

A coordinated UAV deployment based on
stereovision reconnaissance for low risk water
assessment

Temitope Sam-Odusina

A thesis submitted in partial fulfilment of the requirements of Liverpool John
Moores University for the degree of Doctor of Philosophy

Liverpool John Moores University
September 2018

Acknowledgements

This thesis would not be possible without the great support from my supervisor Dr Frederic Bezombes for his guidance and advices that were necessary for me to proceed in my research and complete my thesis.

Special thanks to my supervisory team Dr. Patryk Kot and Prof Andy Shaw for their support and helpful suggestions.

Members of the Natural Sciences, Psychology School Dr Owen McAree and Prof Serge Wich for providing assistance and helpful suggestions.

Finally, I would like to thank my family for their love, motivation and constant support. Without their constant assurance and assistance, completion of this project would not have been possible, and as a sign of my love and appreciation, I dedicate this work to them.

Temitope Sam-Odusina

Abstract

Biologists and management authorities such as the World Health Organisation require monitoring of water pollution for adequate management of aquatic ecosystems. Current water sampling techniques based on human samplers are time consuming, slow and restrictive. This thesis takes advantage of the recent affordability and higher flexibility of Unmanned Aerial Vehicles (UAVs) to provide innovative solutions to the problem.

The proposed solution involves having one UAV, “the leader”, equipped with sensors that are capable of accurately estimating the wave height in an aquatic environment, if the region identified by the leader is characterised as having a low wave height, the area is deemed suitable for landing. A second UAV, “the follower UAV”, equipped with a payload such as an Autonomous Underwater Vehicle (AUV) can proceed to the location identified by the leader, land and deploy the AUV into the water body for the purposes of water sampling.

The thesis acknowledges there are two main challenges to overcome in order to develop the proposed framework. Firstly, developing a sensor to accurately measure the height of a wave and secondly, achieving cooperative control of two UAVs. Two identical cameras utilising a stereovision approach were developed for capturing three-dimensional information of the wave distribution in a non-invasive manner. As with most innovations, laboratory based testing was necessary before a full-scale implementation can be attempted. Preliminary results indicate that provided a suitable stereo matching algorithm is applied, one can generate a dense 3D reconstruction of the surface to allow estimation of the wave height parameters. Stereo measurements show good agreement with the results obtained from a wave probe in both the time and frequency domain. The mean absolute error for the average wave height and the significant wave height is less than 1cm from the acquired time series data set.

A formation-flying algorithm was developed to allow cooperative control between two UAVs. Results show that the follower was able to successfully track the leader’s trajectory and in addition maintain the given separation distance from the leader to within 1m tolerance through the course of the experiments despite windy conditions, low sampling rate and poor accuracy of the GPS sensors.

In the closing section of the thesis, near real-time dense 3D reconstruction and wave height estimation from the reconstructed 3D points is demonstrated for an aquatic body using the leader UAV. Results show that for a pair of images taken at a resolution of 320 by 240 pixels up to 21,000 3D points can be generated to provide a dense 3D reconstruction of the water surface within the field of view of the cameras.

Contents

Chapter 1: Introduction	1
1.1 Background	1
1.2 State of the Art.....	3
1.3 Scope of Research.....	7
1.4 Novelty of Research	9
1.5 Aim and Objectives	9
1.6 Thesis Structure	10
Chapter 2: Current 3D reconstruction techniques	13
2.1 Introduction	13
2.2 3D reconstruction techniques.....	13
2.2.1 Active techniques.....	13
2.2.2 Passive techniques	17
2.2.3 Choice of technique for 3D reconstruction of waves	22
2.3 Summary	24
Chapter 3: Camera Modelling and Calibration	25
3.1 Introduction	25
3.2 Camera Model.....	25
3.2.1 Rigid body transformation	27
3.2.2 Perspective Projection	28
3.2.3 Lens Distortion	30
3.2.4 Computer Image Frame Coordinates.....	32
3.3 Camera Parameters	33
3.3.1 Intrinsic parameters.....	34
3.3.2 Extrinsic parameters	35
3.4 Camera Calibration	36
3.4.1 Experimental Results.....	38
3.5 Summary	42
Chapter 4: Stereovision.....	43
4.1 Introduction	43
4.2 Standard stereo geometry	43
4.3 Epipolar geometry.....	45
4.4 Computing Fundamental and Essential Matrix.....	48
4.4.1 Fundamental matrix.....	48
4.4.2 Essential matrix.....	50

4.5	Image Rectification	51
4.6	Stereo correspondence problem	53
4.6.1	Feature-based matching	54
4.6.2	Correlation based matching.....	56
4.7	Triangulation	66
4.8	Experimental Results	68
4.8.1	3D reconstruction using feature based methods	68
4.8.2	3D reconstruction using correlation-based method.....	78
4.9	Summary	82
Chapter 5:	Water surface reconstruction and analysis of laboratory Generated water waves....	83
5.1	Introduction	83
5.2	3D Reconstruction of water waves	83
5.2.1	3D reconstruction of still water surface.....	83
5.2.2	Wave height estimation.....	92
5.3	Conclusions	113
Chapter 6:	Coordinated control of two UAVs.....	114
6.1	Introduction	114
6.2	Related Work	114
6.3	Flight Controllers.....	116
6.3.1	Overall Evaluation and Selection	119
6.4	Introduction to Ardupilot flight stack	120
6.4.1	Ardupilot Code Structure	121
6.4.2	MAVLink protocol	124
6.4.3	Software-In-the-Loop.....	127
6.5	Ground Control Station (GCS).....	128
6.5.1	Mission Planner.....	130
6.5.2	MAVProxy	134
6.6	Dronekit	137
6.7	Leader-follower formation.....	141
6.7.1	Motion based Constraints for leader and follower UAV.....	141
6.7.2	Simulation Results.....	144
6.7.3	Hardware Overview	146
6.7.4	Communication between leader and follower	148
6.7.5	Flight Modes.....	154
6.8	Flight Test Results and Discussion	156
6.9	Conclusions	168

Chapter 7:	Case Study: 3D reconstruction and wave height measurement over an aquatic body
	170
7.1	Introduction
7.2	Experimental Set-Up
7.3	Data collection and processing
7.4	Conclusion.....
Chapter 8:	Discussions, Conclusions and Future Work.....
8.1	Discussions and Conclusions.....
8.2	Future Work.....
8.2.1	Variable baseline Stereo
8.2.2	Validation in various aquatic environments
8.2.3	Communication between multiple UAVs.....
Chapter 9:	References

List of Figures

Figure 1-1: Human personnel collecting water sample (Grasshoff et al., 2009)	3
Figure 1-2: El Zacaton sinkhole (Gary and Sharp, 2006)	4
Figure 1-3: UAV based water sampling (Ore et al., 2015)	5
Figure 1-4:AUV Deployment (Blidberg, 2001)	6
Figure 2-1: Pulse based laser-scanning technique.....	14
Figure 2-2: (a) phased based laser scanner (Jang et al., 2013) (b) Phase shift measurement.....	16
Figure 2-3: Planar surface textured with circular texture elements (Rosenholtz and Malik, 1997)....	18
Figure 2-4: Structure from motion technique (Westoby et al., 2012).....	19
Figure 2-5: Stereo human perception (Cooper, 1995).....	21
Figure 3-1:Geometric relation between 3D world coordinate and computer image coordinates – adapted from (Quintana, 2003)	26
Figure 3-2: The pinhole mode with image plane located at focal length f away from camera original (Quintana, 2003)	28
Figure 3-3: Effects of radial distortion. Solid lines: no distortion: dashed lines: radial distortion (b: barrel distortion, p: pincushion distortion)(Quintana, 2003)	31
Figure 3-4: Four step transformation from 3D world coordinates to computer image coordinates ...	33
Figure 3-5: Logitech C930 webcam	38
Figure 3-6: Snapshots of checkerboard pattern taken at various orientations.....	39
Figure 3-7: Average reprojection error for various number of checkerboard images	40
Figure 4-1: Schematic of two cameras in a standard stereo-geometry configuration.....	44
Figure 4-2: Finding a corresponding match in a stereovision setup	46
Figure 4-3: the epipolar constraint	47
Figure 4-4: Stereo pair (top) and rectified pictures (bottom). The right pictures show examples of epipolar lines (Fusiello et al., 2000)	52
Figure 4-5: Left views and ground truth disparity maps of various models (Scharstein and Szeliski, 2012)	53
Figure 4-6: Correlation between left and right images at pixel P	57
Figure 4-7: Example of rank transform using a 3*3 window	59
Figure 4-8: Example of census transform using a 3*3 window	60
Figure 4-9: The nine correlations windows, the pixel for which disparity is computed is highlighted	61
Figure 4-10: Left-right consistency check	63
Figure 4-11: Multi directional search pattern for SGM where eight optimization paths from different directions meet at every pixel (p) (Hirschmuller, 2008)	66
Figure 4-12:3D reconstruction pipeline for feature based matching	69
Figure 4-13: Rexroth beam used for construction of Rig	69
Figure 4-14: Stereo rig for 3D reconstruction.....	70
Figure 4-15: left and right images captured by stereo camera	71
Figure 4-16: keypoint detection - SIFT (left), SURF (middle) and ORB (right).....	71
Figure 4-17: Number of keypoints detected on left and right images	72
Figure 4-18: Correspondence matching before RANSAC filtering	74
Figure 4-19: Correspondence matching after RANSAC filtering	74
Figure 4-20: 3D point cloud generated using SIFT feature based matching.....	77
Figure 4-21: 3D reconstruction workflow for correlation based matching	78
Figure 4-22: Rectified images of stereo pair	79
Figure 4-23: Disparity map acquired from images – BM (left) and SGBM (right).....	79
Figure 4-24: Number of 3D points obtained using feature and correlation based 3D methods.....	80

Figure 4-25: generated Point cloud using BM (top) and SGBM (bottom)	81
Figure 5-1: stereo cameras capturing clear water surface	84
Figure 5-2: image pair captured by stereo cameras	84
Figure 5-3: keypoints detected on clear water surface SIFT (left), SURF (middle) and ORB (right)	85
Figure 5-4: Number of keypoints detected on left image using SIFT, SURF and ORB	86
Figure 5-5: Number of keypoints detected on clear water surface after RANSAC filtering	87
Figure 5-6: Disparity map from clear water surface	88
Figure 5-7: planar surface	89
Figure 5-8: disparity map of planar surface	90
Figure 5-9: reconstructed planar surface.....	90
Figure 5-10: Measurements performed for various baseline configurations. Top (baseline = 9.2cm), middle (baseline = 12cm) and bottom (baseline = 15cm).	91
Figure 5-11: Stereo camera setup for wave reconstruction.....	93
Figure 5-12: Adjustable tripod stand	93
Figure 5-13: Coloured water surface	94
Figure 5-14: Thin film added to assist 3D reconstruction.....	95
Figure 5-15: disparity map after applying opaque sheet on water surface (left) before applying opaque sheet (right)	95
Figure 5-16: 3D reconstruction of opaque sheet. Top - top view of reconstructed point cloud and Bottom -side view of reconstructed point cloud	96
Figure 5-17: Processing time comparison for various image resolutions.....	97
Figure 5-18: Raw readings from stereo cameras before applying zero up crossing analysis. Cameras were placed at distance 99cm above the water surface.....	99
Figure 5-19: Zero up crossing analysis to detect trough and crest.....	99
Figure 5-20: average wave height estimated at various distances.....	100
Figure 5-21: wave probe for measuring wave height	102
Figure 5-22: raw readings from ultrasonic sensor. Top –readings before filtering, bottom-readings after filtering noisy data.....	103
Figure 5-23: raw readings of water surface elevation obtained from stereo cameras and ultrasonic sensor. Top=0.16Hz, middle=0.20z and bottom=0.30Hz during the first repetition.....	104
Figure 5-24: crest and trough detection of wave frequency 0.16Hz	105
Figure 5-25: wave height estimation using various sensors. Top - average weight estimation and Bottom - significant wave height	106
Figure 5-26: Time and frequency representation of wave probe measurements for experiments conducted at a wave frequency 0.16Hz (Top- captured time series and Bottom – derived frequency representation)	108
Figure 5-27: frequency domain representation for various sensors at a wave frequency of 0.16Hz (Top- wave probe, middle – stereo, bottom – sonar).....	109
Figure 5-28: frequency domain representation for various sensors at a wave frequency of 0.20Hz (Top- wave probe, middle – stereo, bottom – sonar).....	110
Figure 5-29: wave amplitude estimated by stereo cameras plotted against wave amplitude estimated by wave probe. The R2 value is 0.93 approximately. Peak occurred at 0.15Hz, 0.21Hz and 0.32Hz for stereo cameras.....	111
Figure 5-30: Spectral Estimation via Periodogram for 0.16Hz run.	112
Figure 6-1: Pixhawk flight controller	117
Figure 6-2: Naze32 flight controller	118
Figure 6-3: KKmulticopter mini.....	118
Figure 6-4: Ardupilot solutions: (a) – Ardupilot Mega (b) Pixhawk	120

Figure 6-5: Ardupilot architecture	123
Figure 6-6:MAVLink Header Format (Atoev et al., 2017).....	124
Figure 6-7: GPS MAVLINK message ID=33(Ardupilot Dev Team, 2017).....	126
Figure 6-8: SITL architecture(ArduPilot Dev Team, 2016d)	127
Figure 6-9: SITL execution	127
Figure 6-10: Mission planner firmware selection	131
Figure 6-11: Mission planner Frame Type Selection.....	131
Figure 6-12: Compass calibration.....	132
Figure 6-13: Accelerometer calibration positions.....	132
Figure 6-14: Radio Calibration procedure.....	133
Figure 6-15: MAVProxy Console and map interface	135
Figure 6-16: MAVProxy UAV configuration.....	135
Figure 6-17: MAVProxy screenshot .Top - Start-up and Bottom - Arming vehicle	136
Figure 6-18: Dronekit SITL.....	139
Figure 6-19: Simulated vehicle at home position (left) and proceeding to proceeding to latitude: - 35.363048° and longitude: 149.165824° (right).....	140
Figure 6-21: Definition of the axes set and locations of two UAVs separated by a distance S	141
Figure 6-22: Orientation of two UAVs heading north.....	142
Figure 6-23: Two Drones in a general position and heading relative to the geographic axes set.....	143
Figure 6-23: heading at 0 degree (top-left), 45 degree (top-right), 89degree (bottom)	145
Figure 6-24: Follower at various positions relative to leader (a) - no coding of appropriate solution (left) (b) - coding of appropriate solution (right)	146
Figure 6-25: Two UAVs equipped with remote controller.....	148
Figure 6-26:Connecting Raspberry Pi to Pixhawk (Choi et al., 2016).....	151
Figure 6-27: Leader-follower communication interface	152
Figure 6-28: Logical flowchart for leader (left) and follower (right)	157
Figure 6-29: Flight path for leader and follower during square pattern flight-test.....	160
Figure 6-30: Distance between leader and follower during square pattern flight-test	161
Figure 6-31: Altitude comparison between leader and follower during square pattern flight-test... <td>162</td>	162
Figure 6-32: Flight results in X-Y plane.....	165
Figure 6-33: Distance between leader and follower during flight-test	166
Figure 6-34: Altitude comparison between leader and follower during flight-test	167
Figure 6-35: Snapshot of leader-follower formation flight.....	168
Figure 7-1: Leader UAV attached with stereo cameras	171
Figure 7-2: Top - Roll response of tuned vehicle and Bottom - Pitch response of tuned vehicle	172
Figure 7-3: Snapshot of leader UAV acquiring image frames of the water surface with on-board stereo cameras.....	174
Figure 7-4: Image acquisition setup	175
Figure 7-5: System overview	176
Figure 7-6: Captured 3D scattered point cloud (a) and surface reconstruction of scattered point cloud (b).....	178
Figure 7-7: outlier detection from stereo measurements.....	179
Figure 7-8: water surface elevation measurement acquired from airborne stereo cameras	181
Figure 7-9: Ground stereo camera system mounted on a tripod for measuring water surface elevation	182

List of Tables

Table 2-1:3D method comparison	22
Table 3-1: Intrinsic and extrinsic parameters of a camera pair	41
Table 4-1: Rig setup and specifications.....	70
Table 4-2: keypoints detected for left and right image using various feature based matching techniques.....	72
Table 4-3: Number of keypoints detected in left and right images after RANSAC.....	73
Table 5-1: Number of keypoints detected on clear water surface using various feature detection techniques.....	85
Table 5-2: Number of keypoints detected in left and right images after RANSAC filtering	87
Table 6-1: Evaluation of researched flight controllers.....	119
Table 6-2: Various types of GCS	129
Table 6-3: Configurable radio channels	133
Table 6-4: Selected Dronekit information requests and mission commands.....	138
Table 6-5: Main flight modes	155
Table 6-6: 5m separation distance for fixed heading angle.....	159
Table 6-7:7m separation distance for fixed heading angle	159
Table 6-8:5m separation flight results for varying heading angle	163
Table 6-9:7m separation flight test results for varying heading angle	164
Table 7-1: Comparison of average wave height	182
Table 7-2: Comparison of significant wave height.....	182

Chapter 1: Introduction

1.1 Background

Although often perceived to be ordinary, water is one of the most important natural resources. Comprising over 70% of the Earth's surface, it sets the stage for the evolution of life on earth and is an essential ingredient of all of life today (Marowski, 1992). Most human activities involve the use of water in one way or another. There is no other resource that affects so many areas economically, humanly and environmentally like water does (Tevera and Moyo, 2000). In view of this fact, one would expect humans to treat the resource with the utmost respect and safe guard its cleanliness. However, it is often not the case as water pollution is still one of the major challenges especially in developing countries. The Joint Monitoring Programme (JMP) for Water Supply and Sanitation, implemented by the World Health Organisation (WHO) reports that 783 million people in the world (11% of the total population) have no access to safe water, 84% of whom live in rural areas (World Health Organization, 2015). Because of these facts, a serious need has emerged for comprehensive and accurate assessments of trends in water quality, in order to raise awareness of the urgent need to provide viable solutions to water pollution in lakes, rivers and streams. In addition, by providing accurate assessments of the pollutants in such aquatic environments the necessary legislative frameworks and directives can be implemented to reduce the threat that pollutants bring to both human health and to ecosystems.

Traditional water monitoring consists of a series of observations, measurements and samples to be collected over time and analysed. The most common parameters that affect water quality are physical, chemical or biological (Chapman

and World Health Organization, 1996). The physical properties include temperature, sediments, flow and erosion. Chemical characteristics involve parameters such as dissolved oxygen, pesticides and other pollutants (World Health Organization, 2015).

Finally, the biological indicators can include micro-organisms like algae and phytoplankton. By monitoring these properties, one can identify possible changes or trends in water quality over time (World Health Organization, 2015).

One possible approach for water sampling is with the use of unmanned submersibles. Remotely Operated Vehicles (ROV) and Autonomous Underwater Vehicles (AUV) fall under this category (Blidberg, 2001). The key difference between the two being that ROVs are tethered vehicles, where the tether supplies power and communications to the operating vehicle. In addition, the vehicle is controlled directly by a remote operator. On the other hand, AUVs contain their own power and can perform missions without operator intervention. Since ROVs are tethered, their speed, mobility and spatial range are very limited in comparison to AUVs. Traditionally, AUVs are either hand-launched for lighter vehicles or launched from a ship or large vessel for larger vehicles (Wynn et al., 2014). Such methods are usually very cumbersome or difficult and expensive to deploy. In addition, for areas that are inaccessible by ship or humans, it would be difficult or nearly impossible to deploy an AUV.

This thesis takes advantage of the recent availability and affordability of Unmanned Aerial Vehicles (UAV) to provide innovative solutions to the problems outlined in the previous paragraph. Traditionally, UAVs have been used for applications such as surveillance and reconnaissance (Puri, 2005), mapping (Zongjian, 2008, Nagai

et al., 2008), infrastructure inspection and maintenance (Montambault et al., 2010, M  th   and Bu  soniu, 2015), search and rescue operations (Tomic et al., 2012) and agriculture (Honkavaara et al., 2013).

In this thesis, UAVs are proposed as alternative solutions for AUV deployment and retrieval. Deploying an AUV from a UAV, provides a more convenient, cost-effective and flexible solution in comparison to hand launching approaches or deploying from a boat or other vessel. However, issues arise in fast moving water or waves; under these conditions, it might be very difficult to land a UAV on water for AUV deployment and retrieval purposes. This thesis offers a novel solution that provides a UAV the autonomous decision-making capabilities to land on water for AUV deployment and retrieval purposes.

1.2 State of the Art

Current water sampling methods are based on human samplers (i.e. humans enter the aquatic environment where the sample should be collected and obtain the sample with a bottle, sometimes boats or kayaks are also used to enter into the aquatic environment) (Fornai et al., 2012).

The Figure 1-1 originally presented here cannot be made freely available via LJMU E-Theses Collection because of copyright. The image was sourced at (Grasshoff et al., 2009).

Figure 1-1: Human personnel collecting water sample (Grasshoff et al., 2009)

This methodology has several drawbacks. It is time-consuming and requires personnel who must be highly trained in both sampling techniques and field test procedures. In addition, the samples obtained through this method might not be representative of the entire water body of interest because it may vary with time, location and extension. This can lead to water samples that are insufficient for capturing fluctuations in water quality. Furthermore, these methods are spatially restrictive. In other words, sampling can be hindered by limited accessibility to the site areas. For example, consider the famous El Zacatón located in north-eastern Mexico (Gary and Sharp, 2006) which is regarded as the deepest water-filled sinkhole (over 350 metres deep).

The Figure 1-2 originally presented here cannot be made freely available via LJMU E-Theses Collection because of copyright. The image was sourced at (Gary and Sharp, 2006).

Figure 1-2: El Zacaton sinkhole (Gary and Sharp, 2006)

It is very dangerous for human personnel to go into such deep waters with boats or other vessels in order to obtain water samples. For such dangerous missions UAVs are an excellent approach because they are able to fly towards environments that are not readily accessible by humans. In addition, UAVs are capable of providing real-time data-on-demand. For these reasons, researchers have devoted their efforts into deploying UAVs as an alternative solution for aquatic environment. For example, UAVs have been used for obtaining water samples from pit lakes (Ore et al., 2015). The authors developed a water sampling mechanism tethered to a UAV that collects 20ml water samples per flight (see Figure 1-3).

The Figure 1-3 originally presented here cannot be made freely available via LJMU E-Theses Collection because of copyright. The image was sourced at (Ore et al., 2015).

Figure 1-3: UAV based water sampling (Ore et al., 2015)

With their system, the UAV takes off and automatically proceeds to an assigned GPS location. Once the UAV arrives at the sample location, the UAV descends and captures the water sample in a vial. Whilst the mission was successful, tethering to a UAV comes with several difficulties (Pratt et al., 2008) discussed as follows:

1. UAV can experience entanglement and movement constraints during take-off.
2. Limitations in rate of descent and horizontal manoeuvres in order to allow adequate time for the tether manager to adjust the tension or take-up the slack.
3. Tethered flights require additional crew for tether management and coordination between pilot and tether manager is crucial to ensure vehicle safety.
4. Range of flight is restricted with tethered UAVs.

Another issue that can arise is in aquatic environments with fast moving water or high waves. Under such conditions, it may prove very difficult to successfully grab water samples with a tethered aircraft due to high currents. In addition, UAVs have a short

battery life span of roughly about 10-15 minutes. Hence, in order to utilise an UAV for water sampling in large areas, the UAV has to be flown back to base station often for recharge to avoid running out of power mid-flight. To tackle such problems, researchers have focused their attention in using AUVs and ROVs for similar applications. For example, Karimanzira et al. (2014) developed a guidance system for an AUV for water quality monitoring in a large seabed. Dowdeswell et al. (2008) also deployed a large AUV (6.7m long and weighing 3,600kg) into the Antarctic waters. The AUV carried a compact water sampler that could be triggered at predetermined times or specific locations, thus allowing measurement of a wide range of water properties at various locations and time intervals.

An obvious advantage of such approaches is their extended endurance in comparison to UAVs since they are not limited by short battery life span. Typical AUVs are capable of performing missions for tens of hours, some higher end systems can go for even days without running out of battery power (Blidberg, 2001). However, issues arise with these approaches; they are expensive and difficult to deploy. In addition, heavy machinery, large support vessels and human operators are usually needed for vehicle deployment and retrieval (see Figure 1-4 below).

The Figure 1-4 originally presented here cannot be made freely available via LJMU E-Theses Collection because of copyright. The image was sourced at (Blidberg, 2001).

Figure 1-4:AUV Deployment (Blidberg, 2001)

One possible approach to mitigate many of the restrictions associated with the current technology today is to utilise a hybrid approach. Rather than deploying an AUV from large vessels or ships, an AUV could be deployed from a UAV. Such a solution takes advantage of the longer operational period of an AUV and the higher flexibility gained when working with UAVs. In other words, when utilising a UAV as a platform to deploy an AUV, the use of the AUV can allow rapid acquisition of large spatial data sets and longer range missions to be executed whilst allowing for opportunities for data acquisition in areas previously inaccessible to vessel-based instrumentation. Thus allowing for a cost effective, fast and flexible solution for water sampling in comparison to current technologies.

1.3 Scope of Research

As discussed in Section 1.1 in fast moving water or high waves, it might be difficult to land a UAV on water for AUV deployment and retrieval. Under these conditions, the UAV might tip over whilst trying to land and can be carried away by the high current. This can lead to loss of the aircraft or possible damage in the electrical circuits of the vehicle.

The solution presented in this thesis targets the problem outlined in the previous paragraph and takes advantage of two UAVs flying in a coordinated manner. One UAV, the “leader”, is equipped with two identical cameras that capture 3D information of the wave through non-contact optical techniques (stereovision). The second UAV, “the follower”, can be equipped with the payload (AUV). Once the leader has identified the water’s surface as suitable for landing (low wave-height) the follower UAV can then proceed to the location identified by the leader, land and deploy the AUV into water. Coordinated flight between the leader and follower is achieved through developing a formation-flying algorithm. Given the coordinates and heading of the leader, along

with its pre-set separation distance from a follower, the formation-flying algorithm seeks to find the coordinate values to allow the follower to fly at a fixed distance from the leader. The need for multiple UAVs arises from the various limitations that hinder the performance and flexibility of individual UAVs when undergoing certain tasks. The advantages of having multiple UAVs are discussed as follows:

- 1) When mounting the two stereo cameras on one UAV and the payload on another, one overcomes the weight constraint that would arise especially for smaller vehicles if both cameras and AUV were mounted on a single UAV. Thus, allowing an increase in flight time for wave inspection.
- 2) For narrow baseline configurations, the stereo cameras have to be relatively close to the water surface in order to provide reliable estimates of the wave height. Under such situations, large waves can potentially cause damage to the payload if stereo cameras and payload are mounted on the same vehicle.
- 3) The notion of utilising two UAVs for such a task could be potentially scaled up to a fleet of vehicles where each vehicle in the fleet is equipped with its own AUV payload.

For wave height measurement, stereovision has been chosen as the preferred optical measurement technique. The advantage of utilising such an optical technique is that spatial information can be acquired without disruption of the wave pattern. In comparison to the tethered system developed by Ore et al. (2015), the approach utilised in this thesis provides a more flexible, easier-to-deploy setup for water sampling applications. It is noteworthy to highlight that this thesis does not cover the actual deployment of the AUV into the body of water because it was considered out of scope of the project.

1.4 Novelty of Research

The conceptualisation of this research is guided by an extensive review of literature within a relevant theoretical construct. Earlier studies utilising stereo imaging systems for three-dimensional measurements of the water surface utilised fixed stereo cameras on rigid structures pointing at the water surface. In this study, the novelty lies in the more flexible approach of mounting the stereo cameras on a UAV for dense 3D reconstruction. By utilising this approach, in conjunction with the formation-flying algorithm developed, a UAV is capable of estimating the wave height from the reconstructed 3D point cloud. Thus, allowing for autonomous safe landing in aquatic regions. Such a framework allows for the deployment and retrieval of an AUV from a UAV for the collection of water samples in remote areas that are not readily accessible.

1.5 Aim and Objectives

The overall aim of this research is to develop a framework that provides a UAV with the decision-making capabilities to ensure the autonomous safe landing in an aquatic environments of another UAV. In order to satisfy the research aim, a number of objectives will need to be considered and these are defined as follows:

- Identify the limitations of current technology for wave height estimation and investigate the use of 3D reconstruction techniques as an alternative.
- Developing a robust pipeline that allows for dense 3D reconstruction using the identified reconstruction technique.
- Validation of the reconstructed 3D point cloud against known technologies, determining the accuracy in the context of wave height measurement.
- Development of a formation-flying algorithm to allow for coordinated control of two UAVs that is beneficial in the context of the problem domain.

- Near real time 3D reconstruction and wave height estimation from a series of images taken from a UAV.

1.6 Thesis Structure

This thesis comprises the following chapters:

Chapter 1 - Introduction

This chapter provides an overview of the problem domain, namely water sampling. Furthermore, the chapter discusses the current technology utilised for water sampling and highlights their limitations. It presents an innovative solution; specifically water sampling for regions that are not readily accessible. In doing so, the major challenges are presented, which include 3D reconstruction, coordinated control of multiple UAVs and wave-height measurement from reconstructed 3D data. The scope of the project is clearly described, the aims and the objectives are discussed, and the novel contributions this leads to are claimed. Finally, the chapter is concluded with an outline of the thesis structure.

Chapter 2 - Current 3D reconstruction techniques

3D reconstruction techniques are suitable for providing high spatio-temporal 3D data of a surface. This chapter reviews various 3D reconstruction techniques and presents stereovision as the preferred reconstruction technique for wave reconstruction.

Chapter 3 - Camera Geometry and Calibration

Camera calibration is the first step towards developing a working stereovision system. This chapter begins by introducing sequences of geometrical transformations to determine the camera geometry, which is important for camera calibration. In addition, it discusses the various calibration techniques used in the literature and provides

Zhang's calibration method as the preferred calibration approach. A camera pair is calibrated using this method and the results are presented.

Chapter 4 - Stereovision

Chapter 4 begins by investigating how 3D reconstruction can be achieved from a calibrated camera pair. In order to achieve this, image rectification is important. This chapter discusses the computation of the fundamental and essential matrix to determine the epipolar geometry of the scene which is crucial for image rectification. In addition, the stereo correspondence (stereo-matching) problem is discussed and an extensive literature survey on the existing stereo correspondence algorithms is presented. Finally, this chapter investigates how the various stereo correspondence algorithms affect the quality of 3D reconstruction.

Chapter 5 - Water surface 3D reconstruction

As with most innovations, laboratory based testing is necessary before a full-scale implementation can be attempted. Waves have been reconstructed in a wave tank and a dense reconstruction of the surface has been achieved using the methods described in Chapter 4, based on the reconstructed 3D data. Finally, a method to estimate the wave height from the 3D point cloud is presented. The results have been validated with ultrasonic sensors and a wave probe.

Chapter 6 – Coordinated Control of two UAVs

A detailed discussion of a formation-flying algorithm that is beneficial in context of the problem domain is presented in this chapter. The algorithm aims to allow the follower UAV to track the leader's trajectory at a given distance. Simulation results and experimental results obtained during field tests are also discussed.

Chapter 7 – Wave height estimation from a UAV platform

In this chapter, field experiments were carried out to assess the feasibility of performing 3D reconstruction and wave height measurement of a water surface from the stereo cameras mounted on the leader UAV. Field tests were conducted over a lake.

Chapter 8 – Conclusion and Future Work

This chapter provides an overall conclusion of the thesis and presents recommendations for further study.

Chapter 2: Current 3D reconstruction techniques

2.1 Introduction

In this chapter, a review of popular 3D reconstruction techniques will be undertaken to assess their suitability for addressing the problems of wave reconstruction. The advantages and disadvantages of each technique are summarised in Table 2-1, outlining the most suitable choice available.

2.2 3D reconstruction techniques

Classical observation methods rely on time series retrieved from wave gauges and ultrasonic instruments or buoys to measure wave elevation at a selected point (Goda and Suzuki, 1977, Heys and Bachmayer, 2014). In order to get a better understanding of the dynamics of waves over a larger area rather than at selected points, 3D reconstruction techniques that can reconstruct the 3-D wave surface in both space and time is highly favourable. Methods that are capable of providing high spatio-temporal 3D data of a surface are categorized into two categories: active techniques that require some kind of energy projection and passive techniques that only use ambient illumination and do not consider any kind of projected energy during data acquisition.

2.2.1 Active techniques

The basic idea with active techniques is to measure the surface radiance due to a proposed light source illuminating it. With these techniques, one can rapidly scan an object's surface, recording shape and often visual properties such as intensity. Laser scanning systems fall under this category. In fact, laser scanners have been used for reconstruction of water surface in a wave flume (Streicher et al., 2013). Two popular

laser scanners systems are Time of Flight (TOF) and Phase Shift scanners and are discuss in the following sections:

2.2.1.1 *Time of flight*

These systems emit a pulse of laser energy and measure the time it takes for that energy to travel to a target, bounce off the target and be returned to the sensor.

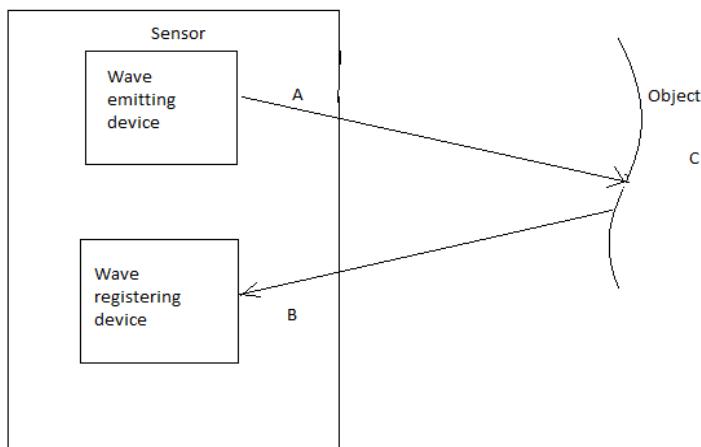


Figure 2-1: Pulse based laser-scanning technique

The pulse is emitted from a source (e.g. laser diode) at point A and reaches the required measurement point on the object at point C. It is then reflected back and is captured by a photodiode within the sensor at point B (Figure 2-1). Assuming the medium within which the pulse travels is homogenous, the law of motion of the pulse is linear, according to

$$s = vt \quad (2-1)$$

Where:

s = Distance to the object(m)

v = Velocity of wave propagation(ms^{-1})

t =Time required to reach the object (s)

Assuming the sensor is built in manner that the distance AC = BC and denoting AC by s, the propagation of the wave during the whole measurement can be described as follows:

$$2s = vt \quad (2-2)$$

Or

$$s = \frac{vt}{2} \quad (2-3)$$

Thus in order to accurately measure distance a precise knowledge of the wave propagation speed v (speed of light $\approx 3 \times 10^8 m/s$) and the ability to accurately determine the time t are required. Because very short pulsed lights are used, these systems require time resolution of picoseconds for obtaining a millimetre resolution during measurement which is difficult to be implemented at low cost (Jang et al., 2013).

2.2.1.2 *Phased based laser scanning*

This technique compares the phase of a transmitted signal with those of a received signal (echo) as illustrated in Figure 2-2. The phase shift between both signals is directly proportional to the distance travelled by the signal (Chen and Ni, 1993). A major shortcoming with this technique is the phase-wrapping ambiguity (Hansard et al., 2012). With such ambiguity, the definite association of an echo and its preceding pulse emission is no longer possible in situations when the round trip time of a pulse is longer than the time between two pulse emissions. In such situations, an echo is

associated with a number of preceding pulse emissions and leads to ambiguous measurement of the phase difference.

The Figure 2-2a originally presented here cannot be made freely available via LJMU E-Theses Collection because of copyright. The image was sourced at (Jang et al., 2013).

(a)

The Figure 2-2b originally presented here cannot be made freely available via LJMU E-Theses Collection because of copyright. The image was sourced at (Jang et al., 2013).

(b)

Figure 2-2: (a) phased based laser scanner (Jang et al., 2013) (b) Phase shift measurement

In comparison to the Time-of-flight based scanners, the phase-shift scanners produce more accurate data at shorter distances. However at longer distances, the Time-of-flight based scanners outperform the phase-shift scanners (San Jose Alonso et al., 2011).

2.2.2 Passive techniques

Passive techniques work with ambient light available so, there is no need for special illumination on the object to acquire 3D information. These techniques acquire thousands of data points simultaneously over a large spatial range and generally differ in the required means of image acquisition i.e. the number and arrangement of cameras and also the method of matching corresponding points in the acquired images. The most common passive techniques are listed below:

- Shape from texture
- Shape from motion
- Shape from stereo

2.2.2.1 *Shape from texture*

This technique was first proposed by Gibson in 1950 (Rosenholtz and Malik, 1997). Shape from texture uses the repetition of an element or the appearance of a specific template over a surface to compute depth properties. One can derive 3D shape information from a single image of a textured surface from the distortion of the texture created by the imaging process. It is based on the idea that neighbouring patches in the scene, which have identical or sufficiently similar texture when projected onto the image plane, appear as patches with different appearances because of distance, orientation and shape of the surface patches with respect to the viewer. This phenomenon is referred to as texture-gradient (Rosenholtz and Malik, 1997). Different texture gradients exist such as the example illustrated in Figure 2-3:

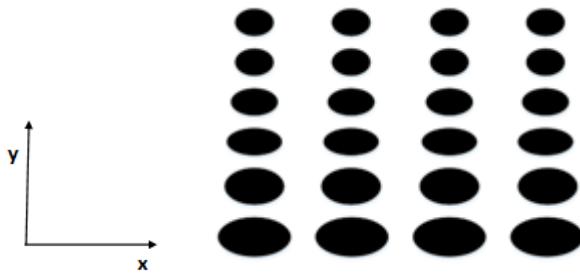


Figure 2-3: Planar surface textured with circular texture elements (Rosenholtz and Malik, 1997)

From Figure 2-3, it can be observed that along the Y-axis, there is a change in the lengths of the major axes of the ellipses due to the fact that they are further away from the observer. This is referred to as the scaling or perspective gradient. In addition, there is also a change in the aspect ratio of the ellipses as we move along the Y-axis (minor axes become smaller at a faster rate than the major axes). This is referred to as the compression gradient. Finally, the areas of the ellipses decrease (the area gradient) and the density increases (the density gradient) as we move along the Y-axis. The mathematical relationship of these different gradients can be used to compute the surface orientation and shape for planar surfaces (Stevens, 1981) and curved surfaces (Gårding, 1992).

For this technique to be effective in 3D recovery, the textures within the scene must be uniform (shape, size and spatial distribution). Such constraints are not necessarily satisfied for moving waves as different parts of the waves could have little or no texture or possibly an entirely different texture depending on the lighting conditions and viewing angle.

2.2.2.2 *Shape from motion*

This technique, commonly also referred to as Structure from motion (SfM), exploits the relative motion between camera and scene. By taking as the only input a set of image correspondences, the technique seeks to find the 3D structure of the scene and also the camera locations where the images were captured in an automated manner. SfM origins can be traced back to the so-called photogrammetry, which since the second half of the 19th century aimed to extract geometric information from images (Westoby et al., 2012). SfM however differs from traditional photogrammetry approaches by determining internal camera geometry, camera position and orientation automatically without the need for a pre-defined set of ground control points or pose and location of the camera. In order to achieve this, the SfM technique has to acquire a multiple set of overlapping images that capture the full three-dimensional structure of the scene as illustrated in Figure 2-4 below.

The Figure 2-4 originally presented here cannot be made freely available via LJMU E-Theses Collection because of copyright. The image was sourced at (Westoby et al., 2001).

Figure 2-4: Structure from motion technique (Westoby et al., 2012)

In order to obtain robust solutions, structure from motion deals with static scenes (Westoby et al., 2012). This is because with SfM technique, the assumption of rigidity in the scene is used to assert that the change in image position of features from one image to another is due purely to the movement of the camera relative to the unknown but static 3D geometry of the scene. This translates into mathematical constraints on the parameters describing camera motion. (Snavely et al., 2008). The 3D point cloud generated from a structure from motion pipeline process suffers from scale ambiguity. In order to compute the absolute scale, one needs the dimension of at least one element in the scene or the baseline between the camera motions (Scaramuzza et al., 2009).

2.2.2.3 *Shape from stereo*

Stereovision is the method utilised by almost all living creatures to achieve a 3D perception of the world. Because of the way our eyes are positioned and controlled, each eye provides a slightly different perspective of the environment; capturing two separate images that are sent to the brain for processing. This disparity between the two images stimulates depth cues in our brains. Fusing the two images recorded by our two eyes allows us to gain a strong sense of depth (see Figure 2-5) (Read, 2014). This can be illustrated through a simple test. When only one eye focuses on an object within close range, the brain provides an idea of that object's position in space; but if the viewing eye changes, the brain is provided with a different information and the object appears to "jump". This "jump" is strongly related to the disparity. Objects that are relatively far away shift only by a small amount in comparison to objects that are closer.

The Figure 2-5 originally presented here cannot be made freely available via LJMU E-Theses Collection because of copyright. The image was sourced at (Cooper, 1995).

Figure 2-5: Stereo human perception (Cooper, 1995)

Marr and Poggio (1979) were the first to point out this form of vision and its application in the field of computer vision. In the field of computer vision, a stereovision system utilises a set of two cameras, to extract depth of a 3-D scene viewed from different viewpoints. To measure the disparity associated with the change in viewpoints the following steps are involved:

- 1) A point of interest in the scene must be selected from one image
- 2) The same point must be identified in the other image
- 3) The disparity in the two corresponding image points must be measured

To find the disparity, the stereo-correspondence problem must be solved. Several solutions have been described in the literature. This is discussed in further detail in Chapter 4. Once the stereo correspondence problem is solved, triangulation techniques (Hartley and Sturm, 1997) can be applied to obtain depth in the scene. A key advantage of stereovision is that the 3D point clouds generated from the process do not suffer from scale ambiguity because the baseline between the cameras is known prior to reconstruction.

2.2.3 Choice of technique for 3D reconstruction of waves

The following table assesses the advantages and disadvantages of the various 3D reconstruction techniques discussed so far in order to determine which method would be the most suitable for 3D data acquisition of a wave.

Table 2-1:3D method comparison

Technology	Advantage	Disadvantage
1. Active <ul style="list-style-type: none"> Time of flight and phase shift Laser scanners 	Performance generally independent of ambient light	Difficult to implement at low cost
	Highly accurate	
2. Passive <ul style="list-style-type: none"> Shape from texture 	Low cost	Heavily relies on textured surfaces
<ul style="list-style-type: none"> Shape from motion 	Low cost	Computationally demanding
		Needs a static environment
		Motion is required to extract 3D data
Stereovision	High accuracy on well-defined targets	Scene can only be recovered up to a scale from only the images acquired
	Low cost and high accuracy on well-defined targets	Can be computationally demanding
	Does not require static environment unlike SfM	
	Absolute scale can be determined from images alone	

The selection of the technique to be used for 3D data acquisition is a very complex task, that must consider the measurement time, cost and quality expected from the measurement technique. Table 2-1 summarizes the main characteristics of various techniques. The advantages and disadvantages of these techniques are strongly application-dependent. For our application, the non-contact measurement technique, stereovision is chosen as the preferred approach for reasons discussed in Table 2-1. Because of the advantages that stereo photogrammetry offers, many researchers have focused their attention on utilising stereo imaging for the reconstruction of water surfaces.

Santel et al. (2002) for example, used a stereo camera system to reconstruct the shape of the water surface on a zone using correlation based matching. Correlation based matching is feasible in this case because large parts of the water surface in the surf zone are covered with foam, which provides surface texture during matching.

To avoid the transparency issue faced during reconstruction of a water surface in a wave tank, Hilsenstein (2005) performed stereo matching using images that had been captured with infrared cameras. The author claims capturing images in this region rather than visible wavelengths reduces the problems associated with transparency, specular reflections and lack of texture. A downside to this approach is that infrared cameras are expensive in comparison to optical cameras (Saari et al., 2017).

Wave Acquisition Stereo System (WASS) (Benetazzo et al., 2012) and Automated Trinocular Stereo Imaging System (ATSiS) (Liu et al., 2008) are successful examples of applying stereovision for three dimensional reconstruction of waves. However, with both systems, the cameras are statically mounted offshore, which reduces flexibility of

operation. Mounting the cameras on a UAV provides a more flexible setup that can assist in beyond line of sight operations.

2.3 Summary

Point based sensors like buoys and ultrasonic sensors measure wave characteristics over small regions. In order to obtain rich information about wave characteristics in both space and time over larger regions, 3D reconstruction techniques are desirable. This chapter has reviewed popular reconstruction techniques in an effort to identify a suitable technique for reconstruction of water waves. Stereovision has been chosen as the preferred approach because it is non-invasive, cost effective and most importantly can be used in a dynamic environment. The next chapter focuses on camera calibration and geometry, which is a first step towards achieving a workable stereovision system.

Chapter 3: Camera Modelling and Calibration

3.1 Introduction

As discussed in the previous chapter, camera calibration is a first step towards achieving a workable stereovision system. Before camera calibration can be achieved, the camera modelling needs to be well understood. This chapter introduces the camera model and how it is gradually obtained through sequences of geometrical transformations. A detailed review of various calibration techniques is presented in an effort to identify a suitable camera calibration technique for this work. Finally, a camera pair is calibrated using the chosen calibration technique.

3.2 Camera Model

The modelling of a camera provides a mathematical formulation, which approximates the behaviour of the camera by using a set of mathematical equations. Camera modelling is based on approximating the internal geometry along with the position and orientation of the camera in the scene. It is a necessary step before camera calibration can be undertaken. The modelling of a camera is broken down into the following four steps (see also Figure 3-4).

The Figure 3-1 originally presented here cannot be made freely available via LJMU E-Theses Collection because of copyright. The image was sourced adapted from (Quintana, 2003).

Figure 3-1: Geometric relation between 3D world coordinate and computer image coordinates –adapted from (Quintana, 2003)

1. The first step consists of a rigid body transformation from the world coordinate system (X_w, Y_w, Z_w) to the camera 3D coordinate system (X_c, Y_c, Z_c) . This transformation is carried out by a rotation matrix and a translation vector.
2. Next is a transformation from the 3D camera coordinate (X_c, Y_c, Z_c) to the ideal image coordinate (X_u, Y_u) by using a perspective projection with a pinhole camera geometry.
3. The pinhole camera geometry assumes no distortion; in the third step, distortion is modelled to account for non-deal lens distortion. Hence a point (X_u, Y_u) is transformed to the real projection of (X_d, Y_d) which are the distorted or true image coordinates on the image plane
4. Finally, the last step involves the transformation from the distorted image coordinates (X_d, Y_d) to the computer image coordinate in pixels (X_f, Y_f)

3.2.1 Rigid body transformation

Changing from the world coordinate system to the 3D camera coordinate system is modelled by using a translation vector and a rotation matrix as shown in Equation (3-1)

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \text{Rot.} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + T \quad (3-1)$$

Where Rot is a 3×3 rotation matrix, as shown in Equation (3-2), that represents the orientation of the camera coordinate system to the world coordinate system.

$$\text{Rot} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (3-2)$$

T is the translation vector, as shown in Equation (3-3), representing the translation between the two coordinates systems and is defined as:

$$T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \quad (3-3)$$

3.2.2 Perspective Projection

The simplest model that provides a very good approximation of a real camera and is also suited for describing the fundamentals of projective geometry is the Pinhole model. In this simple model, all the light rays envisioned as entering the scene travel through a single point known as the “pin-hole” and are then projected onto the imaging plane. The image plane is perpendicular to the optical axis, which goes through the *centre of projection c* (the pinhole). The distance from the centre of projection to the image plane is known as *focal length f*.

The Figure 3-2 originally presented here cannot be made freely available via LJMU E-Theses Collection because of copyright. The image was sourced at (Quintana, 2003).

Figure 3-2: The pinhole mode with image plane located at focal length f away from camera original (Quintana, 2003)

Figure 3-2 illustrates the pinhole camera model. A ray of light from the world $X_wY_wZ_w$ upon entering the camera intersects the image plane, leading to a projected representation X_uY_u at a distance f from the centre of projection. The relationship that maps the 3D real world point to points on the image plane is given as follows (Burger, 2016):

$$X_u = \frac{f \times X_c}{Z_c} \quad (3-4)$$

$$Y_u = \frac{f \times Y_c}{Z_c} \quad (3-5)$$

The next step is a conversion from the Cartesian based coordinate system to a Homogenous coordinate system applied in projective geometry. The relationship between the image and world coordinates in this geometry is shown below:

$$\begin{bmatrix} X_u \\ Y_u \\ Z_u \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \quad (3-6)$$

Where:

$$X_u = f \times X_c \quad (3-7)$$

$$Y_u = f \times Y_c \quad (3-8)$$

$$Z_u = Z_c \quad (3-9)$$

By dividing Equation (3-7) and (3-8) by (3-9) the same solutions to X_u and Y_u are obtained as shown in (3-4) and (3-5). Thus in the homogenous representation, each point in the n-dimensional projective space is represented by an n+1 vector (d_1, \dots, d_2, k) where k is an arbitrary constant that is not equal 0.

3.2.3 Lens Distortion

The third step is the modelling of distortion in lenses. Common lenses used in cameras introduce non-linear distortions resulting in a deviation from the projection described from the ideal pinhole camera model. There are two main forms of lens distortion; radial and tangential distortions (Zhang, 2000).

Radial distortion occurs when rays further from the centre of the lens are bent more than those closer in. This results in magnification at the sides of the lens being different in comparison to the centre. Tangential distortion is due to manufacturing defects and results in lenses not being exactly parallel to the imaging plane. In practice, the effects of tangential distortion are usually negligible and are often disregarded by most calibration methods. Hence, only the radial distortion is accounted for.

The radial distortion occurs in two distinctive forms: the barrel distortion and the pincushion distortion

- Barrel distortion - Occurs when the image magnification is weaker at the edges than that in the centre. Hence, it represents a decrease in image magnification of image points with increasing distance from the image centre. This kind of distortion occurs in both wide-angle lenses and normal view lenses.
- Pincushion distortion - Occurs when the image magnification increases with distance from the centre of the image.

The Figure 3-3 originally presented here cannot be made freely available via LJMU E-Theses Collection because of copyright. The image was sourced at (Quintana, 2003).

Figure 3-3: Effects of radial distortion. Solid lines: no distortion; dashed lines: radial distortion (b: barrel distortion, p: pincushion distortion)(Quintana, 2003)

The relation between the undistorted image coordinates and the distorted image (true) coordinates is as follows (Burger, 2016)

$$X_u = X_d + D_x \quad (3-10)$$

$$Y_u = Y_d + D_y \quad (3-11)$$

X_d and Y_d represent the distorted image coordinates ; D_x and D_y represent an infinite series defined in equations (3-12) - (3-14). The terms k_1 and k_2 represent a vector of distortion coefficients. In practice, when modelling radial distortion only the first few terms k_1 and k_2 are used as adding any more terms can cause numerical instability (Burger, 2016).

$$D_x = X_d(k_1 r^2 + k_2 r^4 + \dots) \quad (3-12)$$

$$D_y = Y_d(k_1 r^2 + k_2 r^4 + \dots) \quad (3-13)$$

$$r = \sqrt{X_d^2 + Y_d^2} \quad (3-14)$$

3.2.4 Computer Image Frame Coordinates

The final step is the conversion from the distorted image coordinates to the computer frame buffer coordinates expressed in pixels shown in Equation (3-15) (Burger, 2016):

$$\begin{bmatrix} X_f \\ Y_f \end{bmatrix} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \end{bmatrix} \begin{bmatrix} X_u \\ Y_u \\ 1 \end{bmatrix} \quad (3-15)$$

The parameters $\alpha = fm_x$ and $\beta = fm_y$ describe the focal length of the camera in pixels. The terms m_x and m_y represent the number of pixels per unit of distance in the X and Y direction. γ denotes the skew between the axes of the coordinate frame and is usually assumed to be zero. u_0 and v_0 represent the location of the image centre with respect to the image coordinate system (i.e. the optical axis).

In summary, the conversion from 3D world coordinates to image coordinates is shown in the flow chart in Figure 3-4:

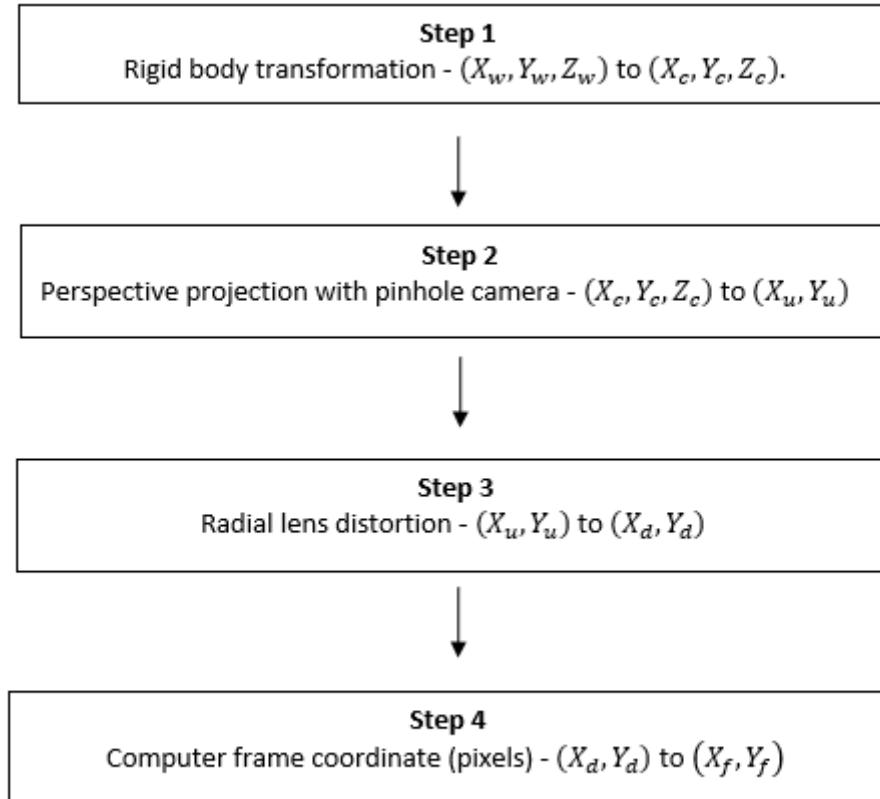


Figure 3-4: Four step transformation from 3D world coordinates to computer image coordinates

3.3 Camera Parameters

The parameters used in the transformation in Figure 3-4 can be categorized into the two classes namely the intrinsic and extrinsic parameters. The combination of both the intrinsic and extrinsic parameters is essential to establish the projective transformation relationship between 3D space and the two-dimensional image plane.

3.3.1 Intrinsic parameters

The parameters in Steps 2-4 in Figure 3-4 are referred to as *intrinsic parameters*. The intrinsic parameters define everything relative to the camera, independently of the outside world. They describe the mapping of the optical rays onto the image plane resulting in two-dimensional pixel coordinates. It is defined by a 3×3 matrix that incorporates the three intrinsic parameters:

- The transformation from 3D camera frame coordinates to computer frame coordinates
- The perspective projection
- The distortion introduced by the lens

Thus, the matrix has the following form shown in Equation (3-16) (Zhang, 2000):

$$K = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3-16)$$

The matrix K can also be referred to as the *camera calibration matrix*.

3.3.2 Extrinsic parameters

The parameters of the rigid body transformation of a 3D object world coordinate to the 3D camera coordinate system are referred to as *extrinsic parameters*. These parameters define the relative position of the camera to the world coordinate system.

The extrinsic matrix takes the form of a rigid transformation matrix: a 3×3 rotation matrix and a 3×1 translation vector. It is possible to concatenate the rotation and translation into a single transformation step of the form shown in equation (3-17)

$$[R|t] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_1 \\ r_{21} & r_{22} & r_{23} & T_2 \\ r_{31} & r_{32} & r_{33} & T_3 \end{bmatrix} \quad (3-17)$$

The extrinsic parameters can also be concatenated with the intrinsic parameters to form the *camera projection matrix* (Burger, 2016) denoted by M :

$$M = K[R|t] \quad (3-18)$$

Thus, given the 3×4 camera projection matrix, it is possible to derive a relationship between a 3D point P_w and its corresponding image projection p_c in pixel coordinates as shown in Equation (3-19):

$$p_c = MP_w \quad (3-19)$$

3.4 Camera Calibration

Camera Calibration is an essential step for 3D reconstruction. Calibrated cameras are needed when metric information is required from the scene (Hartley, 1994). Some potential applications include:

1. *Dense reconstruction*: In a stereoscopic system, a good camera calibration can allow dense 3D reconstruction of objects if the correspondence problem is solved (Hirschmüller, 2005). Each image point is associated with an optical ray passing through the focal point of the camera towards the scene. The crossing of two optical rays allows the derivation of metric positions of 3D points in the scene.
2. *Visual inspection*: Once a dense 3D reconstruction of an object is obtained, the reconstructed object can be compared with a previously stored model to identify any manufacturing imperfections such as dents or cracks in the model of interest.
3. *Camera localization*: By locating some known 3D points in the scene, the position and orientation of the camera in the world coordinate system can be derived. The camera can be placed on a UAV to determine the trajectory of the vehicle. This information can be used in robot control and path planning (Lenz and Tsai, 1989).

Calibration has been a widely researched area in computer vision and the methods found in literature can be classified as:

- Linear versus non-linear
- Photogrammetric versus Self-calibration

The linear versus non-linear techniques can be differentiated depending on the modelling of lens distortion (Ito, 1990). The technique becomes non-linear when lens imperfections are included in the camera modelling. In such cases, the camera parameters are usually obtained through iteration with the constraint of minimizing a determined function (Zhang, 2000).

The photogrammetric technique is performed by observing a calibration object whose 3-D geometry is known with very good precision. An example of such a technique is provided by Tsai (Tsai, 1987). Tsai's approach involves capturing a single image associated with a coplanar calibration object that has two or three planes orthogonal to each other. A two-step algorithm is used, where the first step solves the linear aspects of the calibration problem and the second step addresses the non-linear aspects described by a distortion model.

Finally, in the self-calibration approach, no calibration object is used. By simply moving a camera in a static scene, point correspondence is sufficient to recover the internal and external parameters of the camera (Luong and Faugeras, 1997). While this approach may appear flexible, it is however not yet mature. Due to the large number of parameters to estimate, the results are not always very reliable (Ricolfe-Viala and Sanchez-Salmeron, 2011).

The calibration method used in this research is a hybrid method that lies between the photogrammetric calibration and self-calibration. This method was proposed by Zhang

(Zhang, 2000) and is considered to be more flexible and robust in comparison to other camera calibration techniques. It involves capturing multiple images of the same planar surface in different poses around the cameras' field of view (FOV) to estimate intrinsic and extrinsic parameters as discussed in Section 3.3.1 and 3.3.2. This calibration technique has been implemented within the popular OpenCV framework (OpenCV, 2017) that has been used in this work. The OpenCV library has been used because it is open-source, freely available and most importantly has been tested by other researchers.

3.4.1 Experimental Results

An identical pair of Logitech C930 webcam cameras shown in Figure 3-5 were calibrated using the Zhang's calibration technique. Webcams are used because they are lightweight, low cost, and can be easily integrated on a UAV platform for image capture of the water surface.



Figure 3-5: Logitech C930 webcam

The checkerboard pattern used for the calibration of the camera was printed using a laser printer onto an A0 sheet, and attached to a hard wood planar surface. The checkerboard pattern consisted of 15 columns and 8 rows, giving 120 calibration points. Each square had dimensions of 71mm by 71mm and the origin of the World Coordinate System (WCS) was located at the top left most corner of the checkerboard

pattern. The camera pair captured images of the checkerboard pattern at various orientations as shown in Figure 3-6.



Figure 3-6: Snapshots of checkerboard pattern taken at various orientations

Zhang suggest capturing multiple images of the checkerboard pattern at different orientations for accurate estimation of the intrinsic and extrinsic parameters. To identify the number of calibration images required, the average reprojection error Δe was calculated for a set number of calibration images taken.

$$\Delta e = \frac{\sum_{i=1}^{i=n} E(i)}{n} \quad (3-20)$$

Where $E(i)$ is the reprojection error obtained from image i , and n is the number of calibration images taken. The reprojection error is found by computing the distance between a detected corner-point x_j on the checkerboard in an image plane and a corresponding real world point projected onto the same image plane $M_i(X_j)$. The reprojection error from image i is (Zhang, 2000):

$$E[i] = \frac{1}{k} \sum_{j=1}^k |M_i(X_j) - x_j| \quad (3-21)$$

Where $M_i = K[R_i | t_i]$ represents the projection matrix of the camera for a calibrated grid view i , and k denotes the number of corner-points detected in the image.

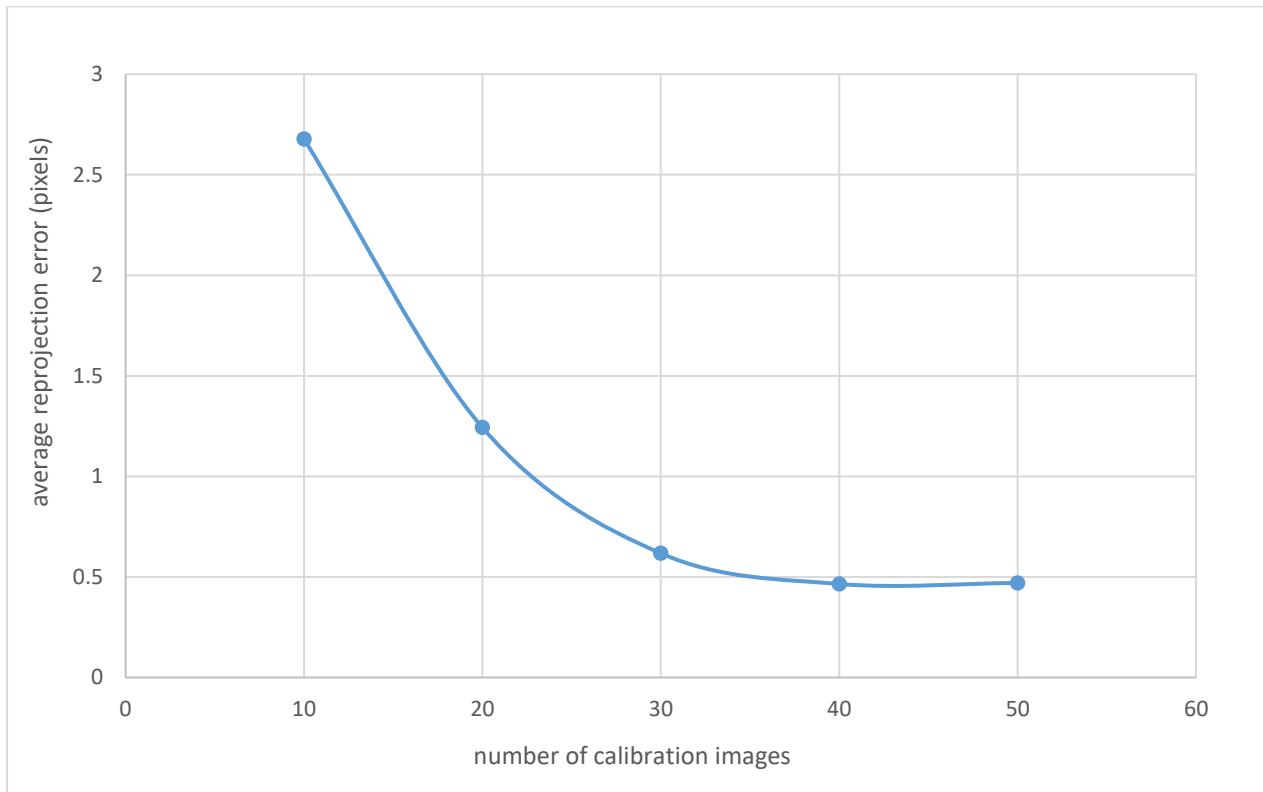


Figure 3-7: Average reprojection error for various number of checkerboard images

For stable calibration, images under varied motions, generally greater than 20 images, should be captured in order to ensure all detected corner points are distributed uniformly (Tan et al., 2017). Figure 3-7 shows the average reprojection error estimated for the various number of checkerboard images taken. Increasing the number of checkerboard images used reduces the average projection error. Forty images were used for calibration because they produced the lowest average reprojection error 0.465 pixels. Increasing the number of calibration images beyond this point has no

significant effect on the average reprojection error. The estimated intrinsic parameters and extrinsic parameters of the camera pair are described in Table 3-1.

Table 3-1: Intrinsic and extrinsic parameters of a camera pair

Left Camera			Right Camera				
Focal length	Horizontal	$\alpha = 509.3683$ (pixel)	Focal length	Horizontal	$\alpha = 509.3683$ (pixel)		
	Vertical	$\beta = 509.3683$ (pixel)		Vertical	$\beta = 509.3683$ (pixel)		
Skew	$\gamma = 0$		Skew	$\gamma = 0$			
Principal point	$u_0 = 305.7052$ $v_0 = 242.1296$		Principal point	$u_0 = 309.3510$ $v_0 = 250.9890$			
Distortion (radial)	$k_1 = -0.1075$ $k_2 = 2.2123$		Distortion (radial)	$k_1 = 0.0467$ $k_2 = -0.0674$			
Extrinsic Parameters	R		$\begin{bmatrix} 0.9993 & -0.0186 & 0.0399 \\ 0.0187 & 0.9998 & 0.0035 \\ 0.0399 & -0.0042 & 0.9992 \end{bmatrix}$				
	T		$T = [-7.8968 \quad 0.1801 \quad 0.02124]$				

3.5 Summary

In this chapter, the camera model was presented and the four step geometric transformation that allows the transformation from 3D world coordinates to computer image coordinates has been discussed in detail. In addition, a comparative study of the most commonly used calibrating methods used over the last few decades has been discussed. Of these methods, it was decided that Zhang's calibration method should be applied because of its flexibility and robustness in comparison to other calibration methods. A camera pair has been calibrated using this method. Forty images of a checkerboard pattern, taken at various orientations, were used for the calibration because they produced the lowest average reprojection error. The estimated intrinsic and extrinsic parameters derived from the calibration procedure would be used for 3D reconstruction as described in Chapter 4.

Chapter 4: Stereovision

4.1 Introduction

This chapter discusses the technique used for recovering the 3D reconstruction of a world scene, or object, from a pair of images, by utilising the estimated intrinsic and extrinsic parameters derived from the camera calibration step. In addition, an investigation into how various stereo matching algorithms affect the quality of 3D reconstruction is undertaken to identify a suitable matching technique that produces a dense 3D reconstruction.

4.2 Standard stereo geometry

The standard stereo geometry describes the ideal configuration of two cameras with the following characteristics:

- Identical internal geometry i.e. the same intrinsic parameters
- Coplanarity and row alignment of both image planes

Coplanarity and row-alignment of image planes imply the optical axes of both cameras are parallel to one another and perpendicular to the baseline of the stereovision system (see Figure 4-1).

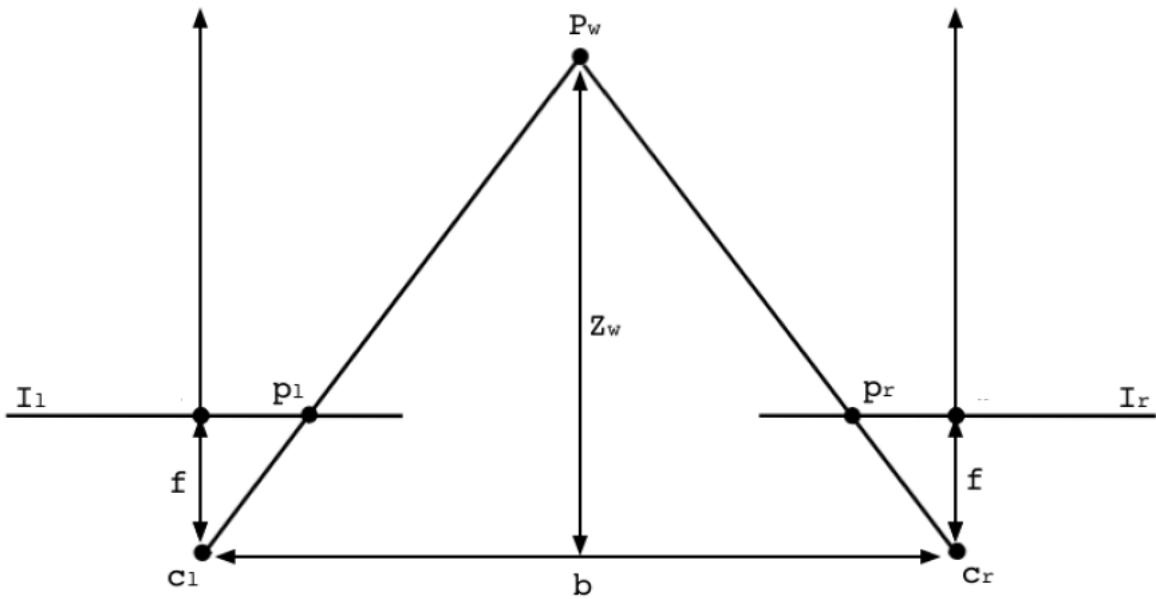


Figure 4-1: Schematic of two cameras in a standard stereo-geometry configuration

Figure 4-1 shows an ideal stereo geometry configuration where both cameras have the same intrinsic properties and where the image planes I_l and I_r are coplanar and row aligned. b represents the distance between camera centres, which is the horizontal translation between the centre of projection c_l of the left camera and the centre of projection c_r of the right camera. Given that the row alignment constraint is satisfied, the projection $p_l = (x_l, y_l)^T$ of a real-world three-dimensional point $P_w = (X_w, Y_w, Z_w)^T$ in the left image plane with the horizontal coordinate x_l , has a corresponding projection $p_r = (x_r, y_r)^T$ in the right image plane with the horizontal coordinate x_r , located in the same row. As a result, the search for corresponding points in a standard geometry system is a one-dimensional problem.

However, it is very difficult to achieve the ideal standard stereo geometry configuration with real world stereovision systems. By estimating the intrinsic and extrinsic parameters of the stereo cameras through camera calibration, it is always possible to

deduce the depth estimation problem to the one described in Figure 4-1 using a process known as rectification (see section 4.5).

4.3 Epipolar geometry

In the real world, the configuration of a general stereo vision system does not satisfy the constraints of the ideal or standard stereovision system. Cameras are usually configured in a convergent configuration towards the scene in order to provide maximum overlap to fully capture the scene (see Figure 4-2).

Besides deriving the internal geometry of the cameras, it is necessary to determine the relative position and orientation of the coordinate of one camera with respect to the other. This is achieved by placing a calibration object at different orientations in both views of the stereovision set up to allow image capture of corresponding sets of points in reference to a calibration object as discussed in the previous chapter. Based on these sets of corresponding points, it is possible to estimate the geometry of the stereovision system with the use of *epipolar geometry*.

Epipolar geometry defines the relationship between two cameras that are related to a set of 3D points and was first published by Longuet-Higgins in 1981 (Longuet-higgins, 1981). In his paper, the author discusses the potential techniques to compute the 3D structure of a scene from a correlated pair of perspective projections when the spatial relationship between the two projections is unknown. Computing the epipolar geometry is also useful in simplifying the correspondence problem in stereo-matching (Kim and Chung, 2004) estimating camera motion (Kim et al., 2010) and scene reconstruction.

Stereovision works by acquiring 3D information from multiple views and solving the correspondence problem between the left and right camera. The challenge however,

is finding potential matches, i.e. - knowing where to look on the right image for a point on the left image that is already known (see Figure 4-2 below).

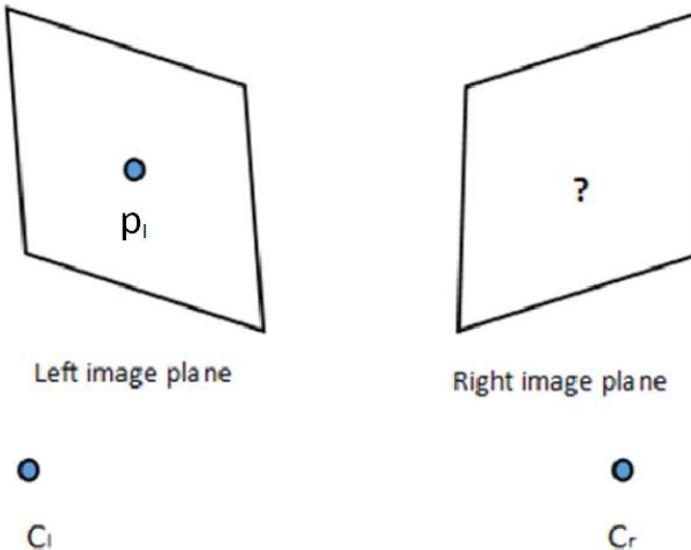


Figure 4-2: Finding a corresponding match in a stereovision setup

Figure 4-2 shows a configuration where two cameras are modelled by their centres C_l and C_r , and p_l is a point formed on the left image plane. In order to find the corresponding match of p_l in the right image plane, the reader is referred to Figure 4-3. Point p_l lies along the projection ray between a 3D point, P , and the left camera centre, C_l . Now projecting this ray onto the right image plane implies the corresponding point on the right image plane, which is denoted as p'_r , must lie on the epipolar line l' . The line l' is referred to as the epipolar line. In other words, the epipolar line, l' , in the right camera gives all possible locations of a 3D point, P , that would project to p_l in the left camera. This setup greatly simplifies the correspondence problem as it reduces the search from 2D to 1D. Hence, rather than having to look at the full 2D image in the right image for a corresponding point, p'_r , the search is simply restricted to corresponding points along the epipolar line.

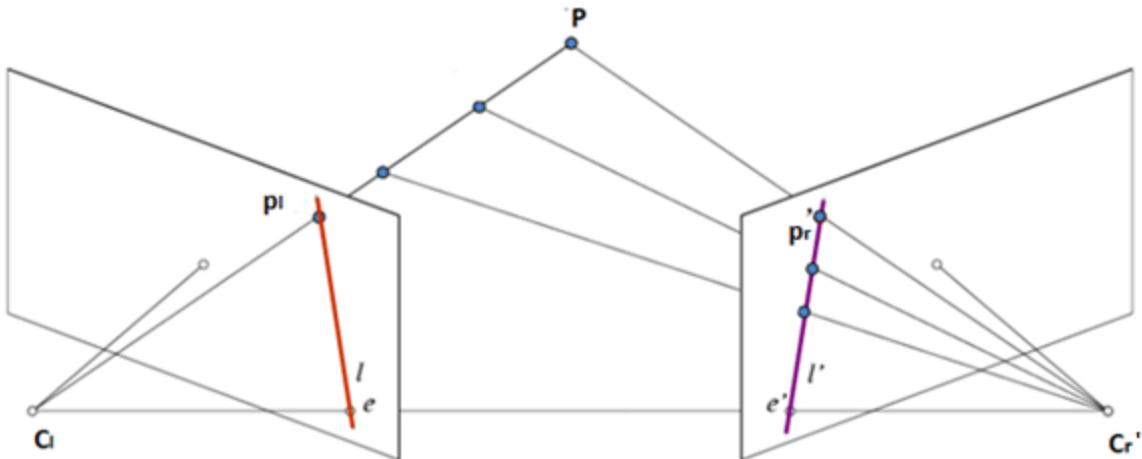


Figure 4-3: the epipolar constraint

The line $\overline{C_l C_r}$ (the base length between the two cameras) intersects the right image plane at the epipole e' as shown in Figure 4-3. The epipole denotes a common point where all the epipolar lines converge. In the case of parallel cameras, the epipolar lines are parallel and hence do not converge at the epipole. In such a configuration, epipolar lines fall along the horizontal scan lines of the image hence, a pixel placed in the n^{th} row in the left image will find a corresponding match in the n^{th} row on the right image. This kind of configuration is best from a computational point of view, but two main issues arise:

- It is practically very difficult to align two cameras perfectly parallel to each other unless a very precisely automated rig is used.
- If the views of the cameras do not converge and the scene is relatively near, it is most likely that both cameras will not look at the same scene, as they would only share a small common Field of View (FOV).

For two cameras located in any position in the world viewing the same scene, the simple geometry of the parallel case can be achieved by *rectification* as will be discussed in Section 4.5. Once image rectification is complete, the pixels

corresponding to point features will lie on the same horizontal scanline and differ only in horizontal displacement.

4.4 Computing Fundamental and Essential Matrix

To compute the epipolar geometry, information about the fundamental or essential matrix is needed. The essential matrix was introduced by Longuet-Higgins (Longuet-higgins, 1981) prior to the fundamental matrix. The realization that the essential matrix could also be applied in uncalibrated situations gave rise to the so-called fundamental matrix (Luong and Faugeras, 1996). As a result, the fundamental matrix captures all the relevant information that is necessary to establish correspondences between two pairs of images without knowledge of the intrinsic parameters of the camera.

4.4.1 Fundamental matrix

The epipolar constraint finds its algebraic expression in the so-called *fundamental matrix* F , which is in the form of a 3×3 matrix (Hartley, 1997). The fundamental matrix can be computed from point correspondences between two images. For every pair of corresponding points $p_l(x_l, y_l)$ and $p_r(x_r, y_r)$ the matrix in Equation (4-1) can be obtained (Hartley, 1997):

$$\begin{bmatrix} x_r \\ y_r \\ 1 \end{bmatrix}^T \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x_l \\ y_l \\ 1 \end{bmatrix} = 0 \quad (4-1)$$

This can be expanded into:

$$x_r(f_{11}x_l + f_{12}y_l + f_{13}) + y_r(f_{21}x_l + f_{22}y_l + f_{23}) + (f_{31}x_l + f_{32}y_l + f_{33}) = 0 \quad (4-2)$$

A set of n pairs of corresponding points forms a linear homogenous system:

$$Mf = 0 \quad (4-3)$$

where M represents a $n \times 9$ matrix with n rows of the form:

$$M_i = [x_l x_r \ x_l y_r \ x_l \ y_l x_r \ y_l y_r \ y_l \ x_r \ y_r \ 1] \quad (4-4)$$

and f represents a vector containing the nine entries of the fundamental matrix, F ; described by equation (4-5):

$$f = [F_{11} \ F_{21} \ F_{31} \ F_{12} \ F_{22} \ F_{32} \ F_{13} \ F_{23} \ F_{33}] \quad (4-5)$$

A common approach that provides a linear solution in estimating f , is the *eight-point algorithm* by Hartley (Hartley, 1997). The eight-point algorithm is sensitive to noise and requires an outlier detection technique to eliminate bad correspondences between images. One proposed solution to remedy this problem is the *RANSAC-algorithm (Random Sample Consensus)*.

The RANSAC-algorithm, proposed by Fischler and Bolles (Fischler and Bolles, 1981) is an iterative algorithm based on data resampling that has the ability to fit a model on experimental data. RANSAC is capable of smoothing data containing a significant percentage of gross errors and thus is ideally suited for the detection of outliers in the correspondence problem. The algorithm workflow is as follows:

1. Sample the number of data points required to fit the model
2. Compute the model parameters using the sampled data points
3. Score by the fraction of inliers within a pre-set threshold of the model
4. Steps 1-3 are repeated until the best model with the highest confidence is found

In combination with the eight-point algorithm, the RANSAC algorithm is used to provide a reliable estimate of the fundamental matrix based on a noisy set of pairs of corresponding points.

4.4.2 Essential matrix

For calibrated cameras, the essential matrix, E , relates corresponding points between the left and right images. Hence, the defining equation for the essential matrix can be described below (Luong and Faugeras, 1996):

$$\hat{p}_r^T E \hat{p}_l = 0 \quad (4-6)$$

Where, \hat{p}_l , represents the normalized image coordinates for the left image and \hat{p}_r , normalized image coordinates for the right image.

The essential matrix can be written in terms of the fundamental matrix as described below (Luong and Faugeras, 1996):

$$E = K_r^T F K_l \quad (4-7)$$

Where K_l and K_r are left intrinsic parameters of the left and right camera respectively.

A unique property of the essential matrix is that it encapsulates the rotation matrix (R) and translation vector(T). Hence, it can be expressed as (Horn, 1990)

$$E = R S \quad (4-8)$$

Where S is an anti-symmetric matrix as shown below:

$$S = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix} \quad (4-9)$$

The essential matrix establishes the link between the epipolar constraint and the extrinsic parameters of a stereovision system.

4.5 Image Rectification

Given a pair of stereo images, the intrinsic parameters of each camera and the extrinsic parameters of the stereo system, image rectification aims to compute the image transformation that makes epipolar lines collinear and parallel to the horizontal axis. In other words, it converts a general stereo configuration to an ideal standard stereo system as discussed in Section 4.2. This rectification process is an essential pre-processing step as it speeds up matching in the correspondence problem because searching horizontal epipolar lines is easier than general epipolar lines.

Different solutions to the image rectification problem exist and can be categorised based on the need for camera calibration. Hartley (Hartley, 1999) describes an approach based on an uncalibrated stereo system. In his work, the author applies methods of projective geometry to determine a pair of 2D projective transformations to be applied to the stereo images in order to enable the epipolar lines to run parallel to the x-axis. Another method proposed by Fusiello et al (Fusiello et al., 2000) is for a calibrated case and the main idea is to define new perspective projection matrices for the left and right camera by rotating the old perspective projection matrices obtained through camera calibration around their optical centre until epipoles are at infinity and the epipolar lines are parallel. The algorithm can be summarised as follows:

1. Obtain old perspective projection matrices for the left and right camera
2. Calculation of the new perspective projection matrices by deducing the suitable rotation matrix that enables images to become coplanar, with the X- axis of both images becoming parallel to the baseline, the Y-axis is orthogonal to the X and the Z-axis is orthogonal to the XY plane.

3. Common intrinsic parameters chosen for both left and right images to ensure row-aligned pair of images
4. Application of the new rectification matrices to the set of images to produce a rectified pair of images

The rectified pair of images produced can be thought of as images produced by a single camera translated along the X-axis of its reference system. Figure 4-4 shows the results of the rectification process applied to a set of stereo images. The resulting rectified images have horizontal epipolar lines that are parallel.

The Figure 4-4 originally presented here cannot be made freely available via LJMU E-Theses Collection because of copyright. The image was sourced at (Fusiello et al., 2001).

Figure 4-4: Stereo pair (top) and rectified pictures (bottom). The right pictures show examples of epipolar lines (Fusiello et al., 2000)

4.6 Stereo correspondence problem

One of the most challenging areas of stereovision is addressing the issue of correspondence between images acquired by left and right camera i.e. determining how to correctly match the points in one image to those in another image. When working with rectified images that have the epipolar lines aligned horizontally, solving the correspondence problem allows the generation of disparity maps. Disparity maps refer to the apparent pixel difference between a stereo image pair (see Figure 4-5 below).

The Figure 4-5 originally presented here cannot be made freely available via LJMU E-Theses Collection because of copyright. The image was sourced at (Scharstein and Szeliski, 2012).

Figure 4-5: Left views and ground truth disparity maps of various models (Scharstein and Szeliski, 2012)

Approaches to stereo correspondence matching can be broadly classified into two categories, Feature-based matching (Ohta and Kanade, 1985, Baker and Binford, 1981) and correlation based matching (Hirschmüller, 2011).

4.6.1 Feature-based matching

Featured based stereo correspondence algorithms attempt to establish a correspondence by extracting feature points such as edges, lines, corners and performing the correspondence search to only include these features. The number of points matched is related to the number of image features identified. Since only feature points are matched, the computation cost is greatly reduced (Mistry and Banerjee). However, a major drawback to this approach is that it generates sparse disparity maps as shown in Section 4.8.1. The general steps of feature detection are outlined as follows (Lowe, 1999):

- Interest points, alternatively known as keypoints are selected from an image at distinctive locations in the images such as corners, blobs, edges etc. Good interest points are highly repeatable, i.e. the same interest points should be found in the image under different viewing conditions.
- Next, a feature vector describes the neighbourhood of the interest point. The description procedure should be invariant to noise, rotation and scale.
- Finally, the feature vectors are matched between images. Matching is often based on computing the Euclidean distance between the vectors (Lowe, 2004).

A popular feature detection algorithm that works by extracting corners in images is Harris corner detector (Harris and Stephens, 1988). The corners are points formed from the intersection of two or more edges. Since the technique is based on extracting corners from an image, it is deemed unsuitable for water surface reconstruction.

Another major limitation with the Harris corner is that it is very sensitive to changes in image scale, so it does not provide a good basis for matching images of different sizes (Schmid et al., 2000). A popular method for dealing with changes in image scale is the Scale-invariant feature transform (SIFT) algorithm, developed in 1999 by David Lowe (Lowe, 1999).

The SIFT algorithm has four major steps for feature detection outlined as follows:

1. Scale space extrema detection
2. Keypoint localization
3. Orientation assignment
4. Keypoint descriptor

The first step involves convoluting the input image with a Gaussian filter at different scales to identify possible keypoints. However, many keypoints generated in the first step are unstable; the second step filters out keypoints by rejecting low contrast features. After identification of keypoints, orientations are assigned to each keypoint so that invariance to image rotation is achieved. The assignment is done based on local image pixel gradients and magnitude. Finally, in the last step a descriptor of the detected keypoint is computed to create invariance to illumination. SIFT is generally computationally intensive, a speeded up version SURF (Speeded Up Robust Feature) was created to reduce the computational time for finding keypoints (Bay et al., 2006). Similar to SIFT, in SURF, images are also convoluted with a Gaussian filter at different scales in order to find keypoints. However, they are done with integral images to reduce computational time. The Gaussian blurring used in SIFT and SURF algorithms does not respect the natural boundaries of the object since image details and noise are smoothed. To make blurring adaptive to image features, the KAZE algorithm was

developed (Alcantarilla et al., 2012). In comparison with the SIFT and SURF techniques, the KAZE algorithm is however computationally more expensive (Alcantarilla et al., 2012).

An effective alternative, that has similar matching performance with SIFT and SURF and is not as computationally expensive as the KAZE algorithm, is the ORB algorithm (Oriented Fast and Rotated BRIEF algorithm) (Rublee et al., 2011). The ORB algorithm is a fusion of the FAST keypoint detector (Rosten et al., 2010) and the BRIEF descriptor (Calonder et al., 2010). The FAST keypoint detector is a corner detection method that attempts to classify whether a pixel is a corner based on a circular neighbourhood around the pixel of interest. The BRIEF descriptor relies on a small number of intensity difference tests to represent an image patch as a binary string. Computing the descriptor of the keypoint detected by the FAST technique offers an improvement in speed and performance.

In this thesis, the SIFT, SURF and ORB matching techniques have been investigated for 3D reconstruction as they have been proven to exhibit similar matching performance and are robust to image noise, rotation and scale (Rublee et al., 2011).

4.6.2 Correlation based matching

In contrast to extracting features from the image for matching as discussed in the previous section, correlation based matching methods attempt to match pixels in one image, with pixels in another image, by exploiting a number of constraints. These methods can be divided into Local and Global methods (Hirschmuller and Scharstein, 2007).

4.6.2.1 Local Methods

Local methods compute the similarity between pixels by comparing windows around the pixels of interest. They are commonly referred to as block matching methods. These methods seek to estimate the disparity of a pixel in one image by comparing a fixed rectangular window about that pixel (the template) with a series of rectangular windows extracted from the other image (the search region). These techniques assume that the neighbour pixels surrounding the pixel of interest all have similar colour intensities. Moreover, they also assume that the colour intensities of the neighbours of a pixel in the left image are close to those of the same neighbours of its homologous pixel in the right image. Figure 4-6 shows a square window, defined by its centre pixel, P , and its width, w . Pixels within the window in the two images are compared and aggregated throughout the entire window.

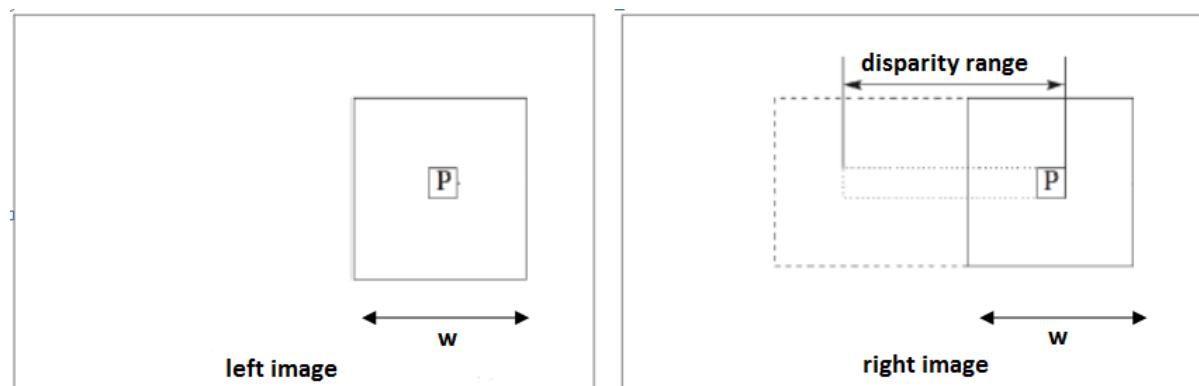


Figure 4-6: Correlation between left and right images at pixel P

Some common correlation based metrics used for comparison between the rectangular windows in the left and right images include the Sum of Absolute Differences (SAD), the Sum of Squared Differences (SSD), Normalised Cross Correlation (NCC), Rank Transform (RT), and Census Transform (CT) (Patil et al., 2013).

Techniques such as NCC are based on statistical methods that determine similarity between the left and right rectangular windows. A major limitation with NCC is that it

tends to blur depth discontinuities compared to any other matching function (Hirschmuller and Scharstein, 2007). NCC is calculated as shown in Equation (4-10) (Shetty et al., 2016):

$$NCC = \frac{\sum_w (p_l(x, y) - mean(p_l)) * (p_r(x, y) - mean(p_r))}{\sum_w (p_l(x, y) - mean(p_l))^2 * \sum_w (p_r(x, y) - mean(p_r))^2} \quad (4-10)$$

Where $p_l(x, y)$ represents the pixel intensity of a pixel in the left image, and $p_r(x, y)$ the pixel intensity of a pixel in the right image. The same notations are used for Equation (4-11) and (4-12).

The SAD metric is based on the intensity difference between pixels and is computationally simpler than the NCC correlation technique. In this method, the correspondence is achieved by selecting a window of the required dimensions between the left and right images and adding the difference between the elements over the entire window (Hamzah et al., 2010).

$$SAD = \sum_w |p_l(x, y) - p_r(x, y)| \quad (4-11)$$

Another metric based on intensity difference is the SSD. In this technique, the differences between pixels are squared and aggregated within a window. This method has a higher computational complexity compared to SAD algorithm as it involves numerous multiplication operations (Baydoun and Al-Alaoui, 2014).

$$SSD = \sum_w (p_l(x, y) - (p_r(x, y)))^2 \quad (4-12)$$

The methods discuss so far above are very sensitive to the intensity variance between the two images. Zabih and Woodfill (Zabih and Woodfill, 1994) propose an alternative method for computing correspondence by applying local non parametric transforms to the images before block matching techniques are applied. The RT for a local region

about a pixel is defined as the number of pixels in that region for which the intensity is less than the pixel of interest. The resulting values are based on the relative ordering of pixel intensities rather than the intensities themselves. As a result, the magnitude of the pixel values is compressed. This technique is robust to outliers that can occur because of occlusion. The rank transform can be defined as (Zabih and Woodfill, 1994):

$$RT(P) = \sum_{n \in W} T[I(P_n) < I(P_c)] \quad (4-13)$$

$$\begin{matrix} 89 & 63 & 72 \\ 67 & 55 & 64 \Rightarrow 2 \\ 58 & 51 & 49 \end{matrix}$$

*Figure 4-7: Example of rank transform using a 3*3 window*

The term, P_c , in Equation (4-13) denotes the central pixel and P_n , the neighbouring pixels. The function $T[]$ is defined to return 1 if its argument is true and 0 otherwise. Figure 4-7 shows an example, because 51 and 49 are smaller than 55 (the pixel of interest), the sum would be 2 based on Equation (4-13). This implies that the rank cost is 2.

One issue with the rank transform is that information is lost as the relative ordering of all of the pixels surrounding the pixel of interest is encoded in a single value. Zabih and Woodfill (Zabih and Woodfill, 1994) proposed a variation of the rank transform called the census transform which preserves the spatial distribution of the rank by encoding them in a bit string. Each bit in the bit string corresponds to the neighbouring pixel surrounding the pixel of interest. If a pixel has a lower intensity than the pixel of interest, then the surrounding bit is set to 1 as shown in Figure 4-8. Matching is

performed by computing the Hamming distance (i.e. the number of bits that differ) between bit strings.

89	63	72
67	55	64
58	51	49

$$67 \quad 55 \quad 64 \Rightarrow 00000011$$

*Figure 4-8: Example of census transform using a 3*3 window*

It is therefore possible to solve the correspondence problem by minimising the Hamming distance after the census transform has been applied to the images.

After computing and aggregating the matching costs using the correlation matching techniques discussed, the disparity of a given pixel can be computed by performing a Winner-Takes-All (WTA) optimization, i.e. For each pixel, choose the disparity associated with the minimum cost function. Repeating this process for every pixel in the image could be used to obtain the disparity map of the scene.

A factor to consider when utilising block based matching techniques is the effect of window-size. The window size must be large enough to include sufficient intensity variation for reliable matching between the left and right images, but must be small enough to avoid the effects of projective distortion. Smaller window sizes, which do not cover enough intensity variation, produce detailed disparity maps but at the cost of introducing more noise to the image. On the other hand, larger window sizes, which cover regions in which the depth of scene points varies, can lead to inaccurate matching due to different projective distortions between the left and right images. This consequently leads to smoother but not very well detailed maps (Hirschmuller and Scharstein, 2007).

For these reasons, an adaptive window solution is proposed by Kanata and Okutomi (Kandade and Okutomi, 1994). The authors investigate how disparity and intensity variation within a window affect the uncertainty of disparity estimation at the centre point of the window. This is achieved by searching for possible window sizes and shapes that produces disparity estimates with the least uncertainty for each pixel.

Another solution presented by Fusiello et al (Fusiello et al., 1997) is a multiple window approach. The main idea is that, for each pixel in the image, correlation is performed with nine different windows, and the disparity with the smallest SSD error value is retained as the solution.

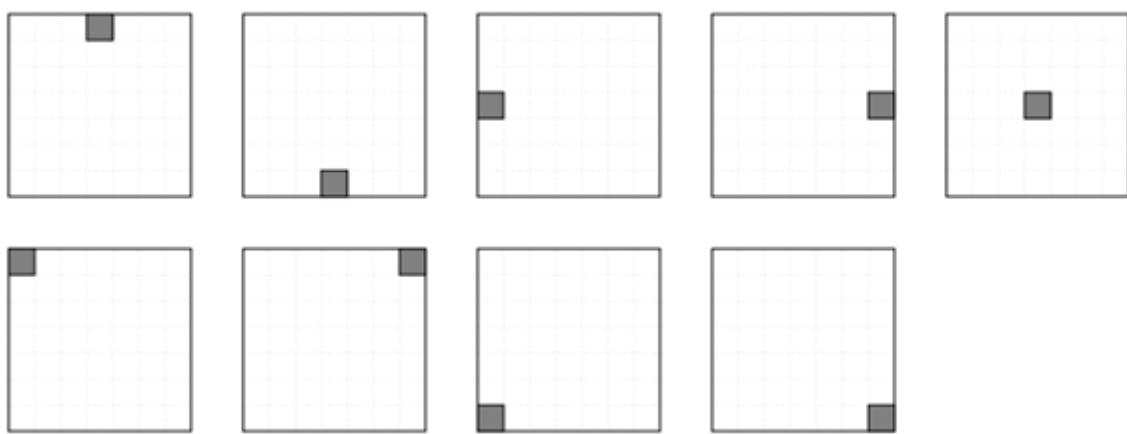


Figure 4-9: The nine correlations windows, the pixel for which disparity is computed is highlighted

Generally, block-matching techniques are suited for real-time applications, however they are considerably sensitive to the presence of image regions characterised by sudden depth variations and occlusions, and can often produce noisy results in low textured regions. In addition, the uniqueness of matching is only enforced for one image (the reference image), while points in the other image might be matched to multiple points (Scharstein and Szeliski, 2002). A solution to this problem is the application of global techniques, which are less sensitive to occlusions and uniform texture regions in images. These are discussed in detail in the following section.

4.6.2.2 Global Methods

Global methods typically skip the cost aggregation step and define a global energy function that includes a data term and a smoothness term. They seek to find the disparity solution d that minimizes the global energy function. This energy function, $E(d)$, consists of two terms: data, $E_d(d)$, and smoothness, $E_s(d)$, and is defined as follows (Scharstein and Szeliski, 2002)

$$E(d) = E_d(d) + \lambda E_s(d) \quad (4-14)$$

The term, $E_d(d)$ measures the degree of similarity between each pixel in the reference image and its corresponding pixel in the target image λ is a constant factor.

Correlation based metrics like SAD or SSD determine this similarity due to the good quality to speed ratio they offer. Performing cost calculations on pixels is generally ambiguous and incorrect matches can easily have a lower cost than correct ones due to noise (Scharstein and Szeliski, 2002). So therefore, an additional constraint $E_s(d)$ is added to support smoothness by penalizing changes in neighbouring disparities. If two pixels are adjacent, they should be displaced by the same amount, i.e. have similar disparities. Thus, the smoothness term is often restricted to only measuring the differences in disparities between neighbouring pixels.

Once the global energy function has been defined, the next step is to minimise it. Several minimisation methods exist in literature. Popular methods are Graph cuts (Vladimir et al., 2014) and Simulated annealing (Barnard, 1989). These optimization methods in general give good empirical results compared to the local methods and form the basis for many of the state-of-the-art stereo correspondence algorithms applied today.

As discussed in Section 4.6.2.1, local techniques suffer from occlusions. A common technique that global methods employ to deal with occlusions is the left-right consistency check. With this technique, an image pair is utilised as follows, for each pixel in the left image the corresponding homologous pixel in the right image is found and for each pixel in the right image the corresponding homologous pixel in the left image is found. This approach yields different sets of conjugate pairs, in which some points are involved in more than one conjugate pair. Points that are involved in more than one conjugate pair do not satisfy the uniqueness constraint and as a result are discarded. Consider for instance point B of Figure 4-10, and take the left image as reference. Although B has no corresponding match in the right image as a result of occlusion, local based methods such as SSD can still return a corresponding match (C'). If the right image is now taken as a reference instead, C' , is correctly matched to its conjugate point C in the left image. Hence, the conjugate pairs (B, C') and (C, C') violate the left-right consistency; in other words, C' does not satisfy the uniqueness constraint and can be disregarded.

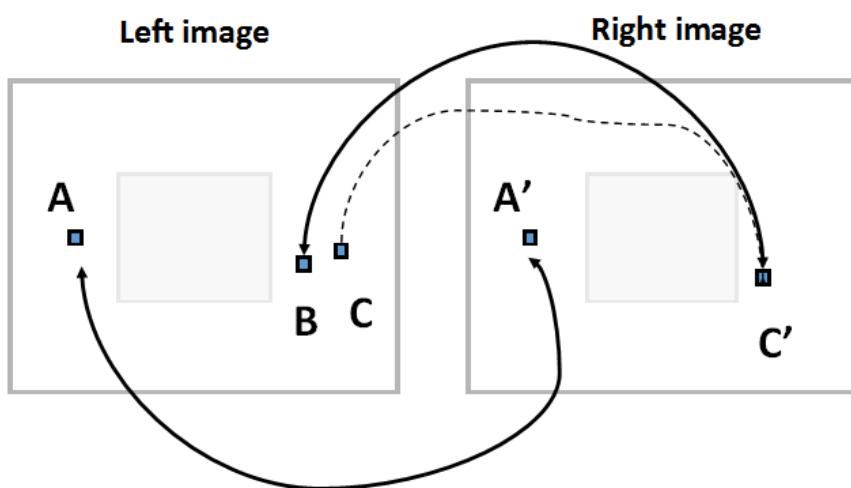


Figure 4-10: Left-right consistency check

In general, global methods provide reliable results. However, image correlation using global methods is slow because of higher computational complexity (Scharstein and Szeliski, 2002). A hybrid approach called Semi Global matching that successfully combines concepts of global and local stereo methods for accurate, pixel-wise matching at low runtime (Hirschmüller, 2005) is addressed in the following section.

4.6.2.3 *Semi-Global matching*

The Semi Global Matching (SGM) is based on the idea of pixel wise matching (Hirschmüller, 2005). The cost calculation for a pair of pixels can be derived using either Equations (4-11) or (4-12). Hirschmuller however adopts a different matching cost based on Mutual Information (MI) because it is insensitive to illumination changes. Mutual Information in general is a measure for model alignment based on the amount of uncertainty (entropy H) in a probability density function. It is defined from the entropy, H , of two images as shown in Equation (4-15):

$$MI_{I_l, I_r} = H_{I_l} + H_{I_r} - H_{I_l, I_r} \quad (4-15)$$

H_{I_l} and H_{I_r} denote the entropy of the left and right images respectively and H_{I_l, I_r} represents the joint entropy. The entropies of the left and right images are calculated from their probability density distributions of intensity across the image. Russakoff et al. (2004) states that for well-registered images the joint entropy H_{I_l, I_r} is low because one image can be predicted by the other. Cost calculation using this technique seeks to maximise the entropy of the individual images to ensure maximum mutual information between images.

The SGM uses a similar energy function, $E(D)$, as shown in Equation (4-14) for penalizing disparity steps and is given in Equation (4-16) (Hirschmuller, 2008):

$$E(D) = \sum_P C(p, D_p) + \sum_{q \in N_p} P_1 T \cdot [|D_p - D_q| = 1] + \sum_{q \in N_p} P_2 T \cdot [|D_p - D_q| > 1] \quad (4-16)$$

The first term in Equation (4-16) sums the matching costs C between a pixel p in the left image and a potential corresponding pixel in the right image for a given disparity D . The second term adds a penalty P_1 to the cost function for all pixels q in the neighbourhood N_p if the difference between disparity D_p and the disparity D_q at a neighbouring pixel q is 1. The third term adds a larger penalty value P_2 to the cost function C if the difference between the disparity D_p to the disparity D_q is higher than one. The *function* T is an indicator function which returns 1 when the condition is true, and zero in all other cases.

Minimizing the energy function $E(D)$ along the image rows (scanline) one at a time suffers from the well-known difficulty of enforcing inter-scanline consistency which results in horizontal “streaks” in the computed disparity map (Hirschmuller, 2008). This is because of the difficulty in relating the 1D optimizations of individual image rows to each other in a 2D image.

The SGM technique overcomes these problems by calculating $E(D)$ along 1D paths from 16 different directions towards each pixel of interest (see Figure 4-11). The costs of all paths are summed for each pixel and disparity. Therefore, at each pixel, the disparity with the lowest cost is chosen as the solution. By performing such an operation for each pixel in the image, a disparity map can be built for the entire image.

The Figure 4-11 originally presented here cannot be made freely available via LJMU E-Theses Collection because of copyright. The image was sourced at (Hirschmuller, 2008).

Figure 4-11: Multi directional search pattern for SGM where eight optimization paths from different directions meet at every pixel (p) (Hirschmuller, 2008)

4.7 Triangulation

After solving the correspondence problem, the 3D coordinates of the scene can be found. In order to find the 3D coordinates of a point X , which project onto the left and right image plane at image points p_l and p_r , one has to find the intersections of the two rays that go through the image points and their respective camera centres. The triangulation problem deals with finding the intersection of these two rays in space. In the presence of noise, the two rays are not guaranteed to cross. The noise arises from the fact that the positions of p_l and p_r obtained from the stereo correspondence algorithms are subject to uncertainties during the search for corresponding points (Hartley and Sturm, 1997). As a result, an approximate solution needs to be found.

A popular method for finding an approximate solution to this problem is the linear triangulation method as described by Hartley and Sturm (1997). The triangulation procedure is discussed in the following paragraph.

A 3D point P_w and its corresponding image projection p_l in the left camera is related by the projection matrix M . This is expressed by Equation (4-17) (as discussed in the previous chapter).

$$p_l = MP_w \quad (4-17)$$

p_l can be expressed in homogenous form $\tilde{p}_l = w(x_l, y_l, 1)^T$, where (x_l, y_l) are the observed point coordinates in the left image and w represents the arbitrary scale factor. Denoting by M_i^T the i^{th} row of the matrix M the equation (4-17) can be written as

$$\tilde{p}_l = M\tilde{P}_w = \begin{bmatrix} M_1^T \\ M_2^T \\ M_3^T \end{bmatrix} \begin{bmatrix} \check{X} \\ \check{Y} \\ \check{Z} \\ \check{W} \end{bmatrix} \quad (4-18)$$

Hence,

$$wx_l = M_1^T \tilde{P}_w \quad (4-19)$$

$$wy_l = M_2^T \tilde{P}_w \quad (4-20)$$

$$w = M_3^T \tilde{P}_w \quad (4-21)$$

Equation (4-21) can be substituted into Equation (4-19) and (4-20) yielding two equations for three unknowns. An analogous substitution can be applied to the equations corresponding to the right camera and the resulting system of four linear equations can be arranged into a matrix of the form shown in Equation (4-22)

$$A\tilde{X} = \begin{bmatrix} x_l M_3^T - M_1^T \\ y_l M_3^T - M_2^T \\ x_r M_3^T - M'_1^T \\ y_r M_3^T - M'_2^T \end{bmatrix} \tilde{X} = 0 \quad (4-22)$$

The least-squares solution for \tilde{X} yields the position of the scene point corresponding to \tilde{p}_l and \tilde{p}_r in homogenous coordinates. Finally, the Euclidean coordinates of the scene point corresponding to the image p_l and p_r are shown in Equation (4-23).

$$P_w = 1/\tilde{W}(\check{X}, \check{Y}, \check{Z})^T \quad (4-23)$$

4.8 Experimental Results

An investigation into how the feature and correlation based matching affect the quality of 3D reconstruction is undertaken to determine a suitable stereo matching algorithm that provides a dense 3D reconstruction. Before proceeding to reconstruction of a water surface, images of bones of various shapes and sizes captured from the calibrated camera pair have been used as a case study in this chapter. The notion behind this preliminary experiment is to ensure the developed software (written in C++) is robust enough to produce a dense 3D reconstruction using a stereovision approach. Bones have been used because they possess various contours and edges, which needs to be reconstructed by the various matching techniques. A matching technique that can successfully capture such level of detail would most likely be successful in providing rich 3D data of the water surface and wave patterns over a given region.

4.8.1 3D reconstruction using feature based methods

The SIFT, SURF and ORB feature detection methods are investigated because they exhibit similar matching performance and in addition, are robust to image noise, rotation and scale as discussed in Section 4.6.1. Figure 4-12 shows the overall 3D reconstruction pipeline using these feature based matching methods.

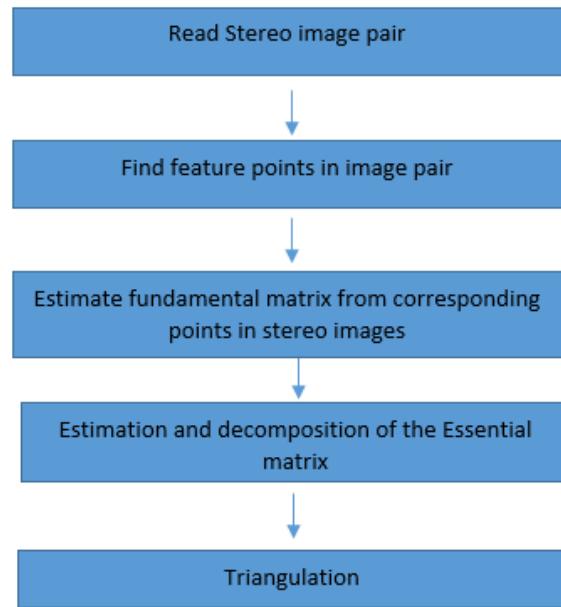


Figure 4-12: 3D reconstruction pipeline for feature based matching

The calibrated camera pair was rigidly mounted on a rig facing downwards to allow image capture of the object of interest. The rig was built using a Rexroth frame beam as shown in Figure 4-13. Specification of the rig is shown in Table 4-1.



Figure 4-13: Rexroth beam used for construction of Rig



Figure 4-14: Stereo rig for 3D reconstruction

Table 4-1: Rig setup and specifications

Overall size (cm) (Length, Width and Height): **79.5 × 104.5 × 102**

Cross section beam size (cm) (Length and Width): **20 × 20**

Material Aluminium

Baseline between cameras 9.2cm

Image resolution 640 x 480



Figure 4-15: left and right images captured by stereo camera

Figure 4-15 shows the left and right images captured by the stereo cameras. Using the OpenCV library, keypoints were extracted individually from the images using SIFT, SURF and ORB detection techniques and compared. Figure 4-16 below shows a snapshot of the keypoints detected in the left image using the various techniques.

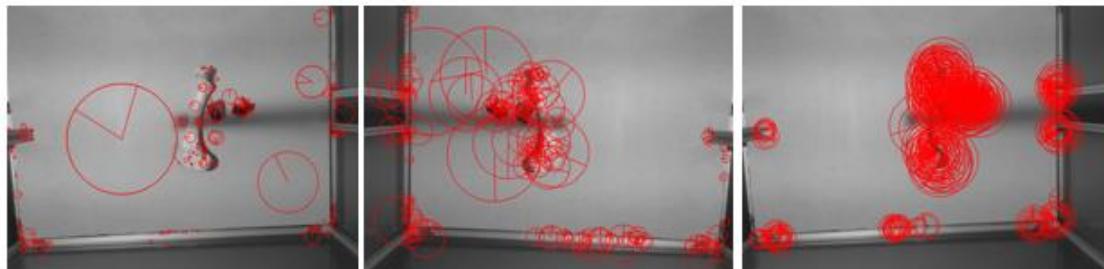


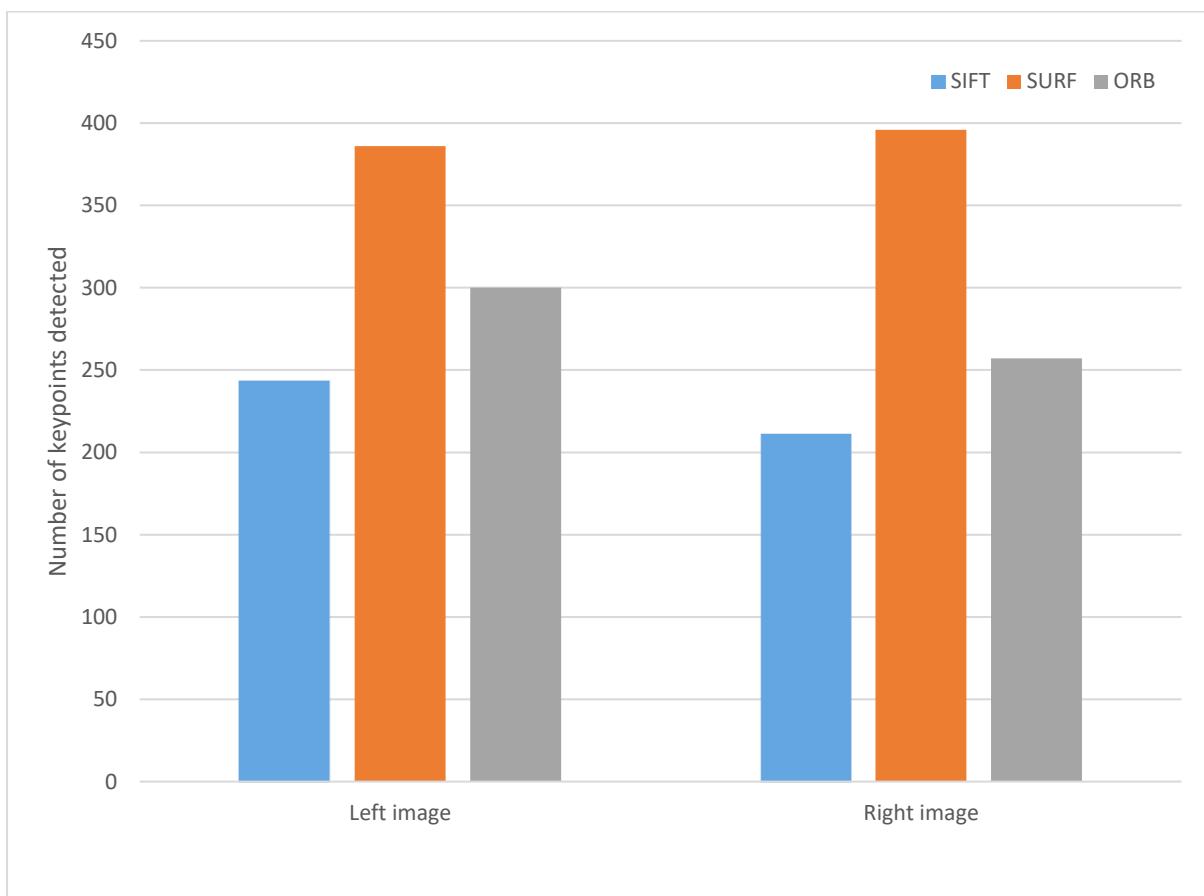
Figure 4-16: keypoint detection - SIFT (left), SURF (middle) and ORB (right)

The keypoints were detected in the left and right images using the SIFT, SURF and ORB matching techniques. Table 4-2 below shows the results.

Table 4-2: keypoints detected for left and right image using various feature based matching techniques

	Number of keypoints detected in left image	Number of keypoints detected in right image
SIFT	243	211
SURF	386	396
ORB	300	257

To provide a visual representation of the results obtained in Table 4-2. Bar plots are provided in Figure 4-17.

*Figure 4-17: Number of keypoints detected on left and right images*

In the left image, SURF produced the highest number of keypoints 386, followed by ORB, which produced 300 and lastly SIFT which produced 243.

The number of keypoints detected in the right image followed the same trend as the left image. Again, SURF produced the highest number of keypoints 396, followed by ORB, which produced 270 and lastly SIFT which produced 211.

Matching corresponding keypoints between the left and right images, is done using the nearest neighbour. The nearest neighbour is defined as the keypoint with minimum Euclidean distance as discussed in Section 4.6.1. The keypoints detected in both the left and right matches are however prone to outliers and can lead to ambiguous matching as shown in Figure 4-18. As a result, the same number of keypoints are not detected in the left and right images (not a one-to-one relationship) as seen in Table 4-2. To eliminate this issue, and filter out outliers that can lead to possible mismatches, the RANSAC algorithm has been used to estimate the inliers due to its ability to tolerate a large fraction of outliers and still produce robust estimated solutions as discussed in Section 4.4.1. Table 4-3 shows the inliers obtained after the RANSAC filter has been applied. It is noteworthy to highlight that the same number of keypoints have now been detected after applying the RANSAC filter.

Table 4-3: Number of keypoints detected in left and right images after RANSAC

	Number of keypoints detected in left image	Number of keypoints detected in right image
SIFT	56	56
SURF	54	54
ORB	40	40

As seen in Table 4-3, SIFT retained the highest number of inliers (56) after RANSAC filtering. After filtering, the nearest neighbour can now be identified for corresponding matching. Figure 4-18 and Figure 4-19 show the matched correspondences between the left and right images before and after the RANSAC filter was applied respectively. Applying the RANSAC filter clearly eliminates the issue of mismatches between the images as shown in Figure 4-19.

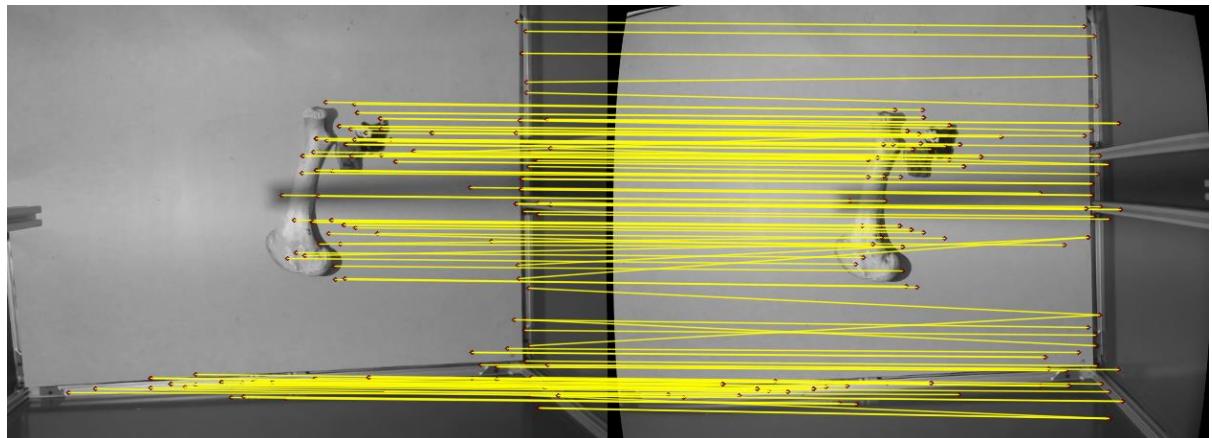


Figure 4-18: Correspondence matching before RANSAC filtering

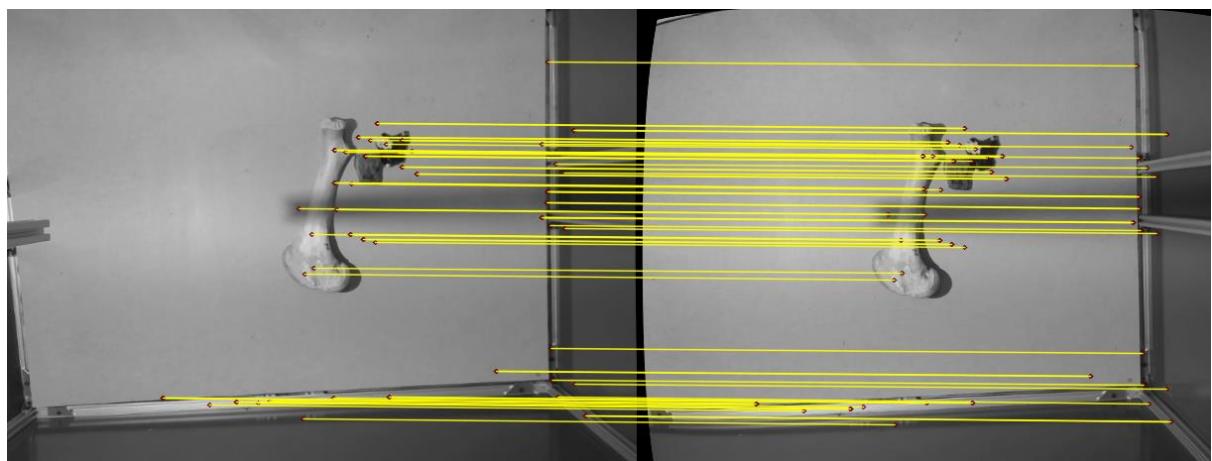


Figure 4-19: Correspondence matching after RANSAC filtering

After finding point correspondences between the left and right images, the fundamental matrix can be estimated using the Hartley's eight-point algorithm. The point correspondences obtained from the SIFT algorithm are used in the estimation of the fundamental matrix because it retained the highest number of corresponding points after the RANSAC filter was applied. As described in Section 4.4.1, F is defined by Equation (4-24)

$$F = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \quad (4-24)$$

The estimated fundamental matrix from the SIFT corresponding points is given by Equation (4-25):

$$F = \begin{bmatrix} 7.5094 \times 10^{-8} & -2.0804 \times 10^{-5} & 3.8333 \times 10^{-4} \\ 2.1095 \times 10^{-5} & -1.4727 \times 10^{-6} & -1.5147 \times 10^{-1} \\ -1.0790 \times 10^{-2} & 1.5220 \times 10^{-1} & 1.0000 \end{bmatrix} \quad (4-25)$$

As discussed in Section 4.4.2, the essential matrix E can be written in terms of the fundamental matrix as shown in Equation (4-26):

$$E = K_r^T F K_l \quad (4-26)$$

Where K_l and K_r are the intrinsic parameters of the left and right camera derived from the calibration procedure in Chapter 3.

$$E = \begin{bmatrix} -2.3892 \times 10^{-3} & 6.6190 \times 10^{-1} & 2.8926 \times 10^{-1} \\ -6.7114 \times 10^{-1} & 4.6856 \times 10^{-2} & 9.0805 \\ -2.6476 \times 10^{-1} & -9.0821 & 4.8590 \times 10^{-2} \end{bmatrix} \quad (4-27)$$

The essential matrix can be decomposed into a rotation matrix (R) and a translation vector (t) as discussed in Section 4.4.2. Given the rotation matrix and translation vector, the projection matrix for the left (M_l) and right (M_r) cameras can be found as described in Equation (4-28) and (4-29) (Hartley and Zisserman, 2003). Where I is a 3×3 identity matrix and 0 is a null 3- vector.

$$M_l = K_l[I|0] \quad (4-28)$$

$$M_l = \begin{matrix} 509.3683 & 0.0000 & 305.7052 & 0 \\ 0.0000 & 509.3682 & 242.1296 & 0 \\ 0.0000 & 0.0000 & 1.0000 & 0 \end{matrix} \quad (4-29)$$

$$M_r = K_r[R|t] \quad (4-30)$$

$$M_r = \begin{matrix} 509.3682 & 0.0000 & 258.3306 & -4.6404 \\ 0.0000 & 509.3682 & 245.1208 & 0 \\ 0.0000 & 0.0000 & 1.0000 & 0 \end{matrix} \quad (4-31)$$

Finally, given the calculated projection matrix M_l and M_r and a pair of corresponding image points, p_l and p_r the 3D coordinate of an image point can be found via the linear triangulation method as seen in Section 4.7.

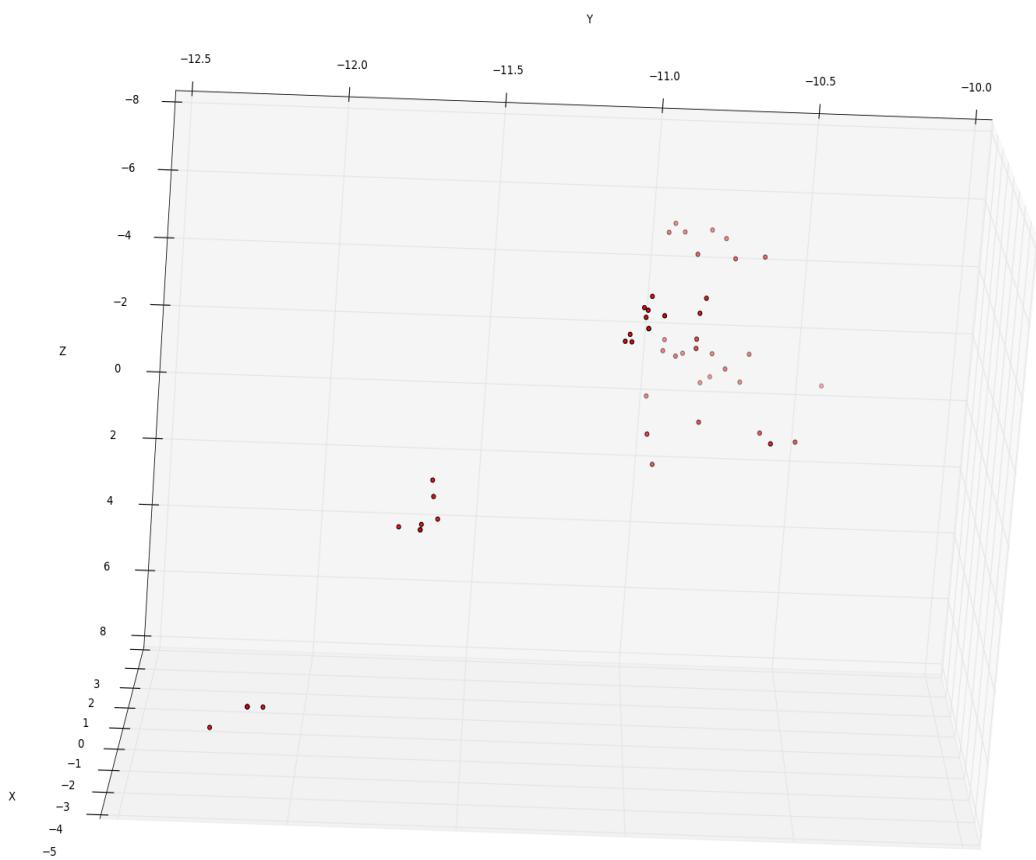


Figure 4-20: 3D point cloud generated using SIFT feature based matching

A single 3D point was generated for each pair of image points (see Figure 4-20). The resulting point cloud generated by the SIFT feature based technique is quite sparse. Fifty-six 3D points were generated in total from the image pair acquired by the stereo cameras. From the results shown here, it can be concluded that matching by extracting only feature points results in a sparse 3D point cloud of the region of interest. This conclusion is in strong agreement with work done by other authors (Li et al., 2015, Shao et al., 2016, De Cubber et al., 2008).

4.8.2 3D reconstruction using correlation-based method

Feature based 3D reconstruction produces sparse 3D point clouds because they match only a sparse set of points in the images as discussed in the previous section. In this section, correlation based matching is performed in an attempt to find a corresponding match in the right image for every point in the left image. The overall workflow is shown in Figure 4-21 below.

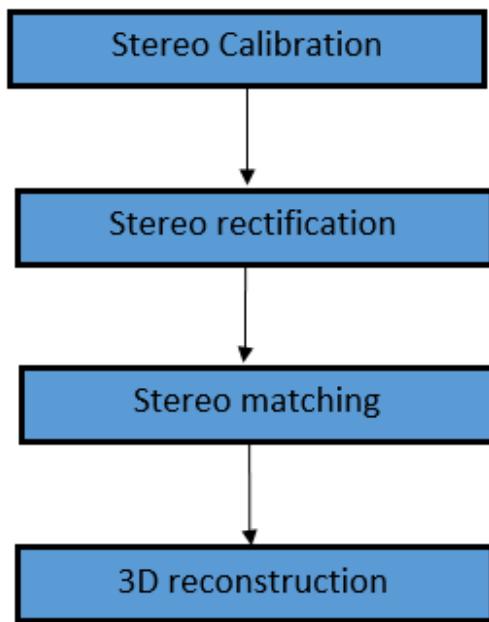


Figure 4-21: 3D reconstruction workflow for correlation based matching

The derived parameters from the calibrated camera pair were used for image rectification as discussed in Section 4.5. With rectified images, the epipolar lines of the corresponding points should lie in the single row of their images. Before applying the rectification matrices, the images were undistorted using the distortion coefficients calculated during camera calibration. Figure 4-22 below shows an example of a rectified stereo pair.



Figure 4-22: Rectified images of stereo pair

After rectification, correlation based matching was run on the image set in order to compute a disparity value for each pixel. Computing the disparity between the left and right images for each pixel led to a disparity map. Figure 4-23 below shows the resulting disparity maps obtained from the Sum of Absolute Differences (SAD) and Semi Global Matching (SGM) correlation based methods.



Figure 4-23: Disparity map acquired from images – BM (left) and SGBM (right)

As seen in Figure 4-23 the SGM technique produces a denser, smoother disparity map with fewer holes in comparison to BM technique.

After disparity map generation, 3D reconstruction was performed, again using the linear triangulation technique. The reconstruction was performed point by point for each valid entry in the disparity map to produce the 3D point clouds.

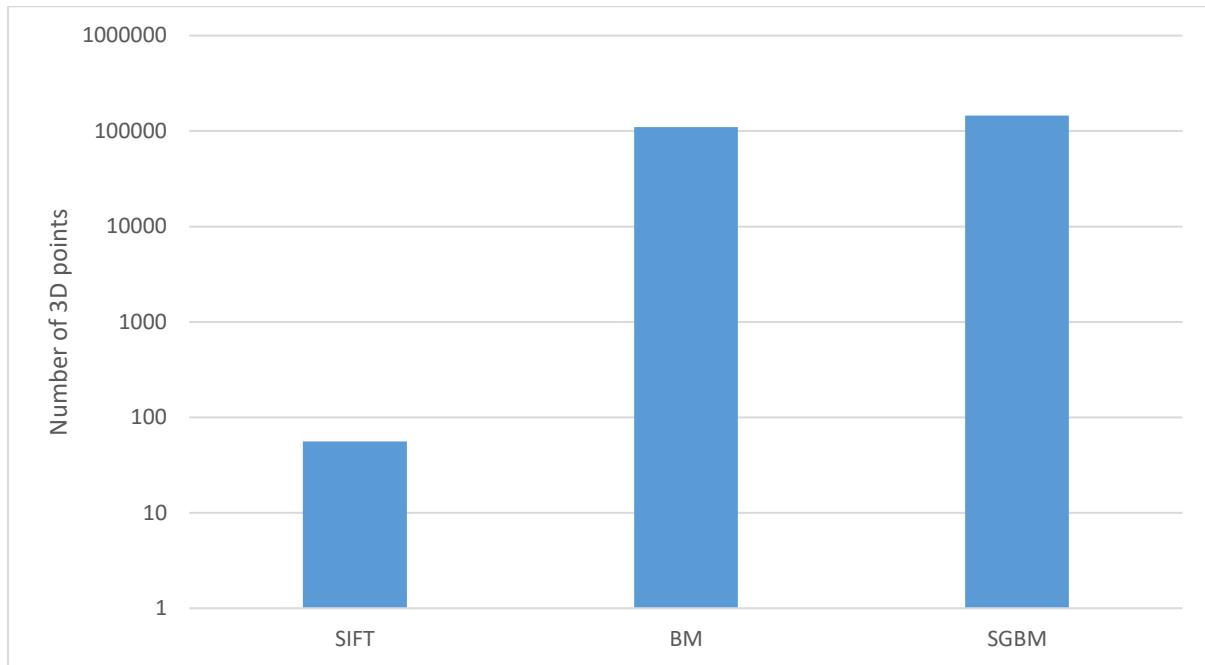


Figure 4-24: Number of 3D points obtained using feature and correlation based 3D methods

Figure 4-24 above shows the resulting number of 3D points obtained from the various matching techniques investigated. The SGM produced the highest number of 3D points (145,445) in comparison to the BM and SIFT methods from the image pair acquired from the stereo cameras. The SGM technique successfully captured the contours and edges of the bone structure. In order to obtain dense 3D reconstruction of a surface, the correlation-based methods, in particular the SGM method, prove to be a reliable matching technique. Figure 4-25 below shows a visual representation of the resulting 3D point clouds obtained from the BM and SGM methods.

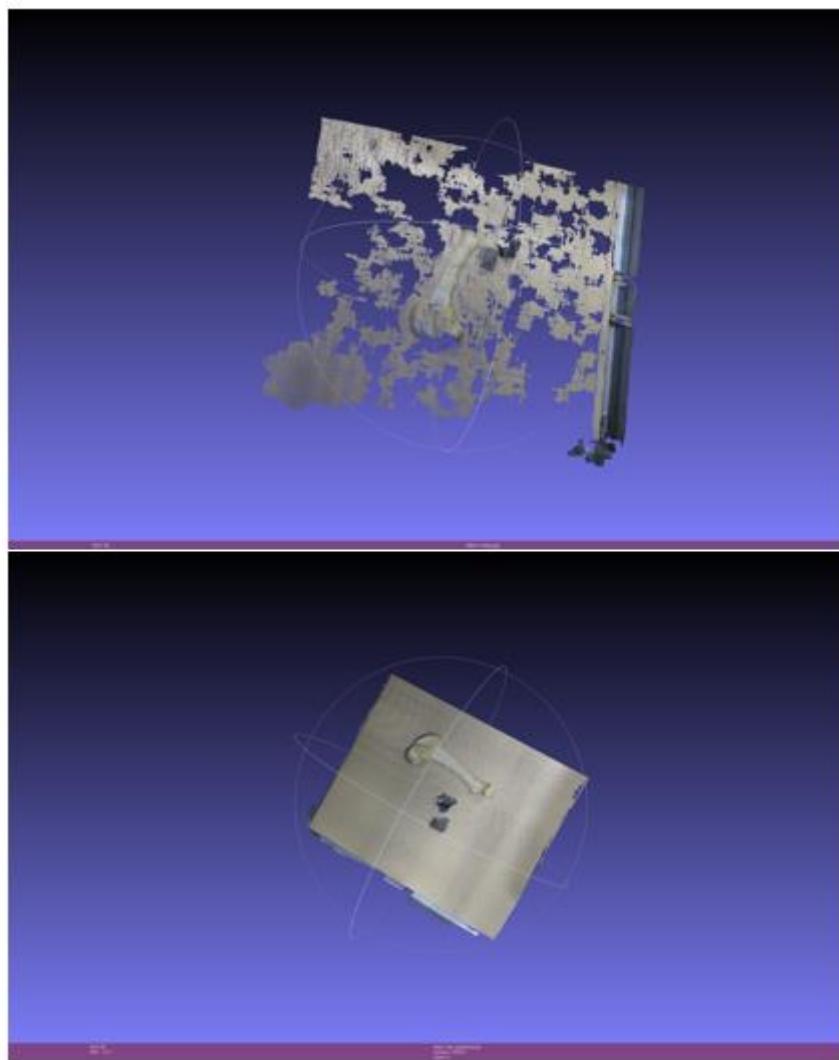


Figure 4-25: generated Point cloud using BM (top) and SGBM (bottom)

As seen in Figure 4-25 the SGM produces a denser and less noisy point cloud in comparison to BM. This is due to SGM producing a better quality disparity map in comparison to BM.

4.9 Summary

By determining the intrinsic and extrinsic parameters of a stereo camera obtained through camera calibration this chapter reveals how a 3D point can be found from a pair of image points. The quality of the 3D reconstruction is strongly dependent on the correspondence algorithm applied. Many approaches exist in literature to tackle the correspondence problem in stereovision. This chapter has discussed the popular techniques that have been developed, outlining their advantages and disadvantages. The performance of the various techniques have been evaluated and preliminary results show that the feature based matching techniques produce sparse point clouds. The SGM method allowed generation of a dense 3D reconstruction in comparison to the other stereo matching algorithms investigated. The next chapter focuses on 3D reconstruction of a water surface and wave height estimation from the reconstructed 3D data.

Chapter 5: Water surface reconstruction and analysis of laboratory Generated water waves

5.1 Introduction

This chapter focuses on 3D reconstruction of a water surface, due to the dynamic nature of the area of interest both feature based and correlation-based methods are investigated. For wave height estimation, waves have been generated in a wave tank and a method for estimating the wave height from the resulting 3D point cloud data is presented. Finally, an accuracy assessment of the estimated wave height is undertaken in time and frequency domain.

5.2 3D Reconstruction of water waves

To further validate the claim that the feature based matching methods produce sparse point clouds as discussed in the last chapter, the various matching techniques are once again performed on a water surface. An investigation into how the baseline affects the reconstruction accuracy is also undertaken.

5.2.1 3D reconstruction of still water surface

Figure 5-1 shows the experimental setup for image capture of the water surface with the stereo cameras.

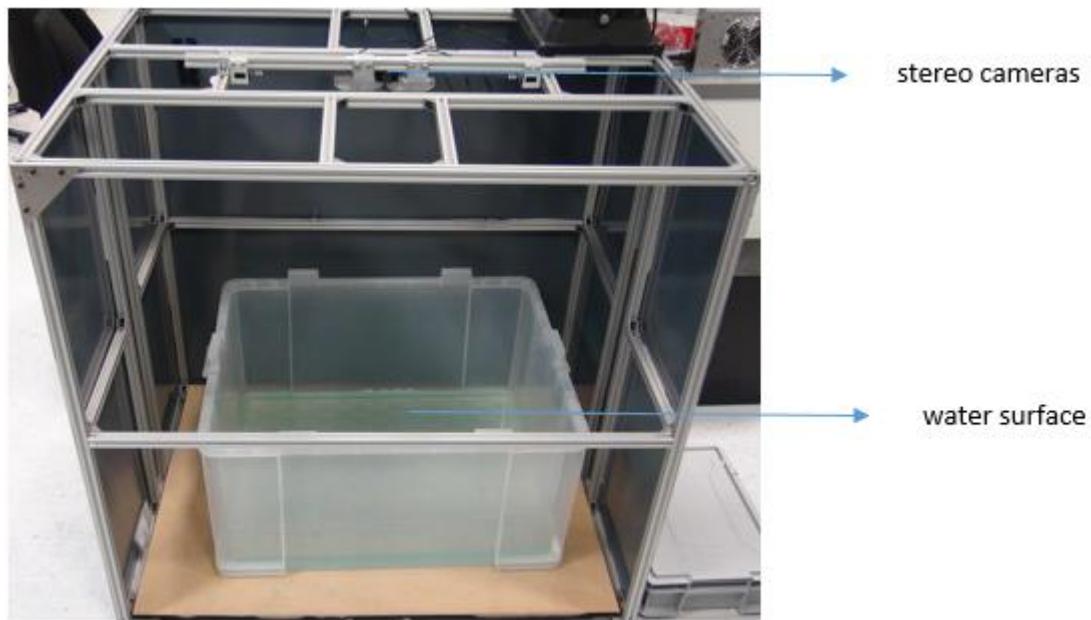


Figure 5-1: stereo cameras capturing clear water surface

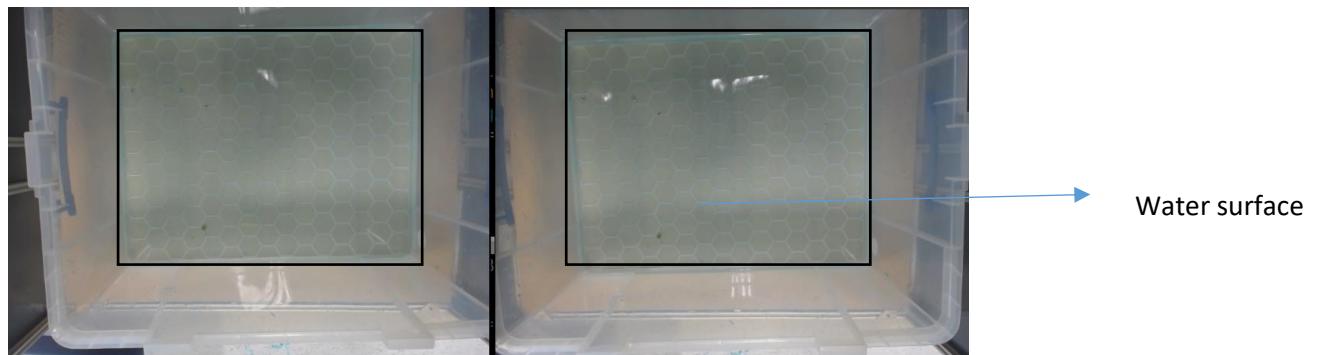


Figure 5-2: image pair captured by stereo cameras

Keypoints were extracted from the image pair captured with the stereo cameras using the SIFT, SURF and ORB techniques. Table 5-1 below shows the results of the number of keypoints detected.

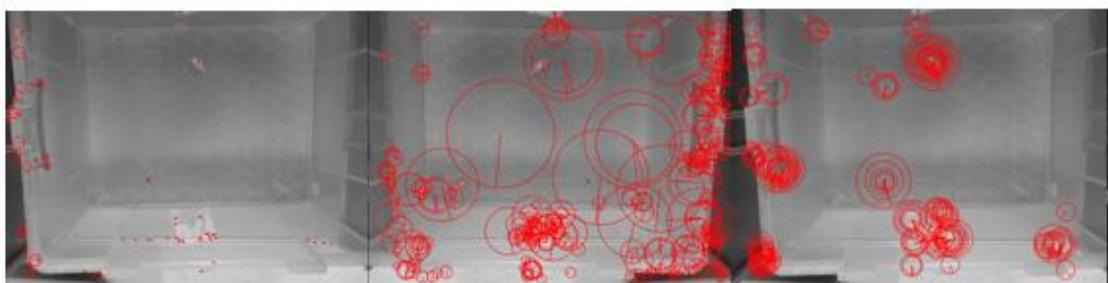


Figure 5-3: keypoints detected on clear water surface SIFT (left), SURF (middle) and ORB (right)

Table 5-1: Number of keypoints detected on clear water surface using various feature detection techniques

	Number of keypoints detected in left image	Number of Keypoints detected in right image
SIFT	100	100
SURF	282	243
ORB	131	210

Bar plots of the results presented in Table 5-1 for the left and right images are shown in Figure 5-4.

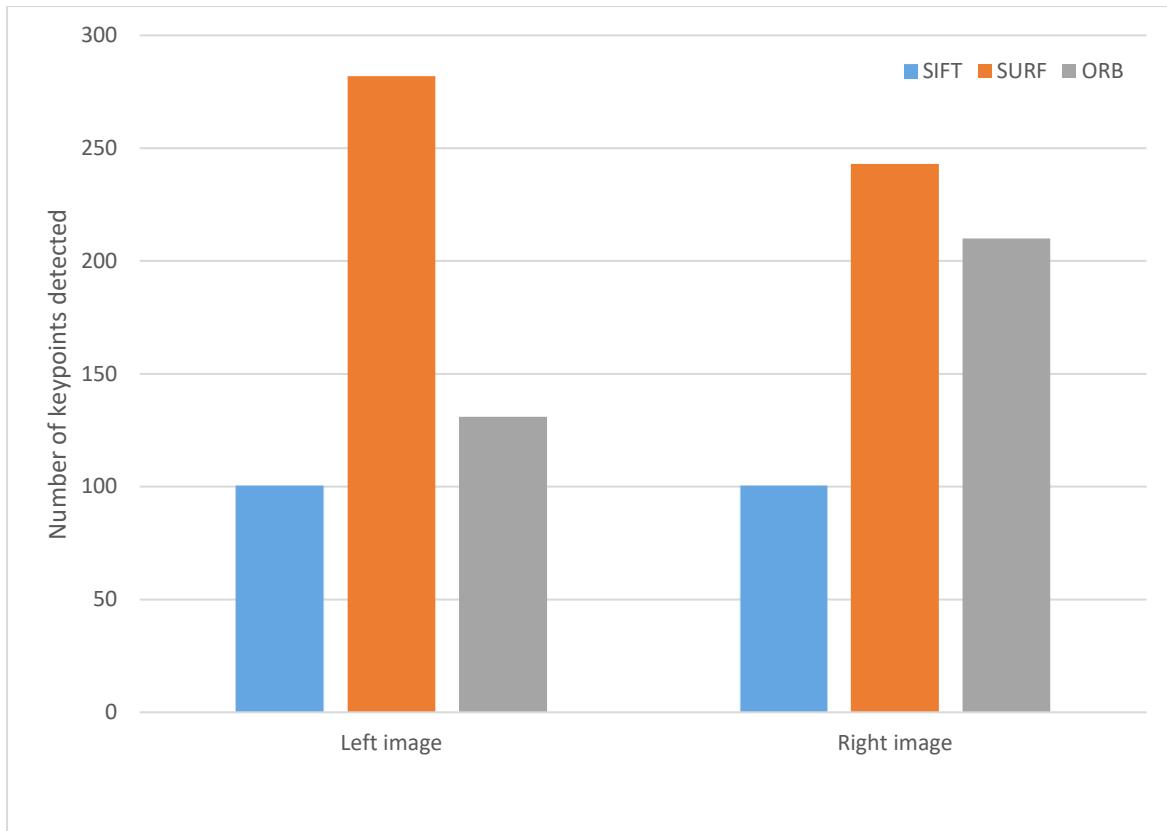


Figure 5-4: Number of keypoints detected on left image using SIFT, SURF and ORB

In the left image, SURF produced the highest number of keypoints, 282 followed by ORB 131 and lastly SIFT 100. The number of keypoints detected in the right image followed the same trend as the left image. Again, SURF produced the highest number of keypoints 243, followed by ORB, which produced 210 and lastly SIFT which produced 100.

RANSAC filtering was then applied to remove outliers. Table 5-2 below shows the detected number of keypoints retained after filtering using the RANSAC algorithm.

Table 5-2: Number of keypoints detected in left and right images after RANSAC filtering

	Number of keypoints detected in left image	Number of Keypoints detected in right image
SIFT	9	9
SURF	9	9
ORB	10	10

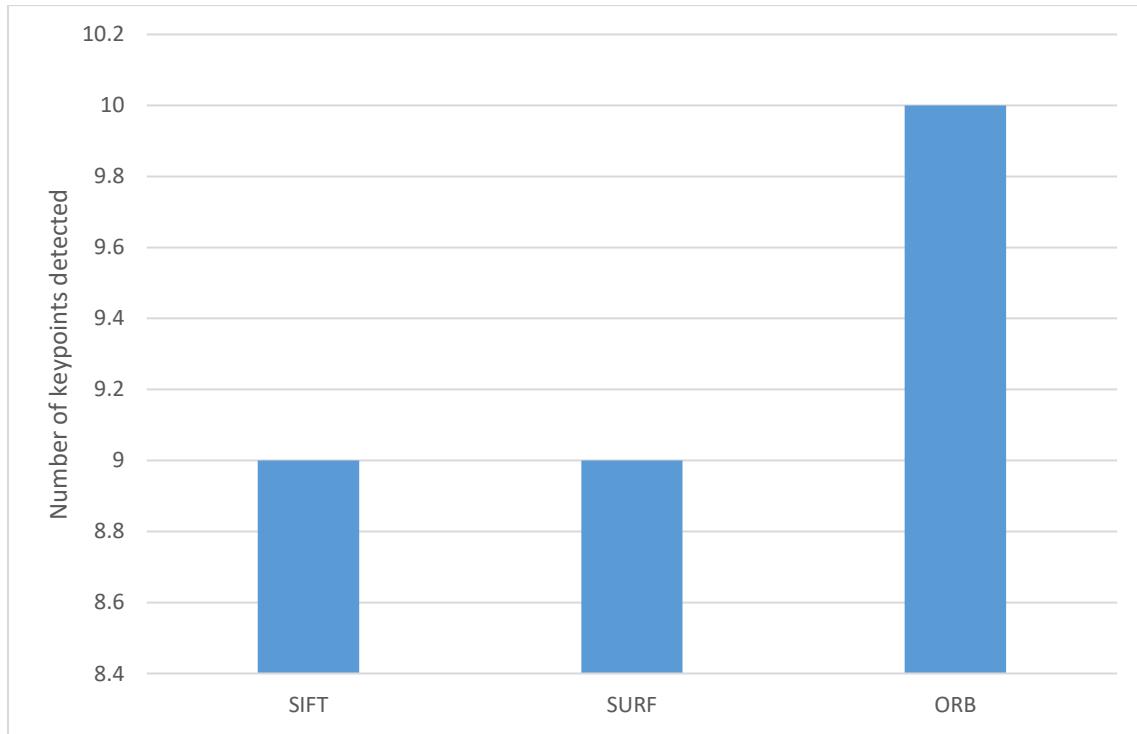


Figure 5-5: Number of keypoints detected on clear water surface after RANSAC filtering

Figure 5-5 shows a bar plot of the results presented in Table 5-2. ORB retained the highest number of keypoints (10) after RANSAC filtering. SIFT and SURF both retained 9 after filtering.

From the results presented in Figure 5-5, it can be concluded that the entire feature based techniques performed poorly when detecting keypoints on the clear water surface. Not enough keypoints are detected to be able to produce a dense 3D reconstruction of the water surface.

Since the feature based matching performed poorly, correlation based matching was performed using the SGM technique in an attempt to produce a dense 3D reconstruction of the water surface. The SGM method was chosen due to its better performance in comparison to the BM method as discussed in the previous chapter.



Figure 5-6: Disparity map from clear water surface

Figure 5-6 shows the resulting disparity map obtained using the SGM method. The disparity map obtained was poor with many holes. This is because of the nature of the water surface. The water surface is transparent by nature and texture-less. As a result, it proved very difficult to perform stereo matching on a water surface. Thus, a major challenge remains providing enough distinguishable surface textures on a water surface to aid reconstruction. Section 5.2.2 provides a solution to this problem when producing a 3D point cloud of the waves in a wave tank.

5.2.1.1 *Investigating the effect of stereo-baseline on reconstruction accuracy*

In this section, an investigation into how the camera baseline affects the reconstruction accuracy is undertaken in order to identify a suitable baseline for 3D reconstruction of waves in the wave flume. Flat-water surfaces are planar in nature. Hence it is reasonable to assume that the best plane fitting the point cloud represents the Mean Water Level (MWL) for that image acquisition (Corgnati et al., 2015).

The experiments undertaken assess the reconstruction accuracy of a planar surface. A planar surface is used in this experiment since it best represents the MWL as discussed in the previous paragraph.

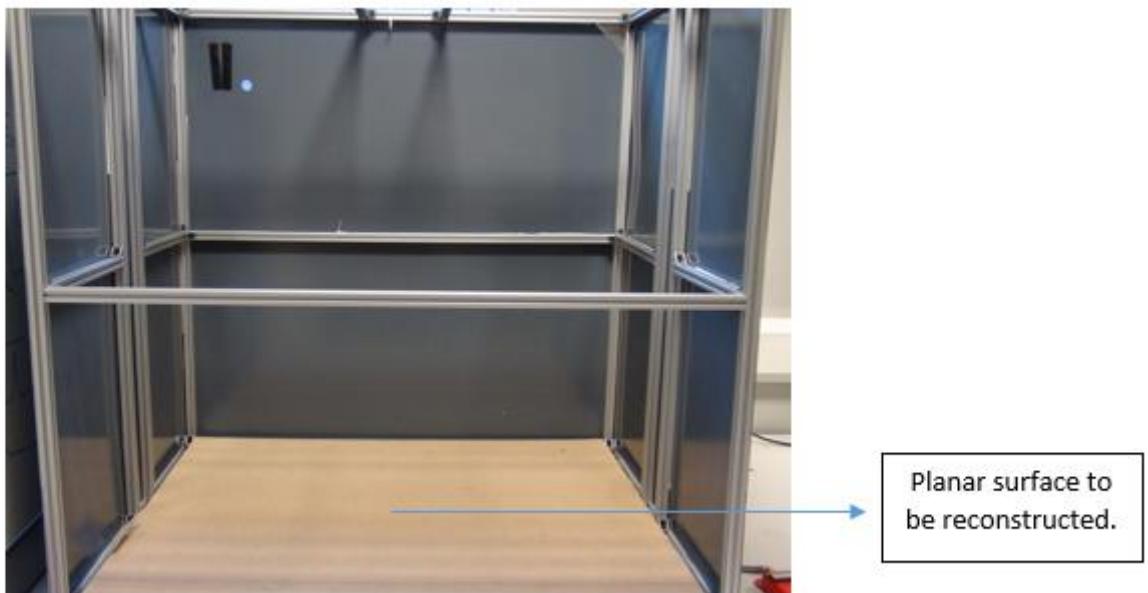


Figure 5-7: planar surface

The SGM technique was used for computing the disparity map of the planar surface. Figure 5-8 shows the computed disparity map of the planar surface and the corresponding 3D point cloud is shown Figure 5-9.



Figure 5-8: disparity map of planar surface

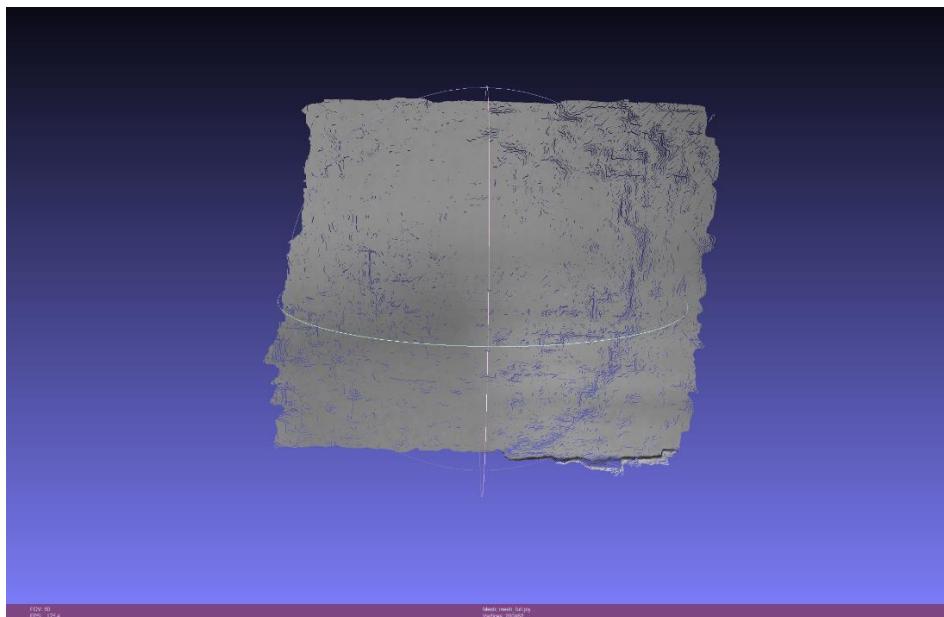


Figure 5-9: reconstructed planar surface

The reconstruction accuracy of the reconstructed planar surface was assessed at various baselines between the cameras. Due to the physical constraints imposed by the camera housing, the minimum separation distance investigated is 9.2cm. Baselines of 12 and 15cm are also investigated to identify a suitable stereo baseline that can be mounted on the UAV without causing stability issues during flight.

Camera calibration has been done for each baseline tested. For each calibrated stereo camera pair, measurements were made at a fixed distance of 50cm in order to accommodate for the minimum working range of the stereo cameras. It was observed that below this distance, the resulting disparity maps were too poor to obtain any decent reconstruction of the planar surface.

The stereo cameras were configured to capture stereo frames for 28 seconds in total. For each image pair captured, the resulting 3D points were averaged across the surface. Figure 5-10 below shows the results for the various baseline configurations tested.

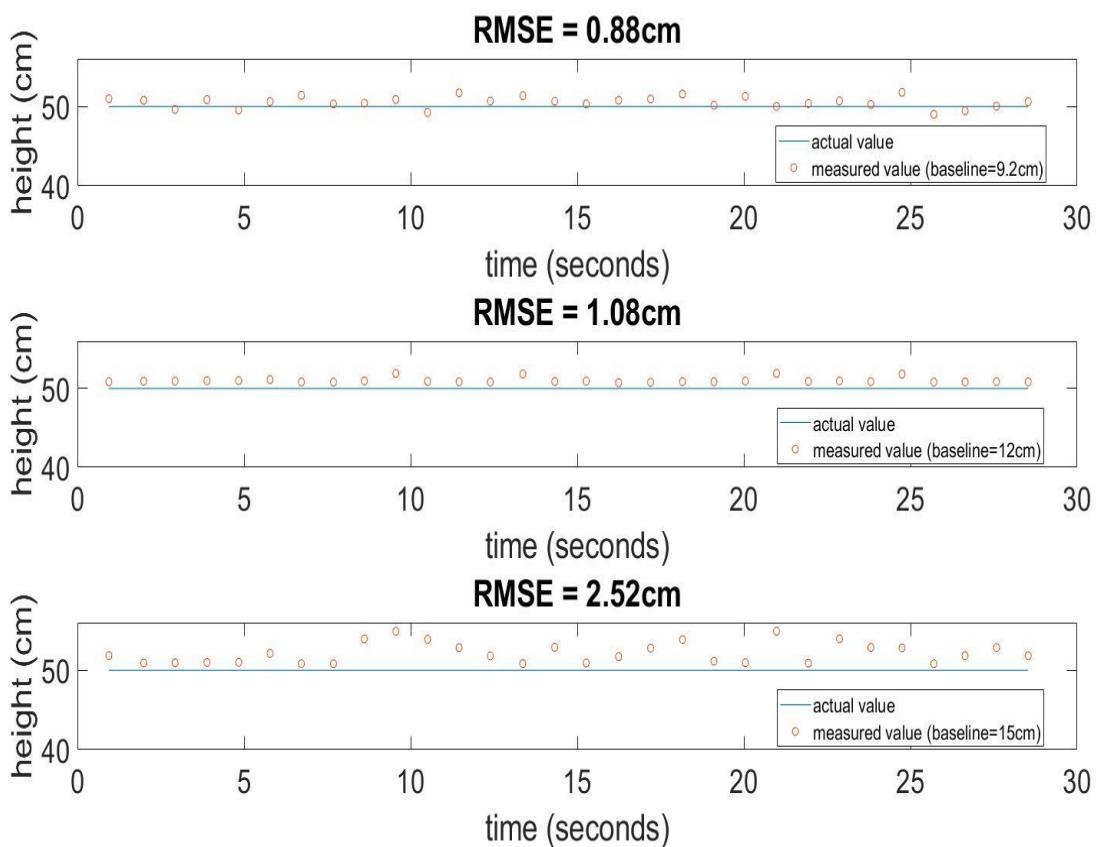


Figure 5-10: Measurements performed for various baseline configurations. Top (baseline = 9.2cm), middle (baseline = 12cm) and bottom (baseline = 15cm).

It is evident from Figure 5-10 that the measurements follow quite well with the actual values for the 9.2cm baseline configuration (Root Mean Square Error (RMSE) = 0.88cm). For a fixed distance, as the baseline increases, the RMSE between the measured and actual value also increases (RMSE for the 12cm and 15 cm baseline configuration are 1.08cm and 2.52cm respectively). For 3D reconstruction of the waves in the wave flume, the 9.2cm baseline configuration will be used because it produces the lowest RMSE in these experiments.

The stereo matching process is easier with small baseline configurations than larger because the two scenes are very similar i.e. less occluded areas (Delon and Roug  , 2007). However, small baseline configurations work over smaller ranges in comparisons to larger baseline configurations (Pritchett and Zisserman, 1998, Delon and Roug  , 2007). Since the ultimate goal would be to mount the cameras on a UAV for wave height detection this is not an issue. The UAV can simply be flown to an altitude that is within the working range of the stereo cameras. The maximum working range for the 9.2cm baseline configuration will be investigated in the next section.

5.2.2 Wave height estimation

In this section, waves have been generated in a laboratory wave flume. The length, width and height (m) of the flume is $5.0 \times 0.30 \times 0.6$ respectively. In order to generate the waves in the wave flume, a displacer is moved up and down by an adjustable crank drive. The height of the waves generated in the flume is varied by changing the lift of the displacer. The speed of the motor sets the frequency of the waves. The stereo cameras were used for estimating the wave height at a range of frequencies. The cameras were rigidly mounted off a metal rod attached to an extendable tripod (see Figure 5-11 below). The extendable tripod was used in order to vary the height of the cameras above the water surface during the experiments. The cameras were

positioned facing downwards in order to capture the wave as it propagates through the wave flume.



Figure 5-11: Stereo camera setup for wave reconstruction

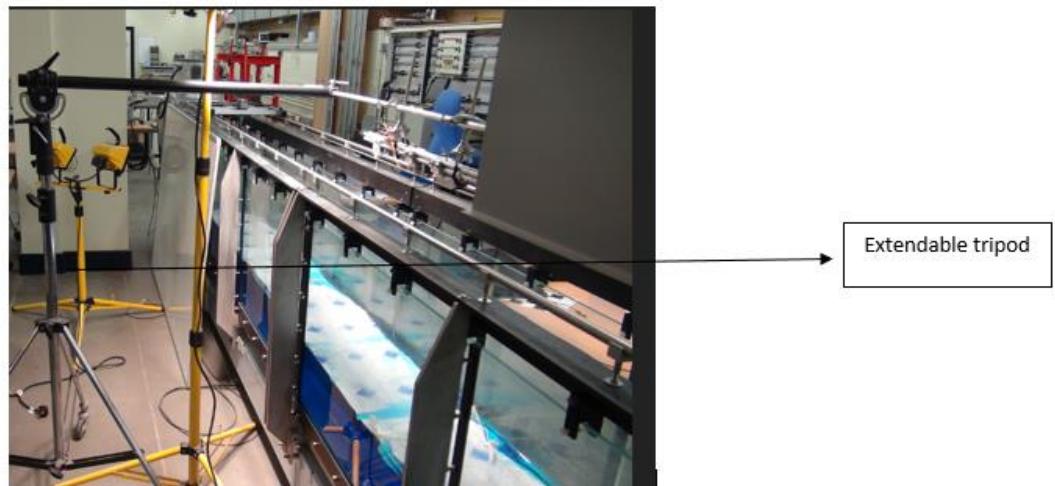


Figure 5-12: Adjustable tripod stand

Unlike most naturally occurring surfaces, the water surface in a wave flume or tank presents a smooth glass surface. From an image-processing standpoint, these waves pose an unusual challenge; both the reflected ceiling and floor of the wave tank are visible in the acquired image. Hence, images of such waves presents one or more haloes of light due to water transparency and reflections. Early attempts to try to

mitigate this issue utilized adding a coloured dye to the water surface to reduce the transparency and reflections. This however proved unsuccessful.

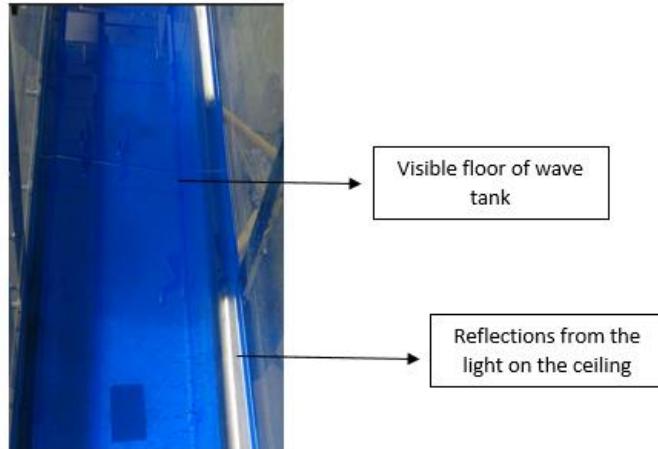


Figure 5-13: Coloured water surface

Researchers have managed to solve this problem by seeding the water surface with floating particles (Perelman et al., 2011). The seeding of particles, accompanied by a careful lighting of the free surface resulted in the successful identification of corresponding points with the stereo matching algorithm. However, the seeding technique presented some limitations. After a while, the particles were agglomerating in small packets, resulting in inhomogeneous distributions. Piepmeyer and Waters (2004) managed to reconstruct the free surface of water waves in a wave tank using a commercial stereovision system at the Davidson Lab at the Stevens Institute of Technology (New Jersey, USA) by texturizing the water surface with a fine mist, reducing the problems of reflection and transparency. The wave heights estimated were however much lower than those measured by wave probes. They argued this could be due to the dampening effect of spraying the water.

To avoid the limitations discussed above, the approach taken in this thesis was addition of a thin flexible opaque film on the water surface (see Figure 5-14). A thin

and flexible material was used to avoid dampening the wave as it propagates through the wave flume.

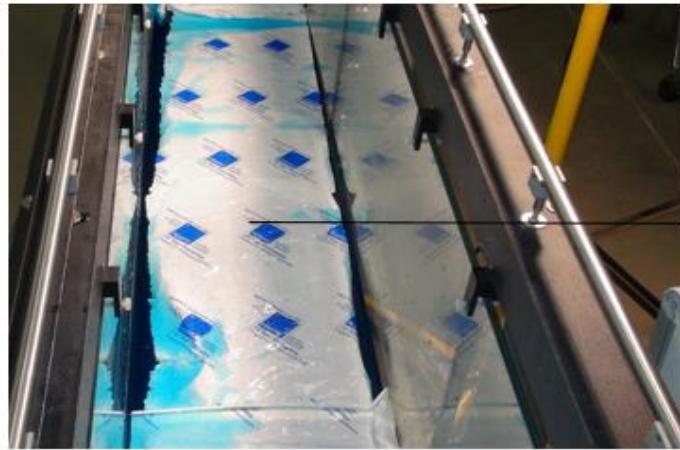


Figure 5-14: Thin film added to assist 3D reconstruction

Such a set-up prevented light penetration and in addition provided enough visible features to allow a successful stereo matching between the left and right images acquired by the stereo cameras.

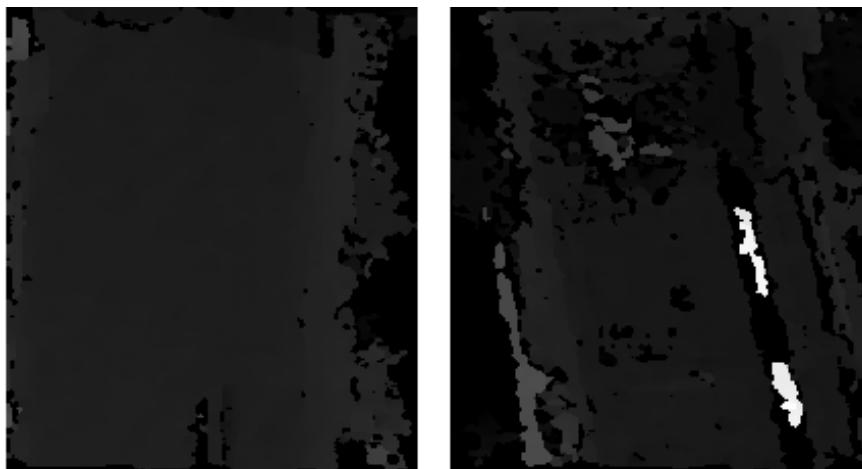


Figure 5-15: disparity map after applying opaque sheet on water surface (left) before applying opaque sheet (right)

As seen in Figure 5-15, the addition of an opaque sheet led to a denser smoother disparity map with fewer holes in comparison to the water surface without application of the opaque sheet. Once a good disparity map was obtained, the next step was projection onto a 3D point cloud.

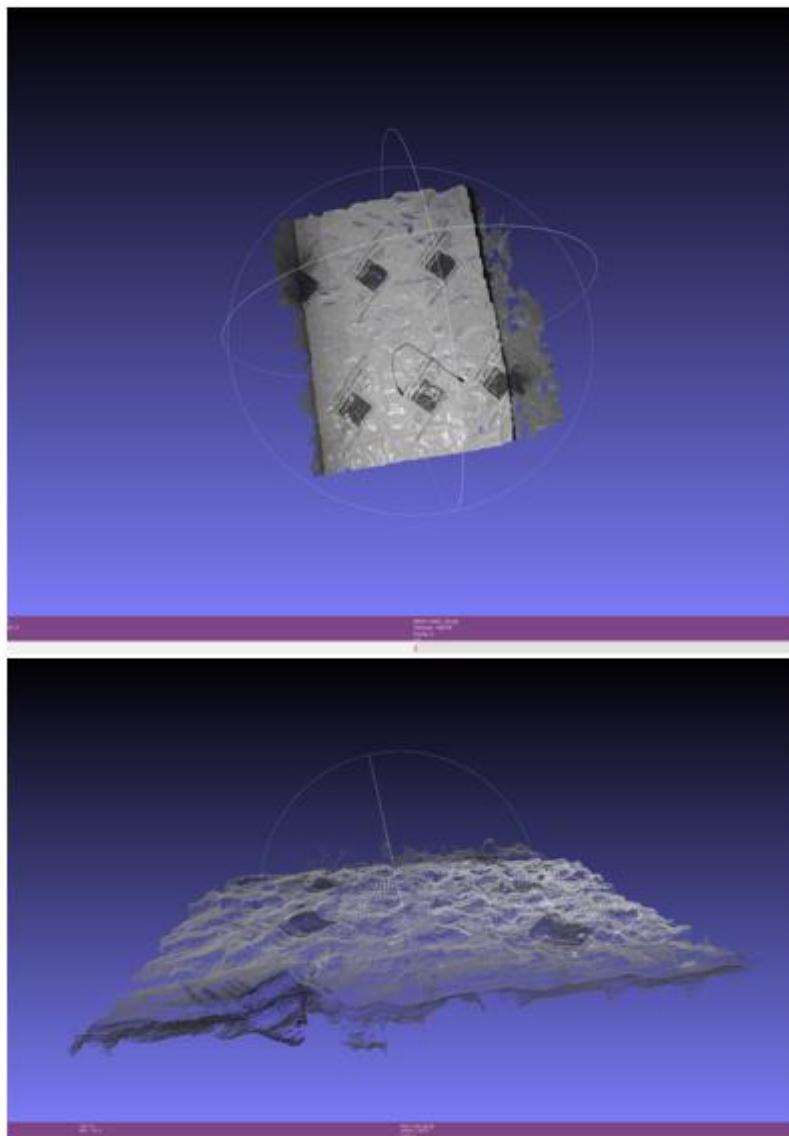


Figure 5-16: 3D reconstruction of opaque sheet. Top - top view of reconstructed point cloud and Bottom - side view of reconstructed point cloud

The result of the extracted 3D points are the continuous water surface elevations at specific time intervals. The time intervals are governed by the overall processing time i.e. time elapsed from grabbing a pair of images from the stereo-cameras, rectification, stereo matching and 3D reconstruction. The resolution at which the stereo images were captured has a significant effect on the overall processing time (see Figure 5-17).

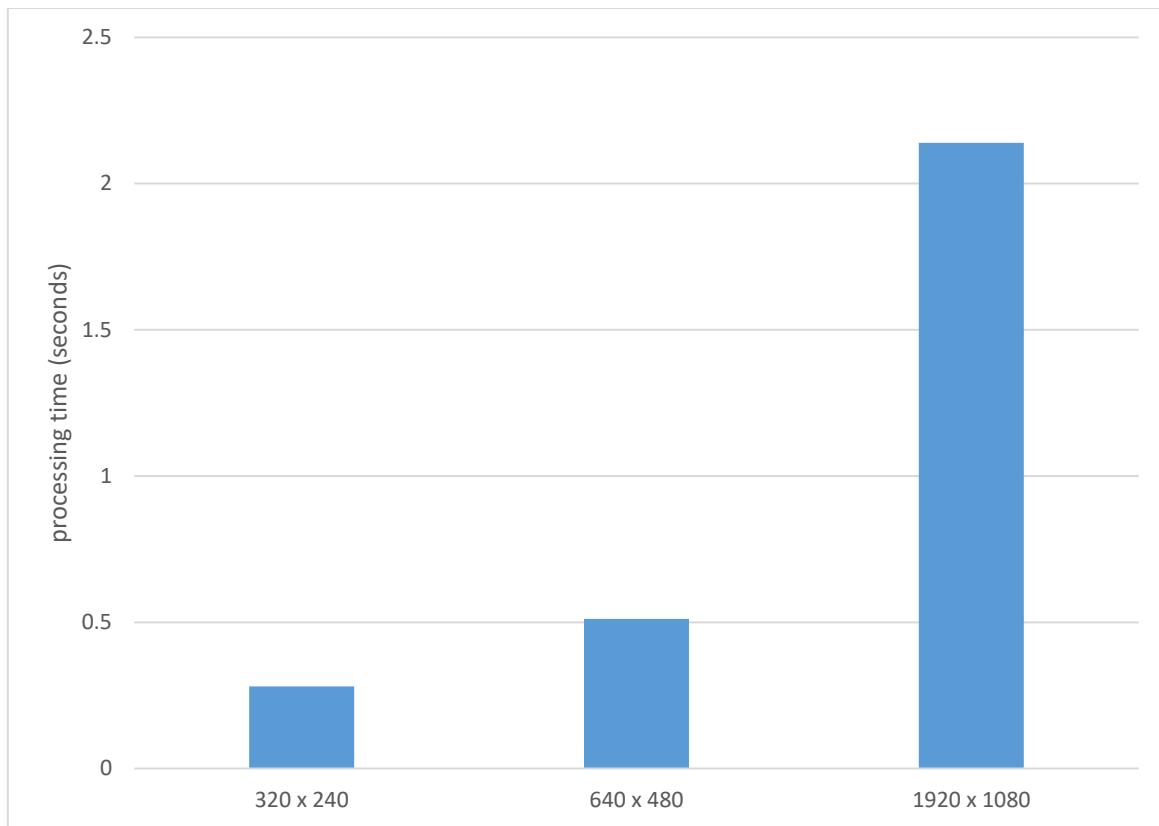


Figure 5-17: Processing time comparison for various image resolutions

Figure 5-17 shows the overall processing time taken for various image resolutions. The average processing time taken for 320×240 , 640×480 and 1920×1080 pixels was 0.281, 0.512 and 2.140 seconds respectively. The increased computation time associated with higher resolution images makes it very challenging for data capture from a UAV since they possess a short battery life span. As a result, the 320×240 resolution has been chosen as the preferred image resolution for wave height estimation.

Zero up-crossing is a common technique used for wave analysis (Viriyakijja and Chinnarasri, 2015, Kamphuis, 2000). The relation of 3D points with time intervals was used to obtain the wave height using this technique. The process of zero up crossing can be described as follows:

- Define a reference line (zero-line) that passes from a negative to positive position of water surface elevation and intercepts with the reference line. Each individual reconstructed point is subtracted from the mean of the data (detrending) so that the measurements are centred around 0.
- The wave height of the first loop is the maximum difference of water surface elevation between the crest and the trough.
- For each wave loop, the previous step can be carried out and the results are averaged in order to find the average wave height for a given time series.

The first series of experiments involved utilising the stereo cameras to capture a still water surface (without waves). Under such conditions, the estimated wave height should be zero. Wave height estimation under these conditions was done using the Zero up crossing technique discussed in the previous paragraph. For example Figure 5-18 , shows the raw readings from the stereo cameras and Figure 5-19 shows the detected crest and trough after the Zero up-crossing technique has been applied. The acquired 3D points were averaged across the entire image captured from the stereo cameras.

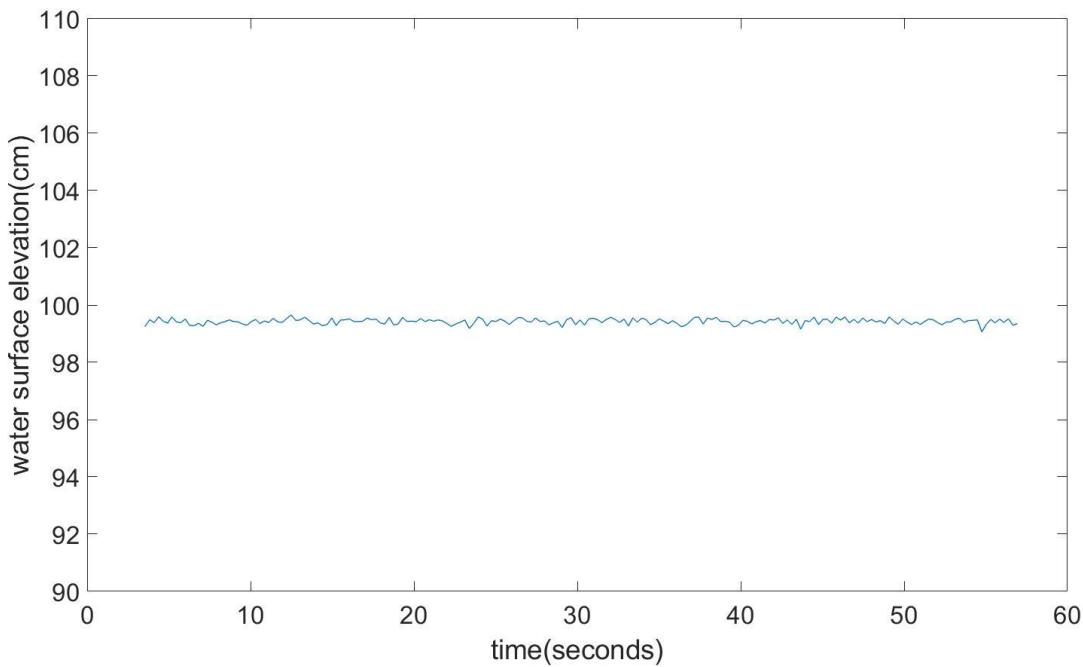


Figure 5-18: Raw readings from stereo cameras before applying zero up crossing analysis. Cameras were placed at distance 99cm above the water surface.

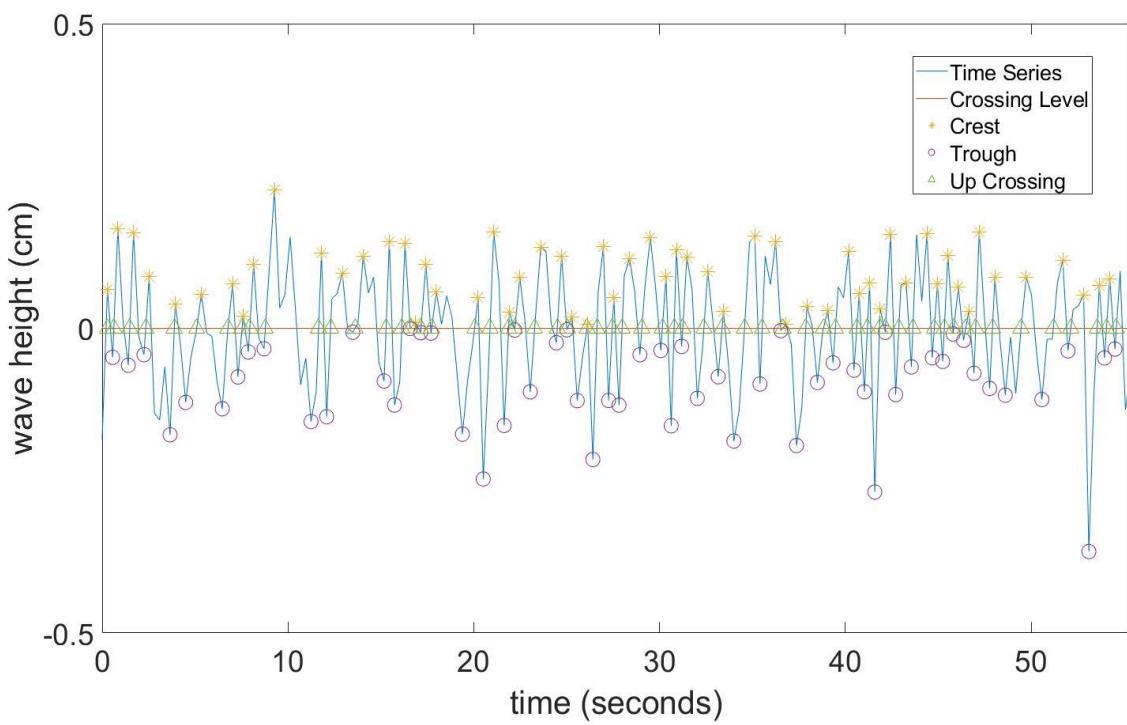


Figure 5-19: Zero up crossing analysis to detect trough and crest

As seen in Figure 5-19, by identifying the crest and the trough of each wave along the reference line, the wave height of each individual wave can be estimated. The average wave height can be estimated by taking the average height of all the waves present in the data captured. Next, measurements were made at various distances ranging from 80cm – 142cm. A minimum distance of 80cm was used in order to accommodate the physical constraints imposed by the experimental setup. Experiments were not undertaken for distances greater than 142cm, because the cameras could no longer capture sufficient area of the free surface in the wave flume due to narrow width of the flume. The aim of this experiment is to investigate how mounting the cameras at various distances above the water surface affected the accuracy of the wave height estimation within the testing parameters. The distances investigated are 80cm, 99cm, 122cm and 142cm. For each distance investigated, three repetitions were made and the results were averaged.

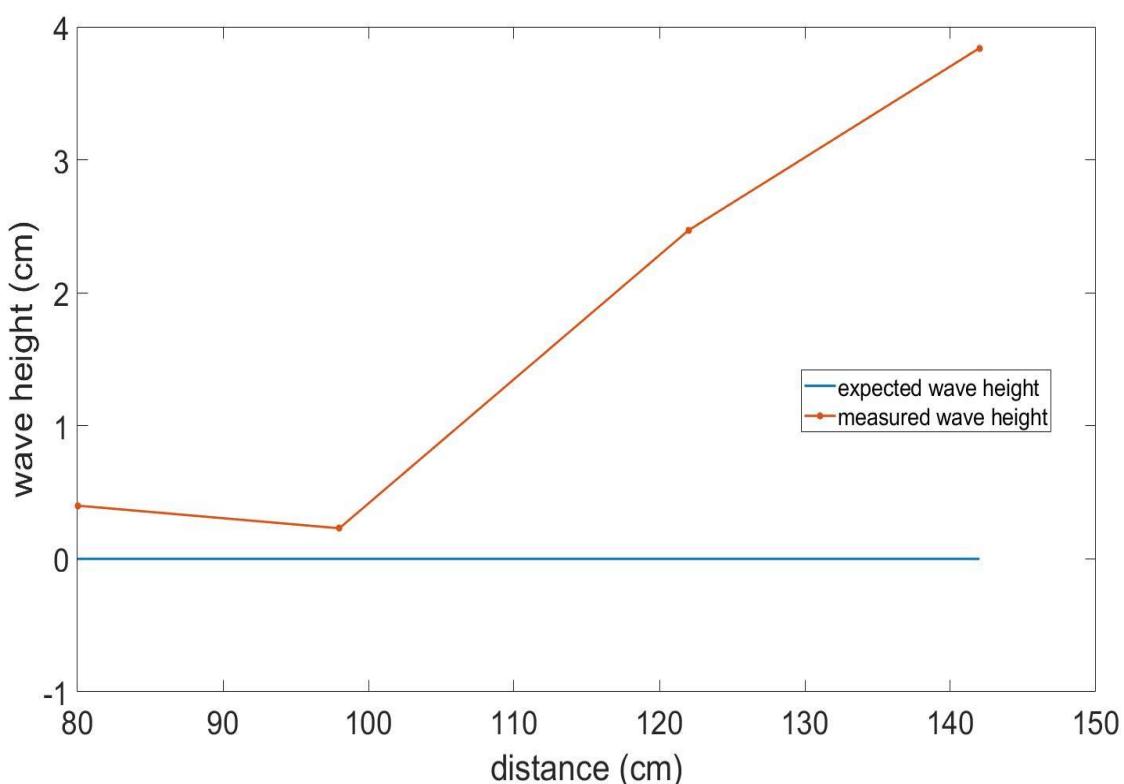


Figure 5-20: average wave height estimated at various distances

Figure 5-20 shows the estimated average wave height at various distances investigated above the water surface. Having the stereo cameras at a distance of 99cm above the water surface produced the lowest absolute error in comparison to the other distance configurations. This suggests that a fixed baseline has its own peak spot where it performs best at a specific distance. This is in strong agreement with the work demonstrated by Boonsuk (2016). The absolute error obtained was 2.3mm.

For the next series of experiments, the stereo cameras were positioned at a fixed distance and waves were generated at various frequencies in the wave tank. The frequencies investigated are 0.16Hz, 0.20Hz and 0.30Hz in order to model possible frequencies that can be observed for wind-generated sea waves as shown by Ferretti (Ferretti et al., 2013). The cameras were mounted at a distance of 99cm above the water surface since it produced the lowest absolute error as discussed in the previous paragraph. The stereo cameras captured two hundred image pairs. For each image pair, the disparity map was estimated and a 3D point cloud generated. A robust plane was then fitted to each 3D point cloud to obtain the MWL for that image acquisition. The resulting 3D points were averaged across the surface. The total execution time taken to run this process for all the frames gathered was approximately 58 seconds. The average wave height for the 3D points was estimated using the Zero up-crossing technique. The results were compared to measurements obtained from a HC-RS04 ultrasonic sensor.

To obtain ground truth i.e. actual wave height, measurements were taken from a wave probe (see Figure 5-21 below).

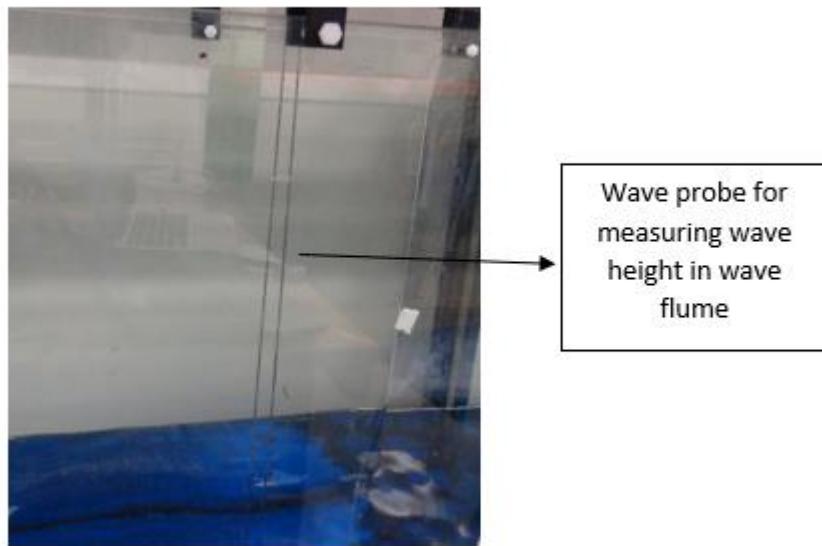


Figure 5-21: wave probe for measuring wave height

The wave probe uses two slightly separated wires that are partially submerged under water. The probe operates by measuring the current that flows between the two wires. The current is converted to an output voltage that is directly proportional to the immersed depth. The voltage can be converted to wave height by measuring the change in output voltage when the probe is raised or lowered by a known amount in still water.

The ultrasonic sensor was interfaced with an Arduino Uno to trigger the sensor. The sensor has a range of 2cm-400cm and an ultrasonic frequency of 40Hz. Readings from the Arduino board were sent via serial interface to a C++ program that captured and processed the images from the stereo cameras. The developed software received data from the Arduino at the exact time a new reading is obtained with stereo cameras to ensure synchronous readings between the sensors. The captured data were stored in a log file for further analysis.

During the course of the experiments, it was discovered that the ultrasonic sensor would occasionally produce large spikes in the data set it produces, these spikes were caused by the ultrasonic sensor taking an out-of-range reading (see Figure 5-22 below). To account for these spikes, data has been extracted only within a certain range to remove any outliers.

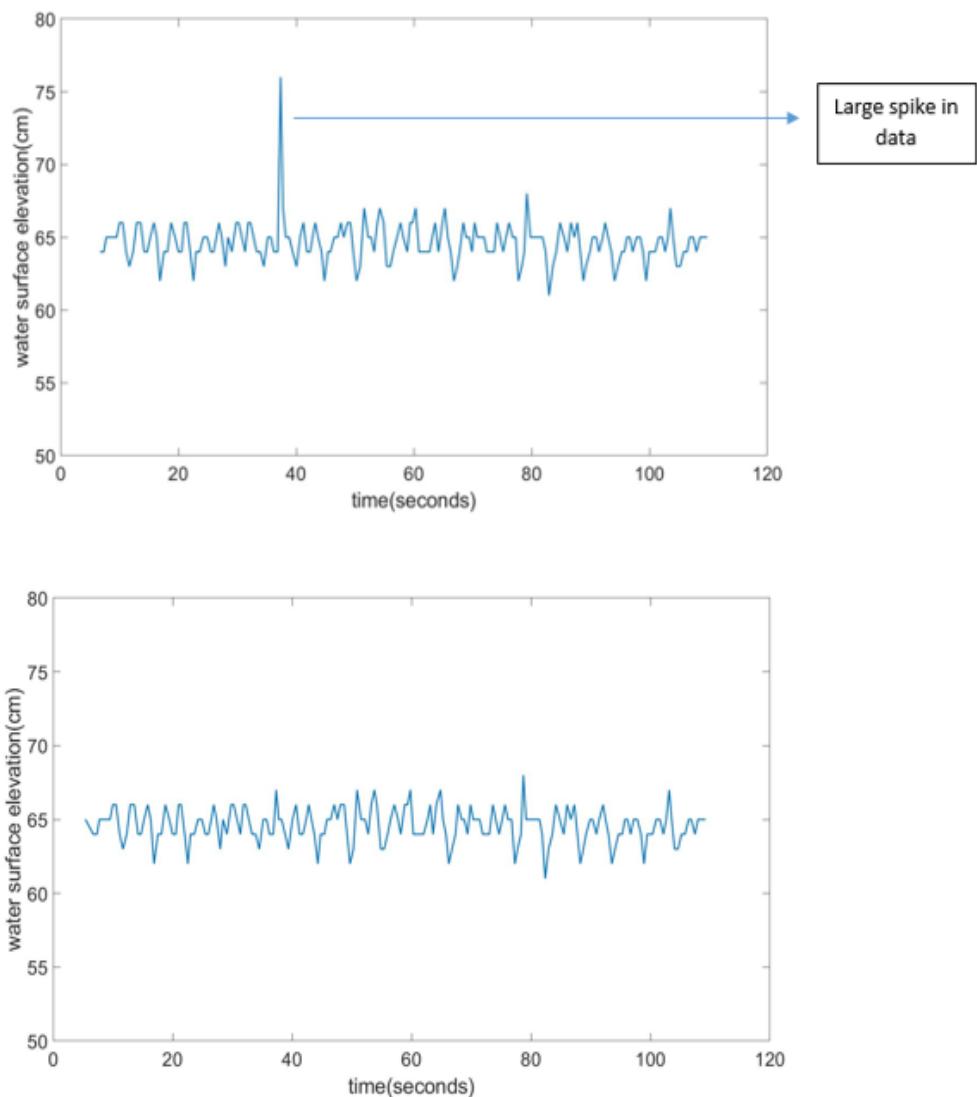


Figure 5-22: raw readings from ultrasonic sensor. Top –readings before filtering, bottom-readings after filtering noisy data

Figure 5-23 below shows the results obtained from both sensors at the various frequencies investigated.

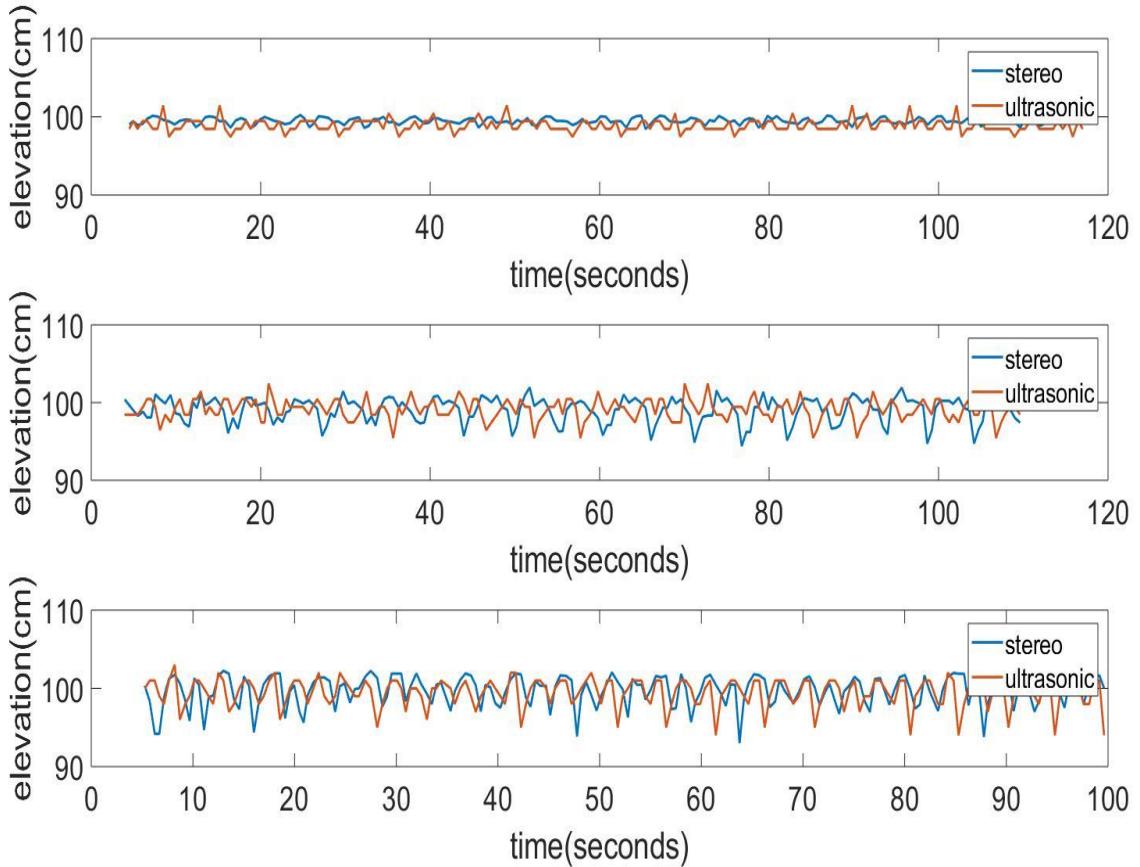


Figure 5-23: raw readings of water surface elevation obtained from stereo cameras and ultrasonic sensor. Top=0.16Hz, middle=0.20z and bottom=0.30Hz during the first repetition

Wave height is estimated as the maximum difference between the detected crest and trough using the Zero up-crossing technique. For example, Figure 5-24 shows the detected crest and trough using the stereo cameras at wave frequency of 0.16Hz.

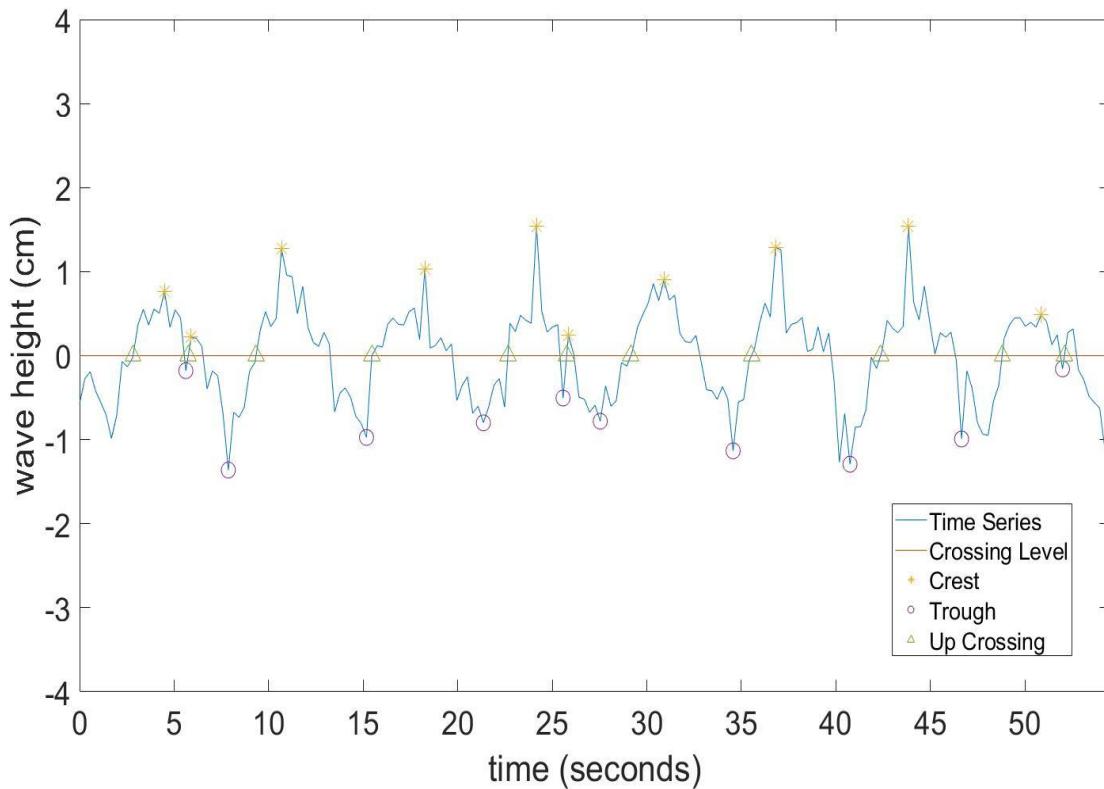


Figure 5-24: crest and trough detection of wave frequency 0.16Hz

The average wave height H_m is computed as the mean of all wave heights detected across the time series. A similar approach was applied to measurements derived from the wave probe and ultrasonic sensor. For a given frequency, three repetitions were performed on each sensor and the results were averaged.

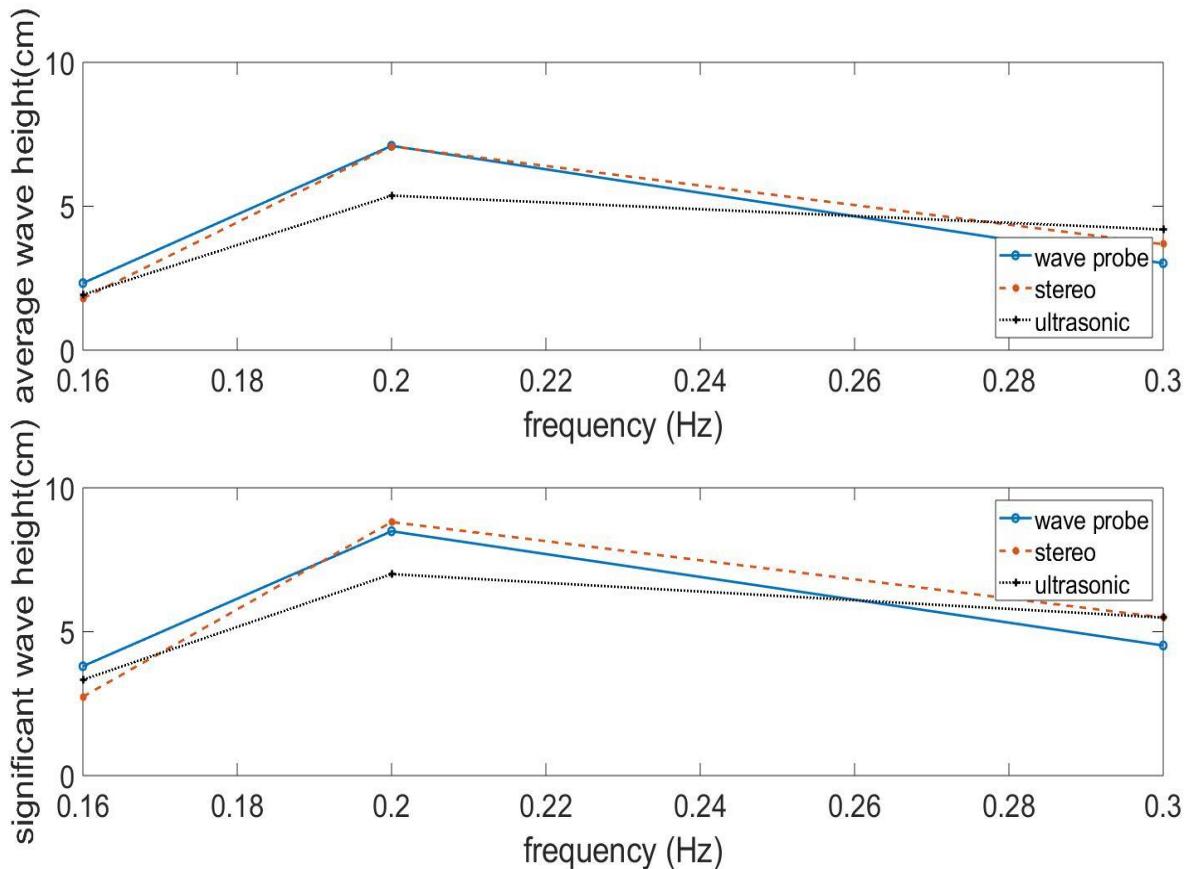


Figure 5-25: wave height estimation using various sensors. Top - average weight estimation and Bottom - significant wave height

Figure 5-25 above shows the wave height estimated by the wave probe, stereo cameras and ultrasonic sensor at the various frequencies investigated. The Significant wave height H_s is computed as the average height of the highest one-third of all waves measured (Thornton and Guza, 1983). Overall, the stereo cameras were able to take measurements that were close to the measurements taken by the wave probe. The Mean Absolute Error (MAE) between the wave probe and the stereo cameras for H_m and H_s is 4.0mm and 7.8mm respectively. The ultrasonic sensor produced a higher MAE in comparison to the stereo cameras. 11.0mm and 9.7mm for H_m and H_s respectively.

The largest difference in measurements between the ultrasonic sensor and the wave probe occurs at 0.20Hz wave frequency (absolute error in $H_m = 17.33\text{mm}$ and $H_s = 14.9\text{mm}$). Possible reasons could be due to the scattering of the acoustic energy i.e. not enough energy is returned to the sensor, which can occur when the angle of incidence between the ultrasonic beam and the water's surface is large enough to cause the beam to be reflected away from the receiving transducer in the sensor. In addition, the sensor also has a resolution of 1cm. This implies at any given point in time if the true wave height recorded by the wave probe is a floating-point value, the sonar would return an integer value instead. The ultrasonic sensor also showed a higher variation between measurements in comparison to the stereo cameras. The largest standard deviation recorded between repetitions was 2.8mm whilst that of the stereo cameras was 1.4mm. Hence, the stereo cameras demonstrated a higher precision in comparison to the ultrasonic sensor.

Further analysis of the results are also carried out in the frequency domain to verify the claims outlined in the previous paragraph. The time series data were converted into the frequency components using the Fast Fourier transform (FFT) algorithm (Cooley and Tukey, 1965). The frequency domain representation contains information of the amplitude and phase information of the signal. Figure 5-26 below shows the time series data of the wave probe and the amplitude distribution at various frequency components in the signal.

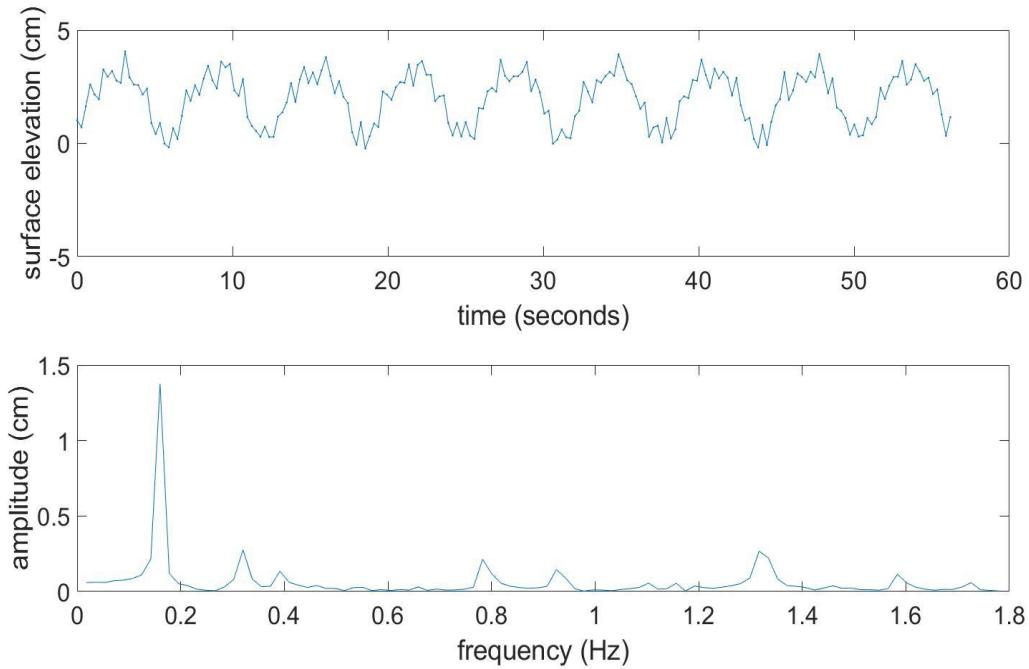


Figure 5-26: Time and frequency representation of wave probe measurements for experiments conducted at a wave frequency 0.16Hz (Top- captured time series and Bottom – derived frequency representation)

In the frequency domain, one would expect the highest peak to occur at the frequency of the wave being investigated i.e. the fundamental frequency. The highest peak in the frequency domain representation of the signal as seen in the bottom plot of Figure 5-26 indeed corresponds to a frequency of 0.16Hz, which was the wave frequency selected during the experiment. The time series data from the stereo cameras and wave probes were also converted into the frequency domain in order to investigate if the highest peak in the signal occurs at the fundamental frequency.

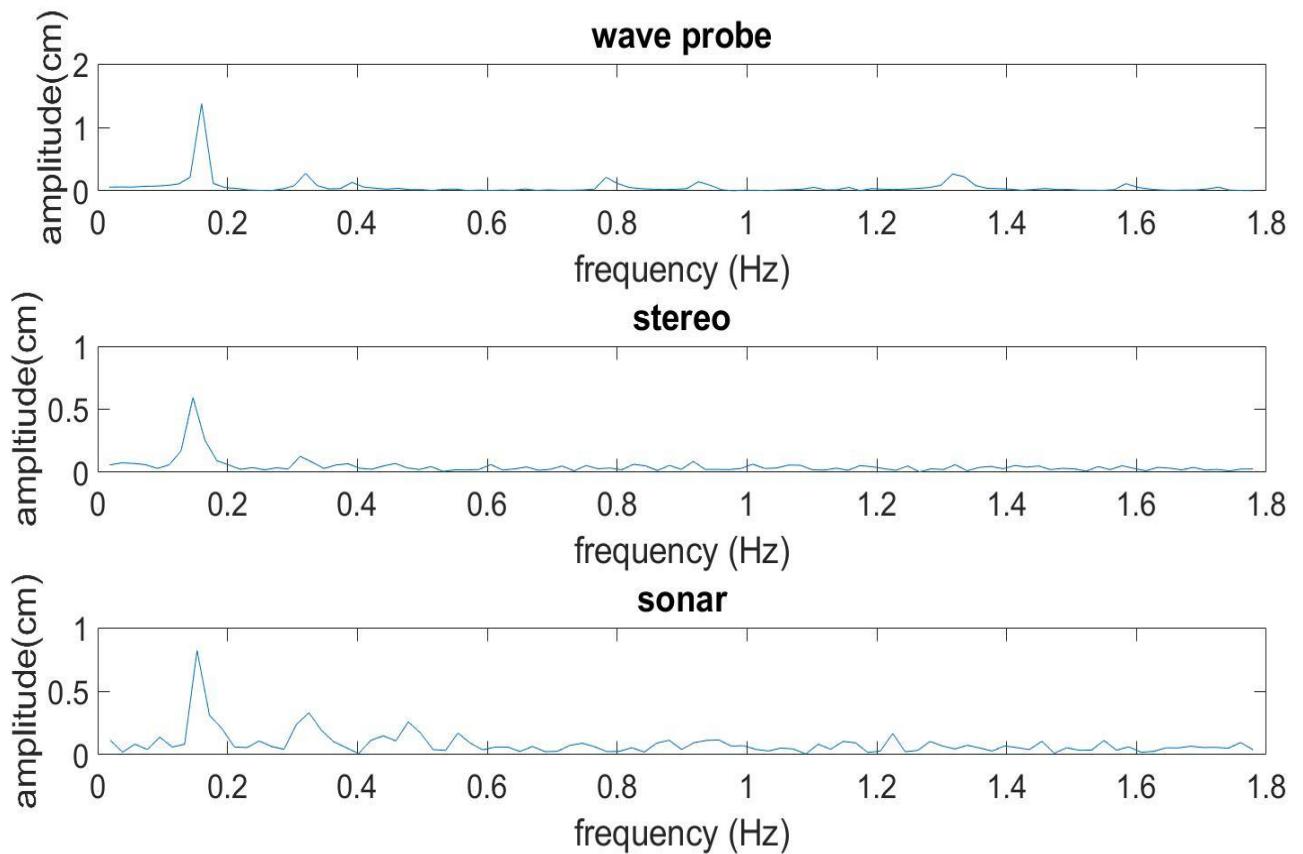


Figure 5-27: frequency domain representation for various sensors at a wave frequency of 0.16Hz (Top- wave probe, middle – stereo, bottom – sonar)

For a wave frequency of 0.16Hz, the highest peak in the signal for both the stereo cameras and ultrasonic sensor occurred at the fundamental frequency as seen in Figure 5-27. Similar behaviour was observed for a wave frequency of 0.30Hz.

However, at a wave frequency of 0.20Hz this is not the case with the ultrasonic sensors. The highest peak in the signal did not occur at the fundamental frequency but at the second harmonic (approximately 0.40Hz) as shown in the bottom plot in Figure 5-28. This suggests the sensor failed to pick up the true waveform at the given frequency and explains why the largest absolute error between the sonar and wave probe occurred at the 0.20Hz frequency.

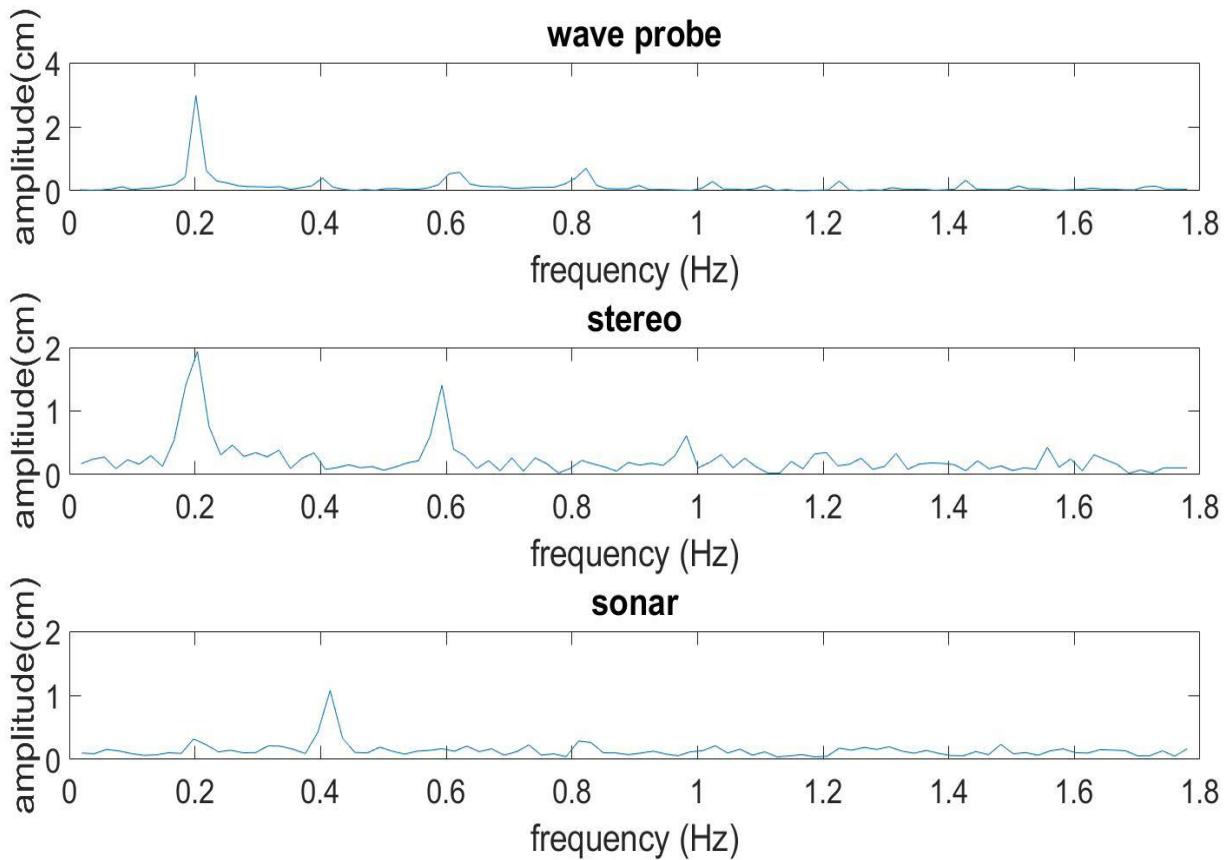


Figure 5-28: frequency domain representation for various sensors at a wave frequency of 0.20Hz (Top- wave probe, middle – stereo, bottom – sonar)

Unlike the ultrasonic sensor, for both wave probe and stereo cameras, the highest peak in the signal in the frequency domain occurred at the fundamental frequency of the system as discussed earlier. The plot in Figure 5-29 shows comparisons between the peak amplitudes observed at the various fundamental frequencies for the stereo cameras and wave probe.

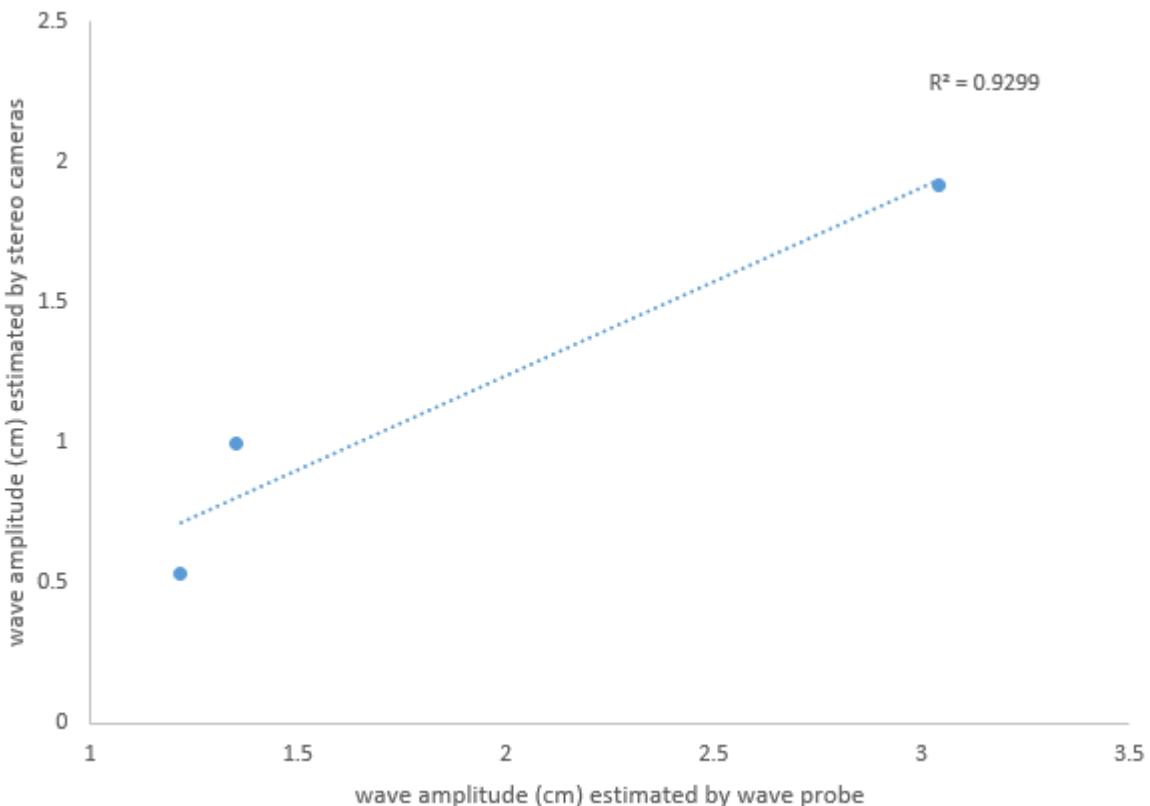


Figure 5-29: wave amplitude estimated by stereo cameras plotted against wave amplitude estimated by wave probe. The R^2 value is 0.93 approximately. Peak occurred at 0.15Hz, 0.21Hz and 0.32Hz for stereo cameras.

The cluster of points are very close to the regression line fitted between the data points. The coefficient of determination R^2 is approximately 0.93. The high R^2 value suggests a strong agreement between the stereo cameras and wave probe measurements. The results should not be surprising since the estimated wave heights between the stereo cameras and wave probes are very similar as indicated by the low MAE between two measurements.

To further support the claim that the stereo measurements fit closely to the wave probe values, spectral analysis is done using a periodogram. Figure 5-30 below displays the power spectra analysis computed from the wave time series measured by the wave probe and stereo cameras in blue and orange, respectively. The power spectrum describes the energy distribution of the time series in the frequency domain (Fulop and Fitz, 2006).

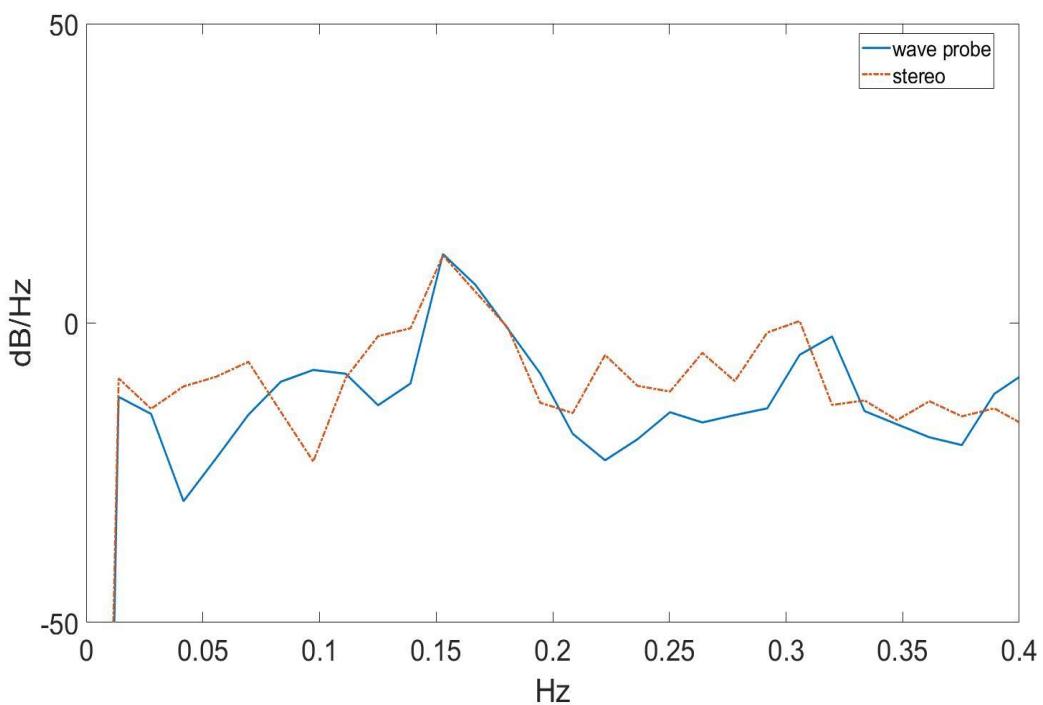


Figure 5-30: Spectral Estimation via Periodogram for 0.16Hz run.

Both spectra fit very well at the dominant wave peak around 0.16Hz as expected, which is the wave frequency, investigated during this run. However, compared to the reference spectrum estimated from the wave probe signal, the stereo spectrum overestimates the spectral energy at the low and high frequencies. Discrepancies at the lower and higher frequencies could be attributed to different wave regions examined by the stereo cameras and wave probe.

5.3 Conclusions

This chapter has investigated how wave height can be estimated from laboratory based water waves. Zero up-crossing analysis was used for estimating the wave height for the time series data extracted from the stereo cameras, ultrasonic sensor and wave probe. Estimated wave height from the stereo cameras showed closer agreement with data acquired from the wave probe in comparison with the ultrasonic sensor. Now that wave height estimation using stereo cameras has been achieved, the next chapter investigates how to control two UAVs for the purpose of mounting the stereo cameras on one UAV (the leader) for wave height detection whilst having the follower track the leader's trajectory at a pre-set separation distance. Such a configuration would allow for the follower to land and deploy the payload on a water surface with low wave height, as detected by the leader as discussed in Chapter 1.

Chapter 6: Coordinated control of two UAVs

6.1 Introduction

The formation-flying algorithm used for controlling two UAVs is presented in this chapter. Before formation flying can be achieved, the UAV needs to be set up or configured. This chapter explores various flight controllers and ground control stations that can be used for configuring an autonomous UAV. After configuration, the system structures and working principles necessary to achieve formation flying is investigated.

The approach has been validated through simulations and filed experiments.

6.2 Related Work

There has been a considerable amount of research that has focused on control of multiple UAVs. For example Kushleyev et al. (2013) created the most remarkable flock of 20 miniature quadcopters with a central computer that calculates the navigational instructions for the robots in the swarm. The scaling down to the size of micro aerial robots enabled the quadrotors to operate in tightly constrained environments in tight formations. Although the performance of the system is very reliable, the system requires the use of high precision camera for localization and as a result is not applicable for outdoor applications.

For outdoor applications, Vásárhelyi et al. (2014) presented an outdoor based GPS decentralized multi-copter flock consisting of 10 autonomous flying robots. Each of these flying robots in the flock is capable of navigating itself based on the dynamic information received from other robots in the vicinity. Their proposed architecture does not require any central computer for data processing or control instead all the necessary computations are carried out by miniature on-board computers.

Quintero et al. (2013) designed a novel flocking algorithm that enabled multiple small, fixed-wing UAVs to flock together in order to distribute a given sensing task among members. The flocking was performed in a typical leader-follower fashion, where the leader was assumed to already have an effective control policy for the task. The followers, however, had no actual interactions between units, they simply flew independently at the same time and the same place, at different altitudes.

Sabine et al. (2011) also demonstrated a GPS-positioning based swarming using 10 fully autonomous fixed-wing UAV in outdoor environment flown with a spacing of 10m as a means to investigate the trade-off between communication range and motion dynamics on aerial swarms.

Most research on swarm UAVs have achieved its goals in a different format. The UAVs in these cases have been flown using pre-programmed flight paths, but what if a pilot wanted maximal control of multiple UAVs at once ? There are many applications where real time control over many UAVs by one pilot is beneficial. For example, flying a UAV over a large aquatic body for landing site assessment for another UAV as described in Chapter 1. For such applications, it is desirable for a pilot to have real time maximal control over multiple UAVs at once. In such context, the problem domain can be simplified to flying of multiple UAVs with one remote controller.

Lazzaro (2015) demonstrated flying multiple UAVs from one remote controller. The UAVs in the swarm sent each other GPS coordinates in order to be aware of the positions of surrounding vehicles.

In this thesis, a similar framework is adopted. A pilot controls a leader UAV with a remote controller, and the GPS position of the leader is sent to the follower in order to

allow the follower track the leader's trajectory at a given distance. Section 6.7 provides a detailed overview of how this achieved.

6.3 Flight Controllers

There are many flight controllers on the market today (Chao et al., 2007) that can be used for controlling UAVs. They are usually equipped with sensors built-in such as an Inertial Measurement Unit (IMU) to determine the vehicles current attitude, position and speed. The IMU generally includes a combination of sensors mainly gyroscope, accelerometers and barometric sensors. Additional sensors such as Global Position System (GPS) can be added to the flight controller to assist in establishing the vehicles current position.

Flight controllers can generally be classified into open and closed source flight controllers (Ebeid et al., 2017). The term open-source refers to software, which is made available to the public and can be modified. In contrast, closed source is usually proprietary software that is not made available to the public and hence cannot be modified. For these reasons, it was decided to utilise open source flight controllers to enable the flight controller functionalities to be easily modified based on the user's requirements. Examples of popular open source flight controllers used today are:

- **Pixhawk**

This flight controller is popular amongst hobby applications and is a rather new platform compared to its competitors. It was developed by 3DRobotics and is the substitute of the Ardupilot Mega (APM).The dimensions of the board are (81.5x 50 x 15.5mm) and it weighs approximately 38 grams. The firmware running on this board is all open source hence it is possible to add new features to the existing implementation. The board offers a range of ports to allow

connection to external hardware. In addition, it offers several flight modes such as stabilize, loiter, guided, circle etc. The guided flight mode has been used in this thesis (see Section 6.7.5 for details of explanation of flight modes) to achieve formation flying between a leader and a follower vehicle. This flight mode has been used because it supports autonomous waypoint capabilities as long as GPS signals are available.



Figure 6-1: Pixhawk flight controller

- **Naze32**

The Naze32 is an incredibly small (single board 36x36 mm and weighs approximately 5 grams) flight controller with a 32-bit processor running at 72MHz. The 10dof Naze32 is composed of 3 axis gyro, 3 axis accelerometer, 3 axis magnetometer and finally a pressure sensor. The board is designed for hobby flights such as fun fliers and acrobatics and lacks reliable GPS functionality for autonomous missions. (Bolam et al., 2017).

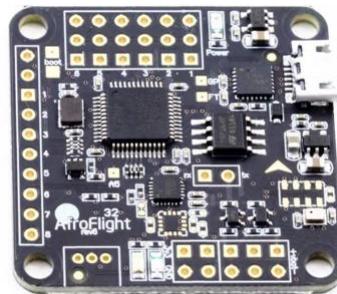


Figure 6-2: Naze32 flight controller

- **KKmulticopter- mini**

The board developed by Rolf Bakke, is suitable for multi copters with up to six rotors. Dimensions of the board are 36 x 36 x11.5mm and it weighs approximately 8.6 g. The board is designed only to stabilise the aircraft during flight by taking the signals from the on-board gyros and passing them through to the Integrated Circuit to process the information according to the KK firmware. It is not suitable for autonomous missions (Kaadan, 2013).



Figure 6-3: KKmulticopter mini

6.3.1 Overall Evaluation and Selection

The classification of each autopilot is presented in *Table 6-1*. Each autopilot is assigned a Boolean value of 0 or 1. A value of 0 indicates the autopilot does not satisfy the proposed decision criteria whilst a value of 1 indicates the proposed decision criteria are satisfied by the autopilot. The objective of the classification is to determine the most suitable autopilot that best fulfils the requirement discussed as follows:

- Small dimensions and weight - In order to maximize payload capacity, UAVs would benefit from autopilots with smaller dimensions and lower weight.
- Waypoint navigation capabilities - Autopilots that support unpiloted (fully autonomous) flight by utilising GPS waypoints can expand flight capabilities and enable complex tasks to be achieved with the UAV.
- Automatic take-off and landing - Autopilots equipped with automatic take-off and landing is beneficial for UAVs during autonomous missions.
- Open-source and Configurable - The ability to add new features to the existing features in the autopilot is vital to allow UAVs greater flexibility.

Table 6-1: Evaluation of researched flight controllers

	Dimensions	Waypoint capabilities	Auto take-off and landing	Open source and Configurable	Total average
Pixhawk	1	1	1	1	4
Naze32	1	0	0	1	2
Kkmulticopter-mini	1	0	0	1	2

Following this decision process, the Pixhawk flight controller is selected as the most suitable flight controller in this thesis. This is because it is relatively lightweight, easily configurable and equipped with autonomous features.

6.4 Introduction to Ardupilot flight stack

The Ardupilot is an open source autopilot system supporting multi-copters, fixed-wing aircraft and rovers. The Ardupilot flight stack has been used in this thesis because the flight control firmware offers very good performance (Bin and Justice, 2009) and is open-source. In addition, support tools are available to allow for debugging in simulation.

Two popular autopilot boards that support the Ardupilot flight stack are the Pixhawk and APM boards (see Figure 6-4). The main advantage of the Pixhawk board over the APM board is that it runs 32-bit software (Meier et al., 2012) architecture compared to the 8-bit used by the APM. The Pixhawk board has more processing power and speed in comparison to the APM board. The APM boards are also becoming increasingly difficult to purchase off-the-shelf. This is why the Pixhawk was the preferred autopilot board chosen in this work.

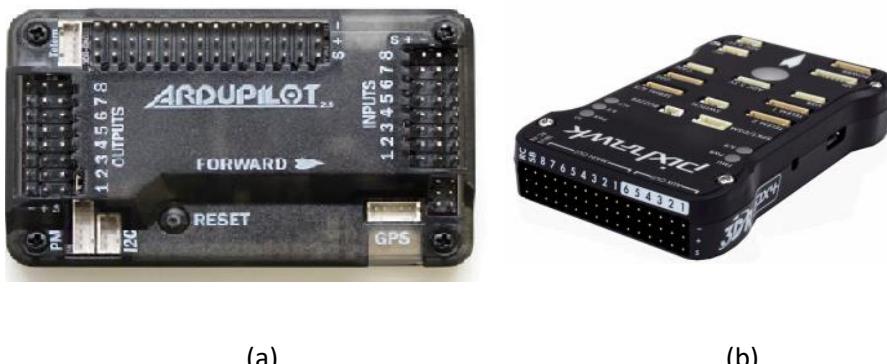


Figure 6-4: Ardupilot solutions: (a) – Ardupilot Mega (b) Pixhawk

6.4.1 Ardupilot Code Structure

The basic structure of ArduPilot is broken up into five main parts (ArduPilot Dev Team, 2016c):

1. Vehicle directories - These directories define the firmware for each vehicle type supported. Currently Ardupilot supports ArduPlane, ArduCopter, ArduRover and Antenna Tracker. The various firmware are briefly explained as follows:

- ArduPlane - The firmware provides fixed wing aircraft full autonomous capability.
- ArduCopter - Provides multicopters full autonomous capability.
- ArduRover - Provides ground vehicles and boats full autonomous capability
- Antenna Tracker - used for tracking vehicles location and utilises the position of the vehicle to correctly align a directional antenna. Utilising such approach significantly improves the range over which signals can be sent and received from the ground station.

In this, thesis the ArduCopter firmware has been used. The motivation behind using multicopters are that they are able to take-off and land vertically, providing the ability to allow the vehicle to operate in a smaller vicinity with no substantial landing/take off site required. In addition, their ability to hover makes them well suited for applications that require precise measurement such as wave height monitoring.

2. AP_HAL - This is regarded as the Hardware Abstraction Layer. This enables the Ardupilot code structure to be portable to lots of different platforms

3. Libraries - These include all the set of libraries used by the project. They include sensor drivers, attitude and position estimation using Extended Kalman Filter (EKF) frameworks and Proportional -Integral-Derivate (PID) control to improve vehicle stability and response.
4. Tools directories – These are miscellaneous support directories. They provide the infrastructure for log replays.
5. External Support code – On some platforms external support code is provided for additional features to the board. For example, the PX4Firmware includes the middleware and drivers used by the Pixhawk boards. In addition, it provides support for the MAVLink protocol and code generators. Section 6.4.2 provides a comprehensive overview of the MAVLink protocol

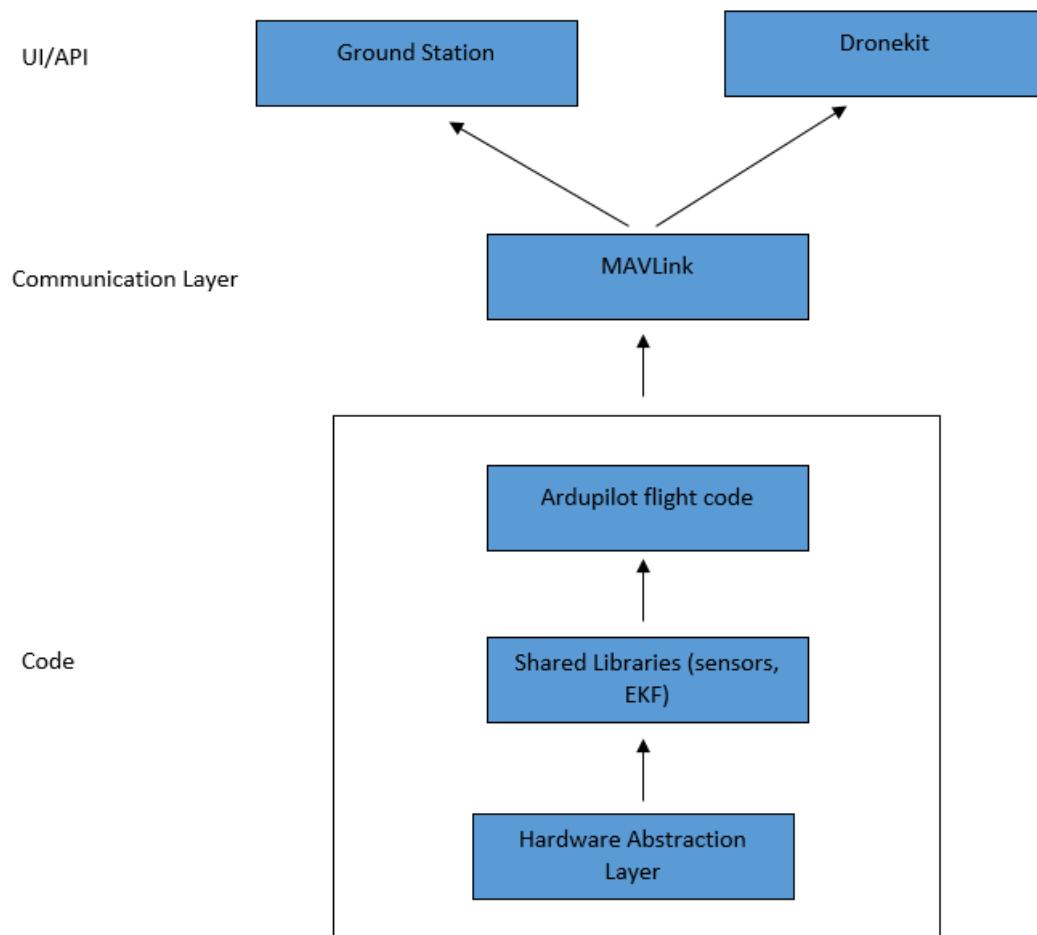


Figure 6-5: Ardupilot architecture

In this thesis, the Dronekit Application Programming Interface (API) shown in Figure 6-5 has been used for coordinated control of the UAVs. This is possible because the Dronekit API as well as the Ground Control Station User Interface communicate with the ArduPilot flight stack through the MAVLink protocol (see Figure 6-5). Section 6.5 and 6.6 provide detailed overview of the Ground Control Station and Dronekit respectively.

6.4.2 MAVLink protocol

The Micro Air Vehicle Communication Protocol is a point-to-point lightweight header communication protocol that allows entities to exchange information. When utilised in UAVs, it allows bidirectional communications between the UAV and a Ground Control Station (GCS). The protocol facilitates the means for the GCS to send commands to the UAV, which allows control of UAVs from a high level. In addition, the protocol also allows the UAV to send status and telemetry information back to the GCS.

MAVLink is part of the DroneCode project, governed by the Linux Foundation. The protocol has been utilised in many Autopilot systems like ArduPilotMega, pxIMU, FLEXIPILOT, SLUGS (Marty, 2013). In addition, it has been utilised in a number of Ground Control Stations (GCS) such as Mission Planner, APM Planner, QGroundControl and MAVProxy.

All MAVLink messages contain a header appended to every payload of the message. The header contains information about the message and the payload contains the data for the message. The minimum length of a MAVLink message is 8 bytes and the maximum length, with full payload is 263bytes (see *Figure 6-6*). Each message also contains a checksum for error detection. If checksum validation at the sending and receiving ends does not match, then it means the message is corrupted and will be discarded.

The Figure 6-6 originally presented here cannot be made freely available via LJMU E-Theses Collection because of copyright. The image was sourced at (Atoev et al., 2017)

Figure 6-6:MAVLink Header Format (Atoev et al., 2017)

A brief description of the fields in the MAVLink packet in *Figure 6-6* is described as follows (Atoev et al., 2017):

- **STX** - This indicates the packet start sign
- **Len** - This indicates the length of the payload
- **SEQ** - this indicates the sequence number of the packet, is useful for detecting packet loss, and thus allows the aircraft or ground control station to take appropriate action.
- **SYS** - ID of sending system. It allows differentiating UAVs on the same network. The maximum number of vehicles allowed on the same network is 255.
- **COMPID** - Represents the on-board component system sending the message.
- **MSGID** - Indicates the type of message that is carried out by the payload. For example, a Message ID of 0 represents a heartbeat message. This message is sent periodically once every second to indicate there is communication between the GCS and autopilot.
- **PAYLOAD** – This contains the actual data of the message which depends on the message ID
- **CKA & CKB** – These last two bytes are useful for checksum validation

MAVLink messages can be split into two main categories: information requests and mission commands. The information requests provide information about the status of the vehicle. For example, there are MAVLink messages that are used to extract the GPS position of the vehicle. (See **Error! Reference source not found.**). There are three types of categories of mission commands:

1. **Navigation commands** - Useful for controlling the movement of the vehicle.

This includes taking off, moving to various GPS waypoints, changing altitude and landing.

2. **DO commands** – These commands do not affect the position of the vehicle and are used to achieve a particular task. For example can be used to trigger camera or setting a servo value.

3. **Conditional commands** – Used to delay a DO command until a certain condition is met.

The Figure 6-7 originally presented here cannot be made freely available via LJMU E-Theses Collection because of copyright. The image was sourced at (Ardupilot Dev Team, 2017).

Figure 6-7: GPS MAVLINK message ID=33(Ardupilot Dev Team, 2017)

MAVLink messages are usually defined in an XML file and are converted to C/C++, Python code via existing code generators. The Pymavlink provides python bindings which allows MAVLink messages to be sent and read from the UAV.

The Dronekit Application Programming Interface (API) and MAVProxy GCS are built upon the Pymavlink python bindings.

6.4.3 Software-In-the-Loop

The Ardupilot Software In the Loop (SITL) allows simulation of an ArduPilot based autopilot and communicates with it using the MAVLink protocol (Psirofonia et al., 2017) (see Figure 6-8). This is essential because before deploying any code on actual hardware it is important to test the code using simulation/modelling techniques to identify and fix any bugs in the code that could lead to unexpected behaviour of UAVs in flight.

The Figure 6-8 originally presented here cannot be made freely available via LJMU E-Theses Collection because of copyright. The image was sourced at (Ardupilot Dev Team, 2016d).

Figure 6-8: SITL architecture (ArduPilot Dev Team, 2016d)

The *ArduCopter.elf* contains the built version of the ArduCopter source-code. It can be run as shown in Figure 6-9.

```
$ ./ArduCopter.elf --home -35,149,584,270 --model quad
Started model quad at -35,149,584,270 at speed 1.0
Starting sketch 'ArduCopter'
Starting SITL input
bind port 5760 for 0
Serial port 0 on TCP port 5760
Waiting for connection ....
```

Figure 6-9: SITL execution

Running this command spawns a simulation for a quadcopter model, at a home location with GPS coordinates of lat = -35° , lon = 149° , alt = 584 m (above mean sea level) and vehicle heading = 270° . The output shows that after the execution of this command; a connection to the SITL can be established using TCP/IP at the network address of the machine running SITL at port 5760. In the normal setup, SITL is used in combination with the MAVProxy GCS (see Section 6.5.2 on details of MAVProxy).

Additional TCP port 5763 is also open for connection.

6.5 Ground Control Station (GCS)

The Ground Control Station (GCS) is the software that gives commands to the vehicle. It is able to pilot, assist and monitor the vehicle. There are various ground control stations on the market today. Some examples are shown below in Table 6-2 (ArduPilot Dev Team, 2016a).

Table 6-2: Various types of GCS

Name	Free/Payment	Operating System	Description
Mission planner	Free	Windows	Developed by Michael Osborne. Provides calibration tools for the various sensors in the UAV. Allows for planning and control of all aspects in flight in real time. Written in C#
QGroundControl	Free and open source	Windows, Linux	Provides full control and mission planning. Written in C++ using the Qt libraries
MAVProxy	Free and open source	Windows, Linux and Mac	Developed by Andrew Tridgell with intent of providing a portable and extendable GCS for UAVs. It is a command-line, console-based app written in Python. Supports loadable modules, moving maps and joysticks
APM Planner	Free and open source	Windows, Linux and Mac	Open source software similar to Mission planner
Tower	Free and open source	Android	3DR android app to control UAV. Intended for end users. Provides a user friendly interface

In this thesis, the ground control station software packages used are Mission planner and MAVProxy. Mission planner has been used in this thesis for configuring the UAV before flight because the ground control station provides a user-friendly interface that allows quick interactions between the ground control station and the aircraft (see section 6.5.1 below for more details). In addition, in the case of any issues or conflicts that can arise during configuration a large open source community is available that can help with resolving such issues or conflicts. On the other hand, the MAVProxy has been used for testing in SITL because the ground control station is compatible with SITL architecture (see Section 6.5.2).

6.5.1 Mission Planner

The Mission Planner version that was used in this work is the version 1.3.38. Mission Planner was used for installation of the firmware, calibration of accelerometer, compass, radio and flight modes. The following section will explain the setup procedure in detail:

- Firmware – The latest firmware can be uploaded into the board through Mission Planner. The GCS provides support to a variety of copters such as quadcopters, hexacopters, octocopters, helicopters, planes and even ground vehicles. The installation of the desired configuration firmware is done by simply clicking on the corresponding icon (see Figure 6-10). Mission planner then prompts the user to confirm the firmware download and installation. In this work, the ArduCopter 3.3.1 firmware has been utilised.

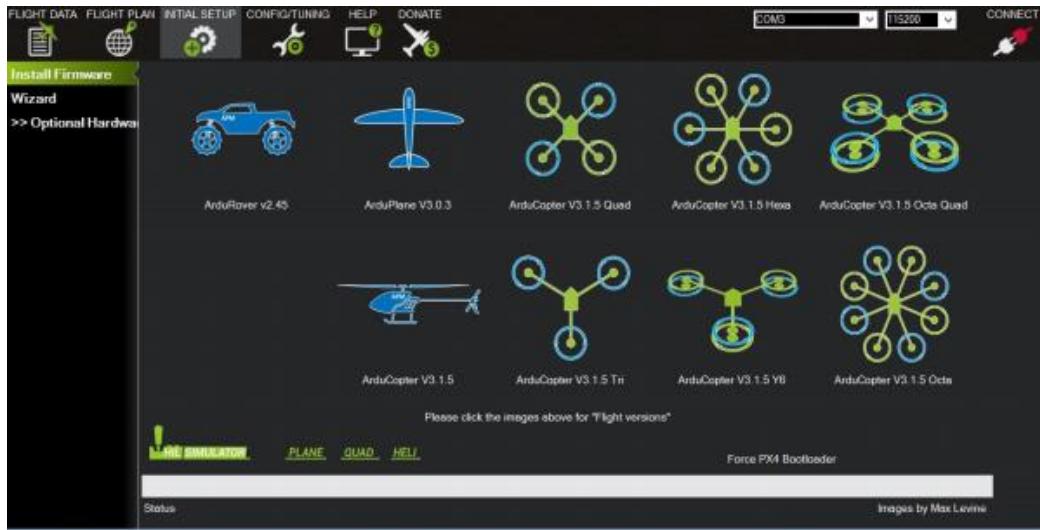


Figure 6-10: Mission planner firmware selection

- Frame orientation – Mission planner supports three types of frame configurations: X, Y and H. The X configuration is selected because X-type frames were utilised in this work.



Figure 6-11: Mission planner Frame Type Selection

- Compass calibration - The compass was calibrated using the “Live calibration” method. During this calibration, the vehicle is rotated around its different axes for a few seconds in order for the vehicle to determine the true orientation.

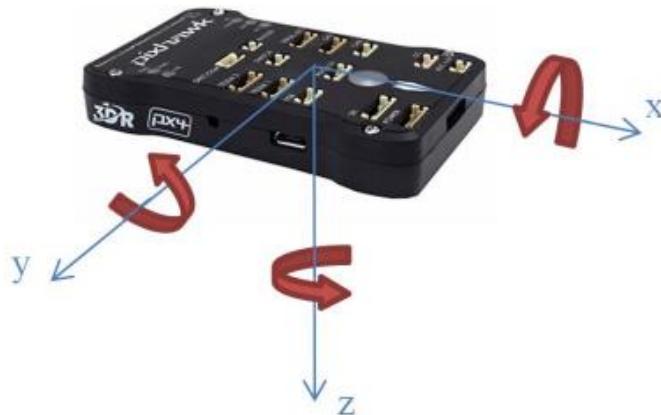


Figure 6-12: Compass calibration

- Accelerometer calibration - In order for the vehicle to determine the right altitude to consider as level while flying accelerometer calibration is essential. This is done by placing the autopilot on all of its six sides respectively i.e. standing, right side, left side, nose down, nose up and lastly on its back.

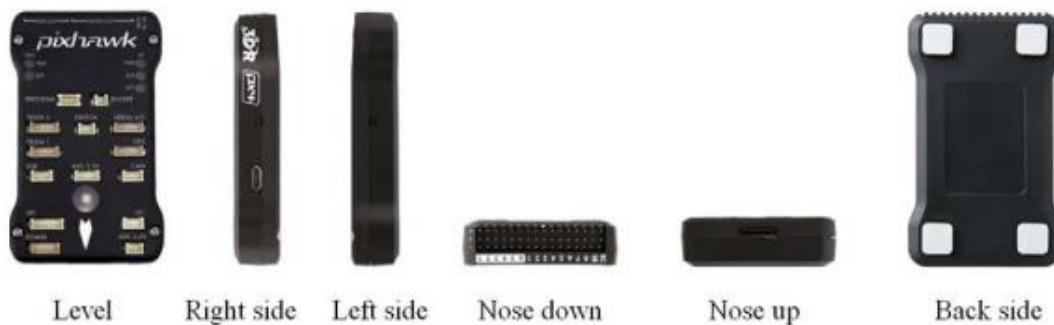


Figure 6-13: Accelerometer calibration positions

- Radio Calibration -This procedure is essential because it establishes the minimum and maximum Pulse Width Modulation (PWM) signals emitted. Calibration is achieved by moving the enabled sticks and switches through their full range to enable the autopilot learn the minimum, maximum and centre positions of the sticks and switches on the transmitter. The red lines across the

calibration bars in Figure 6-14 show the minimum and maximum PWM values in each channel.



Figure 6-14: Radio Calibration procedure

The Radio Control (RC) input channels are shown in Table 6-3. Channels 7 and 8 are currently not used.

Table 6-3: Configurable radio channels

Channel Number	Parameter
Channel 1 : Ailerons	Roll
Channel 2 : Elevator	Pitch
Channel 3 : Throttle	Speed
Channel 4 : Rudder	Yaw
Channel 5	Flight Mode
Channel 6	Fail Safe
Channel 7	-
Channel 8	-

6.5.2 MAVProxy

As discussed in Section 6.5 MAVProxy is a GCS for UAV supporting the MAVLink protocol. The main difference between MAVProxy and traditional GCS software such as Mission Planner is that MAVProxy is built for command line interface and does not need a graphical desktop environment to operate. This means it is lightweight and is suited for embedded systems. Although it is a command line interface, it supports moving maps, joysticks and allows the user to create modules that can extend the functionality of the UAV. These modules are loaded within the MAVProxy environment.

Section 6.5.2.1 below discusses MAVProxy connection to SITL and Section 6.5.2.2 discusses MAVProxy connection to the hardware.

6.5.2.1 [MAVProxy SITL configuration](#)

MAVProxy can be spawned to connect to a virtual vehicle by running the command below:

```
mavproxy --master=tcp:127.0.0.1:5760 --map --console
```

This command allows output of MAVLink messages via the TCP internet protocol. The `--map` argument allows the virtual vehicle to be displayed on a map and the `--console` provides a Graphical User Interface (GUI) to interact with the virtual vehicle and display vehicle parameters. (See *Figure 6-15* below).

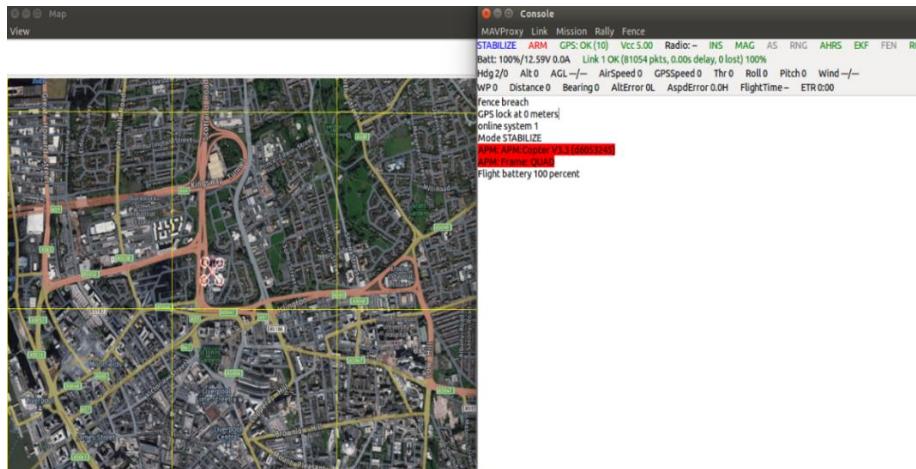


Figure 6-15: MAVProxy Console and map interface

6.5.2.2 MAVProxy UAV configuration

In addition to supporting connection to SITL, MAVProxy also allows connection to a real vehicle. The real vehicle is connected to MAVProxy via a serial link (see Figure 6-16 below).

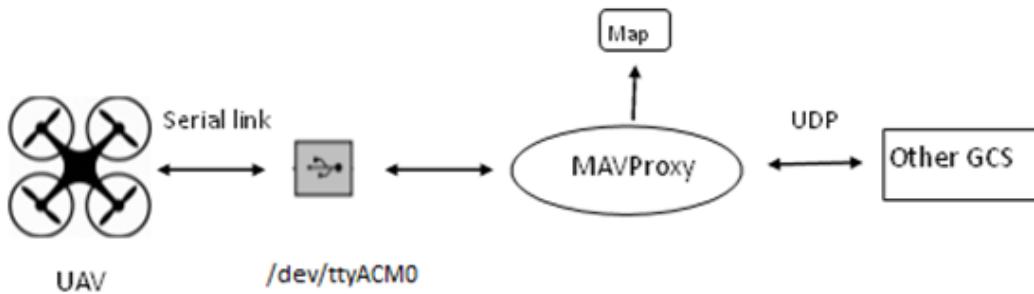


Figure 6-16: MAVProxy UAV configuration

The baud rate setting is an important factor that needs to be considered when connecting a vehicle to MAVProxy. A wrong baud rate setting would not allow successful communication with the vehicle. By default, the Ardupilot software utilises a baud rate of 115200 bd/s when connected via a USB cable and 57600 bd/s when connected via telemetry radio. MAVProxy also allows the redirection of MAVLink messages to other GCS via the User Datagram Protocol (UDP). This allows Mission Planner, QGroundControl and other MAVLink enabled - GCS to obtain live feed of vehicle data provided MAVProxy and the target GCS are on the same network.

The information related to the Pixhawk autopilot after successful connection to MAVProxy is shown in Figure 6-17. As a validation check to see if connection to the vehicle is successful, one can compare the flight mode returned by GCS to the flight mode of the vehicle when it was initially connected to the GCS. For example, Figure 6-17 (Top) shows the right flight mode returned by MAVProxy when it was connected in STABILIZE mode. In addition, the "arm throttle" command can also be executed and one can detect if the vehicle successfully arms by observing the motors spin.

```

tom@temi-MacBookPro:~$ mavproxy.py --master=/dev/ttyACM0 --baudrate=115200
Connect /dev/ttyACM0 source_system=255
Log Directory:
Telemetry log: mav.tlog
MAV> Waiting for heartbeat from /dev/ttyACM0
0 0 online system 1
STABILIZE> Mode STABILIZE
APM: Initialising APM...
APM: Calibrating barometer
APM: barometer calibration complete
APM: GROUND START
Init Gyro****
INS
-----
G_off: -0.42, -0.28, -0.01
A_off: -0.15, -0.35, -1.63
A_scale: 0.99, 0.99, 0.97

*
Ready to FLY ublox fence breach
APM: APM:Copter V3.3.3 (acf2e10c)
APM: PX4: 34e1d543 NuttX: 7c5ef883
APM: Frame: QUAD
APM: PX4v2 00500020 31345119 35313935
Received 508 parameters
arm throttle → Arming vehicle
STABILIZE> APM: GROUND START
Init GyroAPM: Initialising APM...
**
INS
-----
G_off: -0.41, -0.28, -0.01
A_off: -0.15, -0.35, -1.63
A_scale: 0.99, 0.99, 0.97

Got MAVLink msg: COMMAND_ACK {command : 400, result : 0}
[

tom@temi-MacBookPro:~$ mavproxy.py --master=/dev/ttyACM0 --baudrate=115200
Connect /dev/ttyACM0 source_system=255
Log Directory:
Telemetry log: mav.tlog
MAV> Waiting for heartbeat from /dev/ttyACM0 → Waiting for heartbeat message
0 0 online system 1 → UAV mode when connected to GCS
STABILIZE> Mode STABILIZE → UAV mode when connected to GCS
APM: Initialising APM...
APM: Calibrating barometer
APM: barometer calibration complete
APM: GROUND START
Init Gyro****
INS
-----
G_off: -0.42, -0.28, -0.01
A_off: -0.15, -0.35, -1.63
A_scale: 0.99, 0.99, 0.97

Ready to FLY ublox fence breach
APM: APM:Copter V3.3.3 (acf2e10c)
APM: PX4: 34e1d543 NuttX: 7c5ef883
APM: Frame: QUAD
APM: PX4v2 00500020 31345119 35313935
Received 508 parameters → Reading the vehicle parameters
[
```

Figure 6-17: MAVProxy screenshot .Top - Start-up and Bottom - Arming vehicle

6.6 Dronekit

Dronekit provides a higher-level interface to control the UAV through the MAVLink protocol. It enables connectivity in the air with an on-board computer or on the ground for ground control with a laptop or desktop. The API creates a *vehicle object* upon connection to MAVLink stream and stores instantaneous values of important state variables of the UAV such as GPS position (latitude, longitude, altitude), heading, velocity in the body-fixed frame, among others. In addition, it provides some functions, which translate mission commands into MAVLink messages that can be readily sent to the UAV. Table 6-4 shows the relevant information extracted from the UAV and the commands sent to the UAV in this thesis.

The Dronekit API has two major versions. The Dronekit version 1.0 ran as a MAVProxy extension. Hence, it needed MAVProxy in order for the API to be used in various projects. However, Dronekit version 2.0 has undergone a significant change when compared to version 1.0. In version 2.0, Drone kit is a standalone library. Hence, it no longer requires the use of MAVProxy in order for it to run. This is why the Dronekit version 2.0 has been utilised in this work. In general, the library provides classes and helper functions to:

- Connect to a vehicle from a python script
- Get and set vehicle state and obtain telemetry information
- Guide UAVs to a specified GPS position in GUIDED mode
- Send arbitrary custom messages to control UAV movement
- Override RC channel settings.

Table 6-4: Selected Dronekit information requests and mission commands

Information Requests	Mission Commands
GPS latitude	Take off to a designated height
GPS longitude	Land
GPS altitude from sea level	Navigate to GPS location (latitude, longitude, altitude)
Compass heading	Set vehicle mode
Time since vehicle boot	Set vehicle heading
vehicle mode	Shutdown component
Home location	

For completeness, the code snippet below shows how to connect to a real UAV using the Dronekit API. The *connect* method supports both serial and UDP protocols. The *wait_ready* parameter in the *connect* method when set to *True* ensures the *vehicle* parameters are populated before the *connect* method returns.

```
vehicle = connect("/dev/ttyACM0", baud = 57600, wait_ready = True)
```

Dronekit also provides a branch of development that aims to provide other developers with an Ardupilot Simulator. Similar to the Ardupilot SITL (see Section 6.4.3) it does not provide an accurate physical representation of the UAV its main goal is to emulate the Ardupilot firmware so developers can test their code before implementation on the physical platform.

Similar to the Ardupilot SITL, a virtual vehicle can be created at *lat* = -35.363261° , *lon* = 149.165230° , *alt* = 584 m and *heading* = 353° by running the following command as follows:

```
dronekit-sitl copter-3.3 --home=-35.363261,149.165230,584,353
```

Dronekit SITL sets up a TCP connection on the local host on port 5760 (see Figure 6-18) and waits until a GCS such as MAVProxy or QGroundControl connects to the target port to enable MAVLink data streams to be sent to the GCS.

```
temi@temi-MacBookPro:~$ dronekit-sitl copter --home=53.368769,-3.109712,2.88,0
os: linux, apm: copter, release: stable
SITL already Downloaded and Extracted.
Ready to boot.
Execute: /home/temi/.dronekit/sitl/copter-3.3/apm --home=53.368769,-3.109712,2.88,0 --model=quad
Started model quad at 53.368769,-3.109712,2.88,0 at speed 1.0
bind port 5760 for 0
Starting sketch 'ArduCopter'
Serial port 0 on TCP port 5760
Starting SITL input
Waiting for connection .... ←———— Waits until connection from MAVProxy
```

Figure 6-18: Dronekit SITL

Before formation flying between a leader and follower can be achieved, the UAV needs to be able to respond to basic take-off and landing commands and in addition fly towards a GPS position assigned to it. The Dronekit SITL in conjunction with MAVProxy were used to check that these conditions are met in software. For example, Figure 6-19 shows the simulated vehicle proceeding to a GPS position assigned to it. The version of Dronekit-SITL used in this work is 3.2.0 and the version of MAVProxy used in 1.6.2.

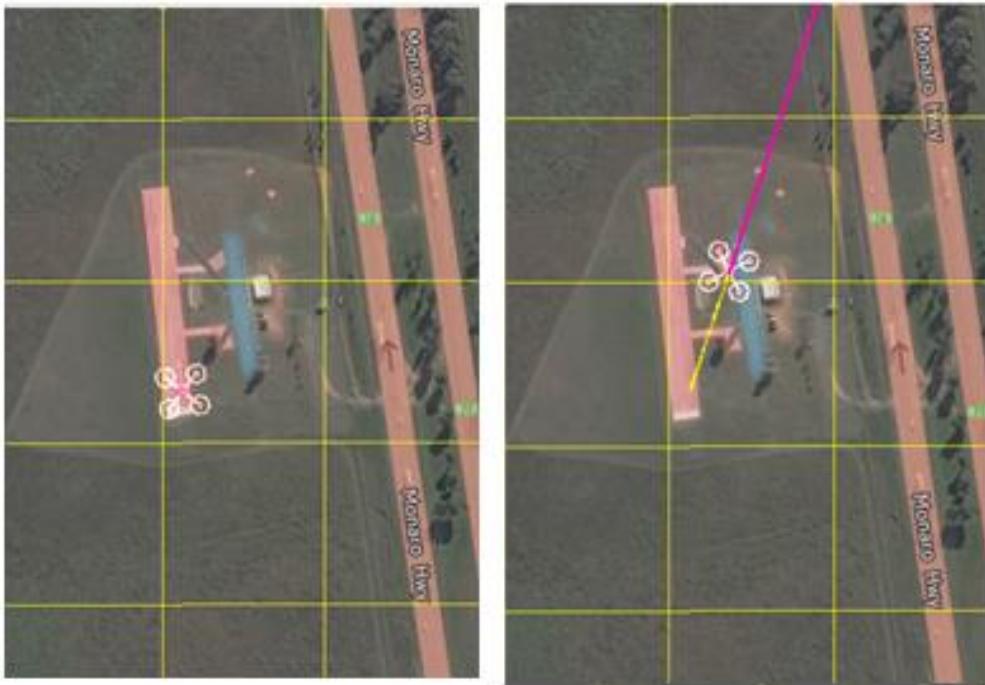


Figure 6-19: Simulated vehicle at home position (left) and proceeding to proceeding to latitude: -35.363048° and longitude: 149.165824° (right).

6.7 Leader-follower formation

The following section discusses the algorithm used for formation flying of two UAVs.

The general idea consists of extracting the GPS position of one UAV, the “leader” and implementing a position controller to determine the appropriate GPS position of the second UAV, the “follower” based on constraints discussed in Section 6.7.1

6.7.1 Motion based Constraints for leader and follower UAV

Knowing the position and heading of one leader UAV, (x_1, y_1, z_1) , its heading, H_1 , and its separation from a second UAV, S , it is possible to determine the position (x_2, y_2, z_2) and heading H_2 of a second, follower, UAV so that it will fly at a specified distance and orientation

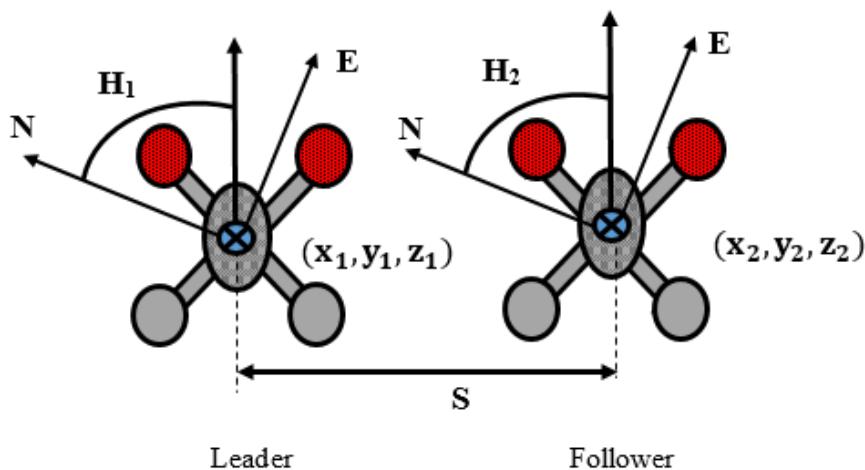


Figure 6-20: Definition of the axes set and locations of two UAVs separated by a distance S

As a first condition, both leader and follower UAV are considered to be on the same heading (H) if both UAVs are to fly parallel, and therefore $H_2 = H_1$

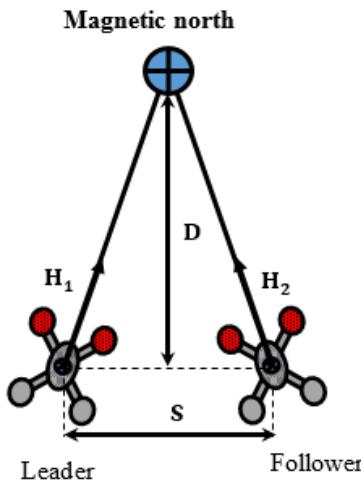


Figure 6-21: Orientation of two UAVs heading north

As shown in Figure 6-22, the separation distance, S is much smaller in comparison to the distance from the baseline of the UAV to the magnetic north D ($S \ll D$). By approximation, one can conclude that the two UAVs are considered to be on the same heading and travelling on parallel paths

Another constraint that must be satisfied is both UAVs must be at the same altitude (z) and so: ($z_2 = z_1$)

Again, because S is small in comparison to the radius of the Earth, the curvature of the Geoid can be neglected.

The third condition can be derived from the restraint that the distance separating the Centre of Gravity (CoG) of each UAV must be fixed at S . Therefore, the Euclidean distance between UAVs is as follows:

$$S = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (6-1)$$

Where (x_1, y_1) is the position of the leader in the Cartesian plane, \mathbb{R}^2 and (x_2, y_2) is the position of the follower.

The fourth condition can be derived from the restraint that the line S must be orthogonal to the heading H . Such a configuration, takes into account deviation in the heading angle of the leader when deducing the coordinates of the follower. Consider Figure 6-23 below which shows the UAVs in a general position relative to the geographic heading axes sets:

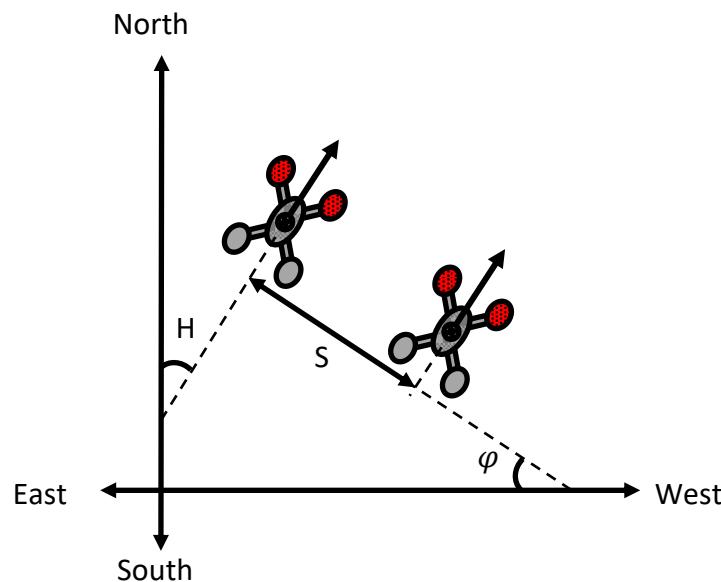


Figure 6-22: Two Drones in a general position and heading relative to the geographic axes set

The angle of inclination, φ , of the separation line S would be given by:

$$\varphi = \tan^{-1} \left(\frac{y_2 - y_1}{x_2 - x_1} \right) \quad (6-2)$$

By applying trigonometry, it can be deducted that $\varphi = H$ as shown in Figure 6-23

From equation (6-1) and (6-2) it is possible to deduct that:

$$S^2 = (x_2 - x_1)^2 (1 + \tan^2(H)) \quad (6-3)$$

Expanding (6-3) and rearranging gives rise to the quadratic equation in x_2

$$(\beta)x_2^2 - (2x_1\beta)x_2 + (x_1^2\beta - S^2) = 0 \quad (6-4)$$

Where $\beta = (1 + \tan^2 (H))$

Equation (6-4) contains only one unknown – i.e. x_2 and may be solved via the General Quadratic Formula (GQF):

$$x_2 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (6-5)$$

Where : $a = \beta$, $b = -2x_1\beta$ and $c = x_1^2\beta - S^2$

Hence

$$x_2 = \frac{2x_1(1 + \tan^2(H)) \pm 2\sqrt{x_1^2(1 + \tan^2(H))^2 - (x_1^2(1 + \tan^2(H))^2 - S^2)}}{2(1 + \tan^2(H))} \quad (6-6)$$

The two solutions derived from equation (6-6) constitute the follower UAV being to the right or to the left of the leader.

Having found x_2 , y_2 can be derived as follows:

$$y_2 = (x_2 - x_1) \tan (H) + y_1 \quad (6-7)$$

If both UAVs are commanded to be at the same altitude, then

$$z_2 = z_1 \quad (6-8)$$

Thus the main objective has been achieved, Given the position of the leader UAV denoted by (x_1, y_1, z_1) and its heading (H) , the corresponding position (x_2, y_2, z_2) of the follower UAV can be deduced so that it will fly at a fixed distance and orientation.

6.7.2 Simulation Results

The proposed algorithm was implemented in MATLAB to demonstrate and validate the formation algorithm. A test case was considered where the leader was set at the position $(0,0)$ (see Figure 6-23). For a given separation S , the resulting dynamic simulation model showed that for headings $0^\circ - 360^\circ$ the UAV pair models are fixed on the same horizontal plane on the same heading at constant separation.

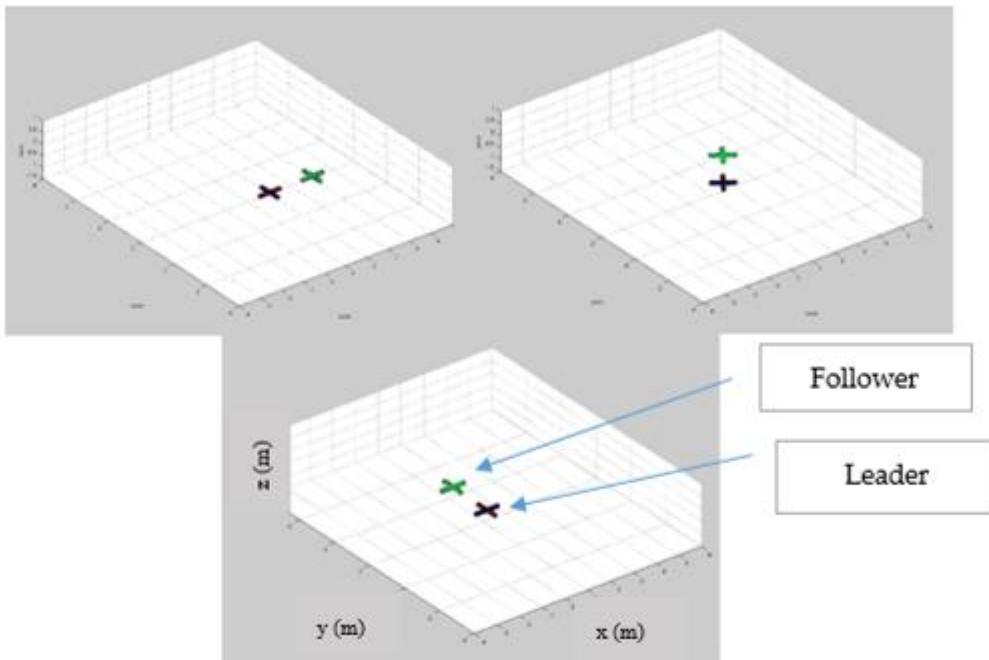


Figure 6-23: heading at 0 degree (top-left), 45 degree (top-right), 89degree (bottom)

From Figure 6-23, it can be seen that at various headings (H), the distance between the models of the leader (denoted by a black cross) and the follower (denoted by a green cross) is always orthogonal to its heading and also the orientation and distance between both UAV models remain fixed. There are however singularities in the solution at $H=90^\circ$ and 270° as a result of $\tan(H) \rightarrow \infty$. The approach to avoid these singularities case scenarios was to reject H which gave rise to mathematically undefined solutions and select $H(\pm 1^\circ)$ in order to obtain finite results.

Equation (6-6) produces two solutions due to the fact it is quadratic in nature. These two solutions refer to the cases where the follower is located at a distance S to the right of the leader or a distance S to the left of the leader. It is vital to select a consistent and appropriate solution to avoid the UAV pair colliding into each other at specific headings. The positive solution is chosen when the follower is required to be to the right of the leader($0^\circ - 90^\circ$ and $270^\circ - 359^\circ$) and the negative solution is applied when the follower is required to be to the left of the leader ($91^\circ - 269^\circ$). For example, at a

heading of 91° the follower is required to be to the left of the leader; hence, the negative solution of the quadratic formula in equation (6-6) is applied to deduce the X-position and the corresponding Y-position is obtained with equation (6-7). Figure 6-24 (a) and (b) below show different locations of the follower (denoted by the green cross) in comparison to the leader. Implementing coding of the appropriate solution as shown in Figure 6-24 b (follower to the left of leader) has proven to provide consistent results for computing real-time position of the follower vehicle.

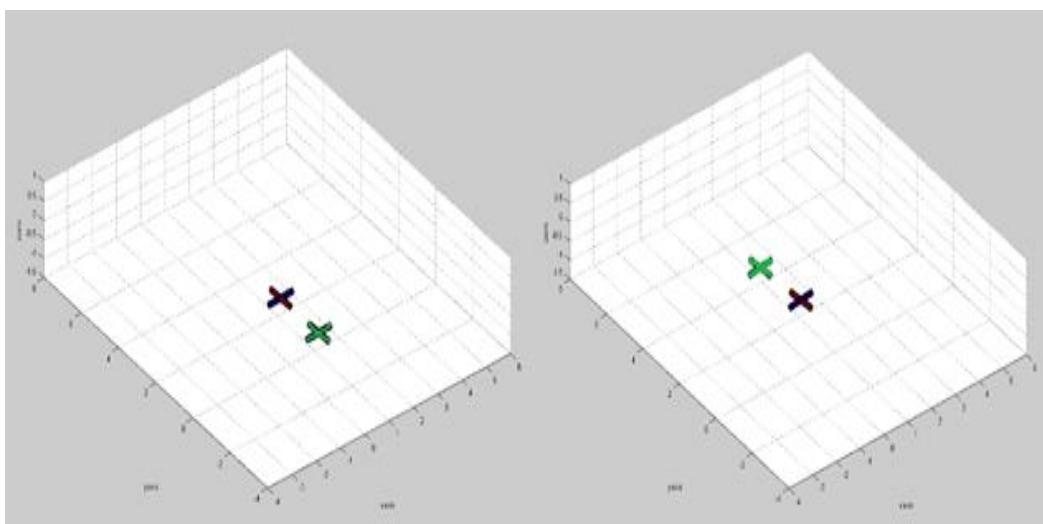


Figure 6-24: Follower at various positions relative to leader (a) - no coding of appropriate solution (left) (b) - coding of appropriate solution (right)

6.7.3 Hardware Overview

In this work, DJI F450 quadcopter frame is utilised (see Figure 6-25 below). This quadcopter is chosen because it is small in size, safe to use and most importantly, it weighs below 7kg. Having the weight below 7kg is crucial because the licenses available to the qualified pilots operating the UAVs are only valid up to 7kg. The dimensions of the quadcopter frame are 450 mm from motor to motor and weighs approximately 282g. For achieving a stable flight, the quadcopter was equipped with the following items:

- 3DR power module -A power module that supplies a stable 5.37V and 2.25Amp power supply needed to power the pixhawk flight controller (Ardupilot Dev Team, 2016b).
- GPS compass module -A UBlox Neo 7M GPS integrated with a compass module (U-blox, 2017) for determining the quadcopters position. This module has been used because it is compatible with the flight controller. It has been mounted on a mast to avoid interference from the Electronic Speed Controllers (ESC) and Power Distribution Board (PDB).
- FRsky Taranis X9D -Used to control the quadcopter manually using a 2.4GHz transmitter and receiver link which transmit signals to the flight controller.
- 2200mAh 3S Lipo Battery -The quadcopter is equipped with a 3S lipo DC battery, which has a capacity of 2200mAh and a 25C discharge rate. The battery lasts for about 6-8 minutes without any payload, which was sufficient to complete the test.
- Electronic Speed Controllers (ESCs) -Responsible for spinning the motors at the speed requested by the flight controller. The ESCs used are the DJI 4 x 15a E300 Opto (DJI, 2017) because they provide a high thrust to weight ratio to allow increased stability and agility in flight.
- Motors and propellers -The quadcopter uses four 2312E 960Kv motors and four 10 x 4.5inches (DJI, 2017). Two motors run counter clockwise and the other two run clockwise in order to produce pitch, roll and yaw control.



Figure 6-25: Two UAVs equipped with remote controller

6.7.4 Communication between leader and follower

In order to send the positional information from the leader to the follower vehicle, the leader needs to be able to communicate and send data to the follower. The initial approach was to adopt a centralized control scheme. A laptop running the Dronekit software was used to connect to both leader and follower UAV by calling the *connect* function as discussed in Section 6.6 on each vehicle respectively. However, it was observed on frequent occasions when attempting to connect to both vehicles the first vehicle connected successfully and the second vehicle failed to connect and raised a "*Timeout in initializing connection*" exception. This error was attributed to the Dronekit API being in its early stages of developments and as a result has some bugs associated with the software (Dronekit, 2018). In order to overcome the lack of implicit multi vehicle support associated with Dronekit, the approach taken in this thesis utilises User Datagram Protocol (UDP) socket connections (see Section 6.7.4.1 for more detail) and Wi-Fi to enable communication across a wireless network.

6.7.4.1 *UDP vs TCP*

Two major protocols for transporting data across a network are the Transport Control Protocol (TCP) and the User Datagram Protocol (UDP) (Xylomenos and Polyzos, 1999). The following are the differences between the two:

1. TCP is a reliable, connection-oriented protocol, and a connection needs to be established before it transmits data as a stream of bytes. UDP on the other hand is an unreliable, connectionless protocol that requires no connection to be established before packets are sent over the network in chunks called datagrams.
2. TCP provides error checking but UDP does not have an option for error checking.
3. Stream of bytes in TCP arrive in the same order they are sent whereas in UDP this might not be the case.
4. Application layer protocols like Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP) and Telnet utilise TCP to transmit data because it is reliable whereas UDP is used by protocols like the Voice over Internet Protocol (VoIP).

The UDP protocol has been used to allow information to be shared from the leader to the follower vehicle (see Section 6.7.4.2) because it does not require the existence of connections, lighter weight and faster protocol in comparison to TCP (Deyah and Bhaya, 2017).

6.7.4.2 *Client-Server Architecture*

The Python *socket* module is used to create network sockets. UDP Sockets can be created by calling the *socket.socket* constructor. Once a UDP socket object has been created, the sever process can be spawned by calling the *socket.bind* function. The *bind* function expects two arguments which are the *hostname* and *port number*. Once the server is spawned, the client can send message to the server by calling the *socket.sendto* function. The *sendto* function takes in two arguments. The first argument is the data to be sent across the network and the second is a tuple data structure that contains the *hostname* of the server and the allocated *port number*.

One can take advantage of this client server relationship to allow the leader UAV share its vehicle parameters with the follower UAV. This can be achieving by equipping the UAVs with embedded systems that have wireless capabilities to allow the vehicle parameters to be sent across a network.

Section 6.7.4.3 below discusses the embedded platform chosen and how it is connected to the UAV in order to allow information sharing from one UAV to another.

6.7.4.3 *Raspberry pi connection to UAV*

The embedded system utilised in this thesis is a Raspberry pi version 3.0. A raspberry pi was chosen because it is lightweight (41.2g), thus ensuring a minimal payload added to the UAV. In addition, the Pi is compatible with the Dronekit software discussed in Section 6.6. Finally, it supports wireless communication and interfacing. The pi has an ARM board, with a 1.2GHz processor and 1GB RAM. It utilises a Linux based operating system to handle computationally intensive tasks such as image processing. Other main key features of the system are as follows:

- GPU: Broadcom Video Core IV
- Four USB ports
- 5V DC-input
- Operating System- Raspbian
- Storage: SD card
- Network: 10/100 MBPS, 802.11n Wireless LAN, Bluetooth 4.0

The Pi is connected to the Pixhawk flight controller through a serial connection (see Figure 6-26). With this setup, MAVLink messages can be sent between the on-board computer and the Pixhawk flight controller.

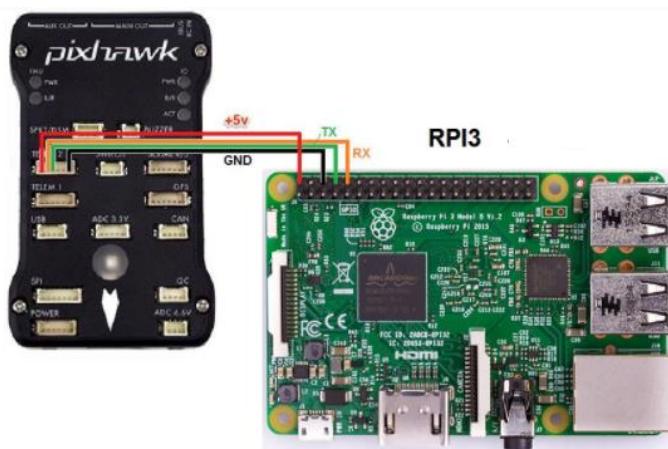


Figure 6-26: Connecting Raspberry Pi to Pixhawk (Choi et al., 2016)

The Pi also runs the Dronekit API discussed in Section 6.6 to allow commands to be sent to the UAV through the MAVLink protocol. Both leader and follower UAV are equipped with these on-board computers and unique static IP addresses are assigned to the aircraft. This setup and the Client -Server architecture discussed in Section 6.7.4.2 allow UDP packets to be sent from the leader to the follower UAV. The UDP packets sent in this work to the follower UAV contain information such as the mode, heading, home-location and current GPS position. After the follower receives the

packet information from the leader (particularly the GPS information), it computes the GPS waypoint that the follower UAV needs to fly towards to ensure formation flying between the UAV pair based on the constraints discussed in Section 6.7.1. Computation of the GPS waypoint the follower needs to fly towards, is done in real-time using the on-board Pi.

Figure 6-27 shows the leader-follower communication interface. The UAVs are connected to the same network through the Access Point. In addition, a laptop is also connected to the network for logging and debugging purposes.

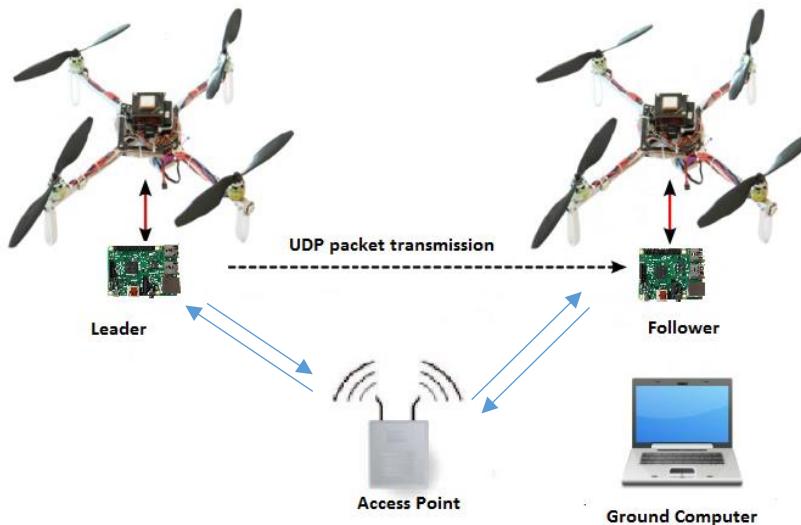


Figure 6-27: Leader-follower communication interface

6.7.4.4 Thread Management

Threads are utilised to allow multiple operations to run concurrently in the same process space. In the python programming language, threads are created by using the *threading* library. The software written utilises three threads in order to allow vehicle parameters to be sent from leader to the follower UAV and they are discussed as follows:

- **Sender-thread:** This thread is run on the on-board computer connected to the leader UAV and is responsible for sending the vehicle parameters to the follower UAV. The UDP client socket is created on this thread. In addition, it generates a checksum for the data that is sent across the network. This is achieved using the *hashlib* library.
- **Receiver-thread:** This thread is run on the on-board computer connected to the follower UAV and is responsible for receiving the incoming vehicle parameters sent by the leader UAV. The UDP server socket is created on this thread. A checksum validation is done on this thread to ensure the packets sent by the Sender-thread match the packet received by the Receiver-thread. This is important because the UDP protocol does not provide an option for error checking when transmitting data. Once the checksum validation has been performed, the parameters are sent to a *Queue* data structure.
- **Task-handler:** This thread is also run on the on-board computer connected to the follower UAV. It extracts the parameters inserted into the *Queue* data structure by the *Receiver – thread*. Once the parameters have been obtained, the thread runs the formation flying algorithm discussed in Section 6.7 to allow formation flying between the UAV pair.

Both the Sender and Receiver-threads are responsible for closing their socket connections depending on certain conditions being satisfied. In the Sender thread, the socket connections are closed when the mode of the leader UAV is no longer in LOITER mode, whilst in the Receiver thread the socket connections are closed if either of the following conditions is true:

- The Receiver-thread has not received any incoming message from the Sender-thread within the last 5 seconds.
- The mode of the follower vehicle is no longer in GUIDED mode. Section 6.7.5 below discusses the various flight modes

6.7.5 Flight Modes

Depending on the flight mode, the vehicle will react differently to the RC signals sent to it. Some flight modes require GPS lock and other flight modes do not require GPS lock. The most important flight modes are

- Stabilise
- Altitude Hold
- Loiter
- Return to Launch (RTL)
- GUIDED

Table 6-5 below provides a high-level overview of the main flight modes.

Table 6-5: Main flight modes

Stabilize	Allows the user to fly vehicle manually, but self - levels the roll and pitch axis. The user needs to frequently input roll and pitch commands to keep the vehicle in position.
Altitude Hold	Vehicle maintains a consistent altitude while allowing roll, pitch and yaw to be controlled manually.
Loiter	Vehicle automatically attempts to maintain the current location, heading and altitude provided a good GPS fix is available. The user may still fly the vehicle in Loiter mode as if it were in Stabilize or Altitude mode
Return to Launch (RTL)	In this mode, the vehicle navigates from its current position and returns back to its home position
GUIDED mode	This mode allows the aircraft to be flown without setting a mission. It can be used to dynamically guide the vehicle to a target location wirelessly using a GCS or telemetry radio.

In addition to these primary flight modes there are some secondary modes such as Circle, Acro, Sport and Land.

6.8 Flight Test Results and Discussion

Figure 6-28 shows the overall system architecture represented by flowcharts. As discussed earlier, the leader has the capability to be programmed to follow a series of GPS waypoints by sending mission commands or can be controlled manually by an R/C UAV pilot. In the following set of experiments the leader UAV is controlled by an R/C pilot and the follower UAV receives positional information from the leader. The follower is required to take off to a safe height (5m) before receiving positional information from the leader. This was done as a safety measure in order to prevent collisions at low altitudes due to external disturbances such as wind.

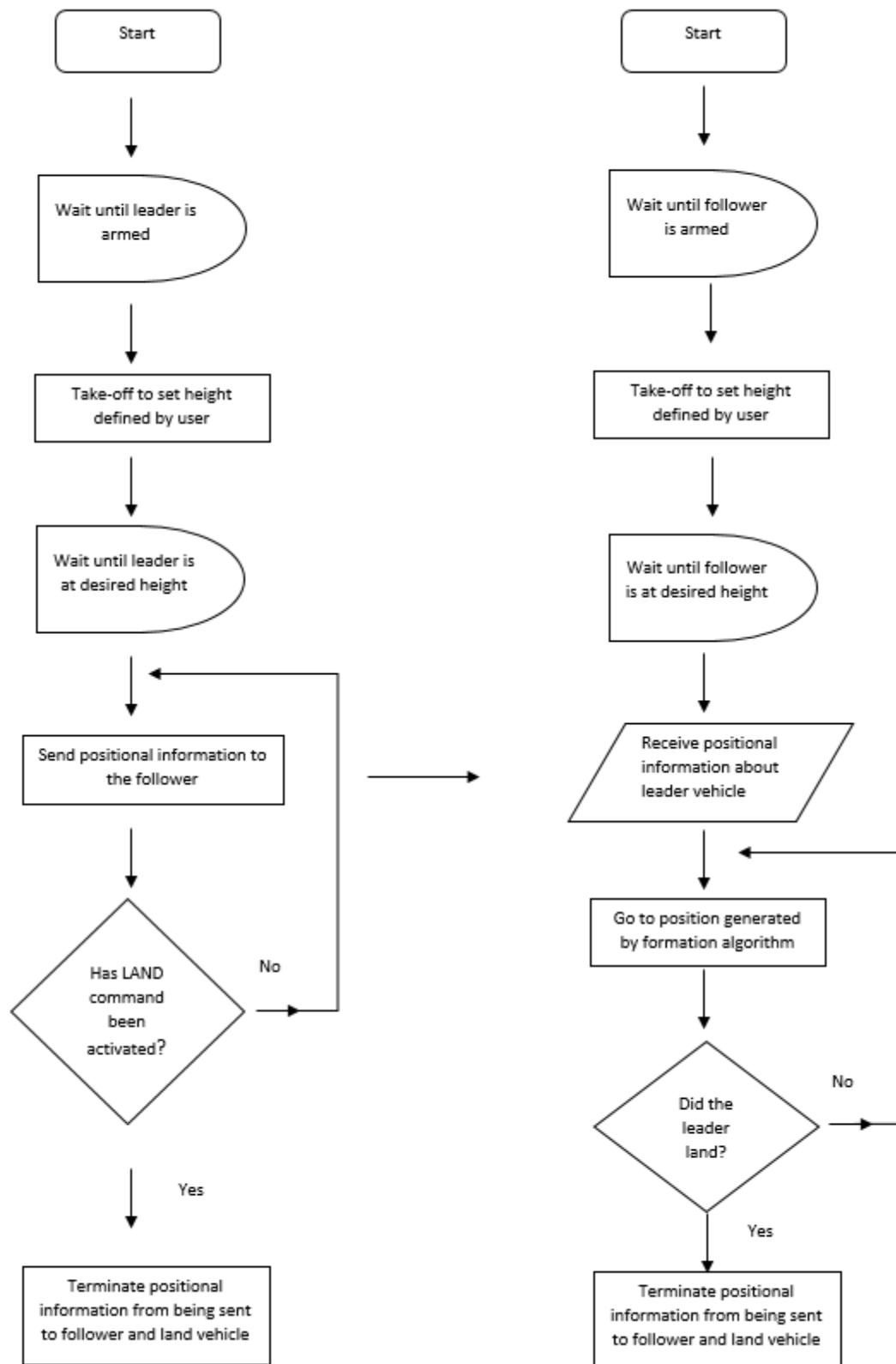


Figure 6-28: Logical flowchart for leader (left) and follower (right)

Experiment 1:

The objective of this series of experiments is to determine if the follower can autonomously track the leader's trajectory whilst maintaining a given separation distance using the proposed formation flying algorithm described in Section 6.7.1 . In this series of experiments, the pilot was flying the leader aircraft in a square pattern with a fixed heading angle (0°). Due to the positional accuracy of the GPS receiver being approximately 2.5m (U-blox, 2017) it was decided the safest minimum separation distance for formation flying should be $\geq 5\text{m}$. This would prevent the two UAVs colliding into each other as a result of external disturbances such as wind and in addition give enough room for pilot manoeuvre in the case of GPS glitches or possible loss of communication between the leader and follower.

Two distances were considered 5m and 7m. Distances greater than 7m were not investigated due to limited range of the Wi-Fi connection. For each distance, three flights were conducted at a wind speed of 8mph. Table 6-6 and Table 6-7 show the results for each distance respectively. The target distance denotes the expected separation distance between the leader and follower. The measured distance is the actual distance between the leader and follower. The measured or actual distance is calculated by using the GPS information of each vehicle as input into the haversine's formula (Mahmoud and Akkari, 2016) which is expressed as:

$$d = 2rsin^{-1} \sqrt{\sin^2\left(\frac{\theta_2 - \theta_1}{2}\right) + \cos(\theta_1)\cos(\theta_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)} \quad (6-9)$$

Where d = distance between the leader and follower in metres, r is the earth's radius (mean radius=6,371km), θ is the latitude expressed in radians and finally, λ is longitude also expressed in radians.

Table 6-6: 5m separation distance for fixed heading angle

	Test 1	Test 2	Test 3	Average
RMS error between target and measured distance (m)	0.70	0.65	0.68	0.68
RMS error for altitude comparison between leader and follower (m)	1.2	1.1	1.15	1.15

Table 6-7: 7m separation distance for fixed heading angle

	Test 1	Test 2	Test 3	Average
RMS error between target and measured distance (m)	0.77	0.79	0.72	0.76
RMS error for altitude comparison between leader and follower (m)	0.55	0.45	0.48	0.49

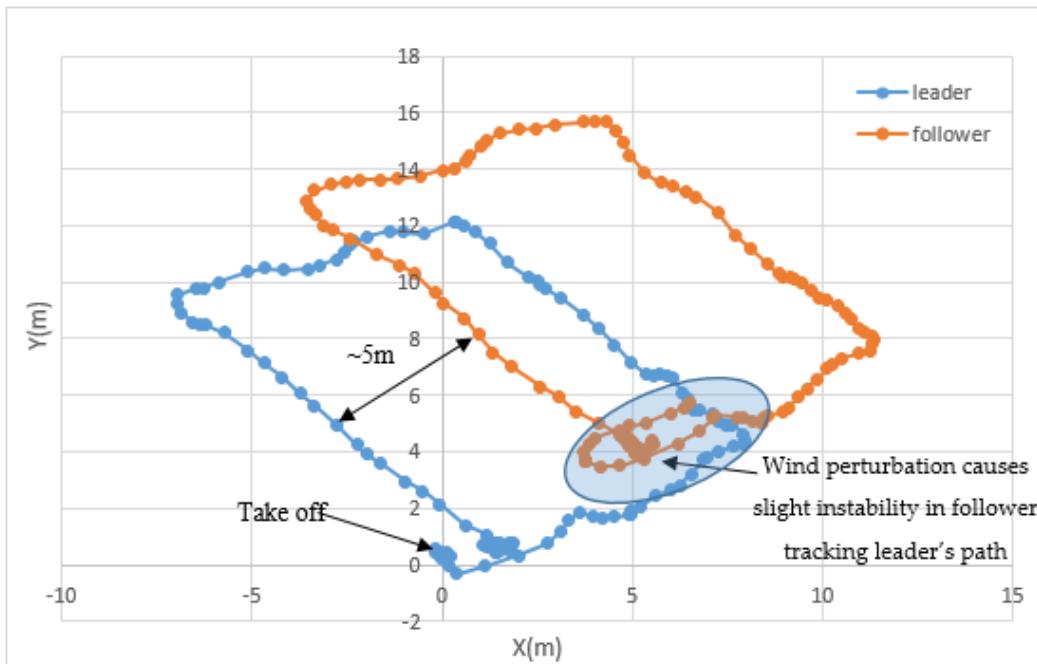


Figure 6-29: Flight path for leader and follower during square pattern flight-test

For both 5m and 7m separation tests, the follower was able to successfully track the leader's flight path and maintain formation despite the windy conditions. Figure 6-29 above shows the flight path of the leader and follower for the 5m separation test. As the leader flew the square pattern trajectory, the follower was able to maintain the same trajectory at the set distance apart despite the presence of external circumstance such as wind and atmospheric pressure, which affected the UAVs trajectory tracking performance.

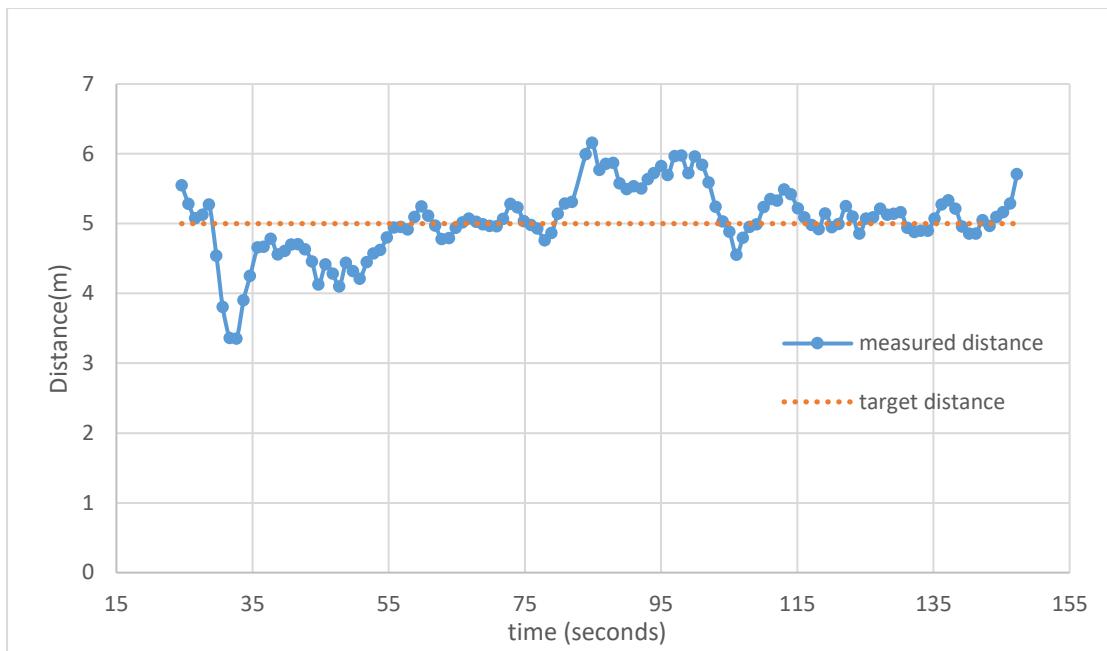


Figure 6-30: Distance between leader and follower during square pattern flight-test

Figure 6-30 above shows the relative distance between the leader and follower. The measured distance was computed using the haversine's formula as shown in Equation (6-9). The average RMS error between the measured distance and the target distance is 0.68m. Besides the possible influence of wind, which can affect performance as discussed in the previous paragraph, error in distance computation, as a result of the low positioning accuracy associated with the GPS receivers can also contribute to the RMS error.

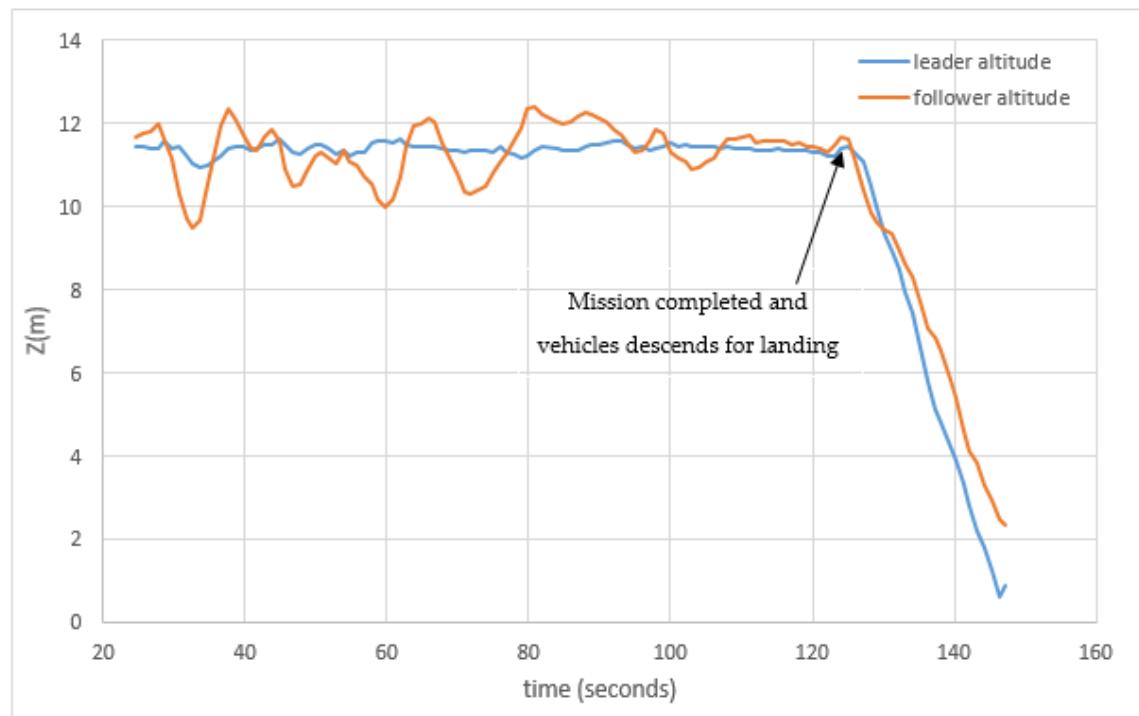


Figure 6-31: Altitude comparison between leader and follower during square pattern flight-test

Figure 6-31 shows the altitude comparison between the leader and follower during the flight test. Lidar-lite sensors were mounted on both the leader and the follower in-order to determine the true height above ground level. Although the follower UAV was programmed to be the same height as the leader in the software, in reality, some oscillation was experienced on the Z-axis due to windy conditions and low sampling rate from the GPS sensors. Hence, the large average RMS error for altitude comparison for both the 5m and 7m tests is shown in Table 6-6 and Table 6-7 respectively.

Experiment 2:

The objective of this experiment was to ascertain if the follower aircraft can maintain a given separation distance, as the heading angle of the leader is dynamically altered from 0°-359° from a stationary position. Similar to experiment 1, the tests were conducted for a 5m and 7m separation distance. During this experiment, the pilot was tasked with keeping the leader in a stationary position whilst altering the heading angle. Flights were conducted at a wind speed of 8mph.

Table 6-8:5m separation flight results for varying heading angle

	Test 1	Test 2	Test 3	Average
RMS error between expected and measured distance	0.55	0.52	0.56	0.54
RMS error for altitude comparison between leader and follower (m)	0.76	0.78	0.72	0.75

Table 6-9:7m separation flight test results for varying heading angle

	Test 1	Test 2	Test 3	Average
RMS error between expected and measured distance	0.51	0.52	0.54	0.52
RMS error for altitude comparison between leader and follower (m)	0.61	0.59	0.63	0.61

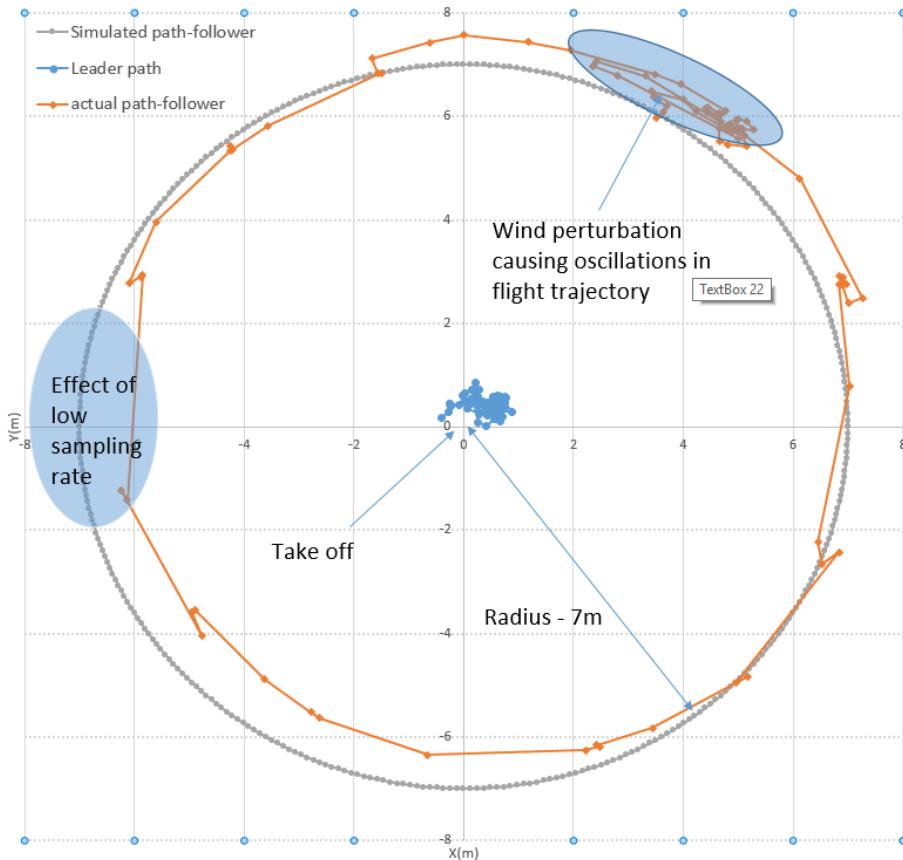


Figure 6-32: Flight results in X-Y plane

Figure 6-32 above shows flight path for the leader and follower for the 7m separation test. For the follower trajectory, the simulated and actual flight path is shown. The slight deviations in the position of the leader were due to the pilot not being able to keep the UAV at the same position because of the windy conditions. Also as seen in Figure 6-32 , the circular trajectory of the follower during simulation does not exactly match the circular trajectory obtained during the field test. The average RMS error between simulation (target) and measured distance is 0.52m. Besides the effect of wind and poor positioning accuracy of the GPS receivers, other sources of errors are the low sampling rate associated with the GPS receivers (10Hz) (U-blox, 2017). For example, if the time interval between consecutive GPS readings is large (low sampling rate) details of the trajectory of the UAV are lost and can lead to uncertainty during measurement. This effect is shown in Figure 6-32 and explains why the follower

trajectory cuts across the x-axis at roughly -6m rather than the expected value of -7m. In general, despite the effect of wind and low sampling rate of GPS sensors, the follower vehicle was able to maintain formation and stay on course during the experiment.

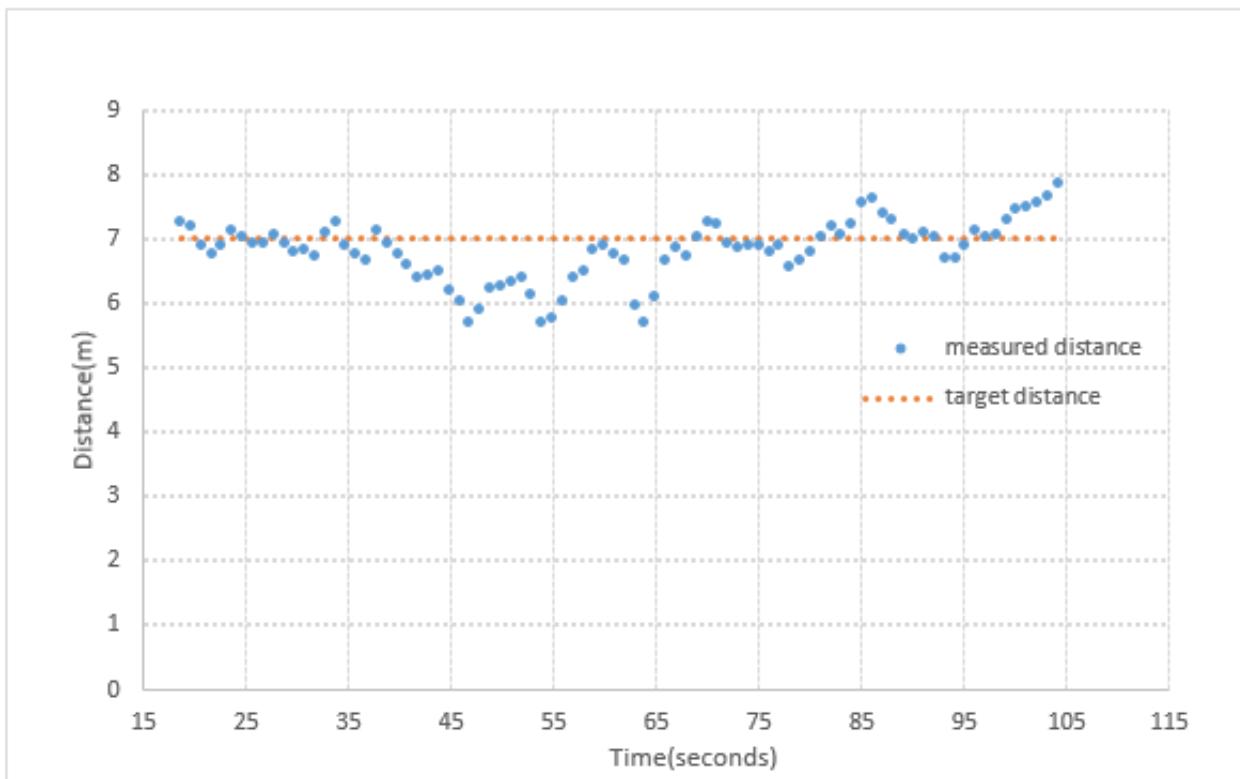


Figure 6-33: Distance between leader and follower during flight-test

Figure 6-33 above shows the comparison between the target (expected) distance and the measured distance. Similar to experiment 1, the distance was computed using the haversine's formula in Equation (6-9). The average RMS error between the target and measured distance is 0.52m as shown in Table 6-9. The error can be attributed to poor positioning accuracy in GPS receivers, low update rate and windy conditions.

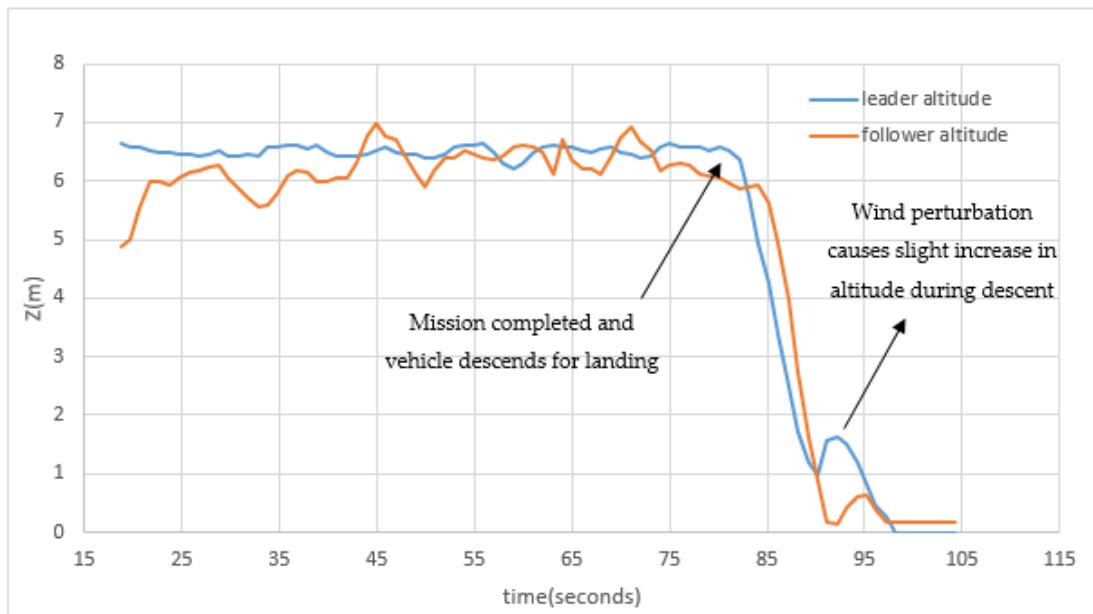


Figure 6-34: Altitude comparison between leader and follower during flight-test

Figure 6-34 above shows the altitude comparison between the leader and follower.

Although both UAVs were programmed to be at the same altitude, oscillations were experienced on the Z-axis similar to Experiment 1. Again, the lidar-lite was used for determining the altitude of the vehicle above ground level. The average RMS error in altitude between the follower and leader was 0.61m for the 7m separation test.

For both the 5m and 7m separation tests, the follower was able to maintain the given separation distance and altitude from the leader to within 1metre tolerance as shown in Table 6-8 and Table 6-9 despite the windy conditions, poor accuracy and low sampling rate of the GPS receivers. Figure 6-35 shows a snapshot of the leader and follower vehicle during a flight.



Figure 6-35: Snapshot of leader-follower formation flight

6.9 Conclusions

This chapter reviewed various flight controllers for UAVs in order to identify a suitable controller. Based on the review, the Pixhawk flight controller, which is an open source, low cost autopilot was chosen and implemented in this work. The chapter also reviews the use of a GCS software called Mission planner in order to set up and configure the Pixhawk autopilot parameters. In addition, it presents the use of Dronekit API and MAVProxy software, which provide a higher-level interface for autonomous UAV control through the MAVLink protocol. In addition, the chapter presents the formation flying algorithm utilised for controlling two UAVs. From the simulation results, at headings of 90 degrees and 270 degrees undefined solutions were obtained due to the nature of the mathematical solution implemented in deducing the coordinates of the follower. Trapping of these undefined cases, in conjunction with coding of the appropriate solution has been implemented to provide consistent solutions at all headings. Actual flights test have been carried out to validate the algorithm.

The follower was able to maintain the given separation distance from the leader at any given heading angle to within 1m tolerance through the course of the experiments despite windy conditions, low sampling rate and poor accuracy of GPS sensors. Tests Communication between leader and follower was achieved through UDP protocol. Now that coordinated control between a leader and follower has been achieved, the next chapter will focus on real-time wave height measurement with the leader using the stereo cameras.

Chapter 7: Case Study: 3D reconstruction and wave height measurement over an aquatic body

7.1 Introduction

Experiments conducted in the laboratory as discussed in Chapter 5, polluted the water surface with the addition of a thin opaque film in order to provide images with high enough contrast to allow successful reconstruction of the free surface. However, in open waters pollution of the water surface is not practicable. The main goal of the field tests is to assess the feasibility of utilising airborne stereo imagery for dense 3D reconstruction in open waters. A calibrated pair of stereo cameras have been mounted on the leader vehicle for the purpose of rapid dense 3D reconstruction of the water surface of a lake.

7.2 Experimental Set-Up

The aerial sensing platform used for 3D reconstruction and wave height estimation was the leader UAV. The calibrated stereo cameras mounted on board the vehicle, captured images at a resolution of 320 x 240 pixels in order to avoid high computational processing times. A narrow baseline of 9.2cm between the stereo cameras was used in order to maximize the overlap in the Fields of View (FOV) of the camera setup.

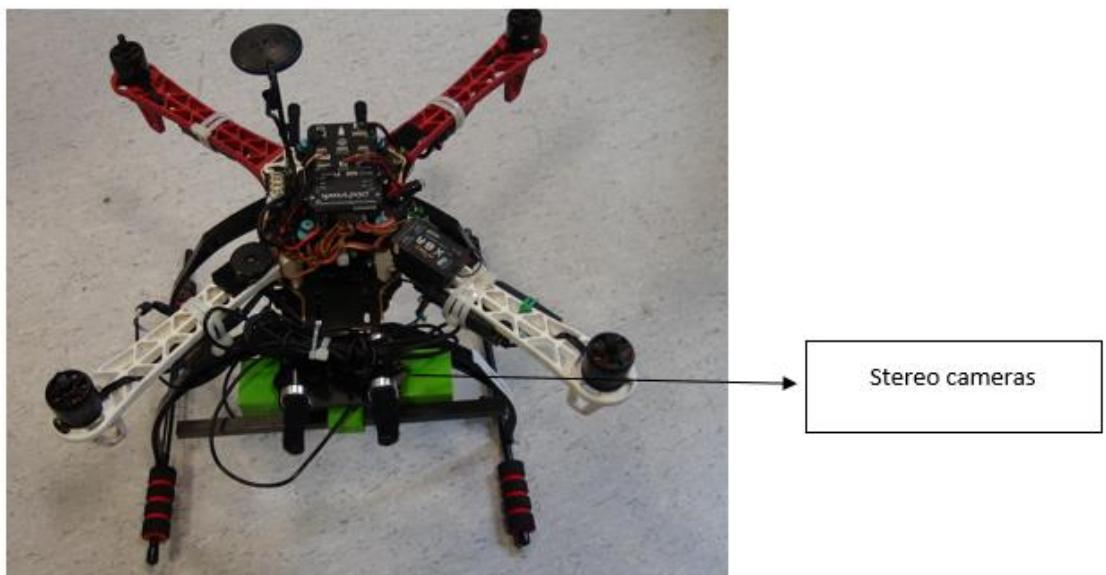


Figure 7-1: Leader UAV attached with stereo cameras

A major difference between the test conducted in the laboratory set up as discussed in Chapter 5 and the setup discussed in this chapter is that stereo cameras are susceptible to motion during data acquisition, which can introduce noisy measurements when taking readings. Two steps were taken to compensate for this

1. Proportional Integrative and Derivative (PID) tuning of the vehicle in order to allow for optimized stabilisation performance.
2. Motion correction using Inertial Measurement Unit (IMU) data.

A well-tuned vehicle allows for good loiter performance. Thus, it mitigates the effect of drift and can allow data acquisition of the water surface from a stationary hover. The vehicle was tuned using the AutoTune mode available in the ArduCopter firmware. In this mode, the flight controller twitches the vehicle to measure its response to figure out the optimum PID settings in order to achieve the least overshoot with the fastest response.

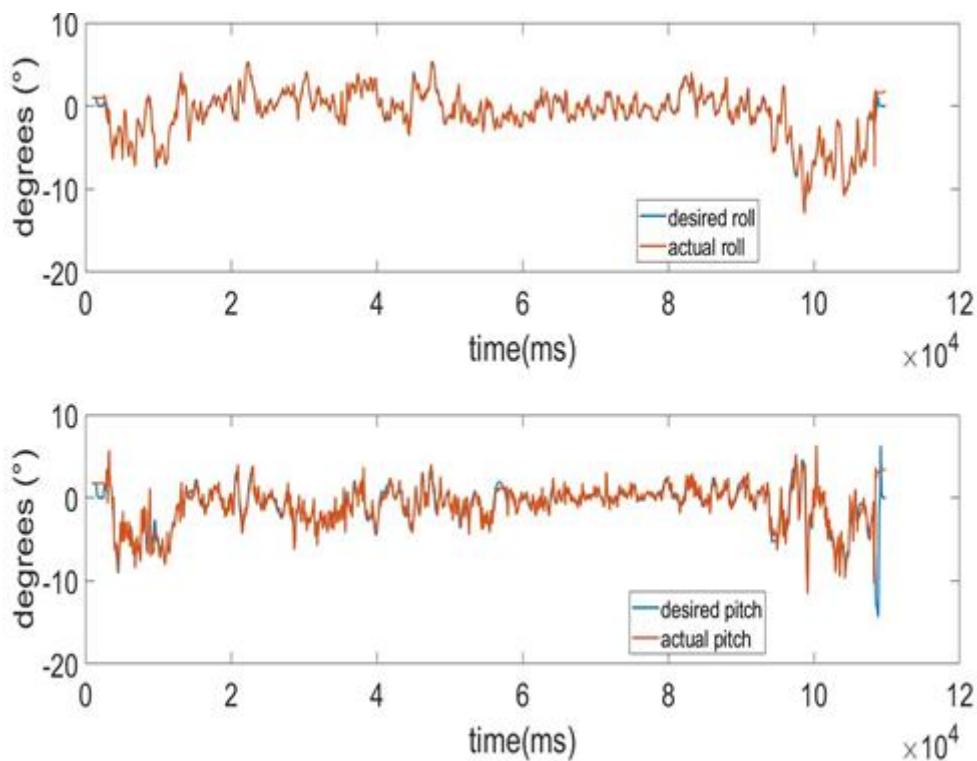


Figure 7-2: Top - Roll response of tuned vehicle and Bottom - Pitch response of tuned vehicle

Figure 7-2 above shows the roll and pitch response from a well-tuned vehicle after utilising the AutoTune feature. The desired roll angle is the angle requested by the pilot (roll left is negative and right is positive). The actual roll angle is the vehicle's actual roll in degrees. Similarly, the desired pitch is the pilot desired pitch angle while actual pitch is the vehicle's actual pitch in degrees (pitch forward is negative, pitch back is positive). As seen in Figure 7-2 both the desired roll and pitch angle follow quite closely to the actual roll and pitch angle indicating that the vehicle was

successfully tuned. Under these conditions, the vehicle is now capable of keeping its current position, heading and altitude during data acquisition with the stereo cameras.

In the presence of external disturbances such as wind, slight adjustments can be made using the throttle, roll and pitch control sticks in order to allow the vehicle to maintain a stationary hover. The rolling and pitching of the vehicle will effectively cause the sensor's angle from vertical to vary with time. In order to compensate for this, a method similar to Christensen et al. (2013) is adopted. In their work, data from an ultrasonic sensor is combined with a motion correction device to provide a time series of a sea surface elevation. The water surface elevation was described as:

$$\partial = S_z - D \cos(\phi_p) \cos(\phi_r) \quad 7-1$$

The symbol ϕ_p denotes the pitch angle and ϕ_r , the roll angle measured by an Inertial Measurement Unit (IMU) attached to the sensor. D represents the distance to the water surface measured by the ultrasonic sensor and S_z is the true vertical displacement which was estimated by integration of the absolute vertical velocity measured by the IMU.

Equation 7-1 was used for deriving the water surface elevation. ϕ_p and ϕ_r angles were obtained from the IMU embedded in the flight controller. This was done using the Dronekit API discussed in the chapter 6. The distance D was obtained from the stereo cameras by averaging the 3D point clouds reconstructed from an image pair. Finally, S_z was obtained as the vertical displacement above the water surface from the UAV. In order to obtain a reliable estimation of the displacement, the Ardupilot firmware provided the functionality for fusing measurement readings taking by the accelerometers, barometer and GPS sensors.

7.3 Data collection and processing

The experiment was carried out on a lake in Pulford Approach, Chester (*lat* = 53.117274 and *lon* = 2.905347). The lake was surrounded by trees, which mitigated the effect of wind during data acquisition. The wind was estimated to be 6mph. With the exception of small capillary waves visible on the water surface, the water surface was calm.

Figure 7-3 below shows a snapshot of the leader flown above a lake in order to capture a series of images for processing during one of the experiments. The UAV was flown at an altitude less than 0.8m in order to accommodate the working range of the stereo cameras, which was investigated in Chapter 5.



Figure 7-3: Snapshot of leader UAV acquiring image frames of the water surface with on-board stereo cameras

The acquired images from the stereo cameras are wirelessly sent to the Ground PC for processing. Processing was done off-board in order to take advantage of the higher processing power available with the Ground PC. The ground PC used was a MacBook

Pro equipped with an Intel Core i5 CPU at 2.7GHz, 8GB RA M and ran a Linux Operating System.

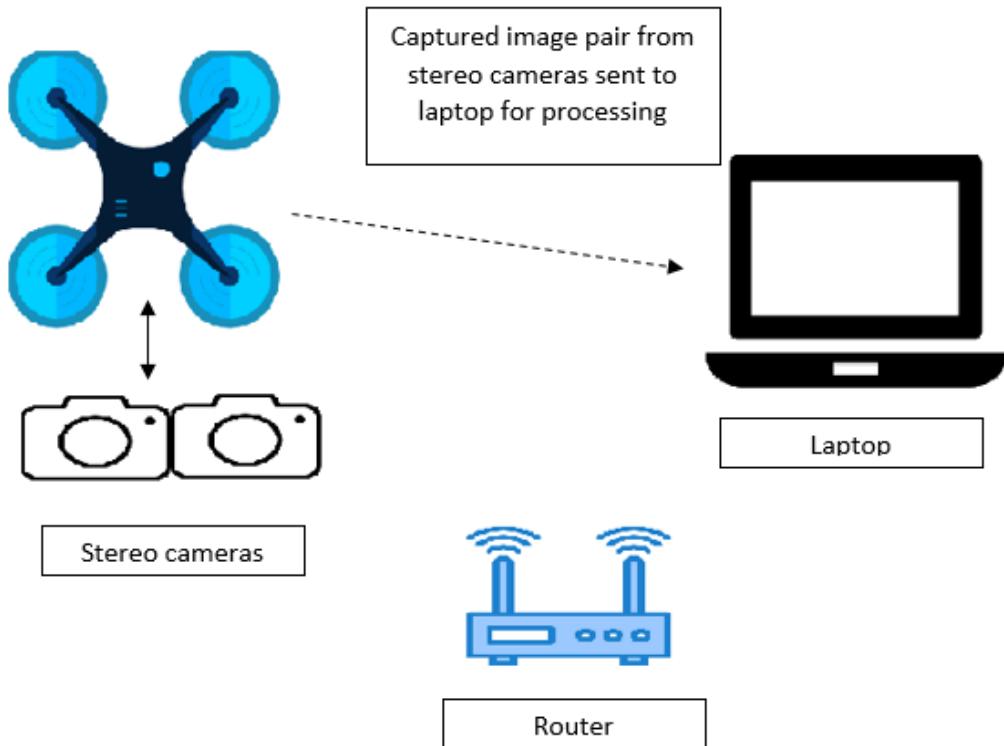


Figure 7-4: Image acquisition setup

Wireless streaming of image frames to the PC was achieved using the ZeroMQ Library (Hintjens, 2014). A 1-2 second lag was noticeable between successive frames acquired by the stereo cameras. During an acquisition session, all the acquired images are processed through the pipeline describe in Figure 7-5 below. The system was implemented in C++ using both OpenCV and PCL libraries.

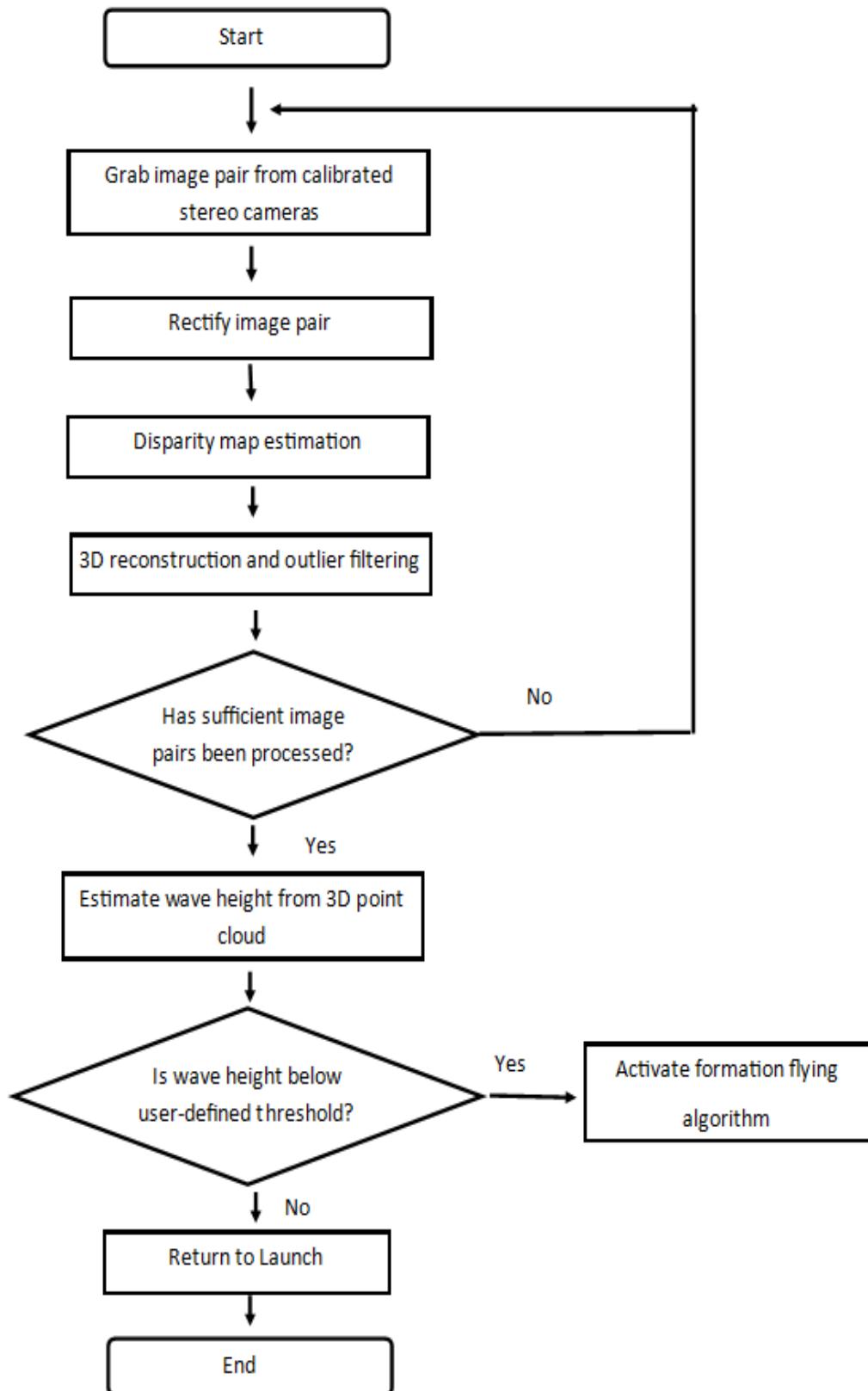
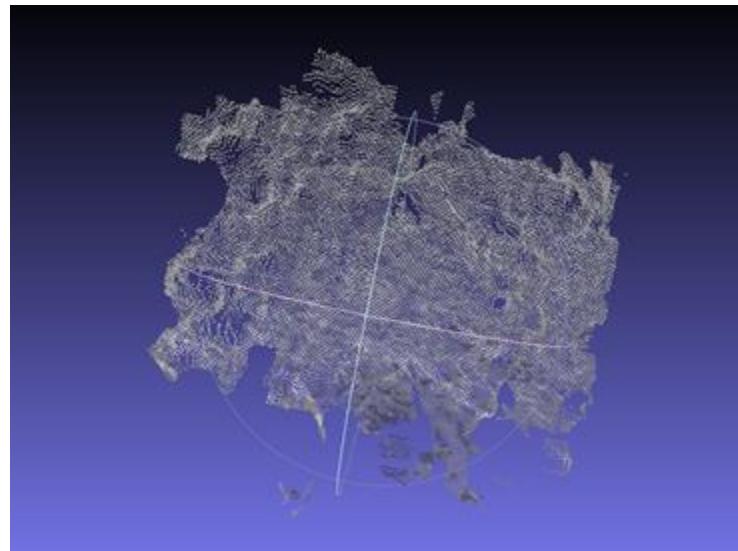
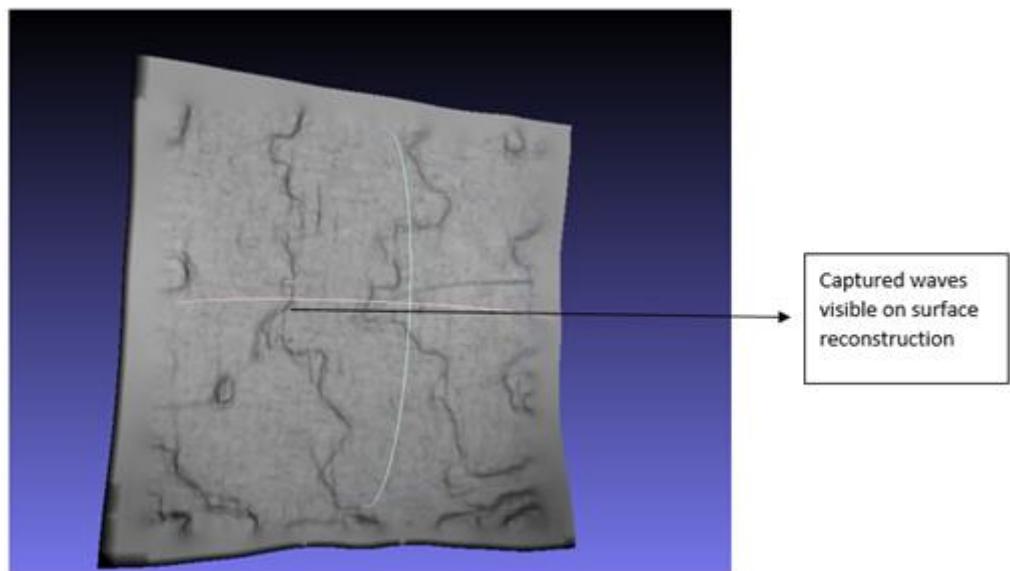


Figure 7-5: System overview

Figure 7-5 above shows a complete system overview of the methodology adopted in this experiment. Each image pair captured with the stereo cameras was rectified using the extrinsic and intrinsic calibration data. At the end of this operation, the epipolar lines are aligned with the image rows. The search for corresponding points between the acquired image pair was performed on the rectified images resulting in a smooth disparity map estimation. The SGM matching method was used for estimation of the disparity map since it produces a dense 3D point cloud as discussed in Chapter 5. Finally, a dense scattered 3D point cloud of the water surface was generated by triangulating all the corresponding pixels identified in the stereo matching process. In order to eliminate the number of point outliers that can still be present, planar-based segmentation is performed on the acquired 3D data set. After segmentation, the resulting point cloud is clean and the residual outliers have been removed. For example Figure 7-6 (a) below shows a 3D reconstruction of the water surface. Up to 21,000 3D points were generated from the images extracted within the field of view of the camera. A mesh reconstruction of the point cloud data is shown in Figure 7-6 (b) in order to show the captured waves on the water surface. The open-source software MeshLab has been used to generate the mesh as network of triangles, which interpolates the surface of the imaged scene.



(a)



(b)

Figure 7-6: Captured 3D scattered point cloud (a) and surface reconstruction of scattered point cloud (b)

The resulting 3D points are averaged across the entire data set. Since the point clouds are formed in the reference frame of the stereo cameras, the resulting average point encodes the distance D from the stereo cameras to the water surface. Now that D has been estimated, the water surface elevation can be deduced using equation 7-1.

Figure 7-7 below shows the water surface elevation captured over a time duration.

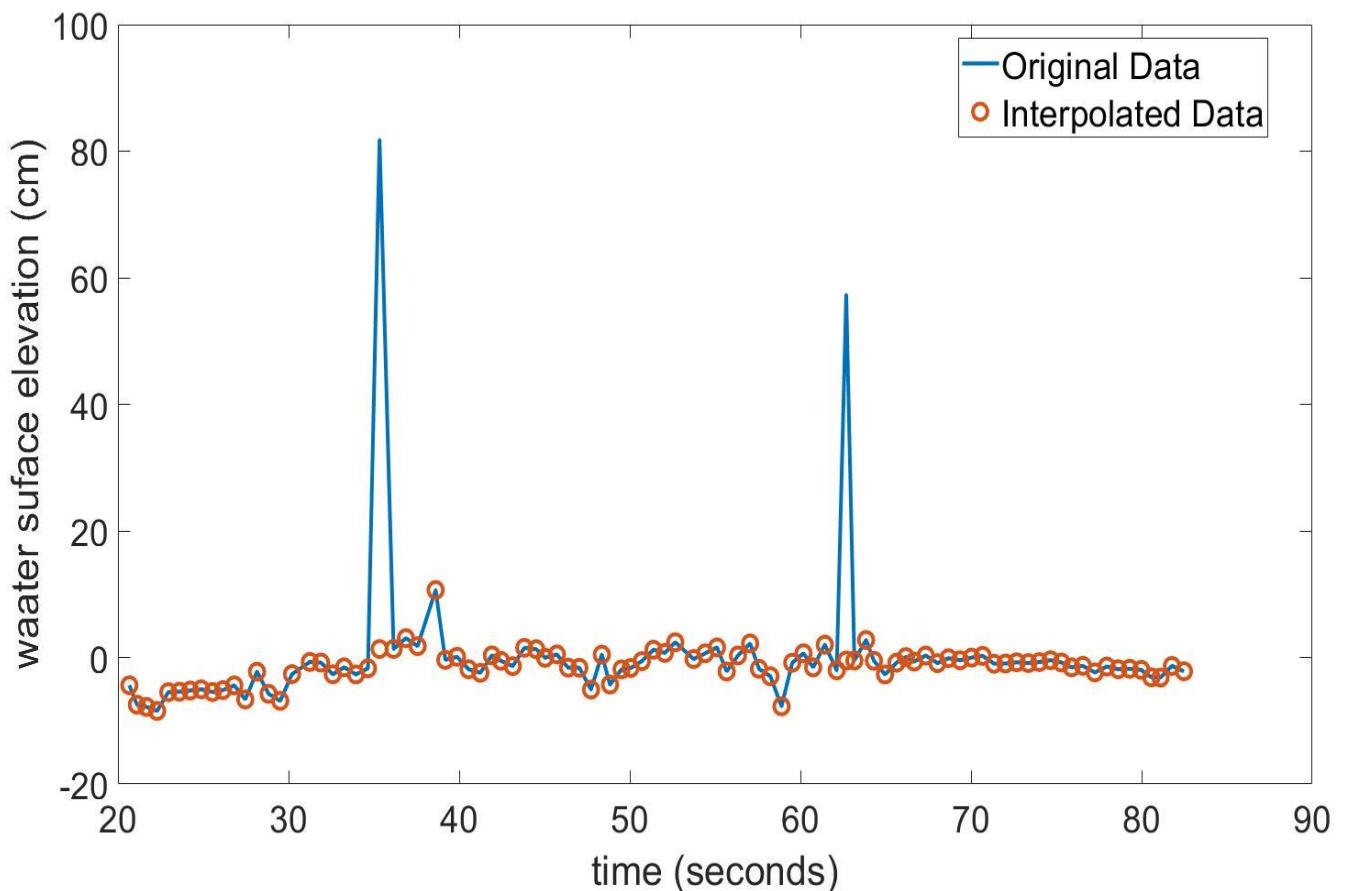


Figure 7-7: outlier detection from stereo measurements

As seen in Figure 7-7, spikes occurred in the time series data because of outliers present in the data set. Outliers occur due to poor disparity map estimation from a set of acquired stereo frames resulting in erroneous values in the derived water surface displacement. A possible explanation for this is because the brightness of the sky is often inhomogeneous. Hence, under such conditions what the camera sees is not the water surface itself, but a distorted reflection of the incident (sky) light. Consequently, the stereo matching approach for estimating the disparity map can cause unwanted errors. In order to eliminate the presence of these outliers during runtime, the outliers were defined as data points that were more than three times the standard deviation from the mean of the acquired data set. The circular markers denoted in Figure 7-7 show the acquired data set after removal of the outliers. As seen in the plot, filtering through this approach allows for reliable outlier detection in an acquired data set. The outlier points detected have been replaced by the neighbouring value.

Now the outliers in the data set have been removed, the Zero Up-Crossing technique can be used to isolate the crest and trough from the water surface elevation over a given time series. The wave height again is calculated as the difference between the maximum elevation of a crest and the minimum elevation of a trough. Figure 7-8 below shows the detected crest and trough over a given time series.

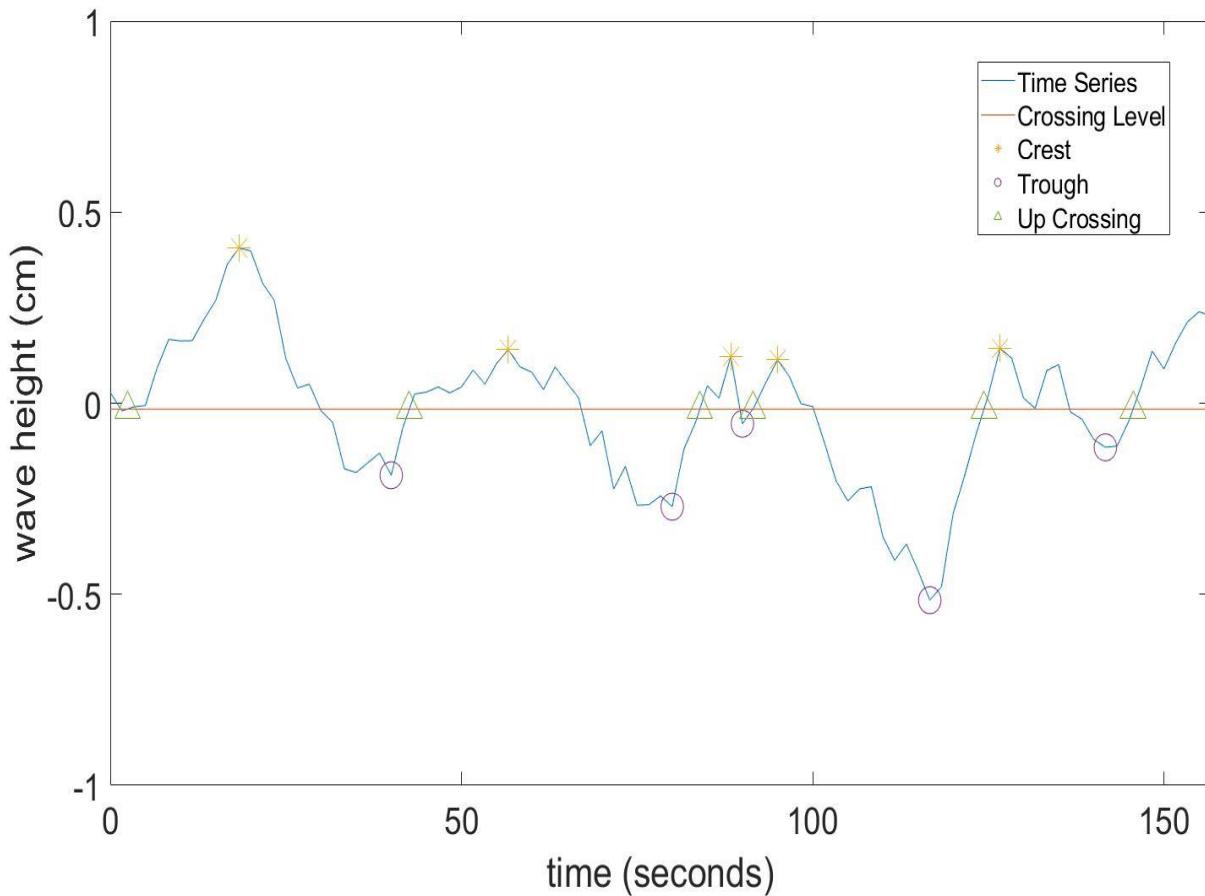


Figure 7-8: water surface elevation measurement acquired from airborne stereo cameras

In order to quantitatively verify the accuracy and reliability of the processing pipeline, the measurements from the UAV are compared to observations obtained from a ground stereo system looking downwards at the water surface as shown in Figure 7-9 below. Identical camera models were used to the ones mounted on the UAV. For the developed prototype, the cameras sit on an aluminium section bar. The section bar is mounted on a tripod to allow fast and flexible deployment during field experiments.



Figure 7-9: Ground stereo camera system mounted on a tripod for measuring water surface elevation

Table 7-1: Comparison of average wave height

Number of repetitions	Ground stereo (cm)	Airborne stereo (cm)	Absolute error (cm)
1	0.243	0.378	0.135
2	0.400	1.393	0.993
3	0.518	1.880	1.362
4	0.302	0.819	0.517

Table 7-2: Comparison of significant wave height

Number of repetitions	Ground stereo (cm)	Airborne stereo (cm)	Absolute error (cm)
1	0.387	0.634	0.247
2	1.034	2.620	1.586
3	0.893	3.338	2.445
4	0.454	0.819	0.365

Table 7-1 and Table 7-2 show comparison between the average wave height and significant wave height estimated for the ground stereo system and the airborne stereo system. Stereo images were for 58 seconds at a rate of 2Hz to yield a complete record of the water surface evolution. The mean absolute error for the number of repetitions taken is 0.752cm for the average wave height and 1.161cm for the significant wave height. The measurements from the stereo cameras mounted on the vehicle were slightly higher than the measurements observed from the fixed stereo system. A possible reason for this is the downwash effect induced by each rotor of the aerial vehicle causes ripples on the water surface that have a signature that differs considerably from the ones observed by the fixed stereo camera system.

7.4 Conclusion

In this chapter, the feasibility of utilising airborne stereo cameras for 3D reconstruction of open waters has been investigated. Experimental results show that open waters possess sufficient texture to allow dense 3D reconstruction of the water surface. Wave characteristics such as average wave height and significant wave height are deduced from the reconstructed 3D data using the Zero Up-Crossing analysis. Motion correction using an IMU has been applied to the water surface elevations obtained from the stereo cameras in order to compensate for slight perturbations of the cameras during data acquisition. Measurement from the airborne stereo cameras have been compared to a ground based binocular stereo camera setup and the results show very good agreement in terms of the average wave height and significant wave height estimated. Since the leader UAV was able to accurately measure the wave height in multiple repetitions, the follower UAV can be commanded to proceed to the leader's position using the control strategy discussed in the previous chapter.

Chapter 8: Discussions, Conclusions and Future Work

Water sampling has become a key activity for managing fresh water resources and maintaining public health. Current water sampling methods to monitor water quality are relatively slow, spatially restrictive or costly to deploy. One approach to solve these problems would be deploying an AUV from a UAV for the purposes of water sampling. Thus providing a fast, flexible and relatively low cost solution to the problem domain. However, a major challenge to overcome with the proposed solution is determining if the aquatic environment is safe for the landing of an UAV for the purpose of AUV deployment. The approach taken in this thesis to overcome this, involves equipping a UAV with the features necessary to estimate the wave height in an aquatic environment. Two UAVs work together in a coordinated manner. One UAV, the “leader”, is equipped with two identical cameras that capture 3D information of the wave through non-contact optical techniques (stereovision). The second UAV, “the follower”, can be equipped with the payload (AUV). Once the leader has identified the water’s surface as suitable for landing (low wave-height) the follower UAV can then proceed to the location identified by the leader, land and deploy the AUV into water. Such an approach provides a UAV with the decision-making capabilities to ensure autonomous safe landing in an aquatic environment.

8.1 Discussions and Conclusions

3D reconstruction has been the subject of considerable research for a number of years and many publications have been produced outlining new techniques, refinements to existing methods and lessons learnt from practical applications. Chapter 2 addresses some of these popular reconstruction techniques. A stereovision approach for acquisition of three-dimensional surface wave measurement provided a cost effective and robust solution. Stereo imaging systems used for reconstruction of water waves have either been tested in a laboratory environment or outdoors with cameras that have been mounted on fixed rigid structures. In this thesis, a novel setup has been introduced by mounting the stereo cameras on a UAV during 3D data acquisition of the water surface elevations. Such a setup allows observations of the wave phenomena from variable viewpoints and altitudes. Thus providing a more flexible spatial coverage in comparison to the traditional stereo imaging systems utilised by previous researchers. This thesis has successfully tackled the challenges faced in developing an airborne stereovision system for measuring the water surface elevations.

The principal conclusions that can be drawn from the work are listed below and cross-referenced against the original objectives:

Objective 1:

Identify the limitations of current technology for wave height estimation and investigate the use of 3D reconstruction techniques as an alternative.

Classical observation methods for wave height estimation rely on time series data retrieved from wave gauges and floating buoys. Besides being invasive and hence disrupting the wave motion, such methods measure the wave characteristics over

fixed positions. Methods that produce 3D representation of a surface can be successfully exploited to obtain rich information of the space-time wave characteristics. After conducting a critical review of existing 3D reconstruction techniques, a stereovision approach for 3D reconstruction was chosen because the technique is low cost, non-intrusive and can be applied in a dynamic environment such as moving waves.

Objective 2:

Developing a robust pipeline that allows for dense 3D reconstruction using the identified reconstruction technique.

In order to create a 3D representation of any surface, stereovision requires some critical stages. The first stage is the camera calibration to identify the intrinsic and extrinsic parameters of the stereo camera setup. The second stage is the identification of similar corresponding points between the images acquired from various perspectives and finally given the camera calibration parameters and identified corresponding points between the images, a 3D representation of the surface in world coordinates (XYZ) otherwise known as 3D point cloud can be found via triangulation. The camera calibration method adopted produced very accurate results of up to 0.5 pixels difference between the real image and a 3D point that has been reprojected using the calibration data. The density of the 3D point cloud produced is strongly dependent on the stereo correspondence algorithm applied to find similar points between the data sets. Feature based and Correlation based stereo algorithms have been investigated in this thesis. Preliminary results show that the feature based matching methods produced sparse 3D points in comparison to the correlation based matching methods in particular the SGM technique, which produced a dense 3D point of the surface of interest.

Objective 3:

Validation of the reconstructed 3D point cloud against known technologies, determining the accuracy in the context of wave height measurement.

Water waves have been generated in a wave tank and dense 3D point clouds of a free surface have been reconstructed using correlation based matching technique. The 3D points are remapped into elevation maps, where the 3D information contained in the point clouds is simplified to a one-dimensional domain containing wave elevations with respect to a fitting plane. The crests and troughs of the waves have been identified using the Zero Up-crossing analysis. Results from the stereo measurements show good agreement with data obtained from a wave probe in both time and frequency domain. Validation results showed that the mean absolute error for the average wave height and significant wave height is less than 1cm from the acquired time series data. In the frequency domain, it was discovered that the highest peak in the signal occurred at the wave frequency that was investigated for both stereo cameras and wave probe. Analysis of the peak signal, showed similar amplitudes observed between the stereo measurement and wave probe, R^2 value of 0.93 for the various frequencies investigated.

Objective 4:

Development of a formation-flying algorithm to allow for coordinated control of two UAVs that is beneficial in the context of the problem domain.

The notion of utilising multiple UAVs has a variety of benefits, with many of those are centred on the increase of efficiency for tasks being tackled by one UAV. Cooperative control of two UAVs has been achieved by developing a formation-flying algorithm. Given a leader's position and heading, the developed algorithm calculates in real time,

the position the follower UAV needs to navigate towards, to track the leader's trajectory from a given separation distance. The results show that the follower UAV was successfully able to track the leader's trajectory within 1m tolerance through the course of the experiments despite windy conditions, low sampling rate and poor positioning accuracy of GPS sensors.

Objective 5:

Near real time 3D reconstruction and wave height estimation from a series of images taken from a UAV.

As a case study, a pair of calibrated stereo cameras have been mounted on the leader UAV for the purpose of 3D reconstruction and wave height estimation of the water surface over a lake. A dense 3D reconstruction of the water surface was achieved generating up to 21,000 3D points from a pair of images taken at a resolution of 320 by 240 pixels. In order to compensate for slight perturbation of the camera position during data acquisition, which can introduce noisy measurements during computation of the water surface elevation, a method for motion correction using an IMU is discussed. The developed system has proven to be easy to deploy and set up; robust and effective in monitoring water surface elevation, providing a satisfactory accuracy and precision in multiple repetitions.

8.2 Future Work

8.2.1 Variable baseline Stereo

The conventional stereovision system adopted in this thesis, rely on two spatially fixed cameras to gather depth information about the scene of interest. As the baseline increases, in theory the estimated 3D information becomes more accurate, making it advantageous to have a large a baseline as possible. However, large baselines have problems whenever the scene of interest approaches the cameras. The region of interest begin to leave the field of view of the cameras, rather than deploy a fixed baseline stereo camera set up on a UAV, a variable baseline stereo camera set up can be adopted, where the cameras can automatically adjust their baseline making it impossible to determine where they are located in 3D space. An auto calibration method can be adopted to determine the pose between the stereo cameras. Such a setup can be achieved by the following steps:

1. Acquiring a subset of stereo frames as calibration images.
2. Extracting SIFT or SURF points from each image acquired.
3. Robustly matching feature points from the acquired stereo pair frames
4. Estimation of the essential matrix by exploiting RANSAC-based epipolar filtering.
5. Decomposition of the essential and performing a Sparse Bundle Adjustment to determine the camera poses

8.2.2 Validation in various aquatic environments

It is freely acknowledged that, experiments conducted outdoors took place in a controlled aquatic environment with very little or no waves, the waves observed were caused by the downward prop wash as the UAV hovers over the water surface during image acquisition. The reason for this is special permissions are required to approve flying over seas and oceans where higher waves can be observed. This is in accordance with the LJMU UAV operation manual since such regions are available to the public.

Accuracy assessment of the wave height obtained from the airborne stereo camera system in open oceans can be compared with recordings from a wave buoy running simultaneously. In addition comparisons of the statistics of the crest and trough heights obtained can be compared with well known theoretical models, such as the Boccotti model (Boccotti, 2000).

8.2.3 Communication between multiple UAVs

A communication infrastructure, relying on the UDP protocol to send and receive data related to mission semantics, has been utilised as a proof of concept in this thesis. To provide a scalable solution, an architecture based on aerial Mobile Ad-hoc Network (MANET) can be adopted for the management of data exchange among all the vehicles in a team. The MANET framework provides a wireless network that is formed by a collection of self-organizing mobile nodes. Each node communicates with its neighbours over a shared wireless medium. Alternative solutions include the use of digital cellular networks.

Chapter 9: References

- ALCANTARILLA, P. F., BARTOLI, A. & DAVISON, A. J. KAZE features. European Conference on Computer Vision, 2012. Springer, 214-227.
- ARDUPILOT DEV TEAM. 2016a. *Choosing a Ground Station* [Online]. Available: <http://ardupilot.org/copter/docs/common-choosing-a-ground-station.html> [Accessed 22/05 2018].
- ARDUPILOT DEV TEAM. 2016b. *Common Power Module* [Online]. Available: <http://ardupilot.org/copter/docs/common-3dr-power-module.html> [Accessed 22/05 2018].
- ARDUPILOT DEV TEAM. 2016c. *Learning the ArduPilot Codebase* [Online]. Available: <http://ardupilot.org/dev/docs/learning-the-ardupilot-codebase.html> [Accessed 21/05 2018].
- ARDUPILOT DEV TEAM. 2016d. *Software-in-the-loop* [Online]. Available: <http://ardupilot.org/dev/docs/sitl-simulator-software-in-the-loop.html> [Accessed 27/09 2017].
- ARDUPILOT DEV TEAM. 2017. *MAVLINK Common Message Set* [Online]. Available: http://mavlink.org/messages/common#GLOBAL_POSITION_INT [Accessed 09/09 2017].
- ATOEV, S., KWON, K.-R., LEE, S.-H. & MOON, K.-S. Data analysis of the MAVLink communication protocol. Information Science and Communications Technologies (ICISCT), 2017 International Conference on, 2017. IEEE, 1-3.
- BAKER, H. H. & BINFORD, T. O. 1981. Depth from edge and intensity based stereo. *Proceedings of the 7th international joint conference on Artificial intelligence*. Vancouver.
- BARNARD, S. T. 1989. Stochastic stereo matching over scale. *International Journal of Computer Vision*, 3, 17-32.
- BAY, H., TUYTELAARS, T. & VAN GOOL, L. Surf: Speeded up robust features. European conference on computer vision, 2006. Springer, 404-417.
- BAYDOUN, M. & AL-ALAoui, M. A. 2014. Enhancing stereo matching with classification. *IEEE Access*, 2, 485-499.
- BENETAZZO, A., FEDELE, F., GALLEGOS, G., SHIH, P.-C. & YEZZI, A. 2012. Offshore stereo measurements of gravity waves. *Coastal Engineering*, 64, 127-138.
- BIN, H. & JUSTICE, A. 2009. The design of an unmanned aerial vehicle based on the ArduPilot. *Indian Journal of science and Technology*, 2, 12-15.
- BLIDBERG, D. R. The development of autonomous underwater vehicles (AUV); a brief summary. Ieee Icra, 2001.
- BOCCOTTI, P. 2000. *Wave mechanics for ocean engineering*, Elsevier.
- BOLAM, R. C., VAGAPOV, Y. & ANUCHIN, A. Curriculum development of undergraduate and post graduate courses on small unmanned aircraft. Universities Power Engineering Conference (UPEC), 2017 52nd International, 2017. IEEE, 1-5.
- BOONSUK, W. INVESTIGATING THE EFFECTS OF STEREO CAMERA BASELINE ON THE ACCURACY OF 3D PROJECTION FOR INDUSTRIAL ROBOTIC APPLICATIONS. Proceedings of the IAJC-ISAM, 2016.
- BURGER, W. 2016. *Zhang's camera calibration algorithm: in-depth tutorial and implementation* [Online]. Available: <http://staff.fh-hagenberg.at/burger/publications/reports/2016Calibration/Burger-CameraCalibration-20160516.pdf> [Accessed March 2018].
- CALONDER, M., LEPETIT, V., STRECHA, C. & FUJII, P. Brief: Binary robust independent elementary features. European conference on computer vision, 2010. Springer, 778-792.
- CHAO, H., CAO, Y. & CHEN, Y. Autopilots for small fixed-wing unmanned air vehicles: A survey. Mechatronics and Automation, 2007. ICMA 2007. International Conference on, 2007. IEEE, 3144-3149.

- CHAPMAN, D. V. & WORLD HEALTH ORGANIZATION 1996. Water quality assessments: a guide to the use of biota, sediments and water in environmental monitoring.
- CHEN, Y. & NI, J. 1993. Dynamic calibration and compensation of a 3d laser radar scanning system. *IEEE transactions on robotics and automation*, 9, 318-323.
- CHOI, H., GEEVES, M., ALSALAM, B. & GONZALEZ, F. Open source computer-vision based guidance system for UAVs on-board decision making. Aerospace Conference, 2016 IEEE, 2016. IEEE, 1-5.
- CHRISTENSEN, RÖHRS, J., WARD, B., FER, I., BROSTRÖM, G., SAETRA, Ø. & BREIVIK, Ø. 2013. Surface wave measurements using a ship-mounted ultrasonic altimeter. *Methods in Oceanography*, 6, 1-15.
- COOLEY, J. W. & TUKEY, J. W. 1965. An algorithm for the machine calculation of complex Fourier series. *Mathematics of computation*, 19, 297-301.
- COOPER, R. 1995. *3D Stereovision* [Online]. Available: <http://www.vision3d.com/stereo.html> [Accessed November 2017].
- CORGNATI, L., MAZZEI, L., MARINI, S., ISOPPO, B., OTTAVIANO, E., BESIO, G. & MAGALDI, M. G. High resolution stereo imaging of sea waves for validation and optimization of wave modelling. OCEANS 2015-Genova, 2015. IEEE, 1-8.
- DE CUBBER, G., NALPANTIDIS, L., SIRAKOULIS, G. C., GASTERATOS, A. & DE, G. Intelligent robots need intelligent vision: visual 3D perception. RISE'08: Proceedings of the EURON/IARP International Workshop on Robotics for Risky Interventions and Surveillance of the Environment, 2008.
- DELON, J. & ROUGÉ, B. 2007. Small baseline stereovision. *Journal of Mathematical Imaging and Vision*, 28, 209-223.
- DEYAH, W. & BHAYA, W. 2017. Balancing between TCP and UDP to Improve Network Performance. *Journal of Engineering and Applied Sciences*, 12, 7755-7758.
- DJI. 2017. *DJI-robotic* [Online]. Available: <https://www.dji.com/flame-wheel-arf> [Accessed].
- DOWDESWELL, J. A., EVANS, J., MUGFORD, R., GRIFFITHS, G., MCPHAIL, S., MILLARD, N., STEVENSON, P., BRANDON, M. A., BANKS, C. & HEYWOOD, K. 2008. Autonomous underwater vehicles (AUVs) and investigations of the ice-ocean interface in Antarctic and Arctic waters. *Journal of Glaciology*, 54, 661-672.
- DRONEKIT. 2018. *Use a single communication link for several vehicles* [Online]. Available: <https://github.com/dronekit/dronekit-python/issues/516> [Accessed 22/05/ 2018].
- EBEID, E., SKRIVER, M. & JIN, J. A Survey on Open-Source Flight Control Platforms of Unmanned Aerial Vehicle. Digital System Design (DSD), 2017 Euromicro Conference on, 2017. IEEE, 396-402.
- FERRETTI, G., ZUNINO, A., SCAFIDI, D., BARANI, S. & SPALLAROSSA, D. 2013. On microseisms recorded near the Ligurian coast (Italy) and their relationship with sea wave height. *Geophysical Journal International*, 194, 524-533.
- FISCHLER, M. A. & BOLLES, R. C. 1981. Random sample consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24, 381-395.
- FORNAI, F., BARTALONI, F., FERRI, G., MANZI, A., CIUCHI, F. & LASCHI, C. An autonomous water monitoring and sampling system for small-sized ASV operations. 2012 Oceans, 2012. IEEE, 1-7.
- FULOP, S. A. & FITZ, K. 2006. Algorithms for computing the time-corrected instantaneous frequency (reassigned) spectrogram, with applications. *The Journal of the Acoustical Society of America*, 119, 360-371.
- FUSIELLO, A., TRUCCO, E. & VERRI, A. 2000. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12, 16-22.

- FUSIELLO, A., VITO, R. & EMANUELE, T. Efficient stereo with multiple windowing. Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1997 San Juan. IEEE, 858-863.
- GÅRDING, J. 1992. Shape from texture for smooth curved surfaces in perspective projection. *Journal of Mathematical Imaging and Vision*, 2, 327-350.
- GARY, M. O. & SHARP, J. M. 2006. Volcanogenic karstification of Sistema Zacatón, Mexico. *Geological Society of America Special Papers*, 404, 79-89.
- GODA, Y. & SUZUKI, Y. 1977. Estimation of incident and reflected waves in random wave experiments. *Coastal Engineering* 1976.
- GRASSHOFF, K., KREMLING, K. & EHRHARDT, M. 2009. *Methods of seawater analysis*, John Wiley & Sons.
- HAMZAH, R. A., RAHIM, R. A. & NOH, Z. M. Sum of absolute differences algorithm in stereo correspondence problem for stereo matching in computer vision application. Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on, 2010. IEEE, 652-657.
- HANSARD, M., LEE, S., CHOI, O. & HORAUD, R. P. 2012. *Time-of-flight cameras: principles, methods and applications*, Springer Science & Business Media.
- HARRIS, C. & STEPHENS, M. A combined corner and edge detector. Alvey vision conference, 1988. Manchester, UK, 10.5244.
- HARTLEY, R. & ZISSEMAN, A. 2003. *Multiple view geometry in computer vision*, Cambridge university press.
- HARTLEY, R. I. Euclidean reconstruction from uncalibrated views. In: MUNDY, J. L., ZISSEMAN, A. & FORSYTH, D., eds. Applications of Invariance in Computer Vision: Second Joint European 1994 Berlin, Heidelberg. Springer Berlin Heidelberg, 235-256.
- HARTLEY, R. I. 1997. In defense of the eight-point algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, , 19, 580-593.
- HARTLEY, R. I. 1999. Theory and Practice of Projective Rectification. *International Journal of Computer Vision*, 35, 115-127.
- HARTLEY, R. I. & STURM, P. 1997. Triangulation. *Computer vision and image understanding*, 68, 146-157.
- HEYS, W. & BACHMAYER, R. Assessing the viability of ultrasonic distance sensors for a wave profiling system. Oceans-St. John's, 2014, 2014. IEEE, 1-9.
- HILSENSTEIN, V. Surface reconstruction of water waves using thermographic stereo imaging. Image and Vision Computing New Zealand, 2005. Citeseer.
- HINTJENS, P. 2014. *Distributed Messaging* [Online]. Available: <http://zeromq.org/> [Accessed 15/05/2018].
- HIRSCHMÜLLER, H. 2008. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence*, 30, 328-341.
- HIRSCHMÜLLER, H. Accurate and efficient stereo processing by Semi-Global Matching and Mutual Information. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. 807-814.
- HIRSCHMÜLLER, H. 2011. Semi-global matching-motivation, developments and applications. *Photogrammetric Week* 11, 173-184.
- HIRSCHMÜLLER, H. & SCHARSTEIN, D. 2007. Evaluation of Cost Functions for Stereo Matching. *IEEE Conference on Computer Vision and Pattern Recognition*,. Minneapolis, MN, USA: IEEE.
- HONKAVAARA, E., SAARI, H., KAIVOSOJA, J., PÖLÖNEN, I., HAKALA, T., LITKEY, P., MÄKYNNEN, J. & PESONEN, L. 2013. Processing and assessment of spectrometric, stereoscopic imagery collected using a lightweight UAV spectral camera for precision agriculture. *Remote Sensing*, 5, 5006-5039.
- HORN, B. K. 1990. Recovering baseline and orientation from essential matrix. *Journal of Optical Society*, 110, 230 - 240.

- ITO, M. 1990. Robot vision modelling-camera modelling and camera calibration. *Advanced Robotics*, 5, 321-335.
- JANG, J., HWNAG, S. & PARK, K. 2013. Design of Indirect Time-of-Flight Based Lidar for Precise Three-Dimensional Measurement Under Various Reflection Conditions. *The Fourth International Conference on Sensor Device Technologies and Applications*.
- KAADAN, A. 2013. *A study of unmanned aerial systems stability for lasercom applications*. University of Oklahoma.
- KAMPHUIS, J. W. 2000. *Introduction to coastal engineering and management*, World Scientific.
- KANDADE, T. & OKUTOMI, M. 1994. A stereo matching algorithm with an adaptive window: theory and experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16, 920-932.
- KARIMANZIRA, D., JACOBI, M., PFUETZENREUTER, T., RAUSCHENBACH, T., EICHHORN, M., TAUBERT, R. & AMENT, C. 2014. First testing of an AUV mission planning and guidance system for water quality monitoring and fish behavior observation in net cage fish farming. *Information Processing in Agriculture*, 1, 131-140.
- KIM, G.-B. & CHUNG, S.-C. 2004. An accurate and robust stereo matching algorithm with variable windows for 3D measurements. *Mechatronics*, 14, 715-735.
- KIM, J.-S., HWANGBO, M. & KANADE, T. 2010. Spherical approximation for multiple cameras in motion estimation: Its applicability and advantages. *Computer Vision and Image Understanding*, 114, 1068-1083.
- KUSHLEYEV, A., MELLINGER, D., POWERS, C. & KUMAR, V. 2013. Towards a swarm of agile micro quadrotors. *Autonomous robots*, 35, 287-300.
- LAZZARO, S. 2015. Flying multiple drones from 1 remote controller. University of Wisconsin-Madison.
- LENZ, R. K. & TSAI, R. Y. 1989. Calibrating a Cartesian robot with eye-on-hand configuration independent of eye-to-hand relationship. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11, 916-928.
- LI, M., ZHENG, D., ZHANG, R., YIN, J. & TIAN, X. Overview of 3D Reconstruction Methods Based on Multi-view. Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2015 7th International Conference on, 2015. IEEE, 145-148.
- LIU, P. C., SCHWAB, D. J., WU, C. H. & MACHUTCHON, K. R. Wave heights in a 4d ocean wave field. ASME 2008 27th International Conference on Offshore Mechanics and Arctic Engineering, 2008. American Society of Mechanical Engineers, 889-893.
- LONGUET-HIGGINS, H. C. 1981. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293, 133-135.
- LOWE, D. G. Object recognition from local scale-invariant features. *Computer vision*, 1999. The proceedings of the seventh IEEE international conference on, 1999. Ieee, 1150-1157.
- LOWE, D. G. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60, 91-110.
- LUONG, Q.-T. & FAUGERAS, O. D. 1996. The fundamental matrix: Theory, algorithms, and stability analysis. *International Journal of Computer Vision*, 17, 43-75.
- LUONG, Q. T. & FAUGERAS, O. D. 1997. Self-calibration of a moving camera from point correspondences and fundamental matrices. *International journal of computer vision*, 22, 261-289.
- MAHMOUD, H. & AKKARI, N. Shortest Path Calculation: A Comparative Study for Location-Based Recommender System. World Symposium on Computer Applications and Research, 2016. 1-5.
- MAROWSKI, D. G. 1992. *Environmental Viewpoints*.
- MARR, D. & POGGIO, T. 1979. A computational theory of human stereo vision. *Proceedings of the Royal Society of London B: Biological Sciences*, 204, 301-328.

- MARTY, J. A. 2013. Vulnerability analysis of the mavlink protocol for command and control of unmanned aircraft. AIR FORCE INSTITUTE OF TECHNOLOGY WRIGHT-PATTERSON AFB OH GRADUATE SCHOOL OF ENGINEERING AND MANAGEMENT.
- MÁTHÉ, K. & BUŞONIU, L. 2015. Vision and control for UAVs: A survey of general methods and of inexpensive platforms for infrastructure inspection. *Sensors*, 15, 14887-14916.
- MEIER, L., TANSKANEN, P., HENG, L., LEE, G. H., FRAUNDORFER, F. & POLLEFEYS, M. 2012. PIXHAWK: A micro aerial vehicle design for autonomous flight using onboard computer vision. *Autonomous Robots*, 33, 21-39.
- MISTRY, D. & BANERJEE, A. Comparison of Feature Detection and Matching Approaches: SIFT and SURF.
- MONTAMBault, S., BEAUDRY, J., TOUSSAINT, K. & POULIOT, N. On the application of VTOL UAVs to the inspection of power utility assets. Applied Robotics for the Power Industry (CARPI), 2010 1st International Conference on, 2010. IEEE, 1-7.
- NAGAI, M., CHEN, T., AHMED, A. & SHIBASAKI, R. 2008. UAV Borne mapping by multi sensor integration. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, 37, 1215-1221.
- OHTA, Y. & KANADE, T. 1985. Stereo by Intra- and Inter-Scanline Search Using Dynamic Programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 139.
- OPENCV. 2017. *Camera calibration With OpenCV* [Online]. Available: http://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html [Accessed November 2017].
- ORE, J. P., ELBAUM, S., BURGIN, A. & DETWEILER, C. 2015. Autonomous aerial water sampling. *Journal of Field Robotics*, 32, 1095-1113.
- PATIL, S., NADAR, J. S., GADA, J., MOTGHARE, S. & NAIR, S. S. 2013. Comparison of various stereo vision cost aggregation methods. *International Journal of Engineering and Innovative Technology*, 2, 222-226.
- PERELMAN, O., WU, C., BOUCHERON, R. & FRÉCHOU, D. 3D wave fields measurements techniques in model basin: application on ship wave measurement. Advanced Model Measurement Technology for EU Maritime Industry, 2011.
- PIEPMEIER, J. A. & WATERS, J. Analysis of stereo vision-based measurements of laboratory water waves. Geoscience and Remote Sensing Symposium, 2004. IGARSS'04. Proceedings. 2004 IEEE International, 2004. IEEE, 3588-3591.
- PRITCHETT, P. & ZISSERMAN, A. Wide baseline stereo matching. Computer Vision, 1998. Sixth International Conference on, 1998. IEEE, 754-760.
- PSIROFONIA, P., SAMARITAKIS, V., ELIOPOULOS, P. & POTAMITIS, I. 2017. Use of unmanned aerial vehicles for agricultural applications with emphasis on crop protection: three novel case-studies. *International Journal of Agricultural Science and Technology*.
- PURI, A. 2005. A survey of unmanned aerial vehicles (UAV) for traffic surveillance. *Department of computer science and engineering, University of South Florida*, 1-29.
- QUINTANA, X. A. 2003. *Modelling Stereoscopic Vision Systems for Robotic Applications*. PhD thesis, Universitat de Girona.
- QUINTERO, GAEMUS E. COLLINS & HESPANHA, J. P. 2013. Flocking with fixed-wing UAVs for distributed sensing: A stochastic optimal control approach. IEEE.
- READ, J. C. A. 2014. Stereo vision and strabismus. *Eye*, 29, 214-224.
- RICOLFE-VIALA, C. & SANCHEZ-SALMERON, A.-J. 2011. Camera calibration under optimal conditions. *Optics express*, 19, 10769-10775.
- ROSENHOLTZ, R. & MALIK, J. 1997. Surface orientation from texture: isotropy or homogeneity (or both)? *Vision research*, 37, 2283-2293.
- ROSTEN, E., PORTER, R. & DRUMMOND, T. 2010. Faster and better: A machine learning approach to corner detection. *IEEE transactions on pattern analysis and machine intelligence*, 32, 105-119.

- RUBLEE, E., RABAUD, V., KONOLIGE, K. & BRADSKI, G. ORB: An efficient alternative to SIFT or SURF. *Computer Vision (ICCV), 2011 IEEE international conference on*, 2011. IEEE, 2564-2571.
- RUSSAKOFF, D. B., TOMASI, C., ROHLFING, T. & MAURER, C. R. Image similarity using mutual information of regions. *European Conference on Computer Vision*, 2004. Springer, 596-607.
- SAARI, H., AKUJÄRVI, A., HOLMLUND, C., OJANEN, H., KAIVOSOJA, J., NISSINEN, A. & NIEMELÄINEN, O. 2017. VISIBLE, VERY NEAR IR AND SHORT WAVE IR HYPERSPECTRAL DRONE IMAGING SYSTEM FOR AGRICULTURE AND NATURAL WATER APPLICATIONS. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 165-170.
- SABINE, H., SEVERIN, L., MAJA, V., FABIO, R., ANGELO, C., JEAN-CHRISTOPHE, Z. & DARIO, F. Reynolds flocking in reality with fixed-wing robots: Communication range vs. maximum turning rate. *International Conference on Intelligent Robots and Systems*, 2011. 5015.
- SAN JOSE ALONSO, J., MARTÍNEZ RUBIO, J., FERNÁNDEZ MARTÍN, J. & GARCÍA FERNÁNDEZ, J. 2011. Comparing time-of-flight and phase-shift. The survey of the royal pantheon in the basilica of San Isidoro (Leon). *Int. Archives of Photogrammetry and Remote Sensing and Spatial Information Sciences*, 38.
- SANTEL, F., HEIPKE, C., KONNECKE, S. & WEGMANN, H. 2002. Image sequence matching for the determination of three-dimensional wave surfaces. *International archives of photogrammetry remote sensing and spatial information sciences*, 34, 596-600.
- SCARAMUZZA, D., FRAUNDORFER, F., POLLEFEYS, M. & SIEGWART, R. Absolute scale in structure from motion from a single vehicle mounted camera by exploiting nonholonomic constraints. *Computer Vision, 2009 IEEE 12th International Conference on*, 2009. IEEE, 1413-1419.
- SCHARSTEIN, D. & SZELISKI, R. 2002. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47, 7-42.
- SCHARSTEIN, D. & SZELISKI, R. 2012. *The Middlebury Computer Vision Pages* [Online]. Middlebury College. Available: <http://vision.middlebury.edu/> [Accessed January 2018].
- SCHMID, C., MOHR, R. & BAUCKHAGE, C. 2000. Evaluation of interest point detectors. *International Journal of computer vision*, 37, 151-172.
- SHAO, Z., YANG, N., XIAO, X., ZHANG, L. & PENG, Z. 2016. A multi-view dense point cloud generation algorithm based on low-altitude remote sensing images. *Remote Sensing*, 8, 381.
- SHETTY, A. A., GEORGE, V., NAYAK, G. C. & SHETTY, R. 2016. Normalized Cross Correlation for Stereo Matching Under Varying Illumination. *International Journal of Control Theory and Applications*, 9, 39-42.
- SNAVELY, N., SEITZ, S. M. & SZELISKI, R. 2008. Modeling the world from internet photo collections. *International journal of computer vision*, 80, 189-210.
- STEVENS, K. A. 1981. The information content of texture gradients. *Biological cybernetics*, 42, 95-105.
- STREICHER, M., HOFLAND, B. & LINDENBERGH, R. 2013. Laser ranging for monitoring water waves in the new Deltires Delta Flume. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci*, 2, 271-276.
- TAN, L., WANG, Y., YU, H. & ZHU, J. 2017. Automatic camera calibration using active displays of a virtual pattern. *Sensors*, 17, 685.
- TEVERA, D. S. & MOYO, S. 2000. Environmental security in southern Africa.
- THORNTON, E. B. & GUZA, R. 1983. Transformation of wave height distribution. *Journal of Geophysical Research: Oceans*, 88, 5925-5938.
- TOMIC, T., SCHMID, K., LUTZ, P., DOMEL, A., KASSECKER, M., MAIR, E., GRIXA, I. L., RUESS, F., SUPPA, M. & BURSCHKA, D. 2012. Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue. *IEEE robotics & automation magazine*, 19, 46-56.
- TSAI, R. Y. 1987. A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses. *IEEE Journal on Robotics and Automation*, 3, 323-344.

- U-BLOX. 2017. *NEO-7 series* [Online]. Available: https://www.u-blox.com/sites/default/files/products/documents/NEO-7_ProductSummary_%28UBX-13003342%29.pdf [Accessed 12/04 2018].
- VÁSÁRHELYI, G., VIRÁGH, C., SOMORJAI, G., TARCAI, N., SZÖRÉNYI, T., NEPUSZ, T. & VICSEK, T. Outdoor flocking and formation flight with autonomous aerial robots. International Conference on Intelligent Robots and Systems, 02/14/ 2014.
- VIRIYAKIJJA, K. & CHINNARASRI, C. 2015. Wave Flume Measurement using Image Analysis. *Aquatic Procedia*, 4, 522-531.
- VLADIMIR, K., PASCAL, M. & PAULINE, T. 2014. Kolmogorov and Zabih's Graph Cuts Stereo Matching Algorithm. *Image Processing On Line*, 4, 220-251.
- WESTOBY, M. J., BRASINGTON, J., GLASSER, N. F., HAMBREY, M. J. & REYNOLDS, J. M. 2012. 'Structure-from-Motion' photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology*, 179, 300-314.
- WORLD HEALTH ORGANIZATION 2015. *Progress on sanitation and drinking water: 2015 update and MDG assessment*, World Health Organization.
- WYNN, R. B., HUVENNE, V. A. I., LE BAS, T. P., MURTON, B. J., CONNELLY, D. P., BETT, B. J., RUHL, H. A., MORRIS, K. J., PEAKALL, J., PARSONS, D. R., SUMNER, E. J., DARBY, S. E., DORRELL, R. M. & HUNT, J. E. 2014. Autonomous Underwater Vehicles (AUVs): Their past, present and future contributions to the advancement of marine geoscience. *Marine Geology*, 352, 451-468.
- XYLOMENOS, G. & POLYZOS, G. C. TCP and UDP performance over a wireless LAN. INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, 1999. IEEE, 439-446.
- ZABIH, R. & WOODFILL, J. 1994. Non-parametric local transforms for computing visual correspondence. *Proceedings of the third European conference on Computer Vision* (Stockholm, Sweden: Springer-Verlag New York, Inc.
- ZHANG, Z. 2000. A Flexible New Technique for Camera Calibration. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 22, 1330.
- ZONGJIAN, L. 2008. UAV for mapping—low altitude photogrammetric survey. *International Archives of Photogrammetry and Remote Sensing, Beijing, China*, 37, 1183-1186.