TR08 - IMAGE PROCESSING SUBSYSTEM COMPONENT TEST REPORT

<table>
<tr><td>Project: 450G1
WP Name: Component Testing
WP Number: WP-IMP-04</td><td colspan="2">Type of Test: Unit Test</td></tr>
<tr><td>Test Articles:
Raspberry Pi Model 2B+
Raspberry PI USB to 3-pin adapter
USB Wifi Adapter
Raspberry PICamera Module V2</td><td>Part Number:
MICRO COMP
RASP. PI ADAPTER
USB ADAPTER
CAM</td><td>Serial Number:
Raspberry Pi 2 Model B
Camera Module V2</td></tr>
<tr><td>System Requirements to Be Confirmed:
[REQ-04-01-01-U]

[REQ-04-01-U]</td><td colspan="2">Test Equipment:
See Section 4</td></tr>
<tr><td>Test Operators:
Harry Akeroyd</td><td colspan="2">Test Engineer:
Harry Akeroyd</td></tr>
<tr><td>WP Group Manager:
Hope Sneddon</td><td colspan="2">WP Supervisor:
Felipe Gonzalez</td></tr>
</table>

Version History/ Revision Record

Version	File Name	Description of Issue	Date	Approved
1.0	349G1-IMP-TR08-01	Initial Issue	15/04/2019	Hope Sneddon
2.0	450G1-IMP-TR08-02	Semester 2 revisions (Addition of validation table and classifier results)	25/07/19	Hope Sneddon

Contents:

Figures:

Tables:

Definitions

UAV	Unmanned Aerial Vehicle
UAS	Unmanned Aircraft System
OS	Operating System
GCS	Ground Control Station
USB	Universal Serial Bus
HLO	High Level Objectives
CSI	Camera Serial Interface
FPS	Frames Per Second
CMOS	Complementary Metal-Oxide-Semiconductor
UAVUSR	Search and Rescue Unmanned Aerial Vehicle
UAV	Unmanned Aerial Vehicle
QUT	Queensland University of Technology
ConOps	Concept of Operations
GPS	Global Positioning System
PMP	Project Management Plan

1.0 Introduction

The inspection and verification of functionality of each engineering subsystem is imperative to the success of the overall UAV project. Hardware design and functionality must be confirmed at key gateway stages within the project to ensure all high-level objectives and system requirements are achieved at the completion of the project. Specifically, component testing based on final design documentation minimises the accumulation of errors and loss of expected functionality when sub-systems are integrated into one overall design. Component testing is therefore imperative for the success of the project.

1.1 Scope

This document outlines the testing procedure and results for component testing of the Image Processing subsystem in order to verify it can be integrated into the overall UAV design. Testing will verify the correct functionality of components outlined below. This testing will involve a component testing setup, procedure, results and analysis. This is to ensure all components work as intended before the complexity of the UAV design is developed further.

1.2 Background

The QUT Airborne Systems Lab (ASL) is a world-leading research centre based in Brisbane, Australia. QUT ASL conducts research into on autonomous technologies which support the development of autonomous aircraft with on-board sensor systems for a wide range of commercial applications. Group 1 has been tasked to develop a UAVUSR that is able to successfully and autonomously conduct constant search around a simulated urban environment after a natural disaster event. During its mission, the UAV must identify and locate two human targets and deploy the correct emergency medication. Throughout the design and delivery of this project, the UAVUSR will be designed in accordance with industry-standard Systems Engineering practices (a rigorous and disciplined engineering methodology for complex projects) to ensure all customer requirements are met. Refer to RD/1 for full customer requirements and project definition.

2.0 Reference Documents

2.1 QUT Avionics Documents

RD/1	SR19G1-UAVTAQ-2019-Urban Search and Rescue-2019-02-18-Final Customer Needs	UAV for Urban Search and Rescue – UAVUSR: 2019
RD/2	349G1-PM-PMP-05F	Project Management Plan
RD/3	349G1-PM-SysReq-01F	System Requirements
RD/4	349G1-Image_Processing-PD-01F	Image Processing Subsystem Preliminary Design
RD/5	349G1-PM-ICD-02	Interface Control Document

2.2 Non-QUT Documents

RD/5	Raspberry Pi 3 Model B	https://www.inet.se/files/pdf/1974044_0.pdf
RD/6	Raspberry Pi Camera Module V2	http://au.element14.com/raspberry-pi/rpi-8mpcamera-board/raspberry-pi-camera-boardv2/dp/2510728
RD/7	MicroSDHC card with adapter	https://www.jbhifi.com.au/cameras/lexar/lexar-high-speed-microsdhc-16gb-uhs-1-300x45mb-s/586542/
RD/8	Wall Charger and Micro USB Cable	http://mobileshop.amaysim.com.au/aos-chargerusb-cable-2a-microusb.html
RD/9	Laptop (6th gen Lenovo X1 Carbon)	https://www.2compute.net/image-library/LNVO/pdf/20KG0026MB.pdf

3.0 Test Objectives

The testing of this subsystem is to confirm that components provided by the customer to be used in the Image Processing system function as advertised. Should the component testing be successful, these elements will them be integrated into the project as specified in the system architecture (RD/2) and subsystem designs (RD/4). The components must satisfy the relevant system requirements in order to be considered as passable (RD/3).

Relevant System Requirements to Image Processing Testing:

Table 1: Relevant system requirements to be evaluated (RD/3)

Requirement as per RD/3	Description
REQ-03-F	The UAV^{U}SR must successfully locate and correctly identify both injured people using on-board systems.
REQ-03-01-F	The identification of patient location shall be within an accuracy of 50cm.
REQ-03-02-F	The identification system shall have zero tolerance for false positives.
REQ-04-01-U	All sensor information must be displayed in real time on the GCS.
REQ-04-01-01-U	Live single-capture imagery as observed by the UAV^{U}SR shall be displayed to the customer on the control station displays.
REQ-05-01-F	Once a marker is found, the UAV^{U}SR shall immediately stop the flight path and hovers above the marker location.
REQ-05-04-F	The patient injury type shall be correctly identified via the onboard SoC chip.

These requirements will be further verified as testing progresses throughout the project.

4.0 Test Set-up and Equipment

4.1 Image Processing Subsystem

4.1.1 Test Setup

The following diagrams depict the setup and testing relationships relevant to the Image Processing subsystem component testing.

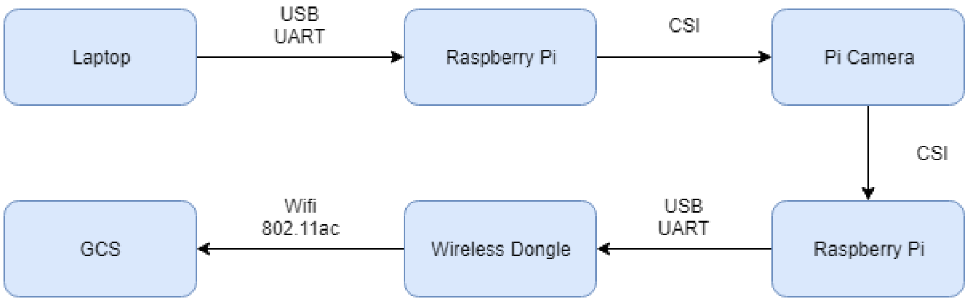


Figure 1: Raspberry Pi & Pi Camera Testing solution.

The supplied hardware is initially visually inspected for any obvious damage or missing components. Any component that looks unfit for use in the relevant application will result in a fail in the hardware integrity test. Should this consequently compromise the overall functionality/ safety of the UAV, component testing of the subsystem will be labelled as a failure overall and any other design/ testing of UAV systems shall not progress until failures have been resolved.

4.1.2 Equipment

Pi Camera V2

Figure 2: Pi Camera V2 (RD/7)

The Pi Camera v2 is light-weight, high quality camera, capable of providing video with a resolution of 1080p at 30 Frames Per Second (FPS). Additionally, the camera interfaces with the Raspberry Pi well, as the camera is built specifically to work with the Raspberry Pi, evident with the one connector for power and data.

Raspberry Pi 2 Model B

Figure 3: Raspberry Pi Microcomputer (RD/6)

The Raspberry Pi is a small form factor, lightweight micro-computer. The Raspberry Pi is used for processing imagery onboard as a standard size desktop or laptop would be too large for in terms of the mass budget for the UAV. It will communicate with the GCS via wifi as described in RD/5.

Laptop

Figure 4: GCS Laptop (RD/10)

This is a generic laptop capable of running the required GCS software. In this case the laptop is specifically a 6th generation Lenovo X1 Carbon.

5.0 Procedure

5.1 Hardware Inspection Test

Each individual component is visually inspected for any damage out of its packaging, this is to ensure that it has arrived in working condition and further action may need to be taken if it is not. Parts supplied with each component are also tested to see if they have been correctly supplied meeting the specifications of the supplier. A pass will be granted to the component provided that it meets specification, otherwise a fail will be assigned, and further action will be considered before utilising the component.

5.2 Assumptions

Before the testing procedure can commence it is assumed that the following steps have been completed and setup correctly:

1) The Raspberry Pi is flashed correctly, is connected to the same WiFi as the GCS and the Pi Camera is connected to the appropriate port on the Pi.
2) The GCS system is setup correctly and connected to a local WiFi network.

The steps required to setup the mentioned systems can be found in the “EGB349 – Practical Guide” document under the “Target Acquisition & Image Processing” and the “Operator Interfaces” respectively.

5.3 Testing Setup

The following figure represents the setup used to test the image processing hardware. As shown the Raspberry Pi and GCS (laptop) are not physically connected as would be in operation, this is evident by the blue USB lead being disconnected. This proves the Raspberry Pi is communication with the GCS using WIFi. The Pi Camera is pointed at a Rubik’s cube, demonstrating the clear image and colour definition of the Pi Camera. The only differences between this testing setup and the final product is the Raspberry Pi is not processing the images and the Pi is connected to a wall outlet whereas onboard the UAV it will be connected to the batteries using the connection pins.

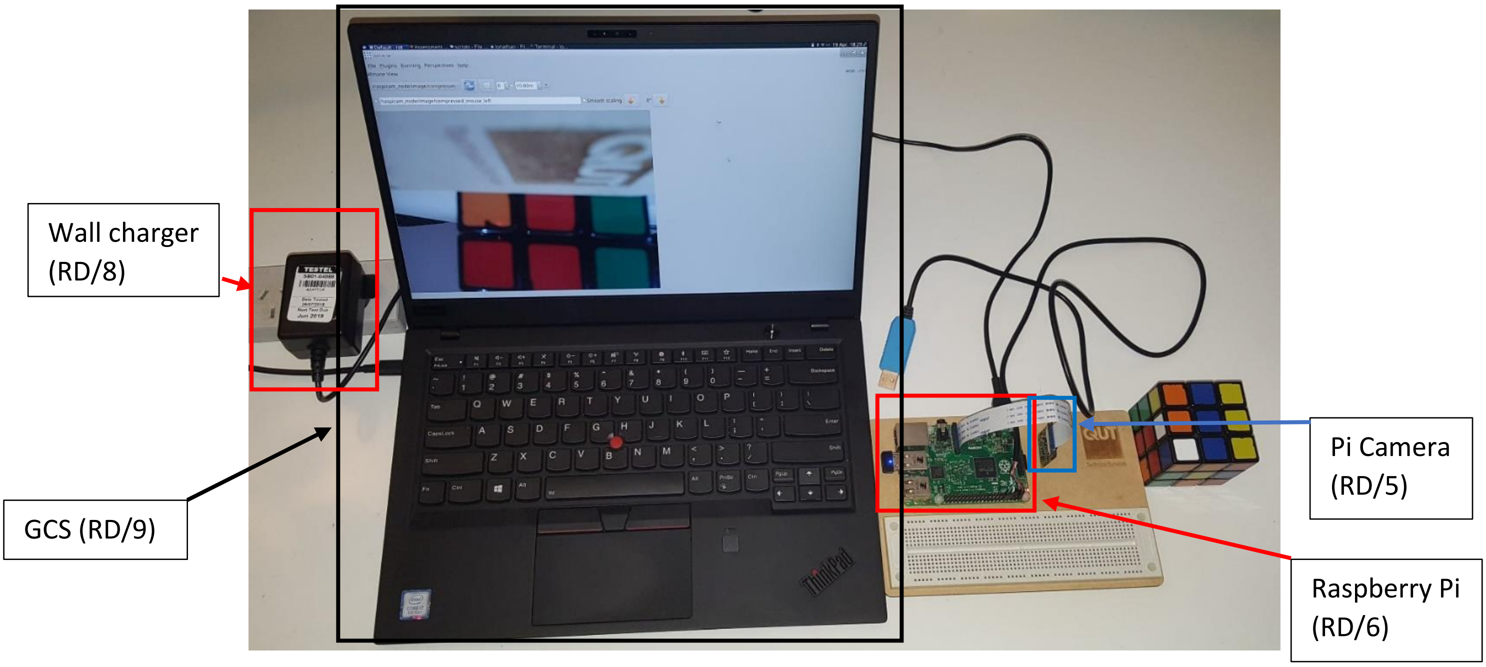


Figure 5: Testing setup for Image Processing subsystem. (RD/ 5-9)

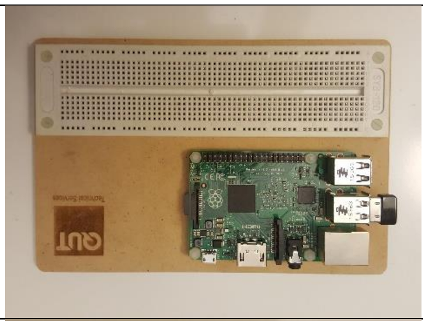
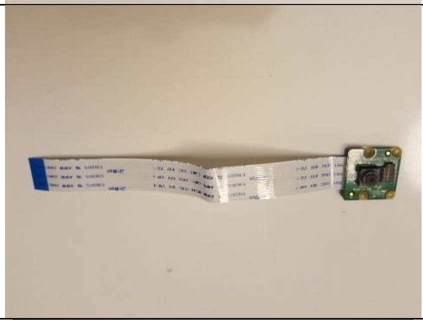
5.4 Hardware Performance Test

Hardware performance tests will be conducted to ensure hardware provided behaves as specified on the component specifications manuals. Hardware inspection tests have been passed, component operational functionality will be tested to ensure performance is as specified.

6.0 Results

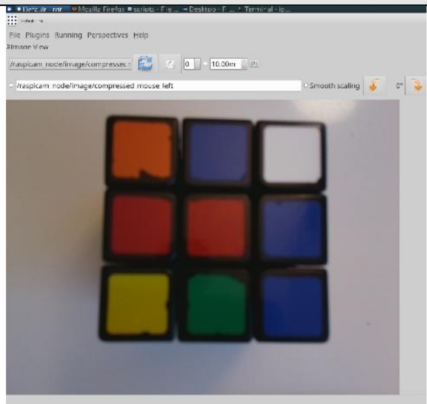
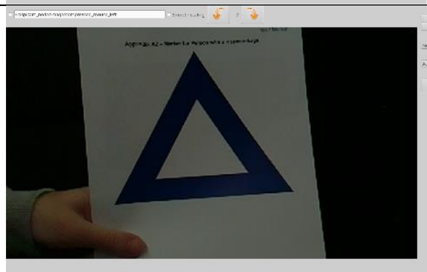
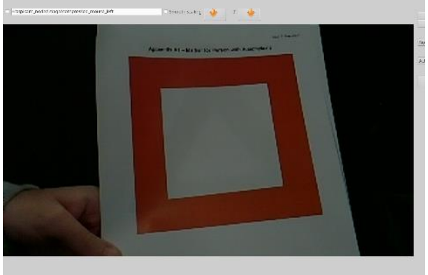
6.1 Hardware Inspection Test

Table 2: Hardware inspection results.

Component	Results (Pass/Fail)	Comments
GCS Laptop	Pass	
Raspberry Pi 2 Model B V1.1	Pass	
Raspberry Pi Camera V2.1	Pass	
Raspberry Pi USB to 3-pin adapter	Pass	
USB Wi-Fi Adapter	Pass	

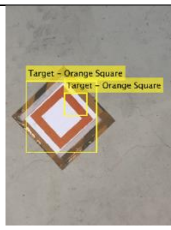
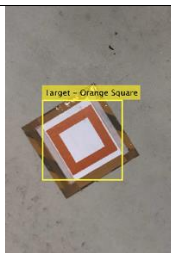
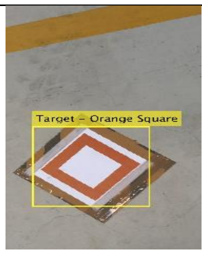
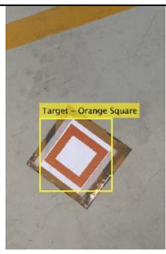
6.2 Hardware Performance Test

Table 3: Hardware Performance Test Results.

Component	Result (Pass/Fail)	Computer Display Confirmation
Raspberry Pi 2 Model B	Pass	
Pi Camera V2	Pass	
	Pass	 

6.3 Software Performance Test

6.3.1 Object Detection Software – Cascade Classifier

Test Number	Results (Pass/Fail)	Photo Evidence
1	Pass	
2	Fail – Identification of too many targets	
3	Pass	
4	Pass	
5	Pass	
6	Pass	

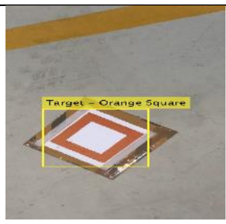
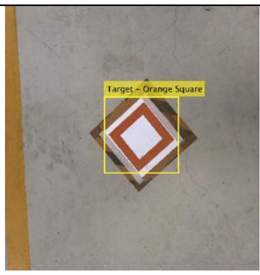
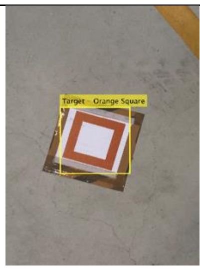
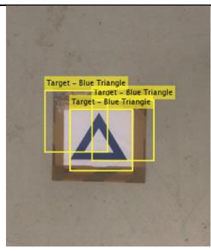
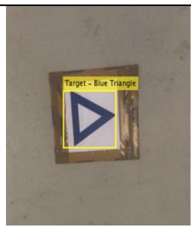
7	Pass	
8	Pass	
9	Pass	
10	Pass	

Table 4: Target Acquisition: Target One: Orange Square

From Table 4 it can be seen that the first classifier had a pass rate of 90%. It can be seen that the ‘failed’ images did recognise and identify the image, however, they also identified a second ROI, this can be seen in test numbers two, five and ten. This was seen as an irregularity as during all final test flight of autonomous flight no false positive or false negatives were identified. Furthermore, this error in the identification was attributed to shadowing of the image due to the UAV being flown in ‘hand-held’ flight when the image was captured.

Test Number	Results (Pass/Fail)	Photo Evidence
1	Pass	
2	Fail – Identification of too many objects	
3	Pass	
4	Pass	
5	Pass	
6	Pass	

7	Pass	
8	Pass	
9	Pass	
10	Pass	

Table 5: Target Acquisition: Target Two: Blue Square

From Table 5 it can be seen that the second classifier had a pass rate of 90%. It can be seen that the 'failed' images did recognise and identify the image, however, they also identified a second ROI, this can be seen in test number two. This was seen as an anomaly as during testing of autonomous flight no false positive or false negatives were identified. Furthermore, this error in the identification was attributed to shadowing of the image due to the UAV being flown in 'hand-held' flight when the image was captured.

2.1.1 Target Localisation Software

To ensure target localisation could be achieved based on the detection of an image: a transform tree was used to display the parent and child relationships such that a connection could be made for each instances of this relationship. The consequence of initiating these relationships was that the target location could be identified in the world, as can be seen in figure 7. For the design of this works please refer to final design document of Image Processing.

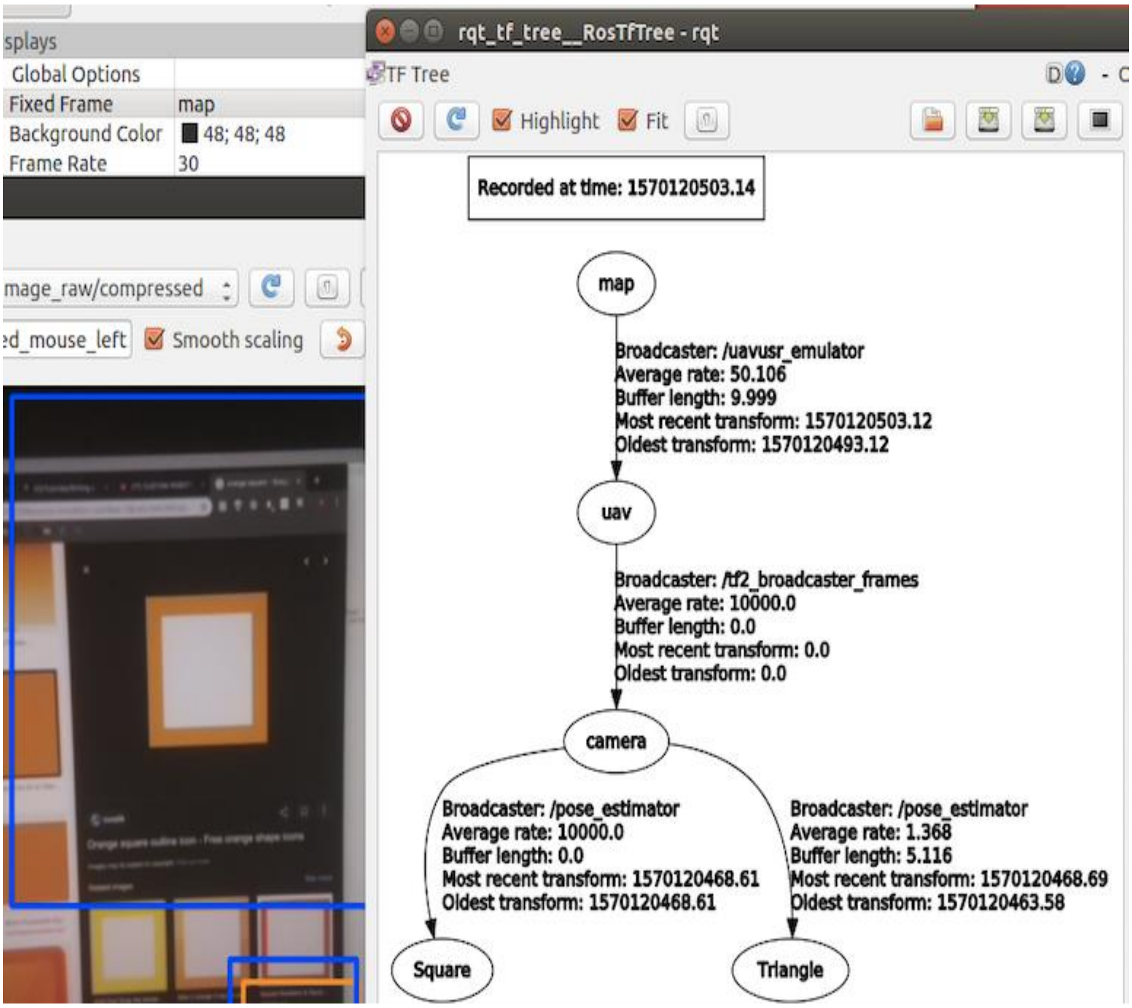


Figure 6: Target identification alert on the left-hand-side indicated by the blue square and the transform tree on the right indicating that the parent-child relationships has been established.

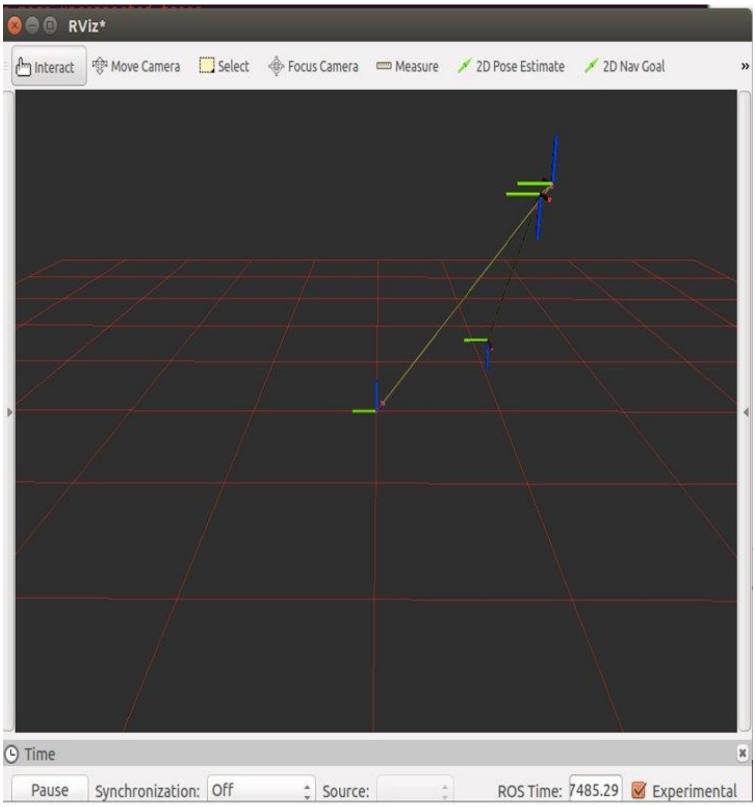


Figure 7: Target localisation of the square (as indicated by the positive identification in figure 6) relative to the camera.

The progression from this localisation works was to transform this camera relative to target location into a real-world location of the target. This step was implemented in the navigation works using a lookup transform based of the child frame names ‘Square’, ‘Triangle’ and the parent name ‘map’.

3.0 Post Test Report and Video Evidence

Subsystem YouTube Link: https://youtu.be/aw5bNTIL-h0

Test Report: Image Processing Subsystems

Test Report Summary: Pass

Subsystem Requirements: [REQ-04-01-U], [REQ-04-01-01-U]

Work Package: WP-IMP-04

Test Operator: Harry Akeroyd

4.0 Conclusions and Recommendations

The completed hardware testing proved that the Raspberry PI camera module V2 can record clear, coloured images at an appropriate frame rate. The target description script was able to be applied to these images and accuracy of detection was achieved to a high standard. The video quality provided sufficient data to create a 3D environment on the GCS. The implementation of the transform library's allowed for target localisation relative to the camera was achieved from the triggering of the detection software.

- [REQ-03-F] required that the image detection system must be able to successfully locate and correctly identify both people using an on-board computer: this was achieved as the indicated accuracy of the
- [REQ-5-F] was able to be satisfied from the perspective of the UAV. From the identification of a target, the publisher was able to be used to alert the payload script as to which injured person had been found and done so using the child frame names.
- [REQ-03-01-F] was able to be satisfied. Referring to figure 7, the patients' locations was able to be found using the image detection software, which was operating on the on-board computer.
- [REQ-04-01-01-U] required that live imagery be displayed on the GCS, this was tested and completed by displaying live imagery of the markers that will be used in the real flight, on the GCS. The camera was moved around like that of a UAV during the testing, verifying the system works.

5.0 Appendix 1: Ongoing Validation Table

Table 4: Ongoing Validation Table as per RD/3

Requirements	**Requirements Confirmed Upon Completion of this test document** Yellow = Previously validated system requirement from previous test reports Green = System requirement confirmed as of this test document
[REQ-01-F]	X
[REQ-01-01-F]	X
[REQ-01-02-F]	
[REQ-01-02-01-F]	
[REQ-01-02-02-F]	
[REQ-01-03-F]	X
[REQ-01-04-F]	
[REQ-01-04-01-U]	
[REQ-01-05-F]	
[REQ-02-F]	
[REQ-02-01-F]	
[REQ-02-01-01-F]	
[REQ-02-01-02-F]	
[REQ-02-01-03-F]	
[REQ-02-02-F]	
[REQ-03-F]	
[REQ-03-01-F]	
[REQ-03-02-F]	
[REQ-04-U]	
[REQ-04-01-U]	X
[REQ-04-01-01-U]	X
[REQ-04-01-02-U]	X
[REQ-04-01-03-U]	
[REQ-04-02-U]	
[REQ-05-F]	
[REQ-05-01-F]	
[REQ-05-02-U]	
[REQ-05-03-F]	
[REQ-05-04-F]	
[REQ-05-05-F]	
[REQ-06-F]	X
[REQ-06-01-F]	X
[REQ-06-02-F]	
[REQ-07-P]	X
[REQ-07-01-P]	X
[REQ-07-02-P]	X
[REQ-08-U]	
[REQ-08-01-U]	
[REQ-08-02-U]	