Flinders
UNIVERSITY
ADELAIDE · SOUTH AUSTRALIA

Maritime RobotX: A Ball Launcher and Target Tracking System

by

Keith Man

Thesis
Submitted to Flinders University
for the degree of

Bachelor of Engineering (Robotics),
Master of Engineering (Electronics)

College of Science and Engineering

Supervised by Professor Karl Sammut and Mr Jonathan Wheare

Submitted 15/10/2018

Declaration

I certify that this work does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person except where due reference is made in the text.

Keith Man, 15/10/2018

Signature: Keith

Abstract

TopCat is a WAM-V unmanned surface vehicle (USV) outfitted to compete in the Maritime RobotX Challenge. The Detect and Deliver task in the 2018 Maritime RobotX Challenge requires a USV to detect a symbol on a floating dock and deliver four racquetballs into the dock. Currently, vision based cameras are used to identify a set of symbols within the cameras' field of view as the USV approaches them. Symbol detection is carried out using a blob detection algorithm that reports the symbol identified. However, the current symbol detection algorithm is prone to misidentifying symbols in changeable lighting environments and does not report the position of symbols within its field of view.

This thesis addresses this issue through the development of a target tracking system that uses blob detection algorithm with automatic parameter configuration in combination with region of interest data from the lidar and simultaneous localisation and mapping systems to identify a symbol's location and pass that information to a pan-tilt servo system. The automatic configuration of internal parameters allows the algorithm to identify symbols in a wide range of lighting environments while region of interest data from lidar sensors is used to reduce the area in an image stream that needs to be searched for symbols. After identifying the position of a symbol within the cameras' field of view, the required servo positions are then sent to a pan-tilt servo system acting as the targeting system of a ball launcher.

The target tracking system developed will enable a ball launcher to be aimed at a target as part of the Detect and Deliver task of the 2018 Maritime RobotX Challenge.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank Professor Karl Sammut and Mr Jonathan Wheare for their kind guidance and significant contribution to this thesis. They have been a great source of knowledge and support.

1 – Introduction

RoboNation, formally the Association for Unmanned Vehicle Systems International (AUVSI) Foundation (RoboNation, 2018b), has run a biennial maritime competition for unmanned surface vehicles (USV) since 2014.

The Maritime RobotX Challenge was first announced in 2013 at AUVSI's Unmanned Systems Conference in 2013. Created in support of advancements in autonomous vehicle technology in land, air and water, and co-sponsored by the Office of Naval Research (ONR), AUSVI Foundation and Singapore's Ministry of Defence, the inaugural competition was held in Singapore in 2014. Three teams each from the United States, Singapore, Australia, Japan and South Korea, were invited for participation (Office of Naval Research, 2013). Flinders University, in cooperation with the Australian Maritime College in Tasmania, was one of the three Australian teams that participated in the inaugural Maritime RobotX Challenge.

To compete in the Maritime RobotX Challenge, a student team has to fit out a 16-foot Wave Adaptive Modular Vessel (WAM-V) with propulsion, power and sensor systems, enabling it to navigate maritime environments autonomously. In the competition, USVs have to complete a series of challenge tasks. These tasks include autonomous navigation, obstacle avoidance and symbol identification and are designed to emulate real world scenarios that an USV might have to deal with. Flinders University's USV, TopCat is shown in Figure 1.

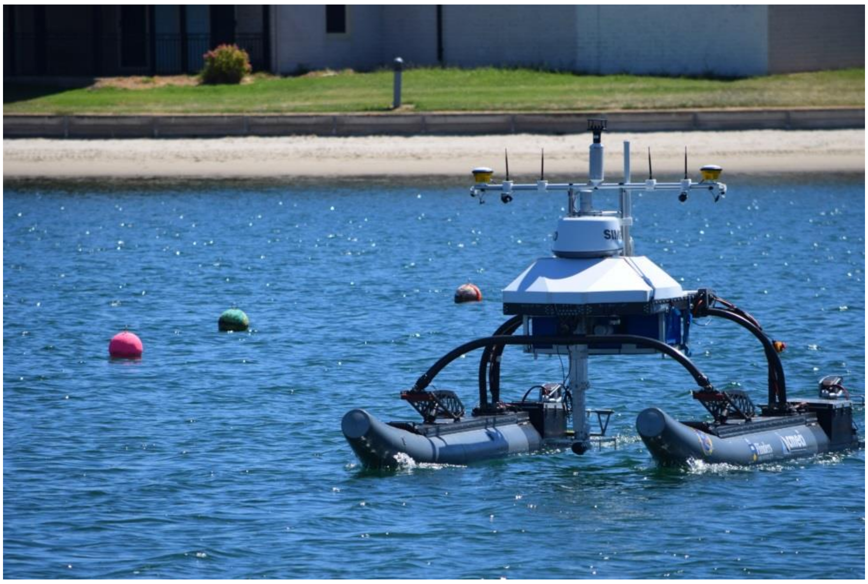


Figure 1. Flinders University's USV, TopCat (Image courtesy: Centre for Maritime Engineering Control and Imaging, Flinders University)

The TopCat is equipped with an on-board computer system that controls the movement of the USV through processing of data collected from its sensors. The on-board sensors include RADAR, lidar, GPS, vision based cameras and an ultrasonic wind sensor. While it can run autonomously, it is also capable of being remotely monitored and controlled by a human operator via a base station.

As of the 2018 Maritime RobotX Competition, there are eight different challenge tasks (RoboNation, 2018b):

- Entrance and Exit Gates
- Avoid Obstacles
- Find Totems
- Scan the Code
- Identify Symbols and Dock
- Detect and Deliver
- Underwater Ring Recovery
- Situational Awareness Reporting

Among these eight tasks, Detect and Deliver was added to the challenge task roster in the 2016 competition (RoboNation, 2018a). Detect and Deliver required a USV to approach a floating dock, identify a coloured symbol on the dock and deposit four racquetballs into a net located above the coloured symbol, Figure 2.

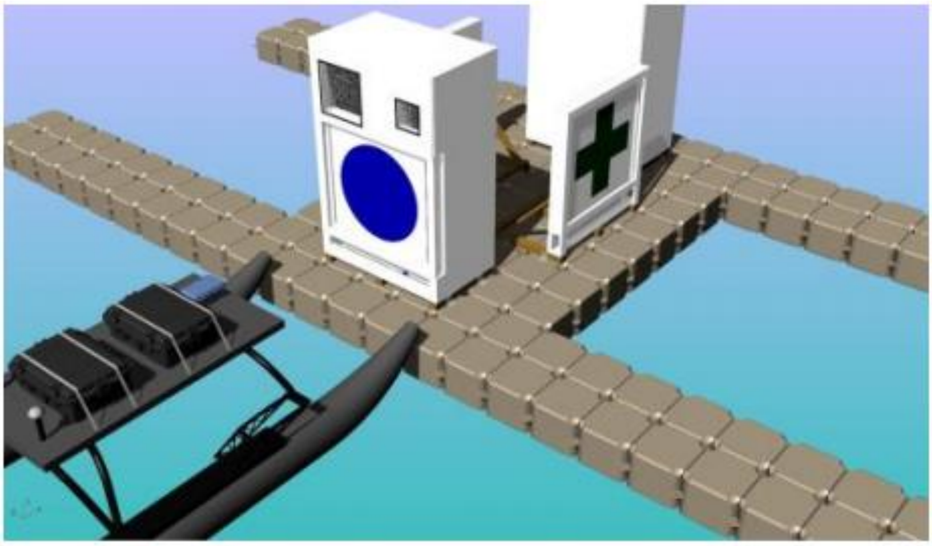


Figure 2. Visualisation of USV approaching floating dock, taken from the task summary (RoboNation, 2018c)

The target, see Appendix A, consists of a coloured target symbol with two square holes located directly above it. There is no requirement to deposit the racquetballs into a specific hole as they are only used to calculate points for the competition. However, depositing the racquetballs into the smaller hole results in more points.

While performing the task, the USV is not permitted to directly interact with either the dock or the target, whether by pushing against the dock to hold position or latching onto the target. While the task permitted the insertion of the racquet balls into the target nets via methods other than propulsion, the task was designed to emulate a fire fighting scenario where a water jet has to be directed through small openings in a vessel or building.

Except for the team from University of Hawai'i at Mānoa, who designed a telescoping mechanical arm to deposit the racquetballs into the target net, all other teams attempted to construct ball launching mechanisms to propel the racquetballs from the USV into the target. While almost all teams that competed in the 2016 challenge documented some form of ball delivery mechanism, the Seoul National University (SNU) team was the only one to demonstrate their system successfully.

As documented in their team paper, the SNU team used a lidar system to find the location of the floating target platform before switching to the camera system to identify the symbol on the platform. Shape recognition was done using a template matching technique. After determining the correct target, the USV switched back to the lidar system and used the RANSAC method to determine its desired position in front of the target. Manoeuvring the USV to its desired position, a Canny edge detector was employed to locate the square regions with the target holes filtered out using geometric constraints. After locating the target, a proportional derivative control system was used to operate the pan and tilt motors to centre the aiming based on image pixel error. The ball launcher would then fire after the target position is determined (Choi et al., 2017).

The use of robotics for hazardous environments is not a new concept. The idea of using remote or robotic systems for such purposes goes as far back as the 1970's. In 1972, Marvin Minsky, a well-known cognitive scientist in the field of artificial intelligence, submitted a proposal to the Advanced Research Projects Agency (ARPA) regarding the development of robotic systems for use in medicine and health, mining and undersea resources, nuclear facilities and space where tasks in the given environments could potentially be hazardous for humans (Minsky, 1972).

In 1979, McGhee's book 'Future Prospects for Sensor-Based Robots' (McGhee, 1979) mentioned the use of robots to replace humans for hazardous environments. It noted that applications included fire fighting, underground mining and ordnance disposal. In 1985, a student design project was carried out at the Naval Postgraduate School in Monterey, California to design a navy robotic fire fighter (Smith et al., 1985). In the paper, it was noted that the goal of developing a robotic fire fighter was not just for the purpose of replacing human fire fighters but to develop a robot that would be able to fight fires more effectively.

The next year, Breaux (Breaux, 1986) explored the possible reactions of a simple robotic system to a fire. It discussed its potential benefits compared to a potentially erratic human action when confronted with catastrophic events. In 1988, a conceptual design study was carried out for the development of a remotely operated system to fight fires (Cox et al., 1988). While it was not an autonomous system, the study demonstrated the feasibility of using robots or remotely controlled vehicles to eliminate human presence in hazardous environments.

In 1991, a report on the United Kingdom's Security and Fire Fighting Advanced Robot (SAFFAR) project investigated the development of a low cost, high performance robot for use in fighting nondomestic fires (Bradshaw, 1991). The use of neural networks to train and control a fire fighting robot was explored in a 1998 paper (Zhou et al., 1998). The paper considered the idea that a fire fighting robot did not necessarily have to be remotely controlled or operate on a fixed set of rules. Instead, it could learn to complete tasks such as identifying the source of a fire, navigating towards it and then combating the fire. The potential use of robotic systems for search and rescue tasks, including fire fighting in the aftermath of the Great Hanshin-Awaji earthquake in Japan 1995 was also investigated (Tadokoro et al., 2000).

Into the 21st century, the development of practical fire fighting robots became more prominent, with the Tokyo Fire Department developing several robots for the purpose of fighting fires remotely under various environment conditions (Miyazawa, 2002). In 2015, the US Navy unveiled a prototype shipboard fire fighting humanoid robot at the Naval Future Force Science & Technology EXPO in 2015. The Shipboard Autonomous Firefighting Robot (SAFFiR), was demonstrably able to walk across uneven floors, identify overheated equipment and deploy a hose to extinguish a small fire in a series of experiments. SAFFiR was developed by researchers at Virginia Polytechnic Institute and State University and was used by the ONR to evaluate the applications of unmanned systems in damage control and naval vessel inspection as part of the navy's focus in supporting autonomy and unmanned systems (White, 2015).

The ONR sponsored several more papers in 2016 that investigated the use of various sensors to track the path of water spray to maximise the capabilities of a robot to suppress a fire. In 'Autonomous Fire Suppression Using Feedback Control for a Firefighting Robot' (McNeil, 2016), a real-time feedback loop was used to identify the spray path of a water jet and apply the necessary adjustments to correct its trajectory. Unlike a ball launcher, water sprays are continuous. As a result, it is possible to utilise a feedback control system for trajectory correction.

From the historical development perspective of robotic fire fighting, the inclusion of a challenge task emulating robotic fire fighting is well founded.

2 – Problem Formulation

In the 2016 competition, the Flinders University team's USV was unable to identify the target holes and propel racquetballs into the net behind the target to complete Detect and Deliver. However, three key issues were identified for further development and research.

2.1 – Problem 1

The USV lacked a fully functional ball launcher. Two prototypes were developed for the 2016 competition but neither reached a satisfactory level of performance prior to the start of the competition. One prototype was too heavy to be mounted on the USV and not weather proofed for marine environments. The other was not sufficiently powerful to propel the racquetballs to the target. This problem was due to a lack of human resources to design and construct a ball launcher.

2.2 – Problem 2

The shape detection algorithm that was developed by Thomas Arbon (Arbon, 2014) for the 2014 competition was not capable of detecting symbols at further distances or the square holes which served as targets. While the blob detection algorithm it utilised was robust, it had difficulty identifying symbols at distances further than 15m and in environments with varying ambient light. A modified algorithm to enable colour detection for the 2016 competition failed to solve the problem. This problem was caused by the algorithm's inability to adjust its own parameters to suit the ambient light environment in each image frame. The lack of square shape detection was because it had not been specifically required for any task in 2014 and had not been implemented in 2016 as Detect and Deliver was not attempted.

2.3 – Problem 3

The last problem was that the shape detection algorithm did not provide any information regarding the location of detected symbols within the camera's field of view. Knowing the target location enables a ball launcher to be in position. Lacking this information made it impossible for a ball launcher system to aim at the target.

The above problems indicate clearly that the USV could not apply its current shape detection algorithm to identify reliably the symbols at distances further than 15m in varying ambient light conditions and supply target position information to a ball launcher.

This thesis mainly investigates the second two areas of research to provide the USV with a viable method to track a target symbol by increasing the video stream resolution, modifying the shape detection algorithm to allow it to change its own image processing parameters and utilising region of interest data that can be supplied from the on-board mapping algorithm.

3 – Objective

The objective of this research is to develop a target tracking system that can aim a pan-tilt servo system towards a target in real time. Region of interest information will be used to assist the location of the target within the camera's field of view, which is then transformed to the pan-tilt system's coordinate frame. The required servo angles to aim the system are then calculated from the target location in the pan-tilt system's coordinate frame.

3.1 – Scope

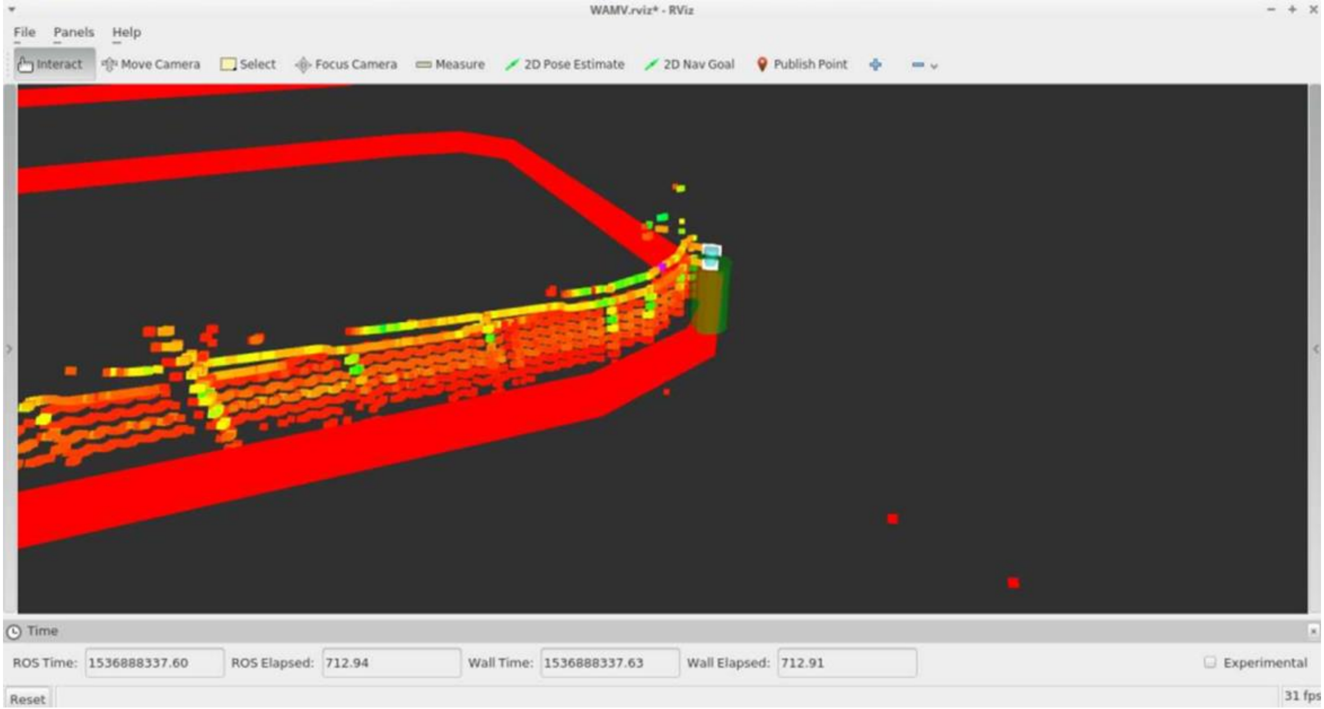
The scope of this thesis covers the following areas:

- Increasing the resolution of the video stream from the on-board cameras to investigate whether it provides significant improvements to shape detection rate at varying distances and what the associated costs may be.
- Modification of shape detection algorithm to allow it to self-configure its shape detection parameters based on parameters of each image frame in question.
- Use region of interest data, supplied by the USV's FastSLAM system using lidar sensors, to narrow the area of interest in each image frame and investigate the potential improvements to shape detection rate and image processing time.
- Supply target position data to a pair of servo motors to control a pan-tilt system that will aim a ball launcher.

4 – System Configuration

4.1 – System Overview

The overall system for a ball launcher and target tracking system is divided into three main sections. The first section uses data from the lidar sensor to map the surroundings of the USV through a simultaneous localisation and mapping (SLAM) algorithm. While the SLAM system is outside the scope of this thesis, region of interest information generated from resulting map is used to provide the shape detection algorithm with information on where objects physically exist.



ROS Time: 1536888337.60 ROS Elapsed: 712.94 Wall Time: 1536888337.63 Wall Elapsed: 712.91 □ Experimental

Figure 3. Visualisation of the USV’s localisation and mapping with region of interest highlighted in the green cylinder (Image courtesy: Dr. Andrew Lammas, Centre for Maritime Engineering, Control and Imaging, Flinders University)

The second section of the system uses region of interest data supplied by the object tracking system to reduce the area a shape detection algorithm has to process to search for symbols. Region of interest are transformed from the world frame to the camera frame, allowing the region of interest in the world frame to be resolved to pixel coordinate in the camera image. Camera images are then cropped around the region of interest to be processed by the shape detection algorithm, ideally locating a target symbol within the region of interest. Once the target symbol has been identified, the position of the target is then passed to the pan-tilt servo system through the Robotics Operating System (ROS).

The third section calculates the required signals to be sent to the pan-tilt servo system by transforming the target position information from the camera coordinate frame pan-tilt system's coordinate frame. The pan-tilt system does not have a camera that shifts with it, so visual servoing cannot be used to control it. After the target is located within the pan-tilt system's effective field of view, it is then used to drive the pan-tilt servos to the desired position.

4.2 – Coordinate Reference Frames and Transformations

4.2.1 – World Coordinate Reference Frame

In this thesis, the East North Up (ENU) coordinate system is used to map the world around the USV. In the ENU system, the world inertial frame is fixed using an origin placed at a known location. Under ENU convention, the z-axis points up from the surface of the earth, the y-axis points east and the x-axis points north.

4.2.2 – Camera Coordinate Reference Frame

The camera system also uses the ENU coordinate system, which allows points located in the world frame to be matched with points in the camera frame. However, the camera is located in the local reference frame and requires a coordinate transform to locate objects from the world frame in the camera frame. For the cameras ENU reference frame, the z-axis points up from the camera plane, the y-axis points to the right and the x-axis points forward.

The images captured by the camera use a pixel coordinate system with its origin at the top left corner of each image. The *u* axis goes across the image, increasing in value from left to right. The *v* axis goes vertically down the image, increasing in value from top to bottom.

4.2.3 – Pan-Tilt Servo Reference Frame

The pan-tilt servo system used a spherical coordinate system with the origin of the reference frame located at the meeting point of the axes of rotation of both servo motors. Angle θ is the polar angle, controlled by the tilt servo, angle φ is the azimuth angle, controlled by the pan servo, and r is the distance from the origin.

4.2.4 – Coordinate Transformations

Rotation matrices allow the mapping of coordinates from one frame to another using Euler angles. Euler angles are a method of describing object orientation using the three angles of rotation. These three Euler angles are defined as the roll angle, Ψ; the pitch angle, Θ; and the yaw angle, Φ. Using ZYX Euler angles, they rotate around the x, y and z axis respectively (Orsag et al. 2018, p. 22-28).

Provided the elementary rotation matrices:

$$\boldsymbol{R}_x(\Psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\Psi) & -\sin(\Psi) \\ 0 & \sin(\Psi) & \cos(\Psi) \end{bmatrix}$$

$$\boldsymbol{R}_y(\Theta) = \begin{bmatrix} \cos(\Theta) & 0 & \sin(\Theta) \\ 0 & 1 & 0 \\ -\sin(\Theta) & 0 & \cos(\Theta) \end{bmatrix}$$

$$\boldsymbol{R}_z(\Phi) = \begin{bmatrix} \cos(\Phi) & -\sin(\Phi) & 0 \\ \sin(\Phi) & \cos(\Phi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The transformation from one frame to another, such as from the world frame to the camera frame, is given as:

$$\boldsymbol{R}_C^W = \boldsymbol{R}_z(\Phi) \cdot \boldsymbol{R}_y(\Theta) \cdot \boldsymbol{R}_x(\Psi)$$

The frame transformation from the world frame to the camera frame is done in ROS using functions from the *tf* ROS library. To check if an object in the world frame lies within the cameras field of view, the *image geometry* ROS library is used in conjunction with the *tf* library to project a reference frame onto a live image stream.

After a shape has been detected in an image, its pixel coordinates have to be transformed into the pan-tilt servo system's spherical coordinate frame. This can be done by treating the 2D image from the camera as a photo sphere (Pla and Traver, 2002). Setting the centre pixel of the image as the origin, the relative horizontal and vertical angle of the pixels around the centre pixel are calculated. These transformations can be performed using the ROS *tf* library, similar to the world to camera frame transform.

After the pixel location of a target has been identified, the pixel’s spherical coordinates are transformed into Cartesian coordinates. Rotation matrices are then applied to transform the Cartesian coordinates from the camera frame into the pan-tilt servo frame. Once the target Cartesian coordinates have been transformed into the pan-tilt servo frame, they are transformed back into spherical coordinates. The horizontal and vertical angles to the target are then used to move the pan-tilt servos to their desired position.

The limitations of this method are that it makes an assumption about the distance from the USV to the target, using the effective range of the ball launcher as the distance. This is to avoid parallax when the camera coordinates are transformed to the pan-tilt system coordinates due to the lateral distance between the camera positions and the proposed ball launcher position.

A second method had been considered as an alternative. Identifying the pixel coordinates of the target, the camera frame is transformed back into the world frame to find the lidar sensor’s distance return at those coordinates. The distance is then passed back to the shape detection algorithm to convert the pixel coordinates to the pan-tilt system, using the same method previously stated. Instead of using the effective ball launcher range as the assumed distance to the target, the second method uses the actual distance to the target to account for parallax. Nevertheless, the second method was discarded in favour of the first as the USV would have to be positioned close enough to launch the balls at the target accurately. It was decided that the overhead processing time required to transform the target location information from the shape detection algorithm to the world frame for distance measurement before transforming the target location to the pan-tilt system frame was not worth the accuracy improvement that might be obtained from it.

4.3 – Camera System

The camera system that is used in the USV is a pair of pole mounted Microsoft LifeCam Studios, see Appendix B. LifeCam Studios are web cameras designed for video call use. The two cameras are mounted on the USV approximately 1.5m apart from each other laterally, facing out from the front of the USV, Figure 4.

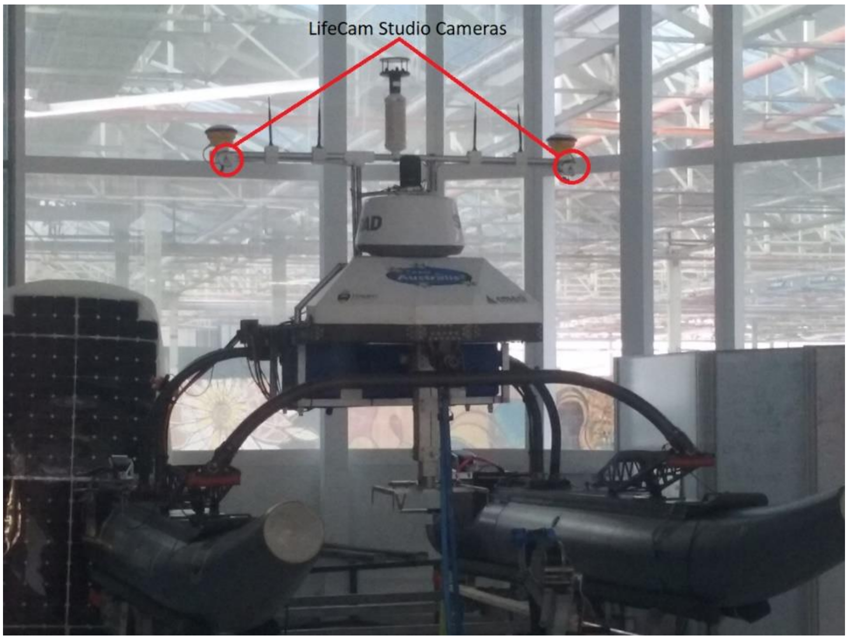


Figure 4. Location of the LifeCam Studios on the TopCat

Their positions along the pole can be adjusted to increase or reduce the distance between them. Each camera can also be adjusted to be tilted at different angles. The cameras use CMOS sensors with a resolution of 1920 x 1080 pixels and wide-angle lenses, providing them with a 75˚ field of view. Images can be streamed from the camera at up to 30 frames per second.

The two cameras are used to provide the USV with a wider field of view than could be provided by a single camera and are not used for stereo vision. The camera system is primarily used to identify any symbols that might be found on the surface of objects as well as the colour of the symbols or objects in question.

Due to bandwidth issues, the cameras had been used to stream only 640 x 480 pixel images during the 2016 Maritime RobotX Competition. However, the bandwidth issues have since been resolved, allowing images to be streamed at the full camera resolution of 1920 x 1080 pixels.

The cameras came with an autofocus function that allowed the camera to autofocus on objects at distances from 0.1m to further than 10m. However, when the autofocus function was on there were often issues when nearby objects, such as birds flying past, entered a camera's field of view. This would trigger the autofocus function, rendering nothing clearly visible for several seconds as the camera ran through autofocus. As a result, this function was turned off and the cameras configured at a constant focus to infinity.

4.4 – Lidar System

The lidar system that is used on the USV is a HDL-32E Velodyne lidar, see Appendix. The HDL-32E is a small, high definition lidar designed for real world industrial applications, including autonomous vehicle control and mobile mapping. It uses 32 lasers, aligned from +10.67˚ to -30.67˚, to scan its 360˚ horizontal field of view. It has a range of 100m and a typical accuracy of ±2cm at a scan rate of 10Hz, generating approximately 700,000 points per second.

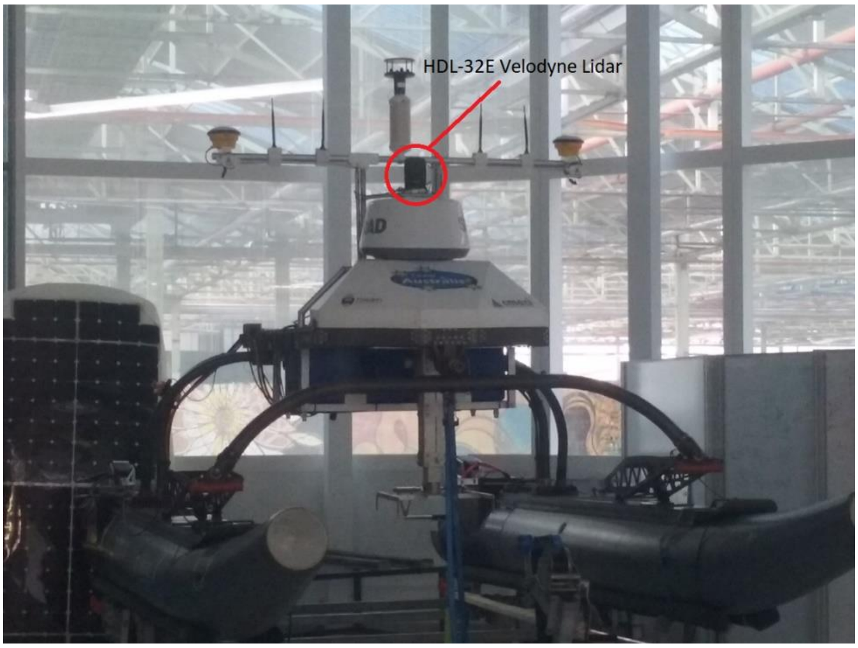


Figure 5. Location of the HDL-32E Velodyne lidar on the TopCat

The USV's FastSLAM algorithm uses the Point Cloud Library to segment the point cloud data from the lidar sensor into objects of interest. FastSLAM was chosen as an algorithm that offered a more robust solution to the data association problem compared to other available SLAM algorithms.

Point cloud data from the lidar is filtered by removing points beyond the USV's operational area as well as points returned from the water surface. The remaining points are then clustered into features of interest. Features of interest that are close to each other are then enclosed inside bounding boxes, which are then used to update the world map (Webb et al., 2014).

To generate region of interest data, whenever objects are detected and tracked by the FastSLAM system, its coordinates are transformed from the world frame to the camera frame. A check is then run to see if the object of interest lies within the cameras' field of view. If the object lies outside the cameras field of view, it is ignored. If the object lies within the cameras' field of view, the camera frame coordinates of the object in interest are then used to identify the corresponding pixel coordinate region of interest in the camera image. This information can then be passed back to the image processing algorithms to reduce the area of the image that needs to be processed.

4.5 – Ball Launcher Pan-Tilt Servo System

Two HobbyKing high torque waterproof servo motors, see Appendix D, are used in a pan-tilt servo system to control the aim of a ball launcher. Waterproof servos were chosen to minimise the effect of wet and damp conditions on the operation of the servo motors, which are expected to be used in marine environments.

Figure 6. HobbyKing HK15328D Servo Motors (HobbyKing, 2018)

The servos have a torque of 12.8kg.cm at 6V and can rotate 180˚. They are mounted perpendicularly to each other on an aluminium frame with one servo motor controlling the pan angle and another controlling the tilt angle, Figure 7. Target positioning information is supplied by the shape detection algorithm and is transformed into the pan-tilt system coordinate frame to drive the servo motors to aim the ball launcher.

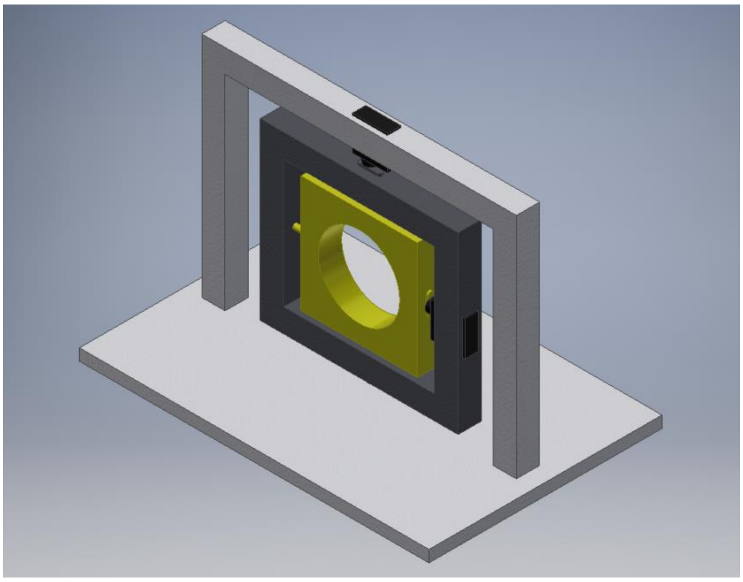


Figure 7. Pan-Tilt system main frame (light gray), pan bracket (dark gray), tilt bracket (yellow) and servo motors (black)

A pneumatic ball launcher that would be mounted to the pan-tilt servo system has been designed and a firearm exemption granted by the South Australian Police as part of South Australian firearm regulations (Firearms Act 2015). However, the complete ball launcher system has yet to be fully constructed. Nevertheless, a pan-tilt servo system is in construction for demonstration in the 2018 Flinders University Thesis Expo in late October 2018.

5– Experimental Data Collection

5.1 – Overview

The main focus of this thesis was to implement a ball launcher target tracking system onto a WAM-V USV. While a WAM-V was accessible for the project, deployment of the USV for live testing and experimentation of the shape detection algorithm was impractical due to the need of a support team to deploy the USV, time required to set up and dismantle the base station for remote monitoring and wash down of the USV after bringing it back up from the water. As a result, two boards with coloured symbols were brought to site on one of the USVs operational field tests and set up so that the symbols could be clearly identified on the USV's camera stream by a human.

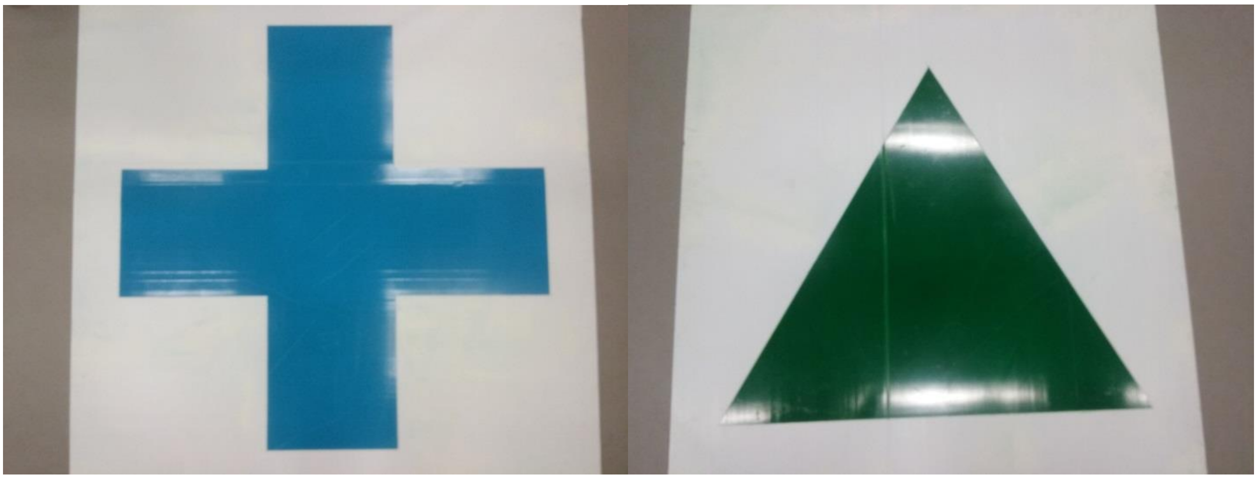


Figure 8. Blue cross and green triangle symbols used to collect data during the operational field test. Symbol sizes and colours are in line with the Maritime RobotX Challenge's task specifications.

Camera and lidar data was then recorded and stored into rosbags, a bag storage file format used by ROS, during the USV's operational test. After the completion of the operational test, camera and lidar were offloaded from the USV and post processed separately. The camera's rosbag data was extracted to a series of JPEG files for each recorded frame while region of interest data was extracted from the point maps created by the lidar sensor.

5.2 – Experimental Setup

Prior to the operational field test, both port and starboard cameras on the USV were configured to stream and record at a resolution of 1920 x 1080 pixels. While the USV is meant to operate autonomously, it was manually controlled for the duration of the operational test for the purposes of collecting specific test data. The weather conditions during the operational field test were partly cloudy with light wind, resulting in relatively consistent ambient light over the course of the test with minimal video distortion from sudden movements of the boat due to wind or waves.

During the operational test, a board with a blue cross was taped to a wooden pole and hung off the corner of a pier within view of the USV and USV operator. The USV was then operated to approach the board at varying angles from distances of 30m to 5m.

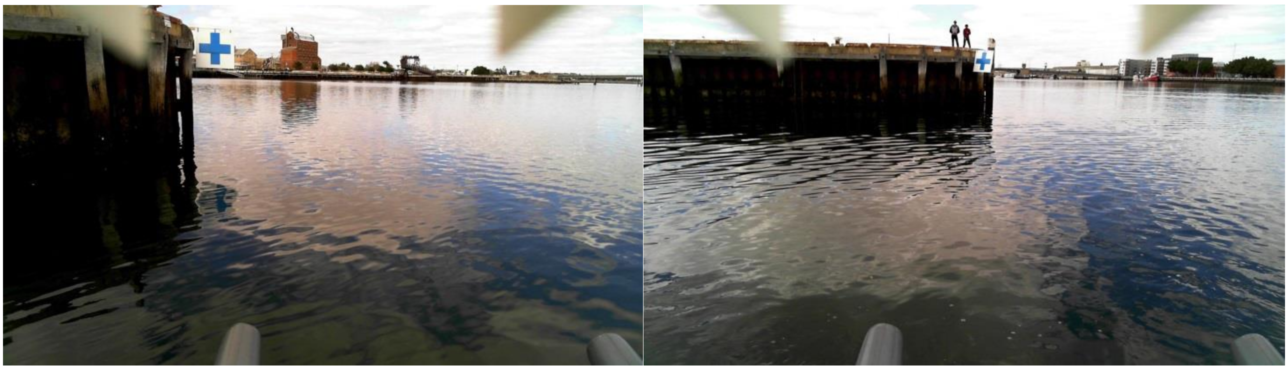


Figure 9. USV vision of the blue cross from different approach angles

After sufficient data was collected, the board with the blue cross was then replaced with a second board displaying a green equilateral triangle pointing vertically. As with the blue cross, the USV was operated to approach the board at varying angles from distances of 30m to 5m.

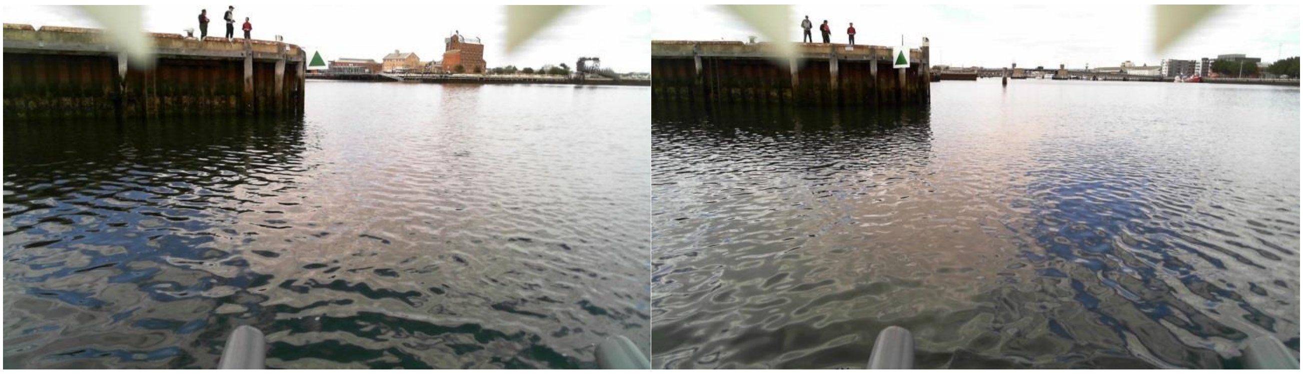


Figure 10. USV vision of the upright green triangle from different approach angles

Lastly, the board with the green equilateral triangle was rotated 90 degrees so that the triangle pointed out horizontally. The USV was again operated to approach the board at varying angles from distances of 30m to 5m.

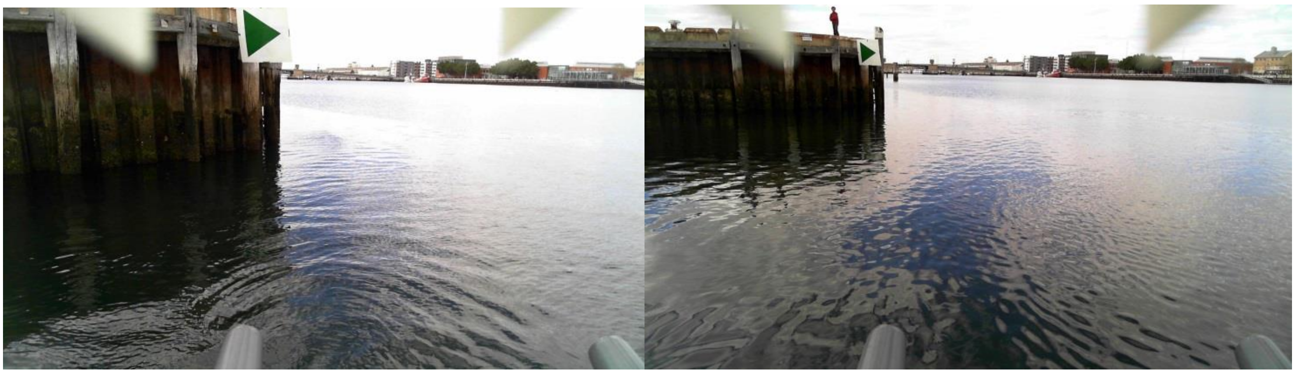


Figure 11. USV vision of the rotated green triangle from different approach angles

5.3 – Data Collection

Six sets of data were collected during the operational field tests of the USV using the two symbols. Two sets of data were collected from the USV directly approaching the blue cross at direct and oblique angles. Four more sets of data were collected from the green triangle by approaching it from direct and oblique angles in both orientations.

The six data sets were then down sampled to 843 x 480 pixels to create another six data sets to compare the effect of increasing video resolution on shape detection. The previous resolution of the video stream had been configured 640 x 480 pixels using a 4:3 aspect ratio. However, since the cameras had been reconfigured to stream at 1920 x 1080 pixels, an aspect ratio of 16:9, the images were down sampled to 843 x 480 pixels instead of 640 x 480 pixels to maintain their aspect ratio.

Regions of interest generated from the FastSLAM algorithm were directly passed to the shape detection algorithm and were not used to generate extra sets of data.

6 – Algorithm Simulation and Results

6.1 – Algorithm Simulation Setup

6.1.1 – Arbon's Shape Detection Algorithm

The original shape detection algorithm used in the USV was developed by Thomas Arbon (Arbon, 2014) for the 2014 Maritime RobotX Challenge. The Detect and Deliver task did not exist in the 2014 competition, so the shape detection algorithm was primarily used to detect plain black symbols on a white background as part of the Identify Symbol and Dock task.

The shape detection algorithm used a blob detection method and was written in C++, utilising functions from the OpenCV open source computer vision library.

The shape detection process developed by Arbon had three main stages. In the first stage, the image frame was converted to grayscale and contrast stretched using OpenCV’s *cvtColor* and *normalise* functions. The conversion to grayscale was required in order to contrast stretch the image. Contrast stretching increased the initial contrast of the image, providing a clearer distinction between the darker coloured shapes and their white backgrounds.

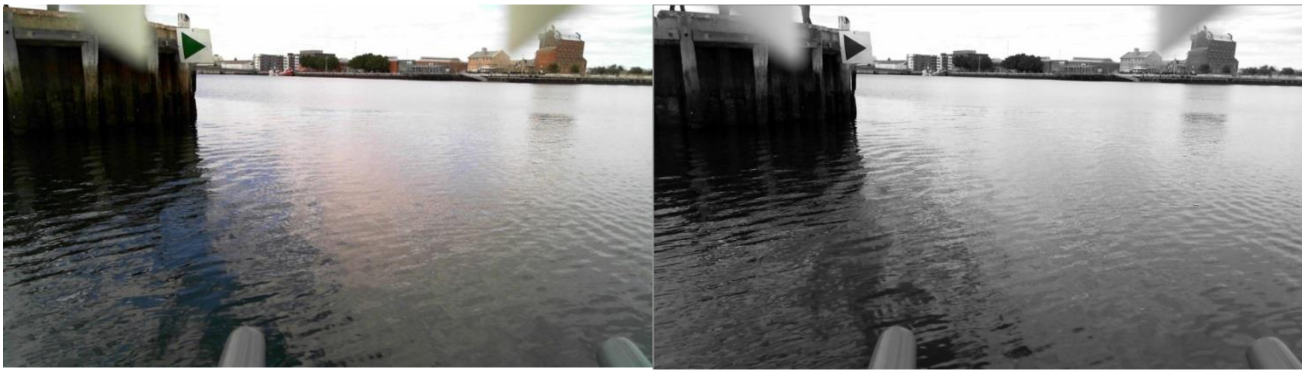


Figure 12. Original source image and image after conversion to grayscale and normalisation

In the second stage, the contrast stretched image is passed through a binary threshold filter, converting the grayscale image to a black and white image using the *inrange* function. After that, the image is morphologically opened and closed to remove unwanted noise from the image. This method isolates shapes using threshold hole finding. At the same time, the image is also run through a Canny edge detection filter to binarise and find closed edges to locate shapes. The Canny edge detector was chosen because the OpenCV implementation performed well at detecting shape edges at varying light conditions and distances. Both methods were found to reliably detect shapes. However, both methods would also often produce false positives. Because of that, the binary AND of the output from the binary threshold and Canny edge detection was used to produce the final image for contour analysis, after a second round of morphological opening and closing.

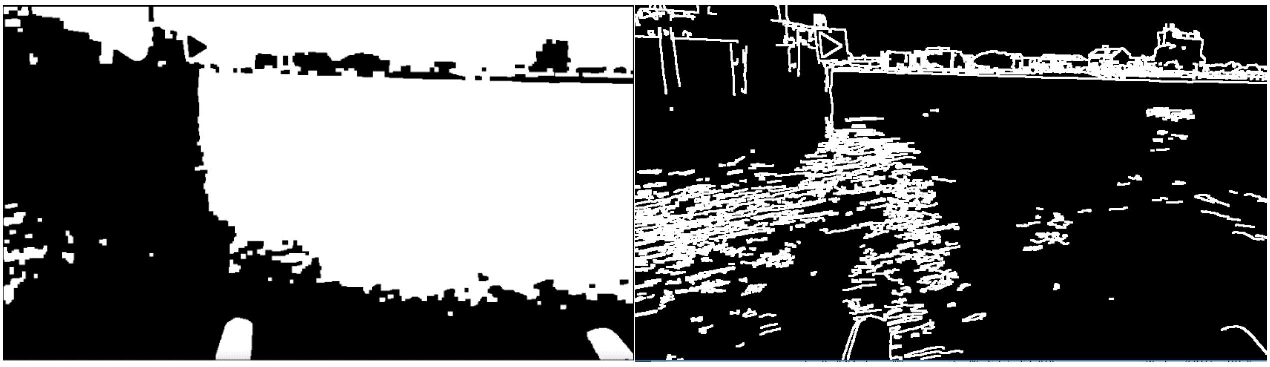


Figure 13. Binary threshold of normalised image (left) and dilated Canny filtered image (right)

In the third stage, the contours of the remaining blobs are analysed to see which of them meet the requirements for being a valid shape. Shape types are identified by dividing the squared perimeter of each blob by its area. This provides a ratio that should be unique within the shape types that can appear in the competition. Valid shapes are filled in with colour to visualise where the algorithm expects a shape to exist.

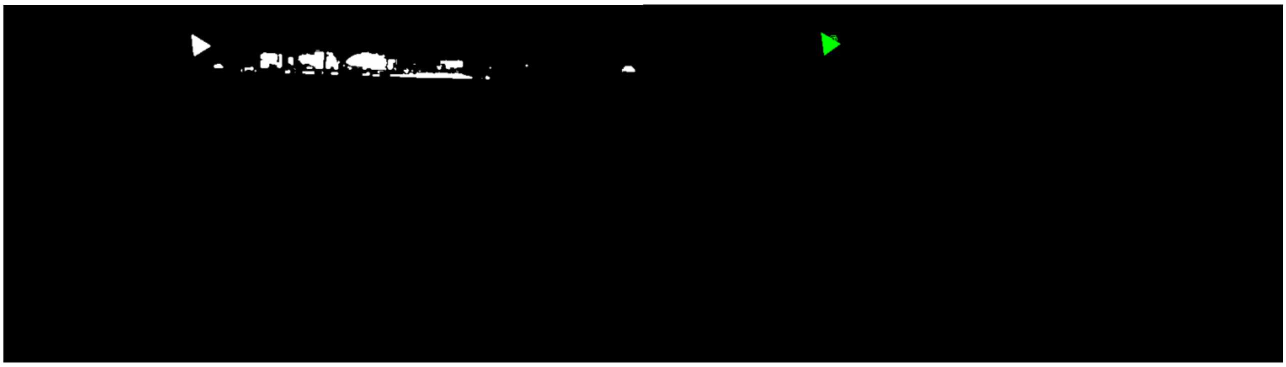


Figure 14. Bitwise AND of the binary threshold and dilated Canny images (left) and detected shape after blob contour analysis (right)

The algorithm was shown to be both size invariant and rotation invariant while also capable of detecting warped shapes to some extent, based on user defined parameters. Its size and rotation invariance stems from its use of contour analysis to identify shapes.

The contour analysis used looks at two main features of each blob, the perimeter of a closed contour and the area enclosed within the contour. As it only uses the perimeter and area within a contour and the orientation of a contour has no effect on the length of its perimeter and enclosed area, this makes contour analysis rotation invariant. Shapes were identified using the ratio of the perimeter of each blob squared and divided by the area enclosed in each blob. The squared perimeter of a blob scales linearly with the area enclosed by a blob allowing shapes of varying sizes to be identified as the same type of shape. The contour analysis is size invariant since it is capable of detecting shapes of varying sizes. The contour analysis used is also capable of identifying warped shapes due to the way it compares ratio of squared perimeter to area of each blob. By allowing a margin around the exact ratio expected for a given shape, warped shapes can still be classified correctly to a certain extent. A larger margin increases the warping allowed at the risk of incorrectly identifying shapes while a smaller margin decreases the amount of warping allowed at the risk of not identifying shapes with slight warping.

6.1.2 – Increasing Video Stream Resolution

One notable change between the USV system configurations when Arbon's algorithm was developed in 2014 and the system configurations prior to the 2018 Maritime RobotX Challenge is that the camera stream resolution was increased from 640 x 480 pixels to 1920 x 1080 pixels.

While Arbon's algorithm was a robust shape detector, it was believed that its effectiveness was limited by the resolution of the camera image stream. At a resolution of 640 x 480 pixels, the pixel difference between symbols and neighbouring objects appeared to be too small differentiate features at wide angles and further distances. Symbols would either be segmented with the region surrounding it as a single blob or merged with other blobs during morphological opening and closing of blobs.

The aim of increasing the camera stream resolution was to increase the pixel difference between colour segments in each image frame to allow for clearer differentiation between objects and their surroundings. For shape detection, the increased resolution provides larger pixel differences between shapes and their surroundings at angles and distances that would previously cause blob merging.

Attempts to increase the camera resolution for the 2016 Maritime RobotX Challenge had previously been hampered by bandwidth issues. However, the bandwidth issues have since been resolved, allowing the LifeCam Studios to output data at their full resolution of 1920 x 1080 pixels.

6.1.3 – Shape Detection Algorithm using Adaptive Parameters

A major issue with Arbon's algorithm is that the parameters used in the algorithm, such as binary threshold value for binarisation, have to be manually configured. This manual configuration could be done while the USV was operational using ROS's dynamic reconfigure but could not be done while the USV was attempting a task as it would no longer be considered completely autonomous.

This thesis explores the use of resolution limited morphological operations, scaling shape size thresholds, contrast limited adaptive histogram equalisation and adaptive binary thresholds to provide better frame to frame shape detection.

With the increase in image resolution provided by the cameras, there were several parameters whose default values were no longer appropriate for higher resolutions. The affected parameters were morphological opening and closing iteration values and shape size cut off.

The morphological opening and closing parameters controlled the number of iterations that the opening and closing operations should be done. This was mainly used to remove 'salt and pepper' noise that would occur in an image due to binary thresholding and edge detection. The main issue with the morphological operation iteration parameters was that as the image resolution increase, the pixel dimensions of the 'salt and pepper' noise also increased. However, as the number of opening and closing iterations remains constant, the noise can no longer be fully removed by the same number of iterations.

To solve this issue, the iteration parameters were changed from constants to variables that scaled with input image resolution. Higher resolution input images would cause the iterations to scale up linearly from the default values, which were designed for 640 x 480 pixel images. This not only allows morphological operation iterations to automatically scale with the increased image resolution but also allows image streams with images of varying dimensions, which would likely occur from region of interest data, to be filtered from image to image.

The second parameter affected by the increase in resolution was the shape size cut off threshold. This parameter filtered the contours of blobs during contour analysis, preventing blobs below the size cut off from being treated as shapes. As with the morphological operation iteration parameters, this parameter was originally configured for 640 x 480 pixel images. As this value did not scale with the image resolution, the increase in input resolution caused the relative minimum shape size to be decreased substantially, increasing the number of false positives picked up by the shape detection algorithm.

To solve this, the shape size cut off was changed from a constant value to a variable that linearly increased with the resolution of the image. Doing this allowed the shape detection algorithm to filter out smaller blobs that may not have been removed by the morphological opening and closing that is used to remove noisy blobs. However, this also caused legitimate symbols in images to be filtered out. This is an unavoidable result when a size cut off filter is implemented. The trade-off that comes with size filtering is that while the false positive rate is decreased, the false negative rate is increased. However, the size filter overall improves the accuracy rate of the shape detection algorithm as it removes most of the remaining false positives that slip through the closing and opening operations regardless of the distance to the shape or if the shape is in the field of view at all. On the other hand, legitimate shapes will only be filtered out if they are too far away from the USV. At closer distances, the size cut off has no effect on the detection of shapes that do exist.

Another issue with Arbon’s algorithm, prior to the increase in image stream resolution, was that objects or features that were very close to each other would often be grouped together during image segmentation. This prevented some individual features from being identified after segmentation. While this problem was alleviated by the increase in image stream resolution, increasing pixel differences between nearby features, the problem that some features could still be in close enough proximity to not be clearly differentiated remained. A combination of three methods was proposed to improve feature differentiation for shape detection, image sharpening filters, contrast limited adaptive histogram equalisation (CLAHE) and adaptive binary thresholding.

Image sharpening is the use of image filters to highlight edges and small details in an image. It effectively runs a high pass convolution kernel across the image. However, directly running a high pass filter over an image also amplifies any noise that might be present in the image, which is not ideal. To sharpen the image while minimising the amplification of any noise present, a technique known as unsharp mask was used. An unsharp mask creates a Gaussian blurred copy and a high contrast copy of the original image. The blurred image is then subtracted from the original image, producing the unsharp mask. Based on the brightness of pixels in the unsharp mask, pixels in the original image are then mixed or replaced with pixels from the high contrast image. As only bright pixels indicated in the unsharp mask are replaced in the original image, the amplification of noise is usually suppressed.

The use of an unsharp mask on an image prior to conversion to grayscale allows nearby features of differing colours to have distinct edges. After image normalisation in grayscale, a second unsharp mask is used prior to binary thresholding. While the first unsharp mask was used to assist with colour segmentation, the second unsharp mask is used to assist in segmenting features in grayscale prior to image binarisation. This attempts to minimise the merging of blobs due to morphological operations after binarisation by clearly defining the edges between features.

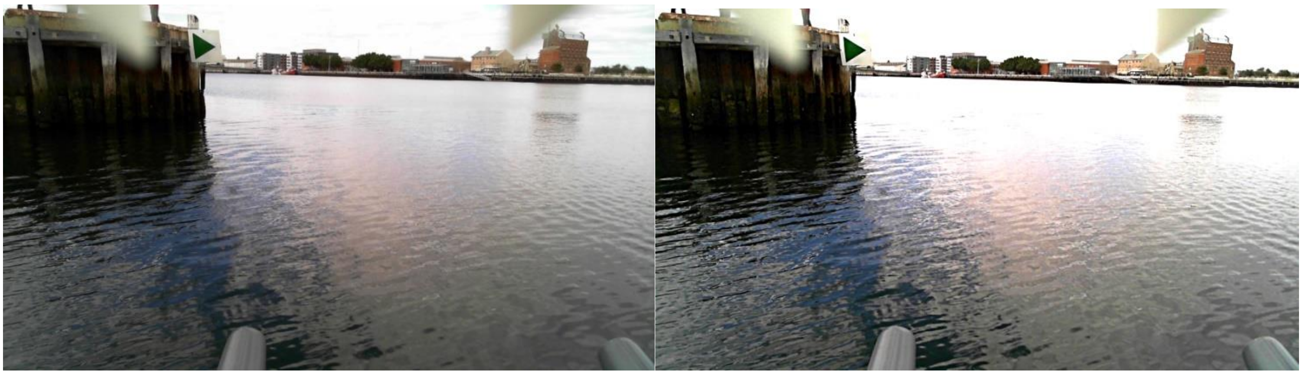


Figure 15. Original source image (left) and unsharp masked image (right)

The second method of improving feature detection was to use a CLAHE filter after grayscale conversion. The use of histogram equalisation had been explored during the development of the original shape detection algorithm to improve image contrast. However, histogram equalisation had been found to produce worse results for shape detection and was discarded in favour of contrast stretching through normalisation.

The reason why histogram equalisation worsened the image contrast for shape detection was due to the way that it works. Histogram equalisation looks at the current brightness of all pixels in the image and stretches them out so that there is a more even spread of pixel brightness from darkest to brightest. However, for images where there is an existing high contrast between dark and bright locations in the image, histogram equalisation tends to brighten up the darker areas while over-brightening the brighter areas.

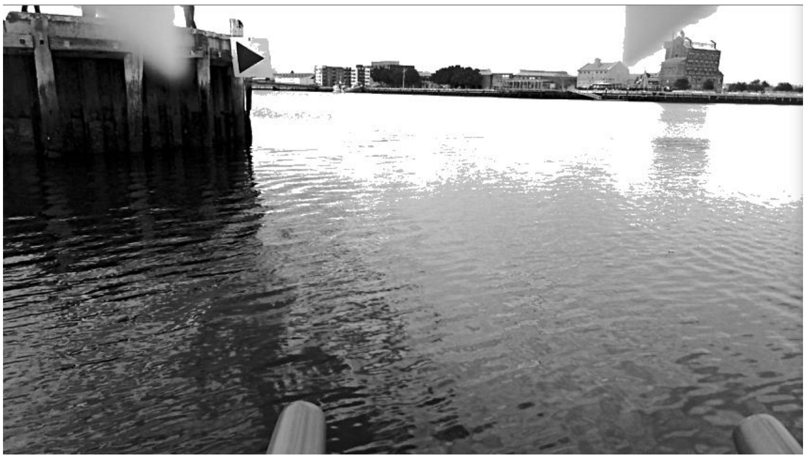


Figure 16. Global histogram equalised image

However, instead of using histogram equalisation, the use of CLAHE was explored. Unlike histogram equalisation, CLAHE divides the image into small blocks of pixels and performs a histogram equalisation on each block. To prevent noise from being amplified, contrast limiting is used to clip any pixels that go above a specified contrast limit. The use of CLAHE allows images to be histogram equalised while preventing over-brightness from occurring by setting a contrast limit.

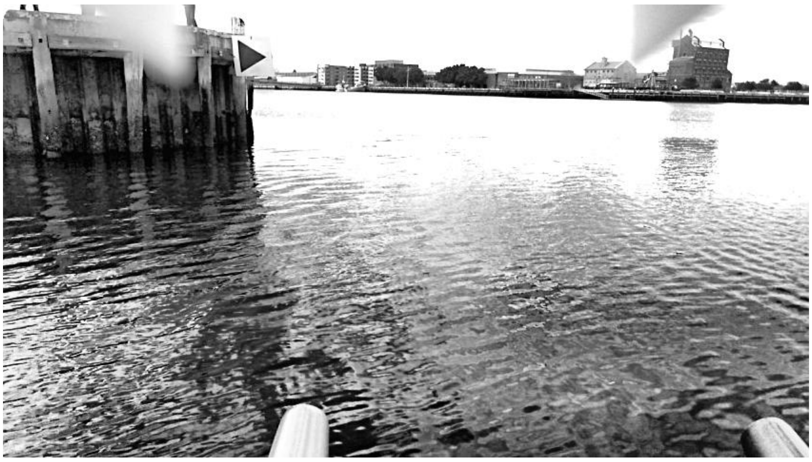


Figure 17. Contrast limited adaptive histogram equalised image

The final method used to assist in the improvement of shape detection was to switch the original fixed binary threshold to an adaptive binary threshold. Arbon's algorithm attempted to use normalisation to contrast stretch the input images to a level where shapes could be reliably identified after applying a fixed binary threshold. However, the problem with using a fixed binary threshold value is that it optimises binary thresholding for a specific range of ambient lights that may be present in an image. While contrast stretching does improve the performance of a fixed binary threshold by attempting to shift the image contrast closer to the one the binary threshold is optimised for, it is not possible to do that for all images. When input images are significantly brighter or dimmer than what the fixed threshold value is optimised for, contrast stretching alone is not enough to produce a good threshold image where the shapes are clearly visible.

The OpenCV libraries have inbuilt functions that can do a variety of adaptive binary thresholds, including adaptive mean thresholding, adaptive Gaussian thresholding and Ostu's binarisation. However, the results from testing adaptive mean and Gaussian thresholds tended to result in highly detailed contours around every small feature, which is unwanted as the algorithm has to analyse each contour to evaluate if it is a symbol.

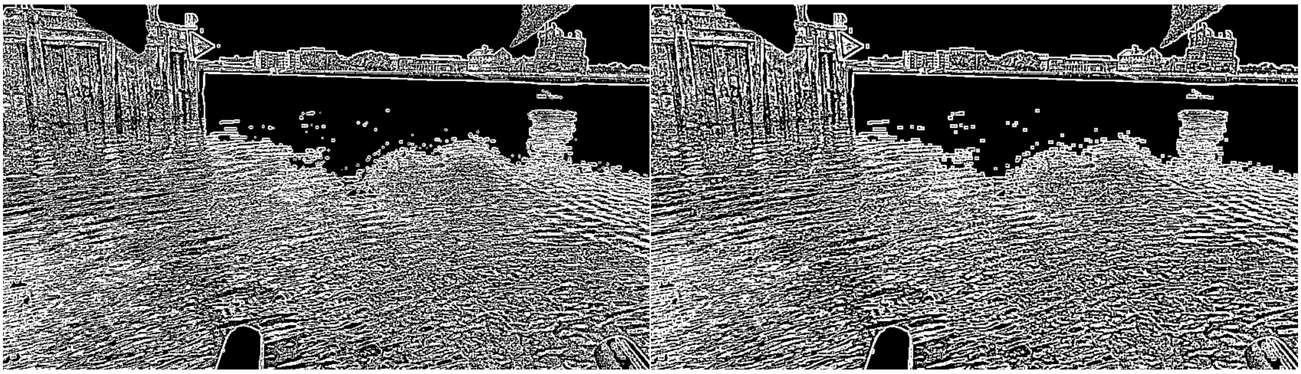


Figure 18. Adaptive Gaussian thresholding (left) and adaptive mean thresholding (right)

Otsu's binarisation produced better results for some images and worse results for others. This was noted to be because Otsu's binarisation assumes a bimodal image and performs significantly worse on non-bimodal images.

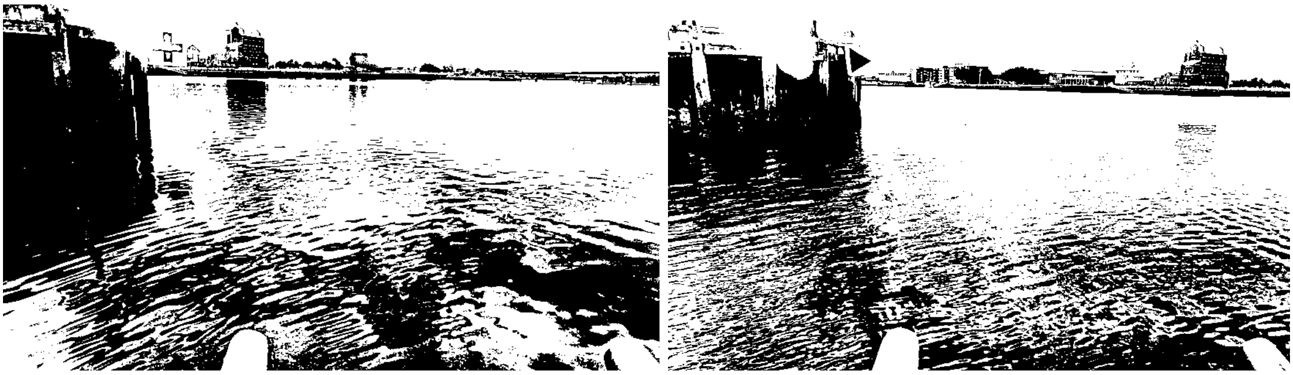


Figure 19. Ostu's binarisation on a blue cross (left) and on a green triangle (right)

An alternative to these methods that succeeded in providing a better image binarisation was to use an offset mean as a binary threshold value. This method took the mean brightness value of the image after normalisation and CLAHE and added an offset to it. The resulting value was then used as the binary threshold value. After running the offset mean binary threshold through several image datasets, it was found that the binary threshold performed best when the threshold value was significantly higher than the mean brightness value. While an offset mean threshold may not necessarily be the best adaptive thresholding method, it was shown that it provided better results from images with larger ranges of brightness than with a fixed binary threshold.

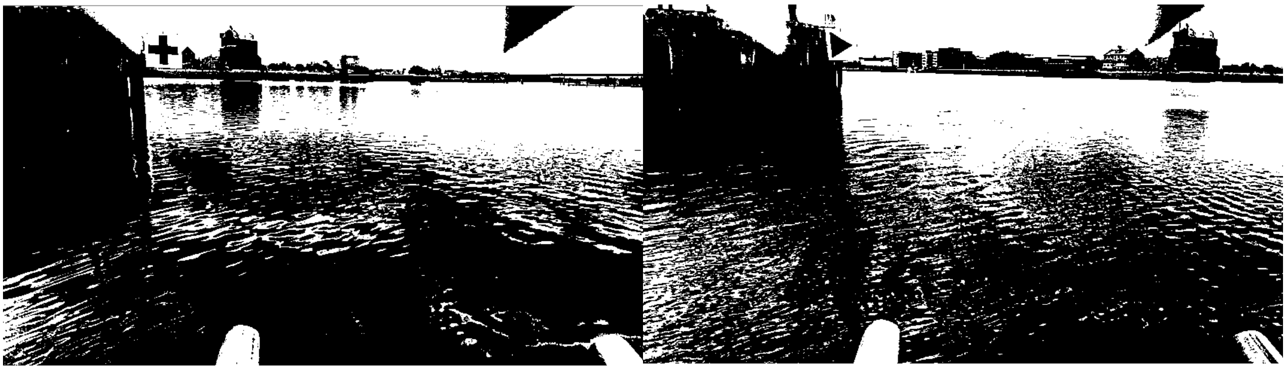


Figure 20. Offset mean thresholding on a blue cross (left) and on a green triangle (right)

6.1.4 – Shape Detection Algorithm using Regions of Interest

The goal of a shape detection algorithm is to maximise the true positive detections, detection of shapes that exist, while minimising the false positive detections, detection of shapes that do not exist. Provided the available sensors on the USV, it was proposed that region of interest data from the lidar and mapping system could be used to limit the shape detection search area in an image and improve the true positive rate while decreasing the false positive rate.

The lidar system is part of a FastSLAM system that maps the surroundings of the USV. Point clouds from the lidar are clustered based on proximity and then grouped into bounded boxes that represent objects. These objects are then tracked as the USV moves around. Objects of interest are defined by their location within the world frame relative to the USV. As the cameras on the USV are forward facing, locations of objects behind it are not reported to the shape detection algorithm. For any objects that are located in front of the camera, their location is transformed from the world frame to the camera reference frame to see if object lies inside or outside the cameras field of view. If an object lies outside the cameras field of view, its location is not sent to the shape detection algorithm. However, if the object lies within the cameras field of view, its location is converted from the camera reference frame to the camera image’s pixel coordinate frame and a region of interest marked out in the image.

The goal of using region of interest data is to limit shape detection to areas in the image where objects physically exist nearby. Since the shape detection algorithm no longer searches areas that contain no objects for shapes, it is less likely to detect false positives from those areas of the image. Vice versa, as the shape detection algorithm is only filtering areas of the image that are likely to contain shapes, removing areas such as the sky and water surface that do not contain objects, the adaptive contrast and thresholding filters will produce a segmented output based on the localised image area rather than the full image.

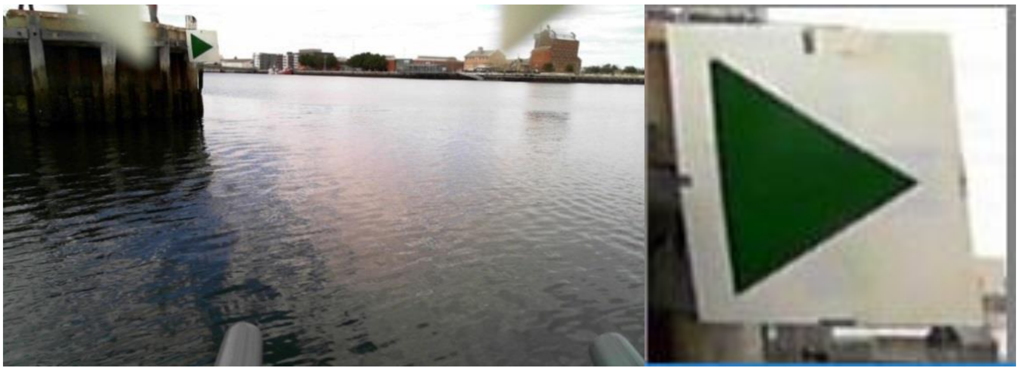


Figure 21. Original source image (left) and region of interest subimage (right)

The use of region of interest data should also assist in offsetting the increase in processing time of the adaptive shape detection algorithm. Reducing the area that needs to be analysed also reduces the time required to apply certain filters such as CLAHE and provides the same benefits of using the adaptive shape detection algorithm while lowering the time cost to implement it.

6.1.5 – Algorithm Simulation Environment

While Arbon’s algorithm was originally written as a ROS C++ node to run on the USV’s Linux based environment, the C++ code was extracted to Visual Studio 2017 and built as a Win32 Console Application to run as an executable in a Windows environment for simulation purposes.

All simulations were run in the same environment, using a Lenovo ideaPad 500S with an Intel Core i7-6500U running at base frequency of 2.50GHz over 4 cores, boosting up to 3.10GHz under higher load, and 8GB of DDR3 1600MHz RAM. The Win32 Console Applications were single threaded and did not utilise more than a single core while running. The laptop had a Nvidia GeForce 920m 2GB video card which was capable of testing CUDA compatible image processing algorithms. However, no GPU acceleration was tested on any of the algorithms.

6.2 – Increased Resolution Simulation Results

6.2.1 Arbon’s Algorithm at 480p

Arbon’s algorithm was tested on six different sets of 843 x 480 pixel images. The six images sets consisted of down sampled versions of direct and oblique approaches to the blue cross, direct and oblique approaches to an upright green triangle and direct and oblique approaches to a rotated green triangle. Each image set consisted of 100 images for a total dataset size of 600 images. The symbol was clearly visible in all images and was not obscured in any way.

The algorithm was configured to its default parameters and was not tuned or optimised in any way to improve shape detection and images were read in at 10 images per second to mimic camera data rate. True positive and false positive detection rates were calculated for each data set with a true positive meaning that a shape was correctly detected and a false positive meaning that at least one incorrect shape was detected in an image. Overall minimum, maximum and average image processing times were also calculated for the total data set.

Table 1. True and false symbol detection rate in 843 x 480 pixel images using Arbon’s algorithm

Image Data Set	True Positive Rate	False Positive Rate
Direct Approach Blue Cross	0%	6%
Oblique Approach Blue Cross	0%	1%
Direct Approach Upright Green Triangle	28%	1%
Oblique Approach Upright Green Triangle	37%	3%
Direct Approach Rotated Green Triangle	37%	15%
Oblique Approach Rotated Green Triangle	34%	3%

Table 2. Image processing time of 843 x 480 pixel images using Arbon’s algorithm

Minimum Image Processing Time (s)	16.7
Maximum Image Processing Time (s)	149
Average Image Processing Time (s)	37.4

4.2.2 Arbon’s Algorithm at 1080p

Arbon’s algorithm was tested on six different sets of 1920 x 1080 pixel images. The six images sets consisted of the full resolution images of direct and oblique approaches to the blue cross, direct and oblique approaches to an upright green triangle and direct and oblique approaches to a rotated green triangle. Each image set consisted of 100 images for a total dataset size of 600 images. The symbol was clearly visible in all images and was not obscured in any way.

The algorithm was kept to its default parameters, as with the 843 x 480 pixel image sets, and was not tuned or optimised in any way to improve shape detection and images were read in at 10 images per second to mimic camera data rate. True positive and false positive detection rates were calculated for each data set. Overall minimum, maximum and average image processing times were recalculated calculated for the total 1920 x 1080 pixel data set.

Table 3. True and false symbol detection rate in 1920 x 1080 pixel images using Arbon's algorithm

Image Data Set	True Positive Rate	False Positive Rate
Direct Approach Blue Cross	0%	34%
Oblique Approach Blue Cross	0%	12%
Direct Approach Upright Green Triangle	32%	21%
Oblique Approach Upright Green Triangle	86%	12%
Direct Approach Rotated Green Triangle	14%	6%
Oblique Approach Rotated Green Triangle	86%	7%

Table 4. Image processing time of 1920 x 1080 pixel images using Arbon's algorithm

Minimum Image Processing Time (ms)	74.5
Maximum Image Processing Time (ms)	581
Average Image Processing Time (ms)	159

6.3 – Automatic Parameter Configuration Simulation Results

6.3.1 Automatic Parameter Configuration Algorithm at 480p

The automatic parameter configuration algorithm tested on the same six sets of down sampled 843 x 480 pixel images that Arbon's algorithm was tested on. Shared parameters with Arbon's algorithm that stayed constant regardless of image resolution were configured to the same values and were not tuned or optimised in any way to improve shape detection. Images were read in at 10 images per second to mimic camera data rate. True positive and false positive detection rates were calculated for each data set. Overall minimum, maximum and average image processing times were also calculated for the total data set.

Table 5. True and false symbol detection rate in 843 x 480 pixel images using the APC algorithm

Image Data Set	True Positive Rate	False Positive Rate
Direct Approach Blue Cross	100%	1%*
Oblique Approach Blue Cross	25%	19%*
Direct Approach Upright Green Triangle	89%	12%*
Oblique Approach Upright Green Triangle	39%	17%
Direct Approach Rotated Green Triangle	17%	16%*
Oblique Approach Rotated Green Triangle	92%	22%

*The left float of the USV, which appears within the camera's field of view, was detected as a shape in most frames of this data set. This is the percentage of false positives that are not due to the left float.

Table 6. Image processing time of 843 x 480 pixel images using the APC algorithm

Minimum Image Processing Time (ms)	22.3
Maximum Image Processing Time (ms)	61.2
Average Image Processing Time (ms)	27.3

6.3.2 Automatic Parameter Configuration Algorithm at 1080p

The automatic parameter configuration algorithm was tested on the same six sets full resolution 1920 x 1080 pixel images that Arbon's algorithm was also tested on. Shared parameters with Arbon's algorithm that stayed constant regardless of image resolution were configured to the same values and were not tuned or optimised in any way to improve shape detection. Images were read in at 10 images per second to mimic camera data rate. True positive and false positive detection rates were calculated for each data set. Overall minimum, maximum and average image processing times were also calculated for the total data set.

Table 7. True and false symbol detection rate in 1920 x 1080 pixel images using the APC algorithm

Image Data Set	True Positive Rate	False Positive Rate
Direct Approach Blue Cross	100%	0%
Oblique Approach Blue Cross	40%	7%*
Direct Approach Upright Green Triangle	91%	15%*
Oblique Approach Upright Green Triangle	55%	3%
Direct Approach Rotated Green Triangle	10%	7%*
Oblique Approach Rotated Green Triangle	89%	11%

*The left float of the USV, which appears within the camera's field of view, was detected as a shape in most frames of this data set. This is the percentage of false positives that are not due to the left float.

Table 8. Image processing time of 1920 x 1080 pixel images using the APC algorithm

Minimum Image Processing Time (ms)	88.1
Maximum Image Processing Time (ms)	222
Average Image Processing Time (ms)	109

6.4 – Region of Interest Simulation Results

The automatic parameter configuration algorithm was tested on a single set of region of interest images. The region of interest images were generated from the direct approach rotated triangle dataset which was the data set the automatic parameter configuration algorithm performed worst on. The reason for using region of interest images from this data set was to explore the effects of shape detection rates on data sets that had performed poorly without region of interest information.

The region of interest data set consisted of the 100 images from the direct approach rotated triangle dataset where the sign containing the symbol has been recognised as a region of interest and sub images has been cropped out for specific shape detection analysis. The automatic parameter configuration algorithm remained the same from the previous data set tests. As with the previous data set tests, minimum, maximum and average image processing time were recorded.

Table 9. True and false symbol detection rate in 1920 x 1080 pixel images using ROI with the APC algorithm

Image Data Set	True Positive Rate	False Positive Rate
Direct Approach Rotated Green Triangle with Region of Interest	50%	2%

Table 10. Image processing time of 843 x 480 pixel images using ROI with the APC algorithm

Minimum Image Processing Time (ms)	0.86
Maximum Image Processing Time (ms)	5.55
Average Image Processing Time (ms)	3.50

6.5 – Pan-Tilt Servo System Simulation

A function added to the automatic parameter configuration algorithm that was not present in Arbon's algorithm is the ability to use symbol position information to drive other systems. Arbon's algorithm located the centres of symbols to visually display where it believed a symbol existed. However, this information was only used internally and could not be used to drive other systems. Using the frame transformations in section 2.2.4, it is possible to convert symbol location data into angles that can be used to drive the pan-tilt servo system.

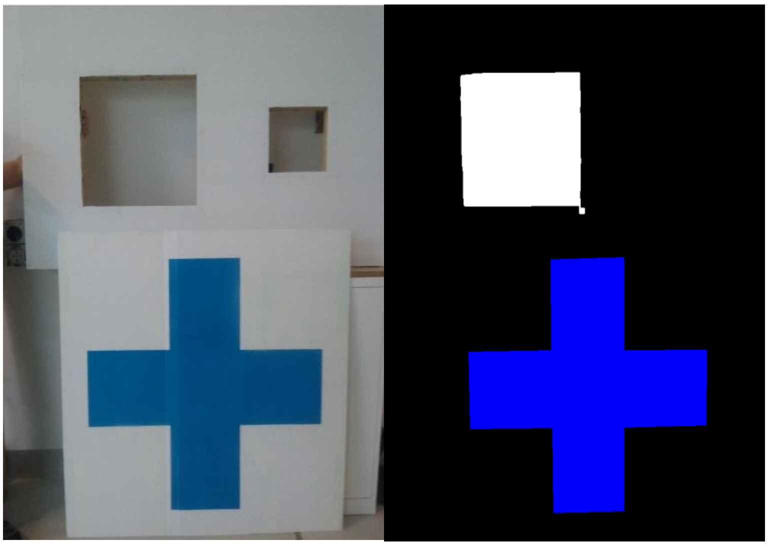
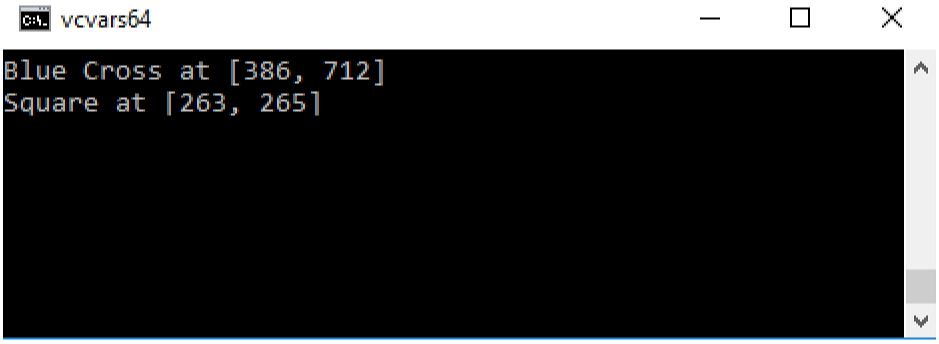


Figure 22. Original source image of target setup (left) and identified symbol and target hole (right). Both square holes were detected by the algorithm but only one was displayed as a target. The other hole can be set as the target in the algorithm by changing the target selection parameters.



```
Blue Cross at [386, 712]
Square at [263, 265]
```

Figure 23. Symbol location output from shape detection algorithm

As of the writing of this thesis, the pan-tilt servo system used to demonstrate the aiming of the ball launcher has yet to be fully constructed. However, it is expected to be completed for demonstration in the Flinders University Thesis Expo in late October 2018.

7 – Discussion

7.1 – Effects of Increasing Resolution on Shape Detection

From the results in section 6.2.1, the limitations of Arbon’s algorithm were clearly shown. Its detection rate of the green triangle, regardless of orientation and approach, is quite low with none of the data sets reaching detection rates above 40%. Of important note is that the detection rate of the blue cross was 0% regardless of approach to the symbol. The reason for this was found to be due to the blue colour of the cross. Compared to the dark green of the triangle symbol, a lighter blue was used for the cross. During binary thresholding, the lighter blue tended to fade into the white around the symbol causing the symbol to be eroded after thresholding to a state where it was no longer identifiable through contour analysis.

From the tests carried out with Arbon’s algorithm, it was found that the default parameter configuration tended to be more suited for darker lighting environments. As a result, it tended to provide poor shape detection for brighter environments. However, false positive rates also tended to be quite low for most datasets, with the exception of the direct approach rotated triangle dataset.

From the results in section 6.2.2, it was shown that even with the increase in resolution, Arbon’s algorithm at default values remained unsuited for detecting the blue cross, with detection rates remaining at 0% for both approaches. However, while the blue cross detection rate did not increase with resolution, the false positive rate increased drastically. As minimum shape size in Arbon’s algorithm did not scale with image resolution, the increased image resolution resulted in more small blobs passing through the size filter without being disregarded.

For the datasets containing the variations of the green triangle, the direct approach upright green triangle data set did not see a significant increase in detection rate with the increase in resolution. However, its false positive rate increased significantly from 1% to 21%. In a reverse case, with the increase in resolution, the direct approach rotated green triangle dataset had a significant decrease in detection rate by more than half from 37% to 14%. Despite this, the false positive rate also decreased by a similar factor from 15% to 6%. While these two data sets did not experience significant benefits from the increase in resolution, the oblique approach upright and rotated green triangle datasets showed significant increases in true positive detection rates with the detection rates of both datasets increasing by a factor of at least 2.

With the increase in image resolution, there is also an increase in processing time. From Table 2 and Table 4, it was shown that the average image processing time increased by a factor of 4.25 while the minimum and maximum image processing times increased by a factor of 4.5 and 3.9 respectively. The increase in processing time does not seem to have increased linearly with image size, with the increase in resolution increasing the pixel count by a factor of 6.75. However, the increase in average image processing time does cause some issues. At 843 x 480 pixels, the average image processing time is 37.4ms. This allows the algorithm to process images at frequencies higher than 10 images per second. At 1920 x 1080 pixels, the average image processing time jumps to 159ms, causing the algorithm to be unable to process 10 images per second, which is the camera and lidar update rate. This means that unless the algorithm can be optimised to process individual images in a shorter period of time, the increase in image resolution cannot be effectively utilised by Arbon’s algorithm without reducing the camera frame rate and lidar scan rate.

From the perspective of Arbon’s algorithm, the increase in image resolution does not appear to provide clear benefits or detriments to shape detection. Several datasets saw no significant increase true positive rates while having significant increases in false positive detection rates. Two other datasets showed significant benefit from the increase in image resolution, with their detection rates jumping from sub 40% to sub 90% with only minor increases to false positive rate. In addition, there was a single data set that suffered from the increase in image resolution with its true positive rate halving. Even considering the increased true positive rates for two datasets, the increased image resolution would still require further optimisation to allow it to process images at the USV’s current camera frame rate and lidar scan rate.

7.2 – Effects of Automatic Parameter Configuration on Shape Detection

From the results in section 6.3.1, the difference between Arbon’s algorithm and the automatic parameter configuration algorithm is shown. At 843 x 480 pixels, the automatic parameter configuration algorithm shows clear advantages as the blue cross datasets show significantly higher detection rates, with the direct approach blue cross dataset reaching 100% true positive rate. It was noted that the automatic parameter configuration algorithm demonstrated higher true positive detection rates across all datasets with the exception of the direct approach rotated green triangle dataset. This was found to be caused by method used by the algorithm to filter the contours found after binarisation. According to the task rules of the 2018 Maritime RobotX challenge, duplicate symbols are not used for any of the challenge tasks. As a result, the algorithm filters out all except the largest contour that may fit a specific symbol. The issue with the rotated green triangle dataset was that the left float of the USV was detected by the algorithm as being a large triangle. As the area of the left float was detected to be larger than the actual triangle symbol at most distances in the dataset, it negatively affected the detection rate of the triangle for that specific dataset. The left float of the USV had been frequently identified as a shape for most of the dataset tests. However, it had mainly been detected as a circle and as a result, did not have noticeable impact on the detection rate of the cross and triangle shapes used. However, if a circle symbol were to be tested, it is likely that it would be impacted more significantly than the other datasets tested.

From the tests carried out with the automatic parameter configuration algorithm, it was found that the automatic parameter configuration provides much higher true positive detection rates across most datasets. Compared to Arbon’s algorithm, it provides better detection rates across the blue cross datasets while providing equal or better detection rates for the green triangle datasets. It had a higher image processing time floor at 22.3ms but a lower average imaging processing time at 27.3ms.

From the results in section 6.3.2, it was shown that with the increase in resolution, the automatic parameter configuration algorithm maintained its high detection rate or performed significantly better across most datasets, again with the exception of the direct approach rotated green triangle dataset. The left float of the USV remained an issue, with the detection rate in the direct approach rotated green triangle dataset decreasing further due to the relatively higher increase in area of the left float compared to the symbol with the increase in image resolution. A note of interest is that the false positive detection rate decreased across most datasets with the increase in image resolution, compared to Arbon's algorithm, which experienced an increase in false positive detection rate with the increase in image resolution.

For the datasets containing the variations of the green triangle, the direct approach upright and oblique approach rotated datasets did not see much variation in true positive rate after increasing the image resolution. However, while the direct approach upright green triangle dataset saw a slight increase in false positive rates, the oblique approach rotated green triangle dataset saw its false positive detection rate drop by half. It was noted that the datasets that experienced issues detecting the USV's left float as a symbol continued to experience those issues with the increase in image resolution. While the relative size of the float to the rest of the image does not change with the increased resolution, its pixel area does increase exponentially with the increase in resolution. This may have caused some of the datasets to appear to have lower true positive rates due to the float increasing the number of false positive detections.

As with Arbon's algorithm, with the increase in image resolution, there is also an increase in processing time. From Table 6 and Table 8, it was shown that the average image processing time increased by a factor of 4 while the minimum and maximum image processing times increased by a factor of 1.4 and 8 respectively. From the large variation in the increase of processing times between the minimum and maximum, it shows that while the average image processing time of the automatic parameter configuration algorithm may be shorter than that of Arbon's algorithm at 1920 x 1080 pixels, the time that is required to process a single image is much less consistent. While the average image processing time of the automatic parameter configuration algorithm is 109ms, close to the 100ms required to process 10 frames per second, its wider range of image processing times makes it susceptible to taking significantly longer than the average 109ms to process a series of images. To utilise this algorithm as it is, while leaving a margin of error if a series of images uses an above average amount of time to process, the algorithm would likely have to be optimised further. However, it is possible to use this algorithm in conjunction with region of interest data to narrow the area for shape detection to improve shape detection accuracy and reduce image processing times.

For the automatic parameter configuration algorithm, the increase in image resolution appears to provide significant improvement in shape detection across most datasets. However, it does suffer from the misidentification of permanent objects within the camera's field of view, such as the USV's floats. Similar to Arbon's algorithm, the automatic parameter configuration algorithm will unlikely be able to process incoming camera images and lidar points of interest at the current frame and scan rate used on the USV. However, apart from further optimisation and streamlining of the code, another alternative is to use the points of interest data to limit the area of an image that needs to be processed by the algorithm. By cropping out the area of interest from the image where objects physically exist and running the algorithm over just the sub-image, it is possible to maintain the higher level of detail from the higher resolution images while decreasing the image pixel size to reduce the individual image processing time.

7.3 – Effects of Region of Interest on Shape Detection

As mentioned in section 7.2, the use of region of interest data for shape detection aims to do two things. Firstly, it reduces the effective area of an image that needs to be search for symbols to areas where objects are physically known to exist. Many false positive symbol detections are due to segmentation of water and distant objects on land into blobs that appear similar to a symbol. Although Arbon's algorithm and the automatic parameter configuration algorithm will only display the most likely shape out of a series of contours, the USV will typically not see any symbols at all as it transverses the Maritime RobotX Competition area unless it specifically nears the area for the Identify Symbol and Dock or Detect and Deliver Tasks. Because of this, it is not useful to process camera images to look for symbols when there are physically no objects nearby that could potentially have a symbol on them. Because of this, the region of interest data from the USV's FastSLAM system is used to limit the algorithm to only processing areas of an image that contain physical objects.

In section 6.4, the effectiveness of using region of interest to limit the area of an image to process was tested. In both the 843 x 480 pixel and 1920 x 1080 pixel images tests, the automatic parameter configuration algorithm had incorrectly labelled the left float of the USV as being a valid symbol while ignoring the actual symbol within the field of view of the camera. This misidentification is caused by the fact that the algorithm is incapable of differentiating between areas where real symbols are more likely to be based on image information alone. Using region of interest information, the area of the image to process can be narrowed down to exclude areas that can be misidentified as symbols, such as the floats of the USV.

From Table 9, it was found that using region of interest information to limit the image area to process, the shape detection rate increased from 10% with 7% false positives to a true positive rate of 50% with 2% false positive detections. This is a substantial increase in detection rate with the use of region of interest information, but also displays two important issues when using region of interest. Using region of interest to limit the area to process does not guarantee that a symbol will be identified in the area, even if it does exist. This is dependent on how clearly the symbol is displayed within the region of interest and how well the algorithm can differentiate the symbol from its immediate surroundings. The second issue is that when a subsection of the image is cut out to be processed, filters such as normalisation and adaptive binary thresholding start to produce more noise due to the reduced sample area to perform filters like normalisation. It is possible to apply all filters up to the binary threshold before using region of interest information to limit the contour analysis area, but that potentially reduces the processing time benefits gained by using region of interest information.

In Table 10, it was shown that for the direct approach rotated green triangle dataset, the average processing time for region of interest areas was 3.5ms, with the minimum and maximum image processing times ranging from 1.86ms to 5.55ms respectively. This is faster than the average processing time of a full image by a factor of 31. From this, it was shown that benefits of using region of interest information for shape detection not only lie in its ability to increase shape detection rates by a significant amount, but also its ability to significantly reduce image processing times.

8 – Conclusions and Recommendations

8.1 – Conclusions

This thesis set out to incorporate a target tracking system for a ball launcher onto a WAM-V USV for the purpose of completing the Detect and Deliver task from the 2018 Maritime RobotX Challenge.

A target tracking algorithm for a ball launcher using on-board cameras and region of interest information was developed in this thesis. The target tracking algorithm was designed for enabling a pan-tilt servo system to aim at and follow a target as well as to account for different ambient light conditions that may alter the relative appearance of the target symbol at varying distances and angles. Using a higher resolution image stream from the cameras and region of interest data from FastSLAM algorithm and lidar sensor, the shape detection algorithm was able to identify a target and produce target position data for a pan-tilt servo system to be aimed. It was also shown that the target tracking algorithm was capable of reliably identifying symbols from varying angle and distances.

To support this thesis with strong evidence, the target tracking algorithm was simulated using real world data collected during an operational field test of the USV at Port River, Adelaide. Running the collected data through Arbon's shape detection algorithm and the automatic parameter configuration algorithm, the results showed that high resolution images provided marginal improvements to Arbon's algorithm but significant performance improvements for the automatic parameter configuration algorithm. Because of the increased image resolution, the increase in image processing time means that both algorithms are unable to process image output at the USV's current camera frame rate. Nevertheless, the use of region of interest information was found to reduce image processing time significantly in addition to increasing shape detection rates. Using the automatic parameter configuration algorithm in conjunction with region of interest information, the target tracking system was able to reliably identify a symbol shape and produce the output required to aim the pan-tilt servo system in the direction of the target with controlled accuracy.

8.2 – Recommendations

It is recommended that a ball launcher system be constructed and used in conjunction with the target tracking system to produce a complete ball launcher and target tracking system that can be mounted and operated on a WAM-V USV. It is also recommended that GPU acceleration be explored into as a method of reducing image processing times.

9 – Bibliography

Arbon, T. (2014). *Computer Vision for RobotX* (Bachelor thesis, Flinders University)

Bradshaw, A., 1991, April. The UK security and fire fighting advanced robot project. In *Advanced robotic initiatives in the UK, IEE colloquium on* (pp. 1-1). IET.

Breaux, J.J., 1986. Initial reactions to a fire from a simple robotic device. *Fire Safety Science*, *1*, pp.571-580.

Choi, H., Jung, W., & Kim, N. (2017, February). Development of autonomous ball launcher system on surface vehicle for detect and deliver mission in 2016 Maritime RobotX Challenge. In *Underwater Technology (UT), 2017 IEEE* (pp. 1-5). IEEE.

Cox, C., Beard, R. and Gates, S., 1988. *Remotely Operated Robotic Firefighter* (No. AMETEK-16.71). AMETEK INC SANTA BARBARA CA OFFSHORE RESEARCH AND ENGINEERING DIV.

Firearms Act 2015, Available at: <https://www.legislation.sa.gov.au/LZ/C/A/FIREARMS%20ACT%202015/CURRENT/2015.46.AUTH.PDF> , Accessed: 12th October 2018

HobbyKing (2018). *HobbyKing HK15328D Servo Motor*., image, accessed 12th October 2018, <https://cdn-global-hk.hobbyking.com/media/catalog/product/cache/1/image/660x415/17f82f742ffe127f42dca9de82fb58b1/legacy/catalog/28972.jpg>

McGhee, R.B., 1979. Future prospects for sensor-based robots. In *Computer vision and sensor-based robots* (pp. 323-334). Springer, Boston, MA.

McNeil, J. G. (2016). *Autonomous Fire Suppression Using Feedback Control for Robotic Firefighting* (Doctoral dissertation, Virginia Tech).

Minsky, M., 1972. Mini-robot Proposal to ARPA.

Miyazawa, K. (2002). Fire robots developed by the Tokyo Fire Department. *Advanced Robotics*, *16*(6), 553-556.

Office of Naval Research 2013, *Students to Face-Off in New RobotX Game*, media release, accessed 29th September 2018, <https://www.onr.navy.mil/en/Media-Center/Press-Releases/2013/Maritime-Robot-X-ONR-AUVSI>

Orsag, M., Korpela, C., Oh, P., Bogdan, S. 2018, *Aerial Manipulation*, Springer International Publishing

Pla, F. and Traver, V.J., 2002. Spherical panoramas for pan-tilt camera motion compensation in space-variant images. In *Topics in Artificial Intelligence* (pp. 375-387). Springer, Berlin, Heidelberg.

RoboNation 2018c, *2016 Maritime RobotX Challenge Task Descriptions*, RoboNation, accessed 8th October 2018, <https://www.robotx.org/images/files/2016-MRC-Tasks-2016-11-28.pdf>

RoboNation, 2018a, *About Us*, RoboNation, accessed 29th September 2018, <https://www.robonation.org/about-us>

RoboNation 2018b, *Task Descriptions and Specifications*, RoboNation, accessed 29th September 2018, <https://www.robotx.org/images/RobotX-2018-Tasks_v2.0.pdf>

Smith, D.L., Petroka, R.P., Yobs, R.L., Lewis, D. and McCarthy, W., 1985. *A mechanical predesign project in robotic fire fighting*. Monterey, California. Naval Postgraduate School.

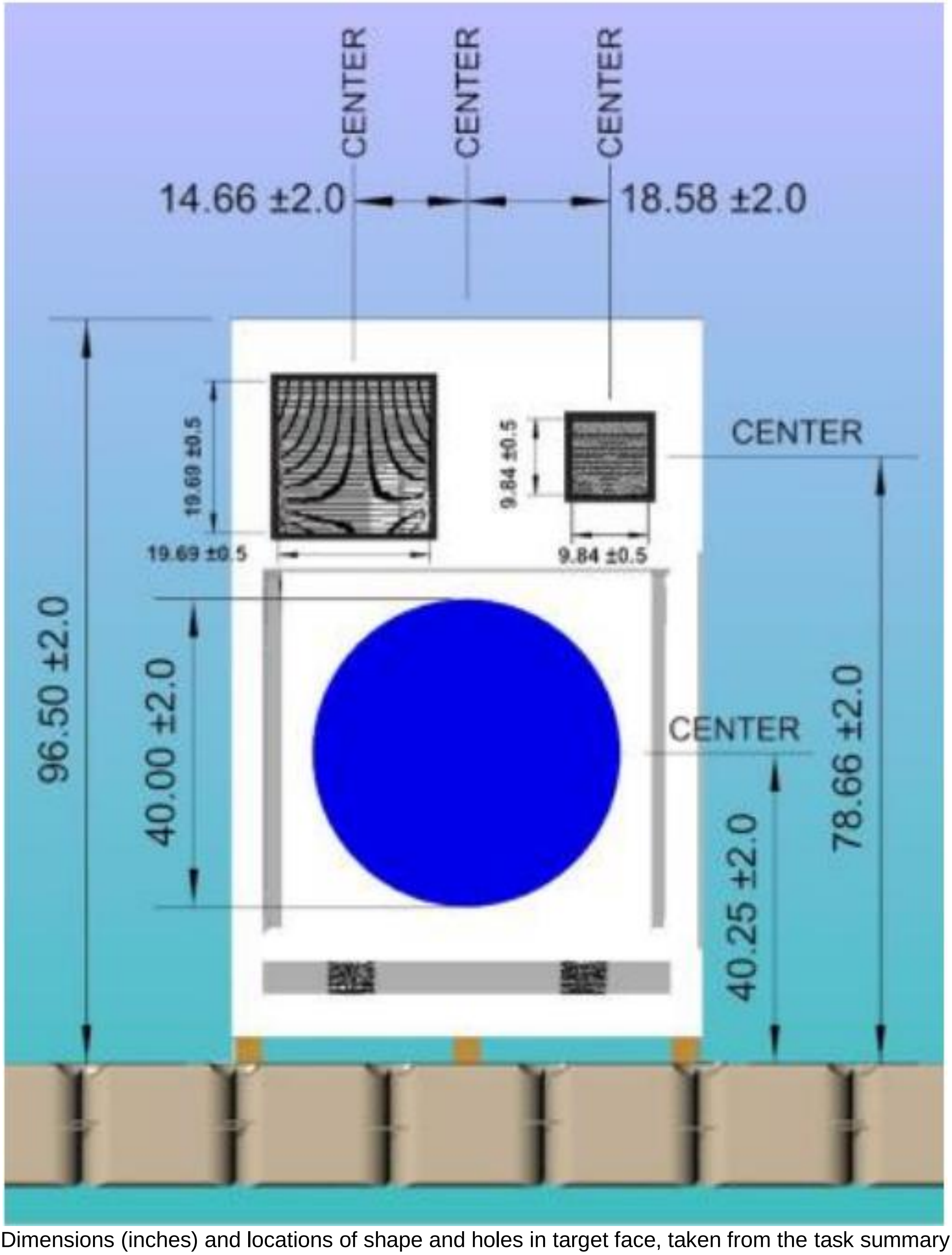
Tadokoro, S., Takamori, T., Osuka, K. and Tsurutani, S., 2000. Investigation report of the rescue problem at Hanshin-Awaji earthquake in Kobe. In *Intelligent Robots and Systems, 2000.(IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on* (Vol. 3, pp. 1880-1885). IEEE.

Webb, A., Donnelly, B., Stewart, J., Wheare, J., Kossatz, M., Hutchinson, S., Geyer, S., Crouch, T., Lammas, A., Sammut, K., 2014, *Development and Testing of the TopCat Autonomous Surface Vessel for the Maritime RobotX Challenge 2016*, accessed 8th October 2018, <https://www.robotx.org/images/files/FlindersUni_2016_RobotX_Journal.pdf>

White, T. (2015). Making Sailors 'SAFFiR' - Navy Unveils Firefighting Robot Prototype at Naval Tech EXPO. Retrieved from https://www.onr.navy.mil/en/Media-Center/Press-Releases/2015/saffir-ship-firefighting-robot-prototype

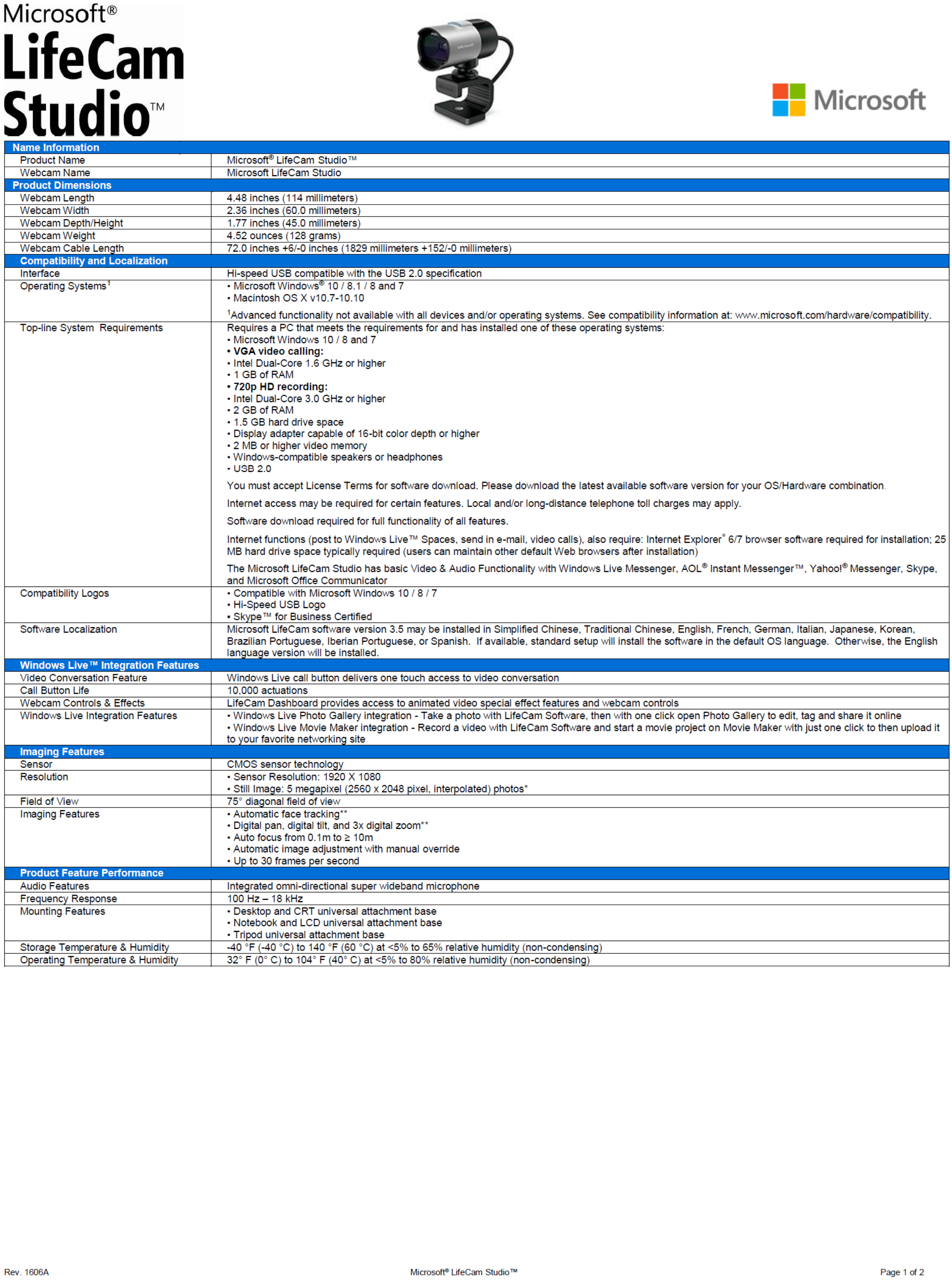
Zhou, Y., Wilkins, D. and Cook, R.P., 1998. Neural network control for a fire-fighting robot. *Software-Concepts & Tools*, *19*(3), pp.146-152.

Appendix A – Detect and Deliver Task Target Dimensions



Dimensions (inches) and locations of shape and holes in target face, taken from the task summary (RoboNation, 2018c)

Appendix B – Microsoft LifeCam Studio Datasheet



Microsoft®
LifeCam
Studio™

Microsoft

Name Information	
Product Name	Microsoft® LifeCam Studio™
Webcam Name	Microsoft LifeCam Studio
Product Dimensions	
Webcam Length	4.48 inches (114 millimeters)
Webcam Width	2.36 inches (60.0 millimeters)
Webcam Depth/Height	1.77 inches (45.0 millimeters)
Webcam Weight	4.52 ounces (128 grams)
Webcam Cable Length	72.0 inches +6/-0 inches (1829 millimeters +152/-0 millimeters)
Compatibility and Localization	
Interface	Hi-speed USB compatible with the USB 2.0 specification
Operating Systems[1]	• Microsoft Windows® 10 / 8.1 / 8 and 7 • Macintosh OS X v10.7-10.10 [1]Advanced functionality not available with all devices and/or operating systems. See compatibility information at: www.microsoft.com/hardware/compatibility.
Top-line System Requirements	Requires a PC that meets the requirements for and has installed one of these operating systems: • Microsoft Windows 10 / 8 and 7 **• VGA video calling:** • Intel Dual-Core 1.6 GHz or higher • 1 GB of RAM **• 720p HD recording:** • Intel Dual-Core 3.0 GHz or higher • 2 GB of RAM • 1.5 GB hard drive space • Display adapter capable of 16-bit color depth or higher • 2 MB or higher video memory • Windows-compatible speakers or headphones • USB 2.0 You must accept License Terms for software download. Please download the latest available software version for your OS/Hardware combination. Internet access may be required for certain features. Local and/or long-distance telephone toll charges may apply. Software download required for full functionality of all features. Internet functions (post to Windows Live™ Spaces, send in e-mail, video calls), also require: Internet Explorer® 6/7 browser software required for installation; 25 MB hard drive space typically required (users can maintain other default Web browsers after installation) The Microsoft LifeCam Studio has basic Video & Audio Functionality with Windows Live Messenger, AOL® Instant Messenger™, Yahoo!® Messenger, Skype, and Microsoft Office Communicator
Compatibility Logos	• Compatible with Microsoft Windows 10 / 8 / 7 • Hi-Speed USB Logo • Skype™ for Business Certified
Software Localization	Microsoft LifeCam software version 3.5 may be installed in Simplified Chinese, Traditional Chinese, English, French, German, Italian, Japanese, Korean, Brazilian Portuguese, Iberian Portuguese, or Spanish. If available, standard setup will install the software in the default OS language. Otherwise, the English language version will be installed.
Windows Live™ Integration Features	
Video Conversation Feature	Windows Live call button delivers one touch access to video conversation
Call Button Life	10,000 actuations
Webcam Controls & Effects	LifeCam Dashboard provides access to animated video special effect features and webcam controls
Windows Live Integration Features	• Windows Live Photo Gallery integration - Take a photo with LifeCam Software, then with one click open Photo Gallery to edit, tag and share it online • Windows Live Movie Maker integration - Record a video with LifeCam Software and start a movie project on Movie Maker with just one click to then upload it to your favorite networking site
Imaging Features	
Sensor	CMOS sensor technology
Resolution	• Sensor Resolution: 1920 X 1080 • Still Image: 5 megapixel (2560 x 2048 pixel, interpolated) photos*
Field of View	75° diagonal field of view
Imaging Features	• Automatic face tracking** • Digital pan, digital tilt, and 3x digital zoom** • Auto focus from 0.1m to ≥ 10m • Automatic image adjustment with manual override • Up to 30 frames per second
Product Feature Performance	
Audio Features	Integrated omni-directional super wideband microphone
Frequency Response	100 Hz – 18 kHz
Mounting Features	• Desktop and CRT universal attachment base • Notebook and LCD universal attachment base • Tripod universal attachment base
Storage Temperature & Humidity	-40 °F (-40 °C) to 140 °F (60 °C) at <5% to 65% relative humidity (non-condensing)
Operating Temperature & Humidity	32° F (0° C) to 104° F (40° C) at <5% to 80% relative humidity (non-condensing)

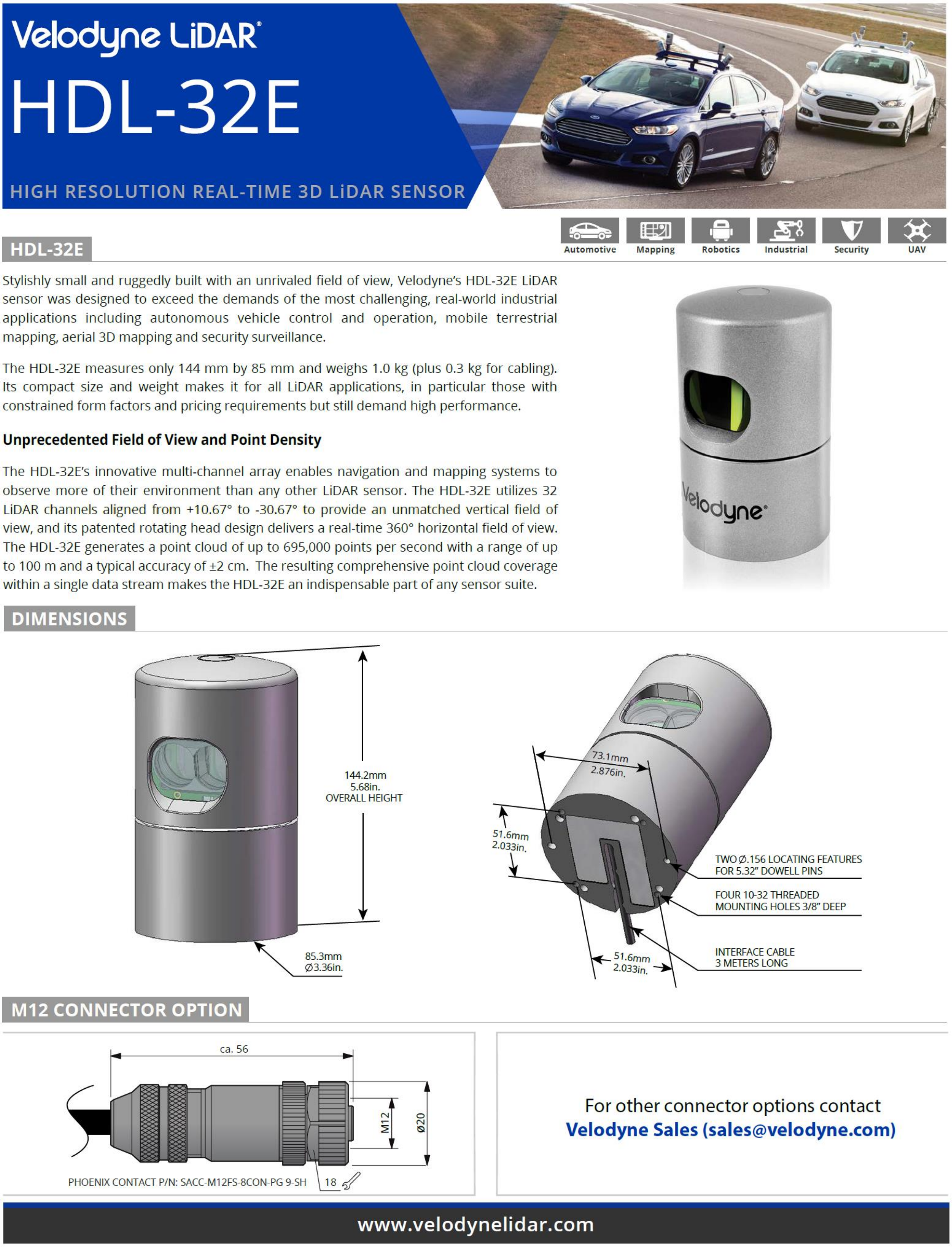
Microsoft® LifeCam Studio™

Microsoft

Certification Information	
Country of Manufacture	People's Republic of China (PRC)
ISO 9001 Qualified Manufacturer	Yes
ISO 14001 Qualified Manufacturer	Yes
Restriction on Hazardous Substances	This device complies with all applicable worldwide regulations and restrictions including, but not limited to: EU directive 2002/95/EC on the Restriction of the Use of Certain Hazardous Substances in Electrical and Electronic Equipment and EU Registration Evaluation and Authorization of Chemicals (REACH) regulation regarding Substances of Very High Concern.
FCC ID	This device complies with Part 15 of the FCC Rules and Industry Canada ICES-003. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation. Tested to comply with FCC standards. For home and office use. Model number: 1425, LifeCam Studio.
Agency and Regulatory Marks	• ACMA Declaration of Conformity (Australia and New Zealand) • ICES-003 report on file (Canada) • EIP Pollution Control Mark, EPUP (China) • WEEE (European Union) • CE Declaration of Conformity (European Union) • VCCI Certificate (Japan) • MIC Certificate (Korea) • EAC Certificate (Russia) • FCC Declaration of Conformity (USA) • CB Scheme Certificate (International)
Warranty	3 years
Windows Certification Kit (WCK)	ID: 1760441 (32-bit and 64-bit) Microsoft Windows 10, 1609318 (32-bit) and 1604420 (64-bit) Microsoft Windows 8.1

Results stated herein are based on internal Microsoft testing. Individual results and performance may vary. Any device images shown are not actual size. This document is provided for informational purposes only and is subject to change without notice. Microsoft makes no warranty, express or implied, with this document or the information contained herein. Review any public use or publications of any data herein with your local legal counsel.

Appendix C – HDL-32E Velodyne Lidar Datasheet



Velodyne LiDAR®

HDL-32E

HIGH RESOLUTION REAL-TIME 3D LiDAR SENSOR

HDL-32E

Automotive Mapping Robotics Industrial Security UAV

Stylishly small and ruggedly built with an unrivaled field of view, Velodyne's HDL-32E LiDAR sensor was designed to exceed the demands of the most challenging, real-world industrial applications including autonomous vehicle control and operation, mobile terrestrial mapping, aerial 3D mapping and security surveillance.

The HDL-32E measures only 144 mm by 85 mm and weighs 1.0 kg (plus 0.3 kg for cabling). Its compact size and weight makes it for all LiDAR applications, in particular those with constrained form factors and pricing requirements but still demand high performance.

Unprecedented Field of View and Point Density

The HDL-32E's innovative multi-channel array enables navigation and mapping systems to observe more of their environment than any other LiDAR sensor. The HDL-32E utilizes 32 LiDAR channels aligned from +10.67° to -30.67° to provide an unmatched vertical field of view, and its patented rotating head design delivers a real-time 360° horizontal field of view. The HDL-32E generates a point cloud of up to 695,000 points per second with a range of up to 100 m and a typical accuracy of ±2 cm. The resulting comprehensive point cloud coverage within a single data stream makes the HDL-32E an indispensable part of any sensor suite.

DIMENSIONS

M12 CONNECTOR OPTION

For other connector options contact
Velodyne Sales (sales@velodyne.com)

www.velodynelidar.com

HDL-32E

High Definition Real-Time 3D LiDAR Sensor

The HDL-32E provides high definition 3-dimensional information about the surrounding environment.

	Specifications:
Sensor:	• 32 Channels • Measurement Range: Up to 100 m • Range Accuracy: Up to ±2 cm (Typical)[1] • Single and Dual Returns (Strongest, Last) • Field of View (Vertical): +10.67° to -30.67° (41.33°) • Angular Resolution (Vertical): 1.33° • Field of View (Horizontal): 360° • Angular Resolution (Horizontal/Azimuth): 0.1° – 0.4° • Rotation Rate: 5 Hz – 20 Hz • Integrated Web Server for Easy Monitoring and Configuration
Laser:	• Laser Product Classification: Class 1 Eye-safe per IEC 60825-1:2007 & 2014 • Wavelength: 903 nm
Mechanical/ Electrical/ Operational	• Power Consumption: 12 W (Typical)[2] • Operating Voltage: 9 V – 18 V (with Interface Box and Regulated Power Supply) • Weight: ~1.0 kg (without Cabling and Interface Box) • Dimensions: See diagram on previous page • Environmental Protection: IP67 • Operating Temperature: -10°C to +60°C[3] • Storage Temperature: -40°C to +105°C
Output:	• 3D LiDAR Data Points Generated: - Single Return Mode: ~695,000 points per second - Dual Return Mode: ~1,390,000 points per second • 100 Mbps Ethernet Connection • UDP Packets Contain: - Time of Flight Distance Measurement - Calibrated Reflectivity Measurement - Rotation Angles - Synchronized Time Stamps (μs resolution) • Orientation: 6DoF Inertial Sensor Measurements • GPS: $GPRMC and $GPGGA NMEA Sentences from GPS Receiver (GPS not included)

For more details and ordering information, contact Velodyne Sales (sales@velodyne.com)

[1] Typical accuracy refers to ambient wall test performance across most channels and may vary based on factors including but not limited to range, temperature and target reflectivity.

[2] Operating power may be affected by factors including but not limited to range, reflectivity and environmental conditions.

[3] Operating temperature may be affected by factors including but not limited to air flow and sun load.

⚠ CLASS 1 LASER PRODUCT

Appendix D – HK15328D Servo Motor Specifications

SKU	9225000014
Voltage	4.5-6V
Speed	0.26sec/60deg(4.8v) 0.22sec/60deg (6.0v)
Torque	10kg.cm (4.8v) 12.8kg.cm (6.0v)
Size	40.9mm x 20mm x37.75 mm
Weight	58gram
Motor	Coreless
Gear Material	Metal
Ball Bearing	2
Type	Analogue
Spline	24