testing phase of the research methodology, a DLNN image segmentation method will be explored, should time permit. Additionally, in the event that the thresholding system proves capable of producing acceptable results, if time constraints allow, a DLNN could be prepared and used as a point of comparison for analysis.

It was also determined that the feature tracking methodology discussed in Stereo Example 1 would not be necessary to implement as part of this project's system. This is due to the fact that the example in question was designed to identify moving objects with a stationary camera, factors which fall outside the objectives of this project as the situations they are drawn from typically involve a moving camera and stationary objects. Even if such functionality was relevant to the project's potential applications the difficulty and time cost associated with implementation of such a system would be impractical with the project's time constraints.



2.5 - Potential Outcomes and Consequential Effects:

Due to the fact that this project's scope is limited to the evaluation of the effectiveness of an obstacle detection system, the consequences of its outcomes are not severe. In a broad scope, this project's methodology would simply provide details regarding the accuracy of the sample obstacle detection system as it analysed different forms of low lying, indoor obstacles. Should the system prove capable of reliably determining the positions of such obstacles, it would provide a precedent for stereo vision to be utilised in indoor environments where such obstacles appear. With this confirmation, the sample system could then be further adapted in order to increase its viability in practical applications. This would likely involve the enhancement of the system to operate in real time and potentially even act as portion of an obstacle avoidance code capable of controlling a mobile robot. Otherwise, if the system is determined to be incapable of producing accurate results, other methods of detecting the analysed types of obstacles would need to be determined. This would prompt further research into such detection to either improve upon the sample system, attempting to correct its flaws, or pursue another method entirely.



Chapter 3: Methodology

3.1 - Methodology Preparation:

In order to conduct the research methodology to complete the aim of this project, a sample obstacle detection system was created. Based upon the factors identified in the literature review, the optimal sensor system to utilise for this project was determined to be a stereo vision system. It was determined that the MATLAB programming environment was capable of operating a stereo vision system, and as such, its documentation was consulted during the preparation of the sample obstacle detection system required for the methodology. (MathWorks 2021)

3.1.1 - Hardware:

The hardware required for the stereo vision system being utilised in the research methodology is relatively simplistic. In order to ensure the accuracy of the stereo vision system, a pair of identical USB webcams were used. The MATLAB features utilised in the calibration of the stereo vision system can be performed using any pair of cameras, so long as their individual field of vision are below 95 degrees. This pair of cameras were mounted on a simple wooden structure with their lenses positioned 80mm apart. This distance was chosen in order to ensure the system was capable of accurately determining the depth of objects in close proximity to the cameras, as per the stereo vision theory identified in the literature review. The only other hardware the system required was a computer capable of running the MATLAB programming platform, on which the system's software aspects are run.



Figure 3.1: Sample System Hardware Setup



3.1.2 - Software:

The software aspects of the sample obstacle detection system were created utilising the MATLAB programming platform, utilising applications included as part of the Image Processing and Computer Vision toolbox.

3.1.2.1 - Stereo Camera Calibration:

In order to be capable of performing the stereo depth estimation required for reliable obstacle detection, the parameters of the stereo camera system must be determined. In order to generate these parameters for the camera setup used, MATLAB's Stereo Camera Calibrator app was utilised. This parameter determination can be conducted on any stereo vision setup provided each camera's field of vision is below the 95-degree maximum. To accomplish this, the app must be provided with a series of image pairs, each featuring a specific checkerboard pattern available in MATLAB. By measuring the disparity between the points on the checkerboard, the application is capable of determining the relative position of the checkerboard with respect to the centre of the stereo vision system. In order to increase the accuracy of these calculations, the provided image pairs should show the checkerboard pattern in positions covering the entire shared region, orientating the pattern at no more than a 45-degree angle to the image plane. The distance at which this calibration is conducted also affects the accuracy of the system, with the distance between the checkerboard and the camera pair during calibration being approximately the same as where objects would be detected.

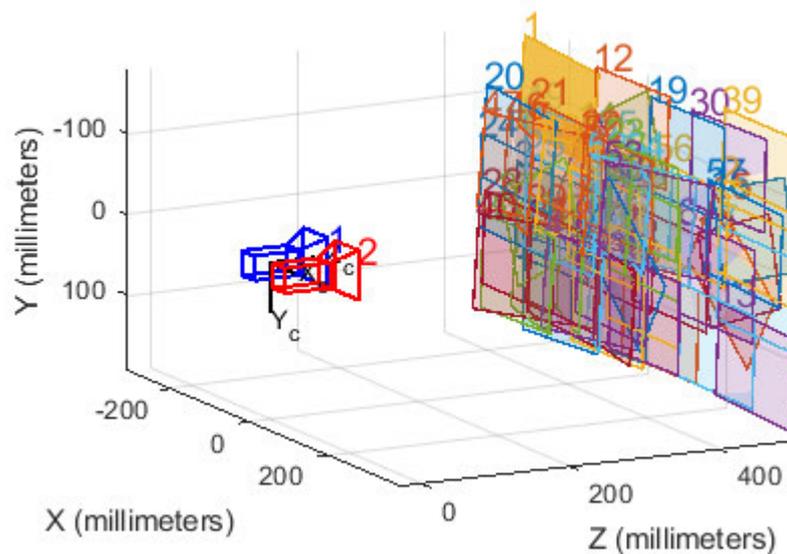


Figure 3.2: Stereo Camera Calibration



Specifically for this project, the stereo calibration was performed using 57 pairs of images. These were taken with the checkerboard covering shared image planes 500mm and 400mm away from the camera, as well as several pairs depicting the checkerboard at angles to the image plane. This series of images produced a mean reprojection error of 0.38 pixels and produced a camera centric view of the calibration images as shown in Figure 3.2.

3.1.2.2 - Image Segmentation:

Concurrently to the generation of the stereo camera parameters, the image pairs are also used to generate individual parameters for each camera. These are utilised in the correction of image distortion using the *'undistortImage'* MATLAB function. These corrected images are then utilised in the image segmentation process, in order to single out the appropriate pixels for depth analysis. The segmentation methods considered for this project all provide their results in the form of a binary image, highlighting the detected regions when used as a binary mask.

The current iteration performs this segmentation using colour thresholding methodology. The code which performs this segmentation was created utilising MATLAB's colour thresholder app. This app allows for the thresholding of images in a manner which interactively demonstrates what result it would produce. The process was performed using a single test image, in which the thresholding was configured to identify the darkest obstacles in the environment. This segmentation was then further processed by methods which clear the borders and fill any holes in the segmented region. In the majority of cases, it is unlikely that this process would exclusively highlight pixels containing the objects in question, as such one further function was utilised. The *'bwareafilt'* function is used to isolate the largest cluster of pixels in a binary image which, in this case, should highlight the obstacle in question.



3.1.2.3 - Depth Calculation:

Once the binary masks for each image have been created, highlighting the obstacles within them, the depth estimation process is conducted. This system's depth estimation process involves the calculation of three separate depth values, these being the minimum, average and central depth of the identified obstacle. The average and minimum depths of the obstacle are determined from depth data calculated on each of the pixels highlighted by the binary mask, whilst the central depth is simply the depth at the centre of the identified area. All these depth calculations are completed using the 'triangulate' MATLAB function which, when provided with the system's stereo parameters and a set of coordinates from each of the input images, calculates the relative depth of the given pixel.

3.1.2.4 - Other Code Functions:

The systems code also includes a series of image outputs which display aspects of the code's functionality. Firstly, using the stereo parameters, a stereo anaglyph and disparity map are generated for the image inputs, displaying the disparity between the left and right camera perspectives. Secondly, the binary image mask generated by the system's segmentation algorithms is overlaid on top of each input image, highlighting both the obstacle being analysed, as well as its identified centre.

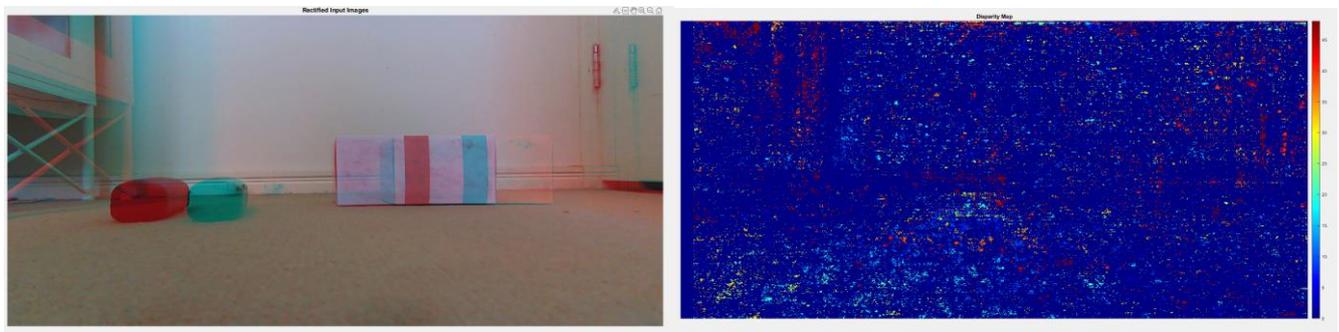


Figure 3.3: Stereo Anaglyph and Disparity Map



3.1.3- Code Depth Estimation Process:

This section will provide a brief outline of the manner in which the sample system's control code functions, detailing the sequence of its operation and providing examples of the image processing performed throughout the depth analysis process. In order to properly perform depth analysis on images, first the sample system's code loads the relevant camera parameters for the stereo camera system and each of its individual lenses. Following this the desired image pair is loaded into the MATLAB workspace and subsequently rectified through the use of the 'rectifyStereoImages' which utilises the stereo parameters to correct the distortion in the image.

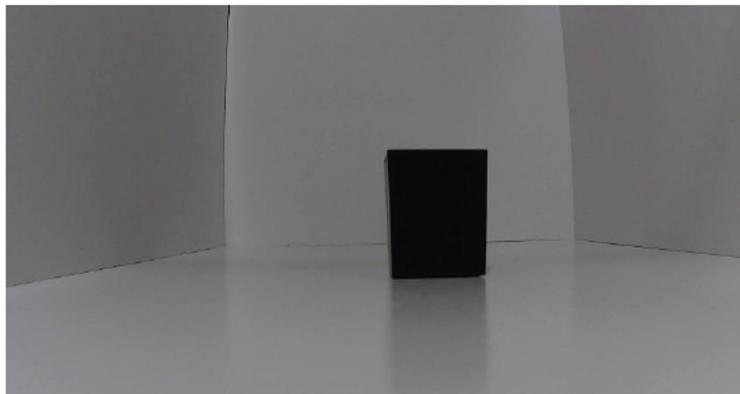


Figure 3.4: Input Image

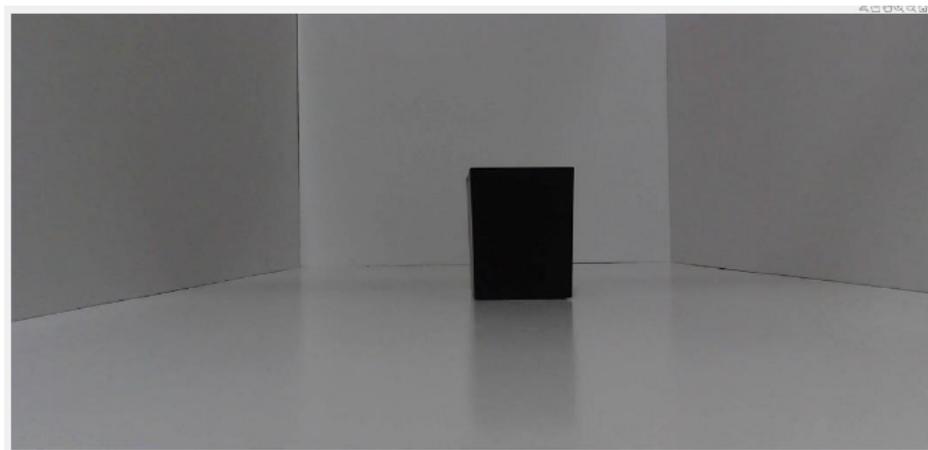


Figure 3.5: Rectified Image

Each of these rectified images are then further processed using the 'undistortImage' MATLAB function which corrects image distortion similarly to the 'rectifyStereoImages' function but instead utilising the camera parameters for the individual lenses.

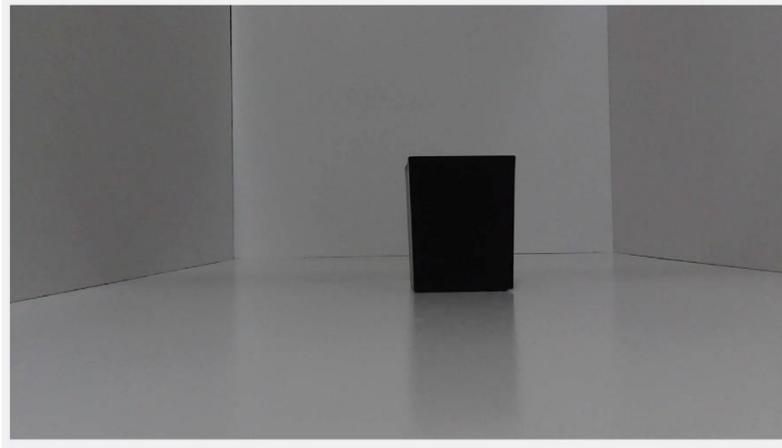


Figure 3.6: Undistorted Image

Following this final correction, the images are then processed through the image segmentation algorithm which produce a binary mask indicating the pixels which meet the specified colour thresholds.

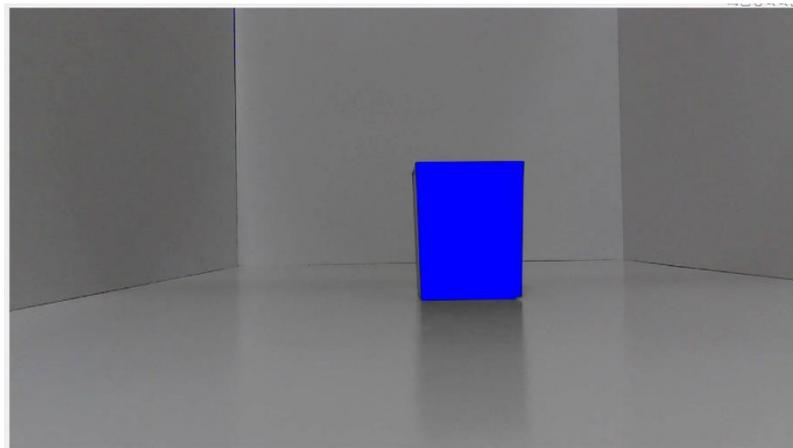


Figure 3.7: Binary Mask (Overlaid on Undistorted Image)

This binary mask is then processed using the ‘bwareafilt’ function to isolate the largest collection of identified pixels which, under ideal circumstances, should indicate the presence of an obstacle. Within Figure 3.7 a small cluster of identified pixels can be seen in the upper left hand corner of the test environment which are not highlighted in the more processed Figure 3.8.

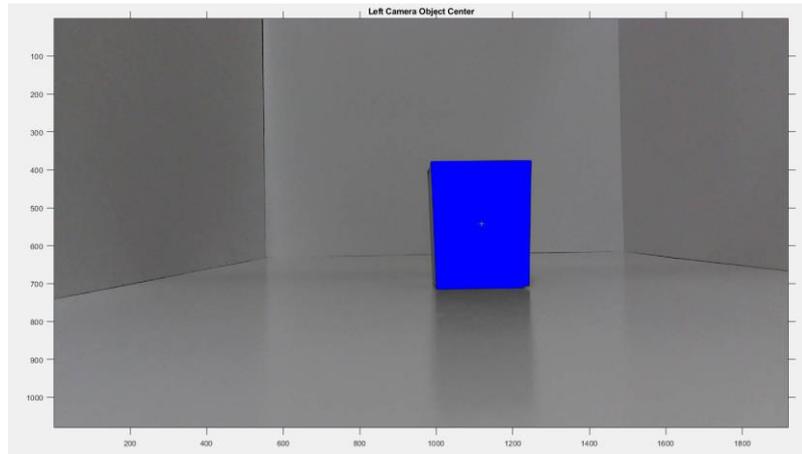


Figure 3.8: Completed Image Processing

The dimensions and relative position of the largest obstacle within the image are determined and provided to the ‘triangulate’ function in order to produce the system’s three depth values as outputs. This includes the calculation of the coordinates centre pixel in the identified shape, which is then used to calculate the central depth, and then the depth of every pixel in the identified cluster is determined, of which the average and minimum values are provided as the average and minimum depths. These three depth values are then displayed in the MATLAB command window.

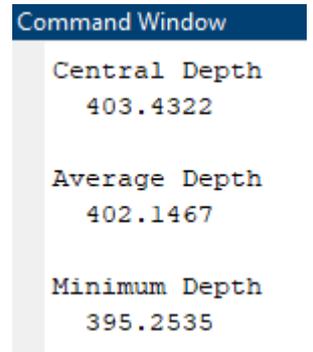


Figure 3.9: System Output



3.2 - Research Methodology:

In accordance with this project's aim, its research methodology revolves around the analysis of an object detection system. As outlined in the methodology preparation section, a stereo camera system was chosen as the basis for the sample obstacle detection code as it was deemed best suited for the forms of obstacles on which this project was focussed.

3.2.1 - Preliminary Testing:

The first portion of the methodology revolves around the confirmation of the sample obstacle detection system's abilities. This will involve a series of tests utilising large and distinct obstacles which fall outside the core aims of the project. Conducting these tests will allow for the evaluation of the baseline capabilities of the sample system, ensuring the baseline reliability of the system. This is vital as to determine whether disparities between the results of the second phase of the research methodology are as a result of the types of obstacles being analysed, or errors in the system's programming. Any major errors identified during this stage of the methodology must be remedied prior to progression to the second phase.

3.2.2 - Controlled Scenario Testing:

Once the effectiveness of the sample system has been evaluated and deemed adequate, the second phase of the research methodology will be undertaken. This second phase will involve the majority of this project's experimentation, testing the validated sample detection system in a series of controlled scenarios. As per the project's aim, a series of tests will be performed covering a variety of low-lying, small scale obstacle types varying in number and location within the test environment. These tests will also include variation in the position of the stereo camera system within the test environment, collecting images of the test obstacles at different elevations and angles for analysis. As the primary focus of this phase of the methodology is the evaluation of the system's obstacle detection abilities, the test environments utilised are tailored to facilitate this detection. This will entail the controlled test scenarios presenting as minimal excess information to the system as possible, removing any potential interference the environment may cause in the detection process.



3.2.3 - Alternate Environment Testing:

The third phase of the methodology focusses on the addition of environmental uncertainty into the testing process. Involving the same general testing factors as the previous phase, the third phase will replace the controlled testing environment with more variable environments. This will still focus on the detection of the same obstacle types, whilst still alternating the locations of both them and the camera system, with the addition of environmental changes. These environmental changes include altering factors such as light levels and the properties of the floor and background in order to further test the system.

3.2.4 – Evaluation:

Once the testing of the second and third phase of the methodology have been completed, the evaluation phase will be conducted. This simply involves the interpretation of the test images, evaluating the sample system's effectiveness for each test. Both the accuracy of the segmentation process and the depth estimation will be considered and recorded for each image pair. Following this interpretation, correlations within the data will be sought with the intent of determining which test scenarios produced the most reliable results and what this data would entail if applied to a practical setting.



3.3 - Preliminary Testing:

The preliminary testing phase of the research methodology serves the purpose of observing the system's object detection capability via a series of basic tests and evaluations. As this project's object detection code utilises the colour thresholding method to perform image segmentation, the preliminary testing environment was constructed in order to best accommodate for its limitations. As previously discussed, thresholding methods of image segmentation are susceptible to noise from environmental factors due to their use of colour as a means of differentiating between pixels and by extension the objects within an image. The elements of the testing environment's design were chosen to allow the segmentation to be performed independent of the system's inherent weaknesses in order to observe the effectiveness of the chosen threshold and the stereo depth estimations. In order to ensure uncertain environmental factors as texture and inconsistent colour interfere with the results of the preliminary testing, the test environment needed to isolate the system's field of view and provide a consistent colour and texture which would cause the minimal impact on the results. A set of four of white MDF boards were chosen to form the testing chamber, encapsulating the entirety of the stereo camera's field of vision whilst still allowing for typical indoor light levels. This not only provides a uniform texture for the test environment but also produces the optimal contrast with the dark colours the thresholding method is best suited to identify.

The testing environment constructed under this logic is shown in Figure 3.10.

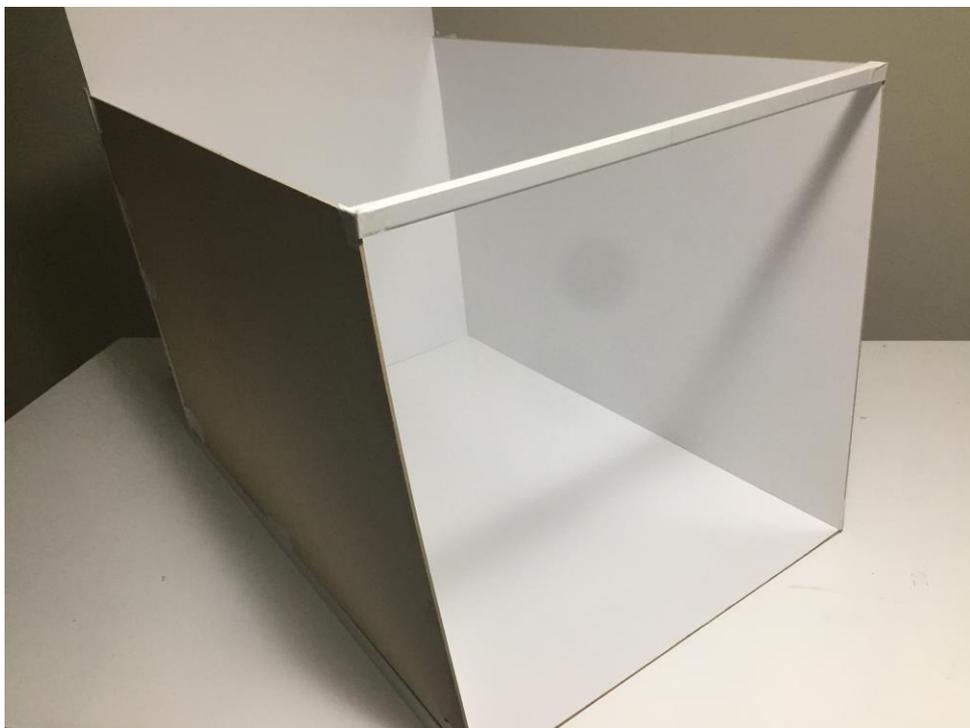


Figure 3.10: Testing Environment



The preliminary testing itself was conducted using a larger, more distinct object than those described in the project's aim. The series of tests vary both the depth and horizontal position of the test obstacle to produce more comprehensive results and determine whether the system's accuracy has a measurable relationship with any of these factors. The MATLAB documentation referenced in the creation of the stereo vision system states that the calibration performed should produce results that are the most accurate at the distance the checkerboard was from the camera during calibration. As such, the preliminary testing will also provide a means of determining the degree to which the accuracy changes outside of the 400-500mm range the system was calibrated at.

The object chosen to act as a sample for the preliminary testing was a trapezoidal pen cup as shown in Figure 3.4. This was chosen not only because of the colour contrast with the test environment, but also the fact that it is an object which may appear within an indoor environment. This test object was placed within the testing environment at a specified position relative to the stereo camera system and an image pair was obtained. This process was repeated, alternating the lateral position of the object 200mm to either side of the camera's centre in increments of 40mm. Once a set of 11 image pairs for the distance were obtained, the process was repeated for 20mm further away from the camera resulting in a set of distances ranging from 380mm to 520mm. The coordinate system the data analysis will be utilising is detailed in Figure 3.11.

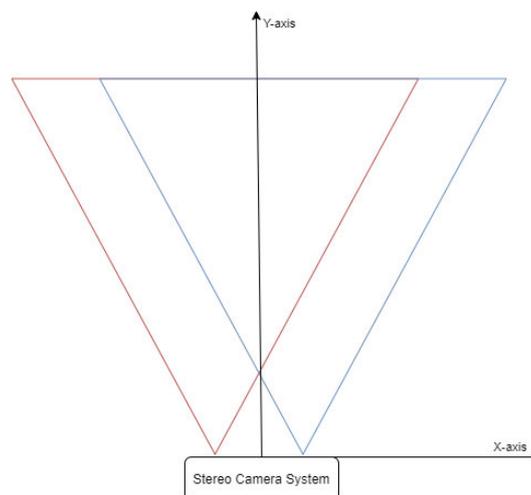


Figure 3.11: Testing Coordinate System



3.4 - Sample Obstacle Testing:

Making use of the understanding of the object detection system obtained through the preliminary testing, the next stages of analysis can be performed. As stated in the project objectives, the final stage of analysis focusses on the observation of the system's ability to detect sample objects which may appear in an indoor environment, in which self-navigating mobile robots may be employed. The sample objects were required to be dark in colour, as to ensure the image segmentation method employed by the sample system was capable of differentiating them from other features in the input images. 4 test objects were chosen which were tested in the controlled environment previously used in the preliminary testing, as well as in alternative indoor environments with different environmental factors.

3.4.1 - Controlled Environment Testing:

The testing of the sample objects undertaken in the controlled environment will constitute the first stage of the sample obstacle testing. This testing will serve the purpose of comparing the identification of the sample obstacles to the trends identified during the preliminary testing. This should allow for the isolation of the effects of the objects themselves on the identification and depth analysis processes, removing any minor abnormalities the sample system naturally has. The sample objects on which this analysis will be performed were chosen as examples which best represent general obstacles which could be on the floor of any given indoor environment, as to ensure the analysis could be applicable to a wide array of indoor environments. The four forms of sample obstacle chosen for analysis include an A4 sheet of paper, an SD card, a corded mouse and more generalized rectangular barrier of various heights. The analysis of the sample systems ability to detect these sample objects will also focus on identifying how certain characteristics of an object can affect the systems readings. As such less variety in the object's positions within the image planes were required, due to data regarding how an objects position affects the readings being available in the form of the preliminary analysis.



3.4.2 - Alternate Environment Testing:

Once the impact the characteristics of the sample objects have on the sample system's detection capabilities, the testing then proceeds to alternate environments in order to determine the impact their conditions may have on the accuracy of object detection and depth estimation. Due to the nature of the segmentation method, the impact that the lighting of an environment can be inferred, as such the alternate testing environments focus on the variation of other factors such as the texture and colouration of the ground on which the sample objects are situated. One area of particular focus will be the observation of how the identification of additional pixels surrounding the sample object will affect the accuracy of the systems results, as such a scenario is amongst the most likely abnormalities to occur when segmenting images by colour thresholding.

3.4.3 - Testing Process:

The process under which the sample object detection system's response to the chosen obstacles was determined was conducted in the following manner. The testing of the SD card done similar to the preliminary testing, with image pairs captured at increments of 50mm within a range of -100mm to 100mm in the x-direction and 400mm to 500mm in the y-direction. These tests were conducted with the same SD card in two different orientations. The A4 sheet of paper was similarly tested at different orientations, being 0, 45, 90 and 135 degrees. Due to the size of the paper relative to the image plane, it was decided that only the distance in the y-direction would be changed between tests and to within a different range than the SD card, specifically between 350 and 450. The corded mouse was tested in 5 different orientations, alternating the y-position between 400mm and 500mm in increments of 50mm. Sample image pairs detailing the different orientations of the relevant objects will be provided along with the raw result and error tables in the results chapter of this report. In order to best conduct the analysis of the image pairs for each sample obstacle, the colour thresholding values of the segmentation code were adjusted as to ensure that the system recognised the sample obstacle within the image.

Chapter 4: Resources, Risk Assessment and Timelines

4.1 - Project Resources:

In order to complete this project's methodology, several resources will be required. Due to the nature of this project and its focus on software for a large amount of its methodology preparation, it does not require an extensive list of resources. The following is a general list of the resources required to complete the project.

Resource	Expected Quantity	Source	Estimated Costs
PC with Windows 10	1	Student	nil
MATLAB	1	Student	nil
MATLAB: Image Processing and Computer Vision Toolbox	1	Student	nil
Microsoft Word	1	Student	nil
USB Webcam	2	Student	<\$60 Each
Construction Materials	~4 Square Meters	Student	<\$50

Table 4.1: Resource Requirements

4.2 -Risk Assessment:

Due to the nature of this project, the conduction of its methodology does not present extensive risks to people or equipment. The following tables are adapted from the USQ safety risk management system, classifying the risks involved in project work by their likelihood of occurrence and potential consequences.

Probability	Consequence				
	1.Insignificant: No Injury 0-\$5k	2.Minor: First Aid \$5k-\$50k	3.Moderate: Med Treatment \$50k-\$100k	4.Major: Serious Injury \$100k-\$250k	5.Catastrophic: Death More than \$250k
A. Almost Certain: 1 in 2	Moderate Risk	High Risk	Extreme Risk	Extreme Risk	Extreme Risk
B. Likely: 1 in 100	Moderate Risk	High Risk	High Risk	Extreme Risk	Extreme Risk
C. Possible: 1 in 1000	Low Risk	Moderate Risk	High Risk	High Risk	High Risk
D. Unlikely: 1 in 10000	Low Risk	Low Risk	Moderate Risk	Moderate Risk	Moderate Risk
E. Rare: 1 in 1 000 000	Low Risk	Low Risk	Low Risk	Low Risk	Low Risk

Table 4.2: Risk Assessment Key



Hazard:	Risk Level:	Precautions:
Construction Equipment during rig or test scenario construction	2C: Moderate	Use of protective equipment when using potentially dangerous tools and assurance that operator is trained in their use.
Trip Hazards during testing	2D: Low	The test scenarios were constructed in a way which does not require people to navigate through them.

Table 4.3: Safety Risks and Precautions

4.3 - Project Plan:

In order to facilitate the completion of the project's methodology within the available time, a timeline was created as an outline of the intended progress. Table 4 outlines the original project plan created at the commencement of the project, conceived as the best-case progression. The timeline divides the entirety of the project's work, allocating certain tasks each week of the first semester, leaving the second semester open should issues arise. As work on the project progressed, it became clear that the initial plan could not be adhered to as closely, due to both issues with completing the allocated tasks on time, as well as the refinement of the project's core tasks. As such it was necessary to create a new project plan as detailed in Table 5.



Table 4.5: Altered Project Plan

	Semester 1: ENG4111															Exams/Break				Semester 2: ENG41112																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	
Phase 1 - Preparation																																				
1.A: Approval																																				
1.B: Initial Consultation																																				
1.C: Resource Acquisition																																				
1.D: Background Research																																				
Phase 2 – Stereo Vision Setup																																				
2.A: Stereo Vision Research																																				
2.B: Stereo Vision Coding																																				
2.C: Stereo Vision Pre-Testing																																				
Phase 3 – Image Segmentation Setup																																				
3.A: Image Segmentation Research																																				
3.B: Image Segmentation Coding																																				
3.C: Image Segmentation Pre-Testing																																				
Phase 4 – Finalisation and Testing																																				
4.A: Preliminary Testing																																				
4.B: Test Scenario Preparation																																				
4.C: Controlled Scenario Testing																																				
4.D: Alternate Environment Testing																																				
Phase 5 – Evaluation																																				
5.A: Results Evaluation																																				
5.B: Graph Construction																																				
Phase 6 – Dissertation and Presentation																																				
6.A: Draft Dissertation																																				
6.B: Project Conference Presentation																																				
6.C: Final Dissertation																																				



Chapter 5: Results and Analysis

5.1 – Preliminary Testing:

The preliminary testing, as described in section 3.1.1 involved the collection of a series of 88 separate image pairs each detailing the same test object shown in Figure 5.1. These image pairs were each processed through the sample object detection system produced in the methodology preparation, and each of the depth values the system returned recorded in the following tables. Organised based upon the position of the object within the test environment for the appropriate image pair, the tables are divided into a series of 1x3 cell groups detailing the depth estimations for the relative positions. These are organised with the first entry of each collection of cells containing the central depth estimation, the second containing the average depth estimation and the third containing the minimum depth estimation. All of these values are in units of millimetres and denote the absolute distance between the stereo camera system and the identified object. This data was then further interpreted to determine the error in the estimation process. This involved direct comparison between the output values and the actual position of the object within the test environment, retrieving an error value in terms of percentage, organised in the same manner as the raw data. These values were then further collated to determine several average error values which detail the average for each estimation method, for the system as a whole, for each half of the effective range and for data excluding outliers. Within the tables, entries listed as ‘ERROR’ produced estimations with error values in excess of 100% and were therefore discounted as relevant data.

This method of interpreting data was performed for three separate tests, analysing the same 88 image pairs with slight variations to the colour thresholds used in the image segmentation, mentioned in section 3.1.3. The following pair of images are an example of the types of image pairs analysed in the preliminary testing, specifically the pair taken with the object positioned at $x = 40\text{mm}$ and $y = 380\text{mm}$.

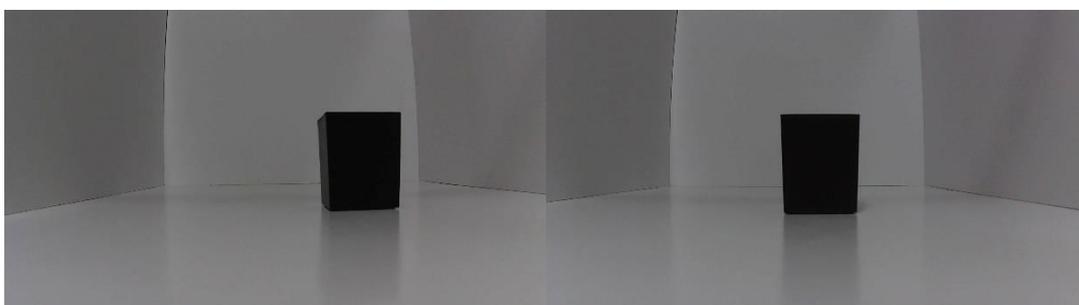


Figure 5.1: Preliminary Testing Sample Image Pair (40, 380)



5.1.1 - Preliminary Test 1:

		X Distance (mm)										
		-200	-160	-120	-80	-40	0	40	80	120	160	200
Y Distance (mm)	380	457.271	449.461	425.427	383.77	385.026	384.912	387.291	400.259	456.346	487.484	ERROR
		397.899	395.03	389.687	382.196	383.722	384.709	385.723	400.065	514.492	571.33	ERROR
		345.339	351.319	374.821	373.134	378.247	377.429	376.154	382.383	480.258	516.73	ERROR
	400	ERROR	474.438	426.306	402.417	401.891	403.929	410.248	415.598	481.906	505.671	ERROR
		ERROR	417.913	408.791	401.403	400.618	401.973	408.871	415.513	535.603	583.182	ERROR
		ERROR	381.407	397.435	391.623	396.473	395.633	400.518	400.149	508.308	539.491	ERROR
	420	475.653	487.247	463.854	422.078	418.682	419.836	426.78	432.477	493.511	493.511	538.827
		446.768	430.195	423.062	421.472	416.822	419.245	426.01	434.792	556.955	556.955	623.55
		421.619	382.745	407.327	412.811	409.673	414.725	416.777	414.691	524.019	524.019	524.814
	440	499.538	519.532	497.801	450.059	446.312	447.522	449.347	457.358	470.097	542.403	575.821
		468.466	463.074	454.33	448.966	445.179	447.322	447.982	456.85	473.893	614.791	681.944
		442.309	420.362	438.427	439.033	439.774	440.955	438.438	441.202	450.305	579.072	608.537
460	517.439	536.396	512.388	476.477	465.393	466.76	470.486	475.892	492.378	506.777	602.761	
	487.074	476.376	470.186	466.006	463.799	464.976	470.978	475.883	494.536	512.456	708.778	
	461.286	427.207	452.87	456.033	455.34	459.872	460.582	462.56	474.872	488.97	577.198	
480	533.77	552.124	500.49	481.127	487.279	485.876	490.299	499.913	515.252	546.952	603.335	
	501.582	492	483.597	479.509	486.038	484.133	489.116	499.271	516.91	570.932	687.155	
	471.663	447.888	469.058	469.977	479.597	477.418	483.385	484.531	497.576	541.028	648.146	
500	548.967	567.917	548.275	503.257	506.088	505.464	507.358	517.277	530.258	596.072	618.802	
	519.201	509.734	504.522	502.015	504.873	503.93	506.515	516.005	533.211	665.814	703.521	
	491.732	470.784	489.437	492.97	498.957	500.872	497.112	502.385	515.743	635.127	664.518	
520	568.823	588.962	566.191	522.576	523.866	525.397	527.253	539.16	549.485	615.328	642.906	
	539.401	531.527	522.387	521.673	521.466	524.84	526.381	539.55	550.406	682.38	733.446	
	512.757	489.559	508.803	513.432	515.121	518.145	518.875	526.193	532.836	654.268	674.303	

Table 5.1: Preliminary Test 1 Raw Data



		X Distance (mm)										
		-200	-160	-120	-80	-40	0	40	80	120	160	200
Y Distance (mm)	380	6.486	9.01	6.758	1.174	0.766	1.293	1.359	3.072	14.517	18.232	ERROR
		7.34	4.191	2.211	1.58	0.425	1.239	0.948	3.022	29.108	38.568	ERROR
		19.58	14.793	5.941	3.913	1.008	0.677	1.556	1.531	20.517	25.325	ERROR
	400	ERROR	10.126	2.082	1.349	0.026	0.982	2.053	1.882	15.396	17.376	ERROR
		ERROR	2.994	2.112	1.598	0.343	0.493	1.711	1.861	28.254	35.368	ERROR
		ERROR	11.468	4.832	3.996	1.374	1.092	0.367	1.905	21.718	25.226	ERROR
	420	2.25	8.411	6.192	1.28	0.763	0.039	1.157	1.152	12.981	9.805	15.83
		3.96	4.283	3.147	1.422	1.204	0.18	0.974	1.694	27.506	23.921	34.042
		9.366	14.84	6.749	3.448	2.898	1.256	1.214	3.008	19.966	16.593	12.818
	440	3.355	10.967	9.15	0.636	1.018	1.709	1.705	2.268	3.076	15.852	19.138
		3.074	1.092	0.382	0.392	0.761	1.664	1.396	2.155	3.908	31.313	41.095
		8.486	10.215	3.868	1.829	0.462	0.217	0.764	1.344	1.264	23.684	25.907
460	3.158	10.136	7.782	2.05	0.792	1.47	1.895	1.925	3.572	4.054	20.168	
	2.895	2.188	1.096	0.192	0.447	1.082	2.002	1.923	4.026	5.22	41.304	
	8.037	12.284	4.738	2.329	1.385	0.028	0.25	0.93	0.11	0.398	15.072	
480	2.648	9.123	1.156	1.129	1.166	1.224	1.793	2.731	4.139	8.101	16.026	
	3.542	2.76	2.259	1.461	0.908	0.861	1.547	2.6	4.474	12.84	32.145	
	9.296	11.478	5.197	3.42	0.429	0.538	0.357	0.429	0.567	6.93	24.643	
500	1.941	8.18	6.627	0.613	0.895	1.093	1.148	2.156	3.123	13.543	14.909	
	3.587	2.903	1.882	0.858	0.653	0.786	0.98	1.905	3.698	26.827	30.641	
	8.688	10.323	4.816	2.644	0.526	0.174	0.894	0.785	0.3	20.982	23.398	
520	2.098	8.253	6.095	0.673	0.447	1.038	1.096	2.479	2.964	13.099	15.395	
	3.183	2.303	2.114	0.845	0.014	0.931	0.929	2.553	3.137	25.424	31.646	
	7.965	10.017	4.659	2.411	1.23	0.357	0.51	0.014	0.156	20.257	21.03	

Mean			RHS	LHS
Error	Raw	Adjusted		
Centre Depth	5.374	3.123	4.019	7.662
Average Depth	7.394	1.960	2.015	14.281
Minimum Depth	6.918	3.876	6.178	9.019
Total	6.562	2.986	4.071	10.321

Table 5.2: Preliminary Test 1 Error

Through depth and accuracy data outlined in Tables 5.1 and 5.2 detail the baseline capabilities of the initial object detection system, highlighting which aspects operate as intended, and which areas require improvement. Overall, the initial system proved capable of producing depth values with an average error of below 7%, however the accuracy data displays trends which indicate how an object's position within the image frame can affect the accuracy of its depth estimation. In general the system produced more accurate data when analysing objects within the left half of the image frame. A specific example of this is the average depth estimations, the error of which rarely exceeded 4% on the left-hand side. In contrast to this, the readings obtained from objects within the right side of the image frame, prominently objects positioned beyond 16 degrees to the right of the stereo camera's field of vision, possessed a high degree of error. This was evident due to the fact that the error values drastically decreased within the 120mm column once the object's y-axis position was more than 440mm, indicating that the x-axis position was not the cause of such data. If beyond 16 degrees is considered outside the system's effective range, the average error of the system becomes less than 3% which may more accurately indicate the accuracy of the system if it was operating as intended. The analysis also indicates that, whilst it may produce the most accurate results on average, the average depth analysis method possesses the largest variation in its accuracy which may prove problematic in certain situations.

By observing the pixels highlighted by the system's segmentation program, several areas of improvement can be identified. Foremost of these areas is related to the image pairs labelled in Tables 5.1 and 5.2 as producing an error. This was assigned to analysis which resulted in an error value exceeding 100%, as was the case for the pairs in question. The image segmentation performed on these image pairs can be seen in figure 5.2, with the object not being recognised in one of the pair's images. As this only transpired for the three instances shown, it is reasonable to

assume that the fact that the object isn't separated from the from the image's edge is the cause of the issue at hand. Secondly, a general trend throughout a number of the processed image pairs is the incomplete segmentation of the test object from the background. This was likely caused by the glossy nature of the test object, reflecting light and appearing a lighter shade in certain positions. In order to resolve such an issue, the thresholding process would need to be tweaked in order to recognise the applicable surfaces of the object.

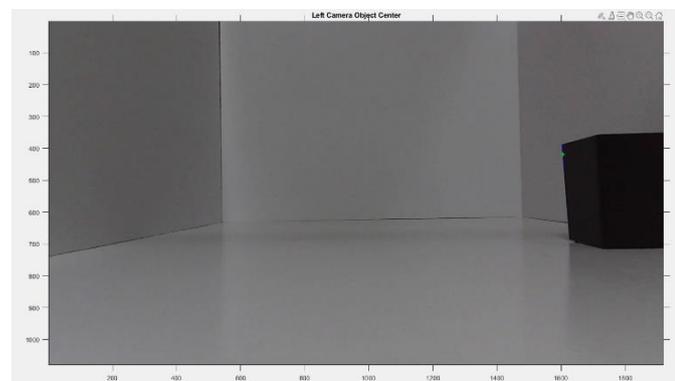


Figure 5.2: Error Image Segmentation



5.1.2 - Preliminary Test 2:

Based on the trends observed from the first preliminary test's data, the system was altered in an effort to improve its overall accuracy. The first alteration was performed on the segmentation code, adjusting each of the colour thresholds to a range of 0 to 60 to increase the range of colour shades the system will identify. This was done to correct the observed inconsistency with the previous test's segmentation, in which sections of the test object were not identified due to appearing a lighter shade as a result of the lighting conditions of the environment and the texture of the test object.

		X Distance (mm)										
		-200	-160	-120	-80	-40	0	40	80	120	160	200
Y Distance (mm)	380	426.763	422.899	417.818	404.1	395.448	385.625	404.34	429.875	437.772	453.601	ERROR
		395.909	392.819	389.18	382.094	383.79	384.512	422.981	462.448	468.095	483.794	ERROR
		373.639	374.126	373.791	372.443	376.786	368.683	407.287	442.321	315.619	450.717	ERROR
	400	ERROR	445.083	437.285	427.143	412.483	404.665	424.566	441.172	462.727	473.557	ERROR
		ERROR	416.094	407.482	402.521	400.782	403.96	438.533	470.303	492.662	504.263	ERROR
		ERROR	394.056	395.24	394.247	393.56	395.722	425.711	355.805	264.298	471.359	ERROR
	420	ERROR	459.569	450.896	449.346	422.953	419.807	441.732	463.05	470.342	486.589	502.186
		ERROR	429.944	422.055	421.542	417.053	418.454	455.323	495.546	498.139	514.387	536.843
		ERROR	409.452	406.39	412.293	412.196	337.888	466.64	478.226	279.857	484.217	496.723
	440	499.544	490.92	482.177	482.9	460.942	447.946	463.193	480.839	502.573	411.975	526.804
		468.674	460.658	452.354	448.803	449.14	447.485	473.126	507.951	545.134	544.915	560.965
		440.519	437.24	434.54	437.193	400.054	418.06	460.355	488.884	519.211	512.706	523.94
460	516.626	505.502	497.433	493.396	478.243	467.55	472.058	496.304	524.194	539.632	551.772	
	487.361	475.297	472.017	470.781	467.027	469.552	469.603	517.999	564.906	583.673	584.674	
	460.025	454.964	460	466.482	355.784	462.797	459.024	501.687	539.394	551.505	548.223	
480	532.309	521.21	512.401	509.739	507.76	488.068	492.103	523.864	545.905	551.443	567.292	
	501.941	492.649	486.588	483.76	481.23	483.801	489.704	550.339	585.367	583.152	603.176	
	477.754	472.678	476.263	478.051	449.796	419.989	479.17	531.646	306.516	555.228	568.165	
500	549.494	539.987	533.499	530.775	521.744	506.631	509.41	539.47	560.654	567.096	582.271	
	516.365	508.837	500.471	505.116	509.684	506.36	511.709	561.741	597.152	597.961	617.523	
	486.159	488.393	481.208	500.604	504.432	489.624	500.843	498.061	297.493	569.139	582.719	
520	568.707	559.414	555.65	550.871	534.868	525.87	529.519	564.712	581.737	585.982	601.567	
	540.837	529.4	511.612	525.204	523.52	526.826	531.588	593.686	621.476	621.167	636.39	
	520.51	510.519	479.955	518.431	517.308	520.15	519.691	576.833	597.952	592.643	601.788	

Table 5.3: Preliminary Test 2 Raw Results



		X Distance (mm)										
		-200	-160	-120	-80	-40	0	40	80	120	160	200
Y Distance (mm)	380	0.618	2.568	4.848	4.061	3.493	1.48	5.821	10.698	9.856	10.014	ERROR
		7.803	4.727	2.338	1.606	0.443	1.187	10.699	19.086	17.465	17.337	ERROR
		12.99	9.261	6.2	4.091	1.391	2.978	6.592	13.904	20.798	9.315	ERROR
	400	ERROR	3.312	4.711	4.712	2.609	1.166	5.615	8.151	10.803	9.922	ERROR
		ERROR	3.417	2.426	1.324	0.302	0.99	9.089	15.293	17.971	17.049	ERROR
		ERROR	8.532	5.357	3.352	2.098	1.069	5.899	12.776	36.712	9.411	ERROR
	420	ERROR	2.253	3.226	5.098	0.249	0.046	4.701	8.303	7.677	8.265	7.953
		ERROR	4.339	3.377	1.405	1.149	0.368	7.922	15.903	14.041	14.449	15.403
		ERROR	8.898	6.963	3.569	2.3	19.55	10.604	11.852	35.931	7.737	6.779
	440	3.356	4.855	5.724	7.98	4.329	1.806	4.839	7.519	10.196	12.007	8.996
		3.031	1.608	0.815	0.355	1.658	1.701	7.087	13.581	19.529	16.388	16.064
		8.856	6.61	4.721	2.241	9.452	4.986	4.196	9.318	13.845	9.508	8.404
460	2.996	3.792	4.636	5.674	3.575	1.641	2.235	6.297	10.265	10.8	10.003	
	2.838	2.409	0.711	0.83	1.146	2.077	1.704	10.943	18.829	19.843	16.562	
	8.288	6.584	3.238	0.091	22.947	0.608	0.587	7.45	13.462	13.238	9.295	
480	2.367	3.013	3.563	4.751	5.418	1.681	2.167	7.653	10.335	8.989	9.095	
	3.473	2.632	1.654	0.588	0.09	0.792	1.669	13.094	18.31	15.256	15.995	
	8.124	6.579	3.741	1.761	6.616	12.502	0.518	9.252	38.049	9.737	9.262	
500	2.039	2.859	3.754	4.822	4.016	1.326	1.557	6.539	9.035	8.023	8.125	
	4.113	3.074	2.67	0.246	1.612	1.272	2.016	10.937	16.133	13.902	14.671	
	9.723	6.969	6.416	1.137	0.565	2.075	0.15	1.639	42.144	8.412	8.208	
520	2.077	2.822	4.119	4.705	2.556	1.129	1.531	7.336	9.008	7.706	7.975	
	2.925	2.694	4.133	0.174	0.38	1.313	1.927	12.843	16.454	14.173	14.225	
	6.574	6.165	10.065	1.461	0.811	0.029	0.354	9.639	12.046	8.93	8.015	

Mean					
Error		Raw	Adjusted	RHS	LHS
	Centre Depth	5.331	4.353	3.725	7.790
	Average Depth	7.191	4.974	2.119	13.522
	Minimum Depth	8.482	7.355	5.914	11.683
	Total	7.002	5.561	3.919	10.998

Table 5.4: Preliminary Test 2 Error



A fact clearly demonstrated by the results of the second preliminary test is the increase in error as a result of the altered colour thresholds. Despite the segmentation identifying the test object more completely, the results for all three depth estimation methods demonstrated a higher degree of error. Maintaining a similar trend to the first example, with regards to the calculations on the left half of the image frame, the new values rarely reached the same error extremes present in the first preliminary test. In spite of this, there is a noticeable increase in the error values calculated on the left side, producing errors exceeding 10% more regularly whilst maintaining a similar average when compared to the initial test.

It is also important to note that, due to the new segmentation identifying the side faces of the object, the error of the centre and average depth calculations slightly increased when compared to the original testing. This is interesting due to the fact that the measurements which the depth estimations were compared to in order to determine the error in their readings were taken with respect to the front face of the object, meaning that the side faces of the test obstacle would always be slightly further away from the stereo camera, and as such should have a more noticeable impact on the average depth calculations.



5.1.3 - Preliminary Test 3:

The third iteration of the preliminary testing procedure, in contrast to the second, reduces the area which the segmentation algorithm identifies. This was done by reducing the range of the colour thresholds to a range of 0 to 20 for all three of the colour channels, ensuring the system only identifies the darkest areas within the image, in theory being exclusively the front face of the test obstacle.

		X Distance (mm)										
		-200	-160	-120	-80	-40	0	40	80	120	160	200
Y Distance (mm)	380	399.819	395.372	389.194	382.568	384.526	384.497	386.147	397.932	410.541	427.501	ERROR
		390.205	397.06	389.971	382.487	383.982	384.46	387.507	397.323	410.765	429.524	ERROR
		372.3	631.801	375.218	373.654	380.225	377.781	374.038	379.852	386.065	400.938	ERROR
	400	ERROR	417.176	408.249	401.334	401.365	403.432	409.323	413.72	435.991	446.663	ERROR
		ERROR	418.515	409.047	401.593	400.205	402.147	408.861	415.849	435.106	451.589	ERROR
		ERROR	396.953	394.302	393.883	394.039	395.254	400.064	396.413	413.08	418.002	ERROR
	420	454.923	431.954	421.925	420.19	418.254	419.588	426.233	430.824	443.68	462.375	475.124
		477.786	432.052	422.649	421.682	416.871	419.815	424.99	429.057	444.74	464.741	476.122
		458.568	412.611	407.518	415.223	409.252	413.726	417.734	414.574	420.568	435.673	440.514
	440	473.377	461.804	452.786	447.349	445.727	447.359	448.53	455.712	466.867	483.171	502.855
		482.908	462.641	454.355	449.987	445.174	446.839	447.391	454.335	468.609	486.978	508.142
		468.623	447.915	440.269	444.843	439.356	441.276	437.361	439.517	445.466	457.321	479.17
	460	490.564	476.659	468.456	463.942	464.523	466.262	470.034	474.742	489.027	502.154	533.787
		496.961	477.871	470.33	465.638	462.925	466.188	469.58	474.174	489.052	505.791	543.321
		483.681	439.551	454.759	459.055	447.078	458.928	461.811	462.69	471.148	483.846	510.629
	480	503.259	492.104	482.869	479.406	486.216	485.868	489.873	498.44	511.496	522.177	539.936
		505.354	493.174	483.691	479.773	485.924	484.139	488.937	496.229	510.203	521.56	544.482
		485.561	475.21	471.022	470.59	480.398	477.076	479.322	482.953	494.514	496.802	514.54
	500	522.408	509.834	504.686	501.853	505.588	505.099	506.587	516.208	526.592	539.334	554.304
		522.235	511.754	504.292	502.001	504.425	503.657	505.185	514.857	525.104	538.823	555.833
		490.463	483.008	487.163	492.97	498.58	498.157	496.822	500.648	510.463	513.004	530.109
	520	539.095	530.75	522.553	521.307	523.373	525.475	526.714	537.862	547.34	558.849	574.603
		541.454	532.08	522.531	521.476	521.875	523.896	525.806	536.872	545.563	557.614	576.783
		518.365	500.971	507.162	513.142	515.191	519.098	518.814	523.035	528.94	535.909	549.768

Table 5.5: Preliminary Test 3 Raw Results



		X Distance (mm)										
		-200	-160	-120	-80	-40	0	40	80	120	160	200
Y Distance (mm)	380	6.893	4.108	2.335	1.484	0.635	1.183	1.059	2.473	3.022	3.684	ERROR
		9.132	3.699	2.139	1.505	0.493	1.174	1.415	2.316	3.079	4.175	ERROR
		13.301	53.234	5.842	3.779	0.491	0.584	2.11	2.183	3.12	2.758	ERROR
	400	ERROR	3.165	2.242	1.615	0.157	0.858	1.823	1.422	4.401	3.679	ERROR
		ERROR	2.855	2.051	1.551	0.445	0.537	1.708	1.943	4.189	4.822	ERROR
		ERROR	7.86	5.582	3.441	1.979	1.187	0.48	2.821	1.085	2.974	ERROR
	420	2.207	3.891	3.407	1.722	0.864	0.098	1.027	0.766	1.574	2.877	2.136
		2.708	3.87	3.241	1.373	1.192	0.044	0.732	0.352	1.816	3.403	2.35
		1.423	8.195	6.705	2.883	2.998	1.494	0.988	3.035	3.718	3.064	5.304
	440	2.058	1.364	0.72	0.03	0.886	1.673	1.52	1.9	2.367	3.2	4.041
		0.086	1.185	0.376	0.62	0.76	1.554	1.262	1.592	2.749	4.013	5.135
		3.041	4.33	3.465	0.53	0.556	0.29	1.008	1.721	2.325	2.321	0.859
460	2.2	2.13	1.46	0.634	0.604	1.361	1.797	1.679	2.868	3.105	6.417	
	0.924	1.881	1.065	0.271	0.257	1.345	1.699	1.557	2.873	3.852	8.318	
	3.572	9.749	4.341	1.681	3.174	0.233	0.016	0.903	0.893	0.654	1.801	
480	3.219	2.739	2.406	1.483	0.945	1.222	1.704	2.429	3.38	3.204	3.834	
	2.817	2.528	2.24	1.407	0.885	0.862	1.51	1.974	3.119	3.082	4.708	
	6.623	6.078	4.8	3.294	0.263	0.609	0.486	0.754	0.052	1.811	1.05	
500	2.991	2.884	1.85	0.89	0.796	1.02	0.995	1.945	2.41	2.735	2.932	
	3.023	2.519	1.927	0.861	0.564	0.731	0.715	1.678	2.121	2.638	3.216	
	8.923	7.994	5.258	2.644	0.602	0.369	0.952	1.128	0.726	2.281	1.561	
520	3.238	2.446	2.082	0.914	0.352	1.053	0.993	2.232	2.562	2.719	3.135	
	2.815	2.202	2.087	0.882	0.065	0.749	0.819	2.044	2.229	2.491	3.526	
	6.959	7.92	4.967	2.466	1.217	0.173	0.522	0.586	0.886	1.498	1.322	

Mean			RHS	LHS
Error	Raw	Adjusted		
Centre Depth	2.124	1.821	1.950	2.528
Average Depth	2.103	1.675	1.808	2.664
Minimum Depth	3.398	3.701	5.696	1.625
Total	2.542	2.399	3.151	2.272

Table 5.6: Preliminary Test 3 Error



The data shown in Table 5.6 clearly demonstrates the increased accuracy of this iteration of the image segmentation algorithm when compared to its predecessors. Similar trends regarding the distribution of error across the image plane, were noted but to a smaller extent than the previous tests. The most noteworthy observation unique to this series of tests is the minimum depth error calculations on the left-hand side of the image plane. In all other testing, including the overall and right-hand side of this same dataset, the minimum depth calculations produced the greatest error, however, the left-hand side's minimum depth error was substantially lower than its centre and average depth.

5.1.4 - Other Changes:

Through a small series of tests, the issue regarding the detection of objects on the edge of the image frame present in all of the preliminary tests was rectified. It was determined that this occurrence was a by-product of the border clearing functionality incorporated into the segmentation code. Due to the benefits the border clearing function provides to the accuracy of the system, during the sample obstacle testing the functionality will be manually deactivated if an applicable error occurs. In applicable situations in the sample object testing, this functionality will be deactivated temporarily to retrieve results from otherwise errored readings.



5.2 - Controlled Environment Testing:

Organised in a similar manner to the preliminary testing, the controlled environment portion of the sample testing involved the collection series of images captured in the test environment using the stereo camera system. Typically, this involved the collection of 15 image pairs for each sample obstacle, with the exceptions of the SD card which involved the collection of 2 sets of 15 pairs and the A4 paper which only involved 12 pairs. The following tables and figures will detail the raw results obtained from processing the image pairs through the object detection algorithm, the accuracy of the results compared to what would be expected and provide a sample image pair from the data collected.

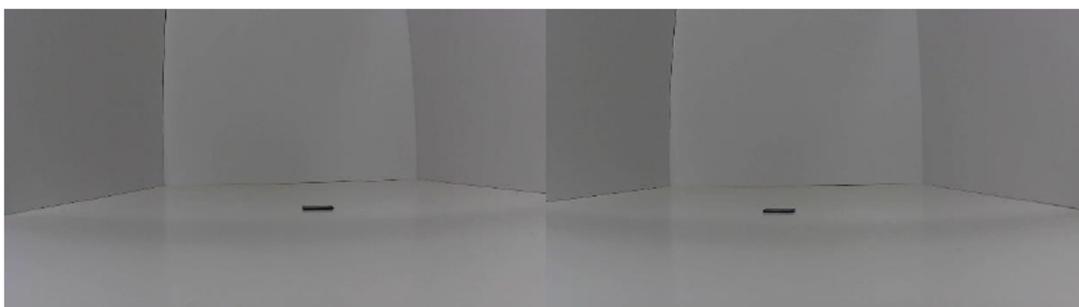


Figure 5.3: Sample Obstacle Testing SD Orientation 1 Sample Image Pair (0, 400)

		X Distance (mm)				
		-100	-50	0	50	100
Y-Position (mm)	400	416.9753	413.0305	408.2328	422.5832	436.6968
		431.0685	400.1065	417.05	434.4013	424.2486
		406.6796	389.9394	408.7971	420.6359	404.2831
	450	463.4816	457.5196	459.6206	466.8933	480.1105
		437.6063	476.6503	438.3321	490.4373	469.0826
		415.8019	448.3963	419.181	466.0728	451.1286
	500	514.6356	509.7675	506.7414	515.9748	532.3263
		539.8404	482.6096	530.7373	501.1712	572.9681
		505.2227	460.9948	502.517	480.0881	534.6948

Table 5.7: SD Sample Obstacle Orientation 1 Readings



		X Distance (mm)				
		-100	-50	0	50	100
Y-Position (mm)	400	1.648015	0.480068	0.914369	1.821661	3.003696
		1.676149	3.594113	1.225728	4.669239	0.067538
		1.365709	3.26794	2.199275	4.346924	1.946946
	450	1.949876	1.544692	0.515022	0.472468	1.567989
		7.423828	2.572112	5.122922	5.538987	0.764977
		9.799903	0.965825	6.848667	2.938261	2.136466
	500	1.349238	0.907424	1.02707	0.299199	2.041901
		3.482283	6.186588	3.659629	2.57844	9.832549
		0.917677	8.258606	0.5034	4.458897	4.862278
Mean Central Error		1.302846				
Mean Average Error		3.893005				
Mean Minimum Error		3.654452				
Average Error		2.950101				

Table 5.8: SD Sample Obstacle Orientation 1 Error

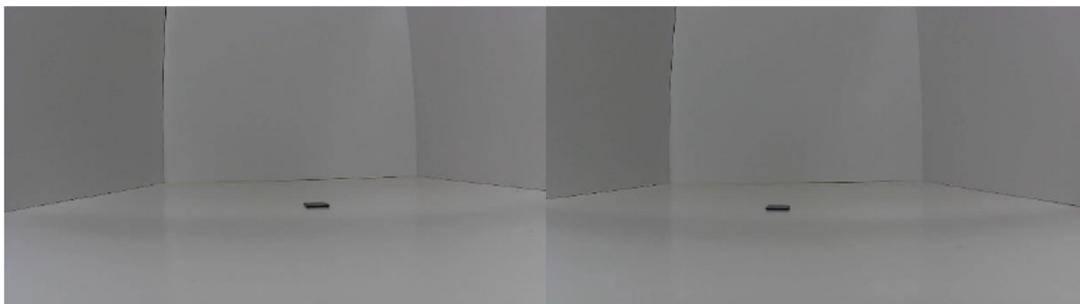


Figure 5.4: Sample Obstacle Testing SD Orientation 2 Sample Image Pair (0, 400)



		X Distance (mm)				
		-100	-50	0	50	100
Y-Position (mm)	400	417.3705	413.6315	416.6169	425.7892	439.9097
		402.0427	419.2373	416.9031	437.8664	451.4849
		389.9234	400.909	411.2048	424.1088	445.4081
	450	462.5089	458.2807	463.7278	460.9325	482.7153
		473.2284	440.6137	478.7958	451.3042	500.6306
		449.2011	427.8993	459.9655	443.4518	490.7775
	500	515.5142	513.2534	509.537	515.6903	528.9073
		496.9459	524.9113	495.3484	539.8963	527.6744
		481.6221	498.785	478.651	522.2983	511.6852

Table 5.9: SD Sample Obstacle Orientation 2 Readings

		X Distance (mm)				
		-100	-50	0	50	100
Y-Position (mm)	400	2.44944	1.279859	0.148293	1.621781	2.818569
		6.031954	0.05806	0.217091	4.504209	5.524
		5.429684	0.546717	2.8012	5.208445	8.027332
	450	2.958391	2.217749	0.487597	1.65194	1.281228
		0.709272	5.987314	2.74588	3.706308	5.040138
		2.5546	5.492855	2.214556	2.057882	6.464588
	500	1.919025	0.995998	1.252519	0.525932	0.629126
		5.451803	1.252752	4.002248	4.143284	0.394556
		5.546135	0.738075	4.2698	3.941247	0.349724

Mean Central Error	1.482497
Mean Average Error	3.317925
Mean Minimum Error	3.709523
Average Error	2.836648

Table 5.10: SD Sample Obstacle Orientation 2 Error

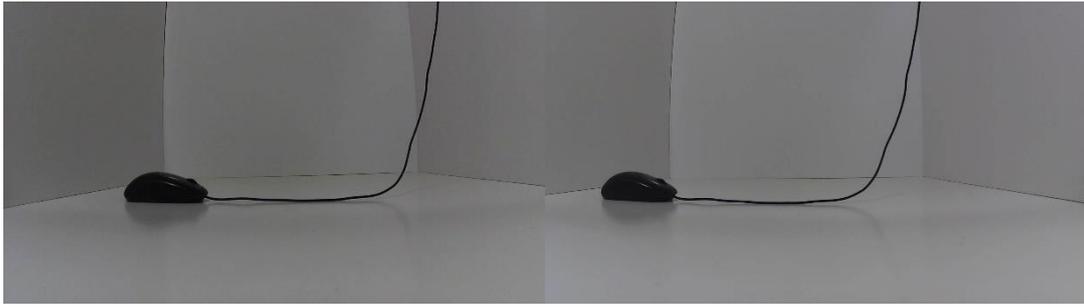


Figure 5.5: Sample Obstacle Testing Mouse Orientation 1 Sample Image Pair (400)

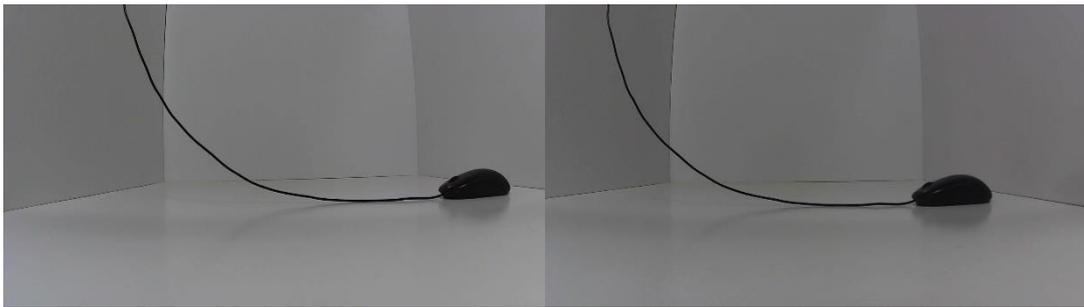


Figure 5.6: Sample Obstacle Testing Mouse Orientation 2 Sample Image Pair (400)

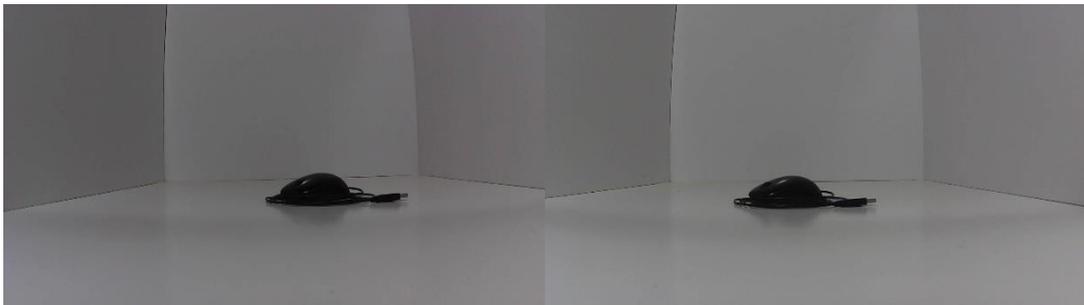


Figure 5.7: Sample Obstacle Testing Mouse Orientation 3 Sample Image Pair (400)

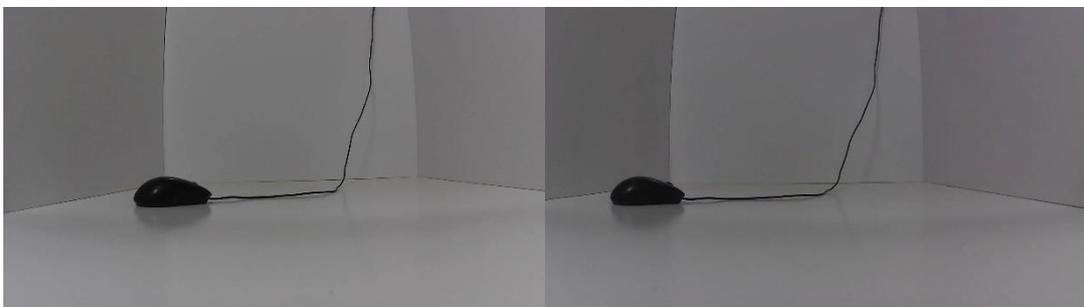


Figure 5.8: Sample Obstacle Testing Mouse Orientation 4 Sample Image Pair (400)

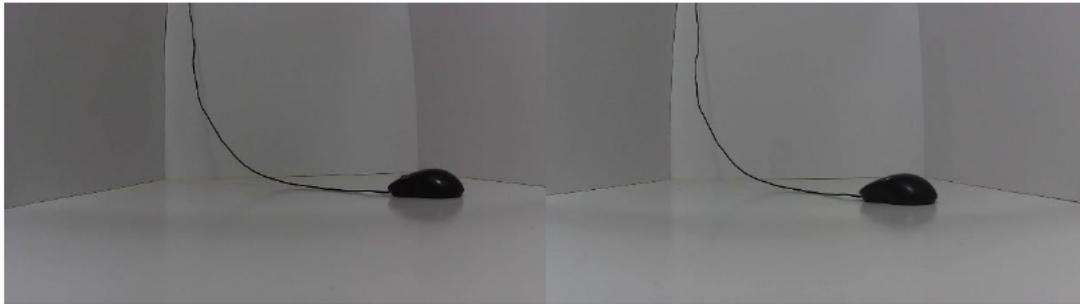


Figure 5.9: Sample Obstacle Testing Mouse Orientation 5 Sample Image Pair (400)

		Configuration				
		1	2	3	4	5
Y-Position (mm)	400	408.2805	421.5957	461.6447	618.328	480.1009
		422.5764	455.623	473.017	418.3154	570.4287
		315.4213	343.1936	450.7287	182.2414	493.8769
	450	468.3174	496.9707	497.4698	499.879	521.291
		445.8159	561.7673	484.2358	459.6413	566.3774
		280.6349	535.0712	420.0256	277.8041	520.5078
	500	508.4512	521.9417	557.9112	563.9052	587.5729
		526.5615	545.241	574.2324	488.5315	691.6588
		430.0974	446.8055	543.7086	257.0482	592.6515

Table 5.11: Mouse Sample Obstacle Readings

		Configuration				
		1	2	3	4	5
Y-Position (mm)	400	2.070125	5.398925	2.587711	54.582	20.02523
		5.6441	13.90575	5.114889	4.57885	42.60718
		21.14468	14.2016	12.68218	54.43965	23.46923
	450	4.070533	10.43793	0.50604	11.08422	15.84244
		0.9298	24.83718	3.15284	2.142511	25.86164
		37.63669	18.90471	6.660978	38.26576	15.6684
	500	1.69024	4.38834	1.4384	12.78104	17.51458
		5.3123	9.0482	4.405891	2.2937	38.33176
		13.98052	10.6389	8.74172	48.59036	18.5303

Mean Central Error	10.96118
Mean Average Error	12.54444
Mean Minimum Error	22.90371
Average Error	15.46978

Table 5.12: Mouse Sample Obstacle Error



Figure 5.10: Sample Obstacle Testing Paper 0 Degrees Sample Image Pair (400)

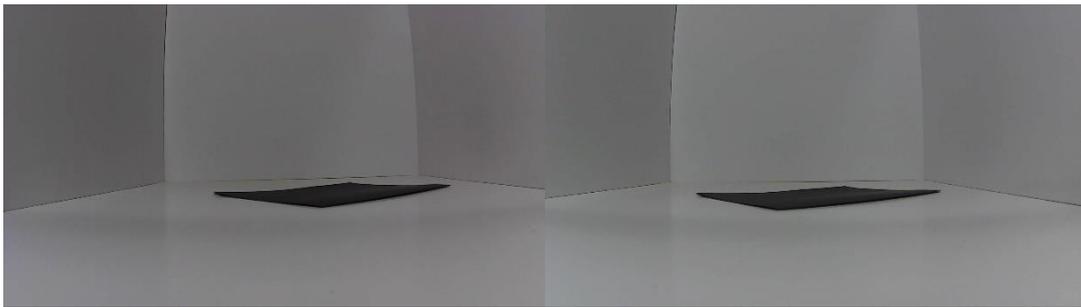


Figure 5.11: Sample Obstacle Testing Paper 45 Degrees Sample Image Pair (400)

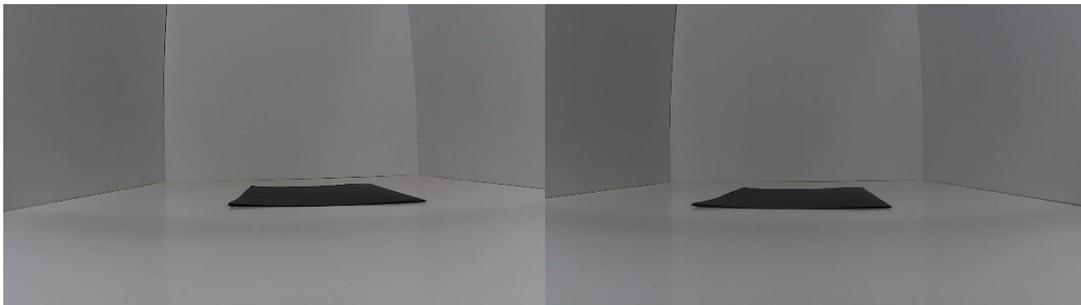


Figure 5.12: Sample Obstacle Testing Paper 90 Degrees Sample Image Pair (400)

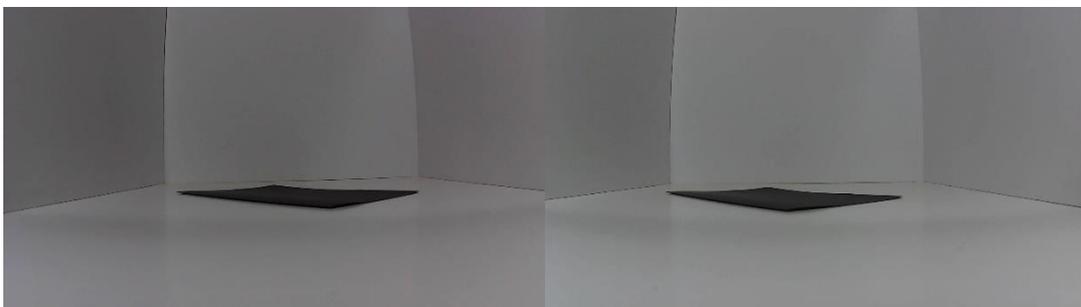


Figure 5.13: Sample Obstacle Testing Paper 135 Degrees Sample Image Pair (400)



		Orientation (Degrees)			
		0	45	90	135
Y-Position (mm)	350	411.3347	457.4654	431.4878	482.2434
		428.0001	536.8907	443.2705	506.0738
		364.9043	449.8065	361.6398	460.1027
	400	461.2407	502.4909	490.9935	535.6187
		575.5383	603.5693	503.2835	570.0308
		409.3112	504.8905	411.0609	520.7388
	450	518.264	558.8436	538.3474	587.9548
		533.6593	680.6574	544.4631	612.2578
		463.8144	563.4799	457.1192	565.6824

Table 5.13: A4 Paper Sample Obstacle Readings

		Orientation (Degrees)			
		0	45	90	135
Y-Position (mm)	350	9.596769	13.98957	13.26878	9.330929
		5.934044	0.943592	10.9004	4.850453
		4.258371	28.51614	3.325657	31.45791
	400	8.665208	13.64236	10.32082	7.949051
		13.96798	3.728879	8.076073	2.035018
		2.3278	26.22263	2.765225	30.1847
	450	6.619099	11.55747	9.900017	6.950332
		3.845171	7.720773	8.876469	3.104141
		3.069867	25.21776	1.582044	25.7072

	0	45	90	135
Mean Central Error	8.293692	13.06313	11.16321	8.076771
Mean Average Error	7.915732	4.131081	9.284315	3.329871
Mean Minimum Error	3.218679	26.65217	2.557642	29.1166
Average Error	10.56691			

Table 5.14: A4 Paper Sample Obstacle Error

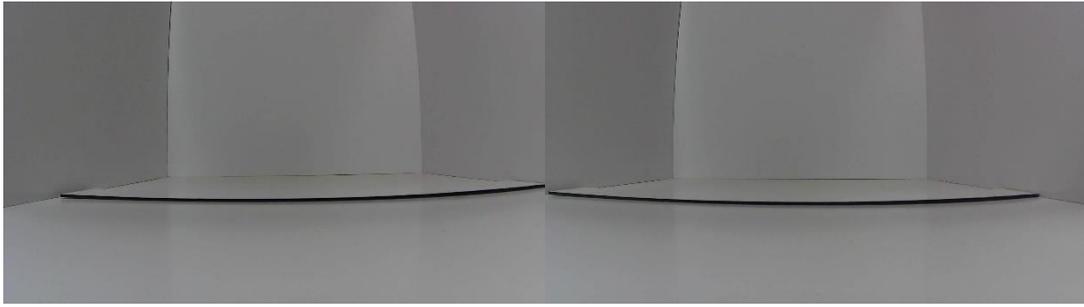


Figure 5.14: Sample Obstacle Testing Barrier 3mm Degrees Sample Image Pair (400)

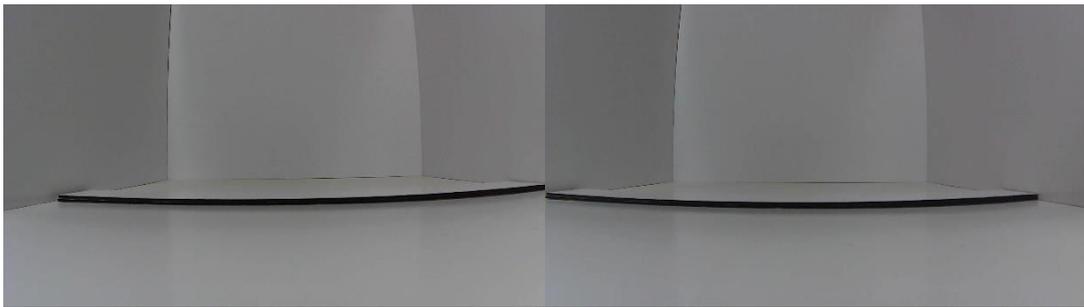


Figure 5.15: Sample Obstacle Testing Barrier 6mm Degrees Sample Image Pair (400)



Figure 5.16: Sample Obstacle Testing Barrier 9mm Degrees Sample Image Pair (400)



Figure 5.17: Sample Obstacle Testing Barrier 12mm Degrees Sample Image Pair (400)



Figure 5.18: Sample Obstacle Testing Barrier 15mm Degrees Sample Image Pair (400)

		Barrier Height (mm)				
		3	6	9	12	15
Y Distance (mm)	400	ERROR	ERROR	ERROR	ERROR	ERROR
		ERROR	ERROR	ERROR	ERROR	ERROR
		ERROR	ERROR	ERROR	ERROR	ERROR
	450	745.4261	810.3095	785.8506	731.8588	755.525
		765.1965	781.7664	712.4167	686.3206	763.6617
		661.6051	728.2076	679.3674	650.8171	724.638
	500	528.9803	517.4936	505.0974	521.5045	511.0704
		585.179	540.3804	523.8737	540.8659	523.7752
		540.728	504.2446	490.2138	503.9642	491.9163

Table 5.15: Barrier Sample Obstacle Readings

		Barrier Height (mm)				
		3	6	9	12	15
Y Distance (mm)	400	ERROR	ERROR	ERROR	ERROR	ERROR
		ERROR	ERROR	ERROR	ERROR	ERROR
		ERROR	ERROR	ERROR	ERROR	ERROR
	450	65.65024	80.06878	74.63347	62.63529	67.89444
		70.04367	73.72587	58.31482	52.51569	69.7026
		47.02336	61.82391	50.97053	44.62602	61.03067
	500	5.79606	3.49872	1.01948	4.3009	2.21408
		17.0358	8.07608	4.77474	8.17318	4.75504
		8.1456	0.84892	1.95724	0.79284	1.61674

Mean Central Error	36.77115
Mean Average Error	36.71175
Mean Minimum Error	27.88358
Average Error	33.78883

Table 5.16: Barrier Sample Obstacle Error



In the sample obstacle testing, the error values for the readings captured using the A4 paper and SD card as sample object were calculated in a slightly different manner than the preliminary tests. This was due to the large disparity between the thickness of the sample objects and their other dimensions, which when viewed by the sample system would be more prominent. To remedy this, instead of being compared to the expected position of the centre of its foremost edge, as with the object used in the preliminary testing, the central and average depth error values were calculated with respect to the centre point of the object's top face instead. Specifically, the expected depths for the SD card readings were offset by 12mm for the first orientation and 16mm for the second whilst the paper readings were offset by 105mm for 0 degrees, 181.872mm for both 45 and 135 degrees and 147.5mm for 90 degrees.

The readings obtained during the testing of the SD card as the sample object are largely in line with those collected during the preliminary testing. The average error in the sample system's readings for both of the tested orientations was below 4%, with each orientation demonstrating a few instances where the error noticeably exceeds the norm. The majority of these outliers are contained within the average depth readings, with the notable exception of the first orientation's '-100 450' minimum depth calculation which approached 10% error.

Through the comparing the error of data collected using the SD card and A4 paper as test objects, it becomes apparent that the system becomes less reliable when identifying larger low-lying obstacles. Despite the fact that the thickness of the SD card is substantially more visible in the captured images and should therefore significantly impact the accuracy of the average and central depth estimation, the average error of the A4 paper's central and average depth readings is more than double the SD card's. This phenomenon's cause can be isolated to the size of the object's top face and not the overall number of pixels being considered in the analysis due to the fact that the preliminary tests, despite analysing a more pixels the SD card tests, retained a similar level of accuracy.



The 45 and 135 degree orientations of the paper, as well as tests conducted using the corded mouse as a sample object highlight a major shortcoming of the sample system. Although the other tests performed with the sample system have demonstrated it's capabilities in recognising simpler shapes with edges in parallel with the stereo camera, the more complex shapes and angles of these tests have resulted in large increases in the system's error. This impact is most clearly visible in the minimum depth calculations which, despite the fact that all previous tests demonstrated an average error below 4%, these tests involving the objects and orientations in question present errors in excess of 20%. Despite this however, the results obtained using the average depth estimation produced the most consistently accurate readings for the situations in question, with the notable exception of situations where the majority of the object is located towards the right edge of the image plane as with mouse configuration 2.

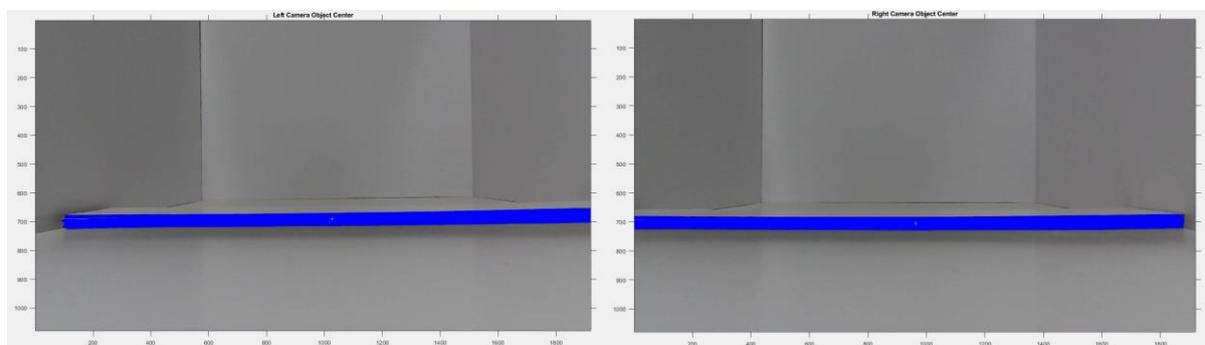


Figure 5.19: 12mm Barrier at 400mm Segmentation

The most obvious characteristic of the barrier testing results is the large levels of inaccuracy present in tests conducted at y-distances less than 500mm. As can be seen in Tables 5.15 and 5.16, despite the 500mm readings demonstrating accuracy more indicative of that seen in other testing, all but two of the 450mm readings possessed errors in excess of 50% with all of the 400mm readings falling into the territory of being considered an unusable reading. By observing the regions of the collected images highlighted by the sample system, it became clear that the fact that this issue is likely as a result of at least one of the edges of the identified region being on the border our outside of the image plane. Due to the way the system operates, this would mean that some of the identified pixels would be outside the regions where it would be possible to perform stereo calculations. This fact offset the comparison of pixels between the images in the pair, leading to the system estimating the feature to be much further away than in reality. As such, only when all of the identified object was visible in both of the images in the par, would the system return to it's previous levels of accuracy. Within the image pair provided in Figure 5.19, it can be seen that each edge of the barrier is only visible in one of the images resulting in the depth estimation readings of 2492.8mm for central depth, 1508.6mm for average depth and 1318.8mm for minimum depth, all with errors in excess of 100%.

5.3 - Alternate Environment Testing:

The tests performed outside of the prepared control environment validated the inherent flaws with the image segmentation method the sample system utilises. Although not unexpected, the sample system's colour thresholding encountered issues resulting from the natural shadows in the environment. As can be seen in Figure 5.20, no suitable threshold value could be determined in order to single out the sample object being used in the test, the system would either identify a shadow within the image or a combination of the object and the shadows it intersects with.

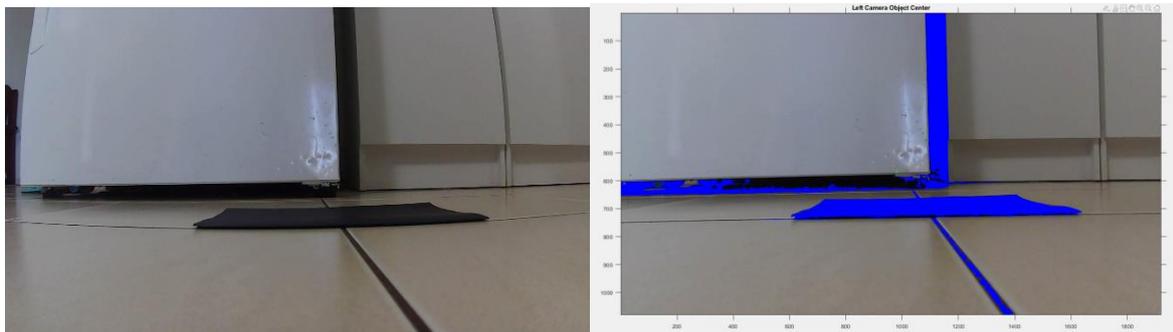


Figure 5.20: Alternate Environment segmentation sample (Raw – Left, Segmented – Right)

Due to this shortcoming, the depth values the sample system determined for each image pair were inherently unreliable and as such, the error in their results were not determined. This issue however could potentially be rectified through the employ of an additional light source aimed in the same direction as the stereo camera system. As with most indoor environments, the three examples chosen for the capture of the test images were lit by light sources mounted within the roof. If the aforementioned additional light source were employed, significantly less shadow would be visible in the stereo camera's image frame, therefore the colour thresholding would be more effective in identifying only the sample obstacle.



Chapter 6: Conclusions and Further Work

6.1 - Conclusion:

In conclusion, this project has fulfilled its core aim and completed each of the objectives outlined in the first chapter of this report. This has involved the creation and subsequent analysis of a stereo vision system as a sample object detection system, a decision born from research into the capabilities of existing systems. The system was prepared using MATLAB, making use of the Image Processing and Computer Vision toolbox to both calibrate the stereo camera hardware as well as perform colour thresholding segmentation on images during the analysis. Once prepared, this system was run through three phases of testing, each designed to allow for the observation of how the accuracy of the system's changes in response to the adjustment of different factors. Specifically, this process first involved a distinct obstacle within a controlled environment, that same controlled environment was then used to analyse example small-scale low-lying obstacles which were then tested in alternate indoor environments with more uncertain backgrounds and lightings. From this data, a general accuracy of the system was established, and factors were identified which attributed to decreases in this accuracy such as the size of flat objects, their orientation relative to the stereo camera and the position of obstacles within the camera's field of vision.

6.2 - Further Work:

Beyond the research has been conducted in the pursuit of this project's aim and objectives, further work could be explored to expand upon and understand the capabilities of the project's object detection system. These potential avenues fall under three categories, the remedy of notable shortcomings of the system, the expansion of the testing on the existing system and the inclusion of supplementary functionality to the system, which in turn would require further testing using a research methodology tailored to the enhanced system.



Of the tests conducted, two main factors were identified which decreased the system's accuracy to the point of making the distance estimations unusable in a practical setting. Namely these issues are the identified problems regarding the perception of objects with features outside of the stereo range, and the separation of object from shadows in the environment as seen in the alternate environment testing. Through additional work it may be possible to find alternate ways of refining the image segmentation produced, isolating it to features specifically visible by both cameras and further fine tuning it's ability to distinguish shadow and dark colour objects. The more feasible solution to this issue would be the implementation of a deep learning neural network to perform image segmentation, a process which although more time consuming, would be more likely to allow the system to produce more accurate depth estimations.

Further testing of the current system would mostly involve the variation of variables in the testing environment which were controlled during initial methodology. Some examples of such factors include the lighting conditions of the environment, the angle and elevation at which the camera is positioned as well as further variation in environmental conditions. The analysis of the results of such additional testing would expand upon the understanding of the system's capabilities and limitations. Another means by which the current methodology could be expanded is through the variation of individual components of the object detection system. To some degree, this was completed as part of preliminary testing phase, but other aspects of the design chosen outlined in the methodology preparation section of chapter 4 could be altered and subsequently tested. First amongst these would be the positions of the twin cameras which constitute the stereo vision hardware. As stereo vision theory dictates, the distance between each lens of a stereo camera determines the area in which depth estimation can be performed. Testing could be conducted comparing the estimated obstacle positions with the cameras at different distances apart, observing what effect, if any, it has on the accuracy of the system's readings. Similar tests could be performed to determine the impact other aspects of the system's design, such as the specifications of the camera, have on the accuracy of the results when compared to the initial design.



The second avenue for building upon this project's methodology revolves around the expansion of the capabilities of the project's system to more closely approximate how similar systems would operate in practical scenarios. The core problem which this project is built upon revolves around the navigation of a mobile robot throughout an indoor environment. Whilst the methodology performed serves as a conceptual test of the problem, in order to better approximate the practical conditions, improvements would need to be made to the system. Chief amongst these improvements would be the adaptation of the system to operate in conjunction with a real-time video feed provided by a mobile robot. Testing of such a system would provide insight as to what extent the motion of the camera can impact the system's readings. To complement this additional image segmentation methods may also be incorporated into the system, such as feature tracking and instance segmentation in order to further adapt the system for practical applications. Beyond these improvements, the methodology could be expanded upon by further study outside of the purview of this project's objectives. This would primarily take the form of interpreting the readings determined by the object detection system and controlling a robot accordingly. This would require the interpretation of the sample system's data in real-time to produce navigational commands

In contrast to the previously discussed options for expanding upon the research conducted during the completion of this project, minor quality of life improvements could be made on the project's sample system to make it easier to utilise. Foremost amongst these would be the implementation of code to assist in the determination of the colour threshold values applicable for each image, either automatically or with small amounts of user input. A similar quality of life improvement would be the automation of the 'clear border' function to toggle on or off in response to whether an object would cross over with the image's border. This would vastly reduce the amount of adjustment needed to ensure the sample system produced the most useable results.



Chapter 7: List of References

1. Garcia-Garcia, B, Bouwmans, T, Silva, A 2020, 'Background subtraction in real applications: Challenges, current models and future directions', *Computer Science Review*, vol. 35, viewed 26 May 2021, <<https://www.sciencedirect.com/science/article/pii/S1574013718303101>>
2. Ghenescu, V, Mihaescu, R, Carata, S, Ghenescu, M, Barnoviciu, E, Chindea, M 2018, 'Face Detection and Recognition Based on General Purpose DNN Object Detector', *2018 International Symposium on Electronics and Telecommunications (ISETC)*, viewed 26 May 2021, <<https://ieeexplore-ieee-org.ezproxy.usq.edu.au/document/8583861>>
3. Inoue, K, Kaizu, Y, Igarashi, S, Imou, K 2019, 'The development of autonomous navigation and obstacle avoidance for a robotic mower using machine vision technique', *IFAC-PapersOnLine*, vol. 52, no. 30, pp. 173-177, viewed 26 May 2021, <<https://www.sciencedirect.com/science/article/pii/S240589631932436X>>
4. Kim, J & Do, Y 2012, 'Moving Obstacle Avoidance of a Mobile Robot Using a Single Camera', *Procedia Engineering*, vol. 41, pp. 911-916, viewed 26 May 2021, <<https://www.sciencedirect.com/science/article/pii/S1877705812026628>>
5. Kulkarni, A, Potdar, A, Hegde, S, Baligar, V 2019, 'RADAR based Object Detection using Ultrasonic Sensor', *2019 1st International Conference on Advances in Information Technology (ICAIT)*, viewed 26 May 2021, <<https://ieeexplore-ieee-org.ezproxy.usq.edu.au/document/8987259>>
6. MathWorks 2021, *Image Segmentation*, MathWorks, viewed 26 May 2021, <<https://au.mathworks.com/discovery/image-segmentation.html>>
7. Nakajima, K, Premachandra, C, Kato, K 2017, '3D environment mapping and self-position estimation by a small flying robot mounted with a movable ultrasonic range sensor', *Journal of Electrical Systems and Information Technology*, vol. 4, no. 2, pp. 289-298, viewed 26 May 2021, <<https://www.sciencedirect.com/science/article/pii/S2314717217300120>>
8. Pan, H, Li, S, Liu, Q, Xu, Y, Ji, P, Kang, X 2020, 'Trajectory Tracking Method of Crawler Robot Based on Improved LOAM', *2020 12th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, viewed 26 May 2021, <<https://ieeexplore-ieee-org.ezproxy.usq.edu.au/document/9204280>>
9. Peng, Y, Qu, D, Zhong, Y, Xie, S, Luo, J, Gu, J 2015, 'The Obstacle Detection and Obstacle Avoidance Algorithm Based on 2-D lidar', *2015 IEEE International Conference on Information and Automation*, viewed 26 May 2021, <<https://ieeexplore-ieee-org.ezproxy.usq.edu.au/document/7279550>>
10. Redmon, J, Divvala, S, Girshick, R, Farhadi, A 2016, 'You Only Look Once: Unified, Real-Time Object Detection', *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, viewed 26 May 2021, <<https://ieeexplore-ieee-org.ezproxy.usq.edu.au/document/7780460>>
11. Sarda, A, Dixit, S, Bhan, A 2021, 'Object Detection for Autonomous Driving using YOLO [You Only Look Once] algorithm', *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, viewed 26 May 2021, <<https://ieeexplore-ieee-org.ezproxy.usq.edu.au/document/9388577>>
12. Sharma, K, Sahoo, S, Manivannan, P 2018, 'A Hybrid Vision System for Dynamic Obstacle Detection', *Procedia Computer Science*, vol. 133, pp. 153-160, viewed 26 May 2021, <<https://www.sciencedirect.com/science/article/pii/S1877050918309633>>
13. SOLAK, S & BOLAT, E 2018, 'A new hybrid stereovision-based distance-estimation approach for mobile robot platforms', *Computers & Electrical Engineering*, vol. 67, pp. 672-689, viewed 26 May 2021, <<https://www.sciencedirect.com/science/article/pii/S0045790616311405>>



14. Wang, L, Chen, X, Hu, L, Li, H 2020, 'Overview of Image Semantic Segmentation Technology', *2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, viewed 7 September 2021, <<https://ieeexplore-ieee.org.ezproxy.usq.edu.au/document/9338770>>
15. Ye, L, Liu, Z, Wang, Y 2017, 'Depth-aware object instance segmentation', *2017 IEEE Conference on Image Processing (ICIP)*, viewed 7 September 2021, <<https://ieeexplore-ieee.org.ezproxy.usq.edu.au/document/8296296>>
16. Young, J & Simic, M 2015, 'LIDAR and Monocular Based Overhanging Obstacle Detection', *Procedia Computer Science*, vol. 60, pp. 1423-1432, viewed 26 May 2021, <<https://www.sciencedirect.com/science/article/pii/S1877050915023455>>
17. Zhang, J, Singh, S 2014, 'LOAM: Lidar Odometry and Mapping in Real-time', *Robotics: Science and Systems*, vol. 2, pp. 9-17, viewed 26 May 2021, <https://www.ri.cmu.edu/pub_files/2014/7/Ji_LidarMapping_RSS2014_v8.pdf>
18. Zhang, Z, Cao, Y, Ding, M, Zhuang, L, Tao, J 2020, 'Monocular vision based obstacle avoidance trajectory planning for Unmanned Aerial Vehicle', *Aerospace Science and Technology*, vol. 106, viewed 26 May 2021, <<https://www.sciencedirect.com/science/article/pii/S1270963820308816>>



Appendix A: ENG4111/4112 Research Project

Project Specification

For: Nicholas Jennings

Title: Identifying Abnormalities in Indoor Environments using Obstacle Detection Programming

Major: Mechatronic engineering

Supervisors: Tobias Low

Enrolment: ENG4111 – ONC S1, 2021
 ENG4112 – ONC S2, 2021

Project Aim: To test the effectiveness of obstacle detection programming in locating terrain abnormalities and small obstacles in an indoor environment.

Programme: Version 3, 6th May 2021

1. Research background information regarding obstacle detection/avoidance programming, focussing on systems which use vision as the primary sensor.
2. Determine the means by which researched methods identify objects in images and how they determine their location relative to the robot.
3. Develop an obstacle detection code (in MATLAB) based upon methods identified, capable of detecting obstacles and terrain abnormalities in still images.
4. Test the capabilities of the obstacle detection code using a series of controlled environments.
5. Perform additional tests in other indoor environments to determine the program's effectiveness outside of the test scenarios.
6. Evaluate the ability for the algorithms to identify different types of abnormalities.

If time and resources permit:

7. Adapt obstacle detection code to work from a real-time data feed provided by the camera(s), creating a real-time obstacle detection code.
8. Incorporate motor control into the real-time obstacle detection code to control a small scale, land-bound robot, to create a real-time obstacle avoidance program.
9. Determine the effectiveness of the real-time obstacle avoidance program in the test scenarios.



Appendix B: MATLAB Code:

Appendix B.1: Main MATLAB Code

```

clear
close all
clc
cd 'C:\Users\nicje\OneDrive\Documents\Research Project\MATLAB Files';

%Parameter and Input Loading
load('stereoParamsV1.mat'); %Loading Parameters from Stereo Calibration
load('cameraParamsL.mat'); %Loading Left Camera Parameters
load('cameraParamsR.mat'); %Loading Right Camera Parameters

cd 'C:\Users\nicje\OneDrive\Documents\Research Project\Controlled
Environment Testing';
inputImage1= imread('B 400 12 L.jpg'); %Left Camera Input
inputImage2= imread('B 400 12 R.jpg'); %Right Camera Input
cd 'C:\Users\nicje\OneDrive\Documents\Research Project\MATLAB Files';

%Stereo Anaglyph Generation
[rectifiedL,rectifiedR] =
rectifyStereoImages(inputImage1,inputImage2,stereoParamsV1,'OutputView','va
lid');
figure;
imshow(stereoAnaglyph(rectifiedL,rectifiedR));
title('Rectified Input Images');

%Disparity Map Generation
dispRange = [0 48];
dispMap =
disparitySGM(rgb2gray(rectifiedL),rgb2gray(rectifiedR),'DisparityRange',dis
pRange,'UniquenessThreshold',20);
figure;
imshow(dispMap,dispRange);
title('Disparity Map');
colormap jet;
colorbar;

%Image Segmentation
UnDistorted_L = undistortImage(inputImage1,cameraParamsL);
UnDistorted_R = undistortImage(inputImage2,cameraParamsR);
CThresh_L = SegmentationCode(UnDistorted_L);
CThresh_R = SegmentationCode(UnDistorted_R);

LargestOBJ_L = bwareafilt(CThresh_L,1);
LargestOBJ_R = bwareafilt(CThresh_R,1);

%Object Centre Calculation

[L_obj_y,L_obj_x]=find(LargestOBJ_L);
[R_obj_y,R_obj_x]=find(LargestOBJ_R);

MatchPoint_L = [mean(L_obj_x), mean(L_obj_y)];
MatchPoint_R = [mean(R_obj_x), mean(R_obj_y)];

```



```
%Segmentation Demonstration
figure;
imshow(imoverlay(UnDistorted_L,LargestOBJ_L,'blue'));
axis on;
hold on;
plot(mean(L_obj_x),mean(L_obj_y),'g+')
title('Left Camera Object Center');

figure;
imshow(imoverlay(UnDistorted_R,LargestOBJ_R,'blue'));
axis on;
hold on;
plot(mean(R_obj_x),mean(R_obj_y),'g+')
title('Right Camera Object Center');

%Depth Calculations
length_Max = [length(L_obj_x), length(L_obj_y), length(R_obj_x),
length(R_obj_y)];
for count = 1:min(length_Max)
    Depth_Full(count)=
norm(triangulate([L_obj_x(count),L_obj_y(count)],[R_obj_x(count),R_obj_y(count)],stereoParamsV1));
end
tri = triangulate(MatchPoint_L,MatchPoint_R,stereoParamsV1);

C_Depth_mm = norm(tri);
A_Depth_mm = mean(Depth_Full);
Min_Depth_mm = min(Depth_Full);

disp('Central Depth' );
disp(C_Depth_mm);
disp('Average Depth');
disp(A_Depth_mm);
disp('Minimum Depth' );
disp(Min_Depth_mm);
```



Appendix B.2: MATLAB Segmentation Code

```
function [BW,maskedRGBImage] = SegmentationCode (RGB)
%createMask Threshold RGB image using auto-generated code from
colorThresholder app.
% [BW,MASKEDRGBIMAGE] = createMask(RGB) thresholds image RGB using
% auto-generated code from the colorThresholder app. The colorspace and
% range for each channel of the colorspace were set within the app. The
% segmentation mask is returned in BW, and a composite of the mask and
% original RGB images is returned in maskedRGBImage.

% Auto-generated by colorThresholder app on 03-May-2021
%-----

% Convert RGB image to chosen color space
I = RGB;

% Define thresholds for channel 1 based on histogram settings
channel1Min = 0.000;
channel1Max = 20.000;

% Define thresholds for channel 2 based on histogram settings
channel2Min = 0.000;
channel2Max = 20.000;

% Define thresholds for channel 3 based on histogram settings
channel3Min = 0.000;
channel3Max = 20.000;

% Create mask based on chosen histogram thresholds
sliderBW = (I(:,:,1) >= channel1Min ) & (I(:,:,1) <= channel1Max) & ...
    (I(:,:,2) >= channel2Min ) & (I(:,:,2) <= channel2Max) & ...
    (I(:,:,3) >= channel3Min ) & (I(:,:,3) <= channel3Max);
BW = sliderBW;

% Fill holes
BW = imfill(BW, 'holes');

% Clear borders
BW = imclearborder(BW);

% Initialize output masked image based on input image.
maskedRGBImage = RGB;

% Set background pixels where BW is false to zero.
maskedRGBImage(repmat(~BW,[1 1 3])) = 0;

end
```