

Bachelor of Engineering
Final Thesis Report
Hexacopter project

Environmental Mapping in 2D/3D
with the use of a Multirotor Unmanned Aerial Vehicle

Thesis Supervisor: Professor Thomas Braunl and Christopher Croft



THE UNIVERSITY OF
WESTERN
AUSTRALIA

Manish Mohanty 20993295

Bachelor of Engineering
Department of Electrical, Electronic and Computer Engineering
The University of Western Australia
October 26, 2015

Environmental Mapping in 2D/3D with the use of a Multirotor Unmanned Aerial Vehicle (October 2015)

M. Mohanty, BE UWA

Abstract—In recent years, unmanned autonomously flying helicopters, airplanes, and multi-rotor copters have gained popularity due to extensive use in fields not limited to; defense, emergency response, live streaming events, mining and agriculture. To be able have a system that can fly autonomously and execute actions, simplifies human need for contribution, increasing its potential uses exponentially.

Through the use of a mini computer (Raspberry pi 2), distance sensor (Lidar) and a Picam (Raspberry branded mono-lens camera) on board the UAV system, users are able to fly autonomously, capture images and store them in the memory of the raspberry pi 2. Later, stored images are extracted and tiled together to create panoramic 2D images with the use of an offline stitching software. Users are also able to create 3D models of any large object by capturing multiple shots covering a 360 degrees' field of view with varying altitudes and model it offline by using a 3D modelling software.

The main focus of this paper is on 3D modeling, by acquiring photogrammetric data and processing it through several modelling software. This paper includes several examples of the 2D offline stitching achieved by using AutoStitch (offline software) and 3D modeling done by two offline 3D modelling software known as PhotoScan and 123 Autodesk, allowing us to compare capabilities and performance whilst highlighting restrictions. All the testing is conducted in autonomous mode, achieved by simply flicking a switch on a remote controller. Through utilizing its autonomous flying and image capturing capabilities, the Hexacopter at The University of Western Australia can perform many intelligent mapping operations.

Key Words—automation, autonomous flying, Hexacopter, 2D modelling, 3D modelling, UAV

I. INTRODUCTION

THIS document outlines equipment, software and programming expertise required to construct an autonomously flying UAV system that can capture images to create environmental maps (panoramic images) and 3-dimensional models utilizing modeling software.

The University of Western Australia has conducted research on the UAV Hexacopter to determine its capabilities and has performed many experiments and tests since 2013 to identify its use in industry. This project is conducted and supervised by Professor Thomas Braunl and Christopher Croft at The University of Western Australia (*Robotics and Automation lab, EECE*). The 2015 Hexacopter team consists of three mechatronics engineering students and one electrical &

electronic engineering student named Brett Downing, Richard Allen, Jeremy Tan and the writer respectively.

A. Unmanned Aerial Vehicle

How much is a human life worth? We ask this question every time we put ourselves in danger. Today we can avoid this question due to advancements in engineering and technology. We are able to have unmanned autonomously flying systems that can carry out many tasks in dangerous situations, decreasing human effort and risk to life.

In the past decade the UAV, or better known as the Unmanned Aerial Vehicle, has received significant attention for its use in various fields. This was only possible after major improvements in electronic equipment and devices used. Small and powerful electronic systems, brushless actuator, MEMS sensors and chips have undergone substantial size and price reductions [1]. Due to these improvements, UAV implementation is made possible and affordable [2].

B. Copter of UWA and its mapping capabilities

Currently the Hexacopter at UWA can fly through pre-assigned way points while capturing images, which can later be tiled (or stitched together) for 2D mapping by using an offline stitching software. In regards to environmental mapping in 3D, the Hexacopter can currently take off the ground autonomously and perform this operation. Before it starts 3D mapping, a user is able to input a desired radius for circular rotation or use the distance value acquired from the LIDAR (distance sensor) on board. This radius is then kept constant as images are captured at varying altitudes. Keeping a stable and constant radius plays an integral role in the construction of a 3D model. Refer to Appendix a to view the complete code written in C++ for 3D mapping.

C. Autonomous flying

Autonomous flying UAV- that is, flying exclusive of human control. Autonomous flying capabilities make drones more effective and useful in situations when human reach is difficult or impossible. Drones can be flown to capture live images, videos and other important information regarding a situation and allow prompt feedback to users gathered at a safe distance. Autonomous flying or autopilot has been a longstanding concept and is widely used today in airplanes to fly from one destination to another without seeking much attention from pilot. This decreases not only human effort, but also human error made by pilots or personnel.

D. Hexacopter flying properties

Hexacopter is a VTOL (Vertical Take-Off and Landing) copter belonging to the class of multimotored helicopters [3]. Breguet-Richet Gyroplane designed the first copter with four rotors with reports of having several successful flying [4]. The current technological advancements such as light batteries, light brushless motors & actuators and advance sensors eased the construction of such machinery [3]. These improvements also reduced, or eliminated in some cases, issues regarding instability of these aircrafts.

The Hexacopter is design in such a way that six blades are spaced equidistant from each other keeping the center of graving conserved [5]. The Hexacopter is controlled by adjusting the angular velocities of the attached rotors. In order to lift of the ground, generated thrust must be greater than the force due to gravity and when thrust is equal to force due to gravity, the multi-copter hovers at a fixed altitude. Another popular design from the same class of multimotored copter is quadcopter [6]. It has four independent equidistantly separated rotors achieving an accurate flight by producing balanced thrust from the four rotors [6].

The autonomously flying system (Hexacopter) that this project focuses on was assembled at The University Australia from scratch. There has been extensive research done over the years to investigate its capabilities in autonomous mode and also concurrently development of software programs to enhance its abilities.

II. ASSEMBLY OF THE UAV SYSTEM

A. Hexacopter at UWA

The Hexacopter system assembled at the University of Western Australia is shown below in Figure 1. There have been many modifications made since 2013, when it was first assembled by the team. Every week, the team is required to meet and work on the Hexacopter as part of their final year project. This section goes through hardware used and changes made in accordance to what was desired and required for this project to remain relevant.



Figure 1: Hexacopter system at UWA

Following hardware's are the major hardware required to have this copter flying and perform environmental mapping. Several other hardware such WIFI router, motors are not listed as they can be chosen according to the project requirements.

1) General information on some components



Figure 2: DJI F550 frame

A frame, which is made of strong fiber glass PCB material board, is 550 mm in diagonal wheelbase and 478 grams[7]. This is the main structure holding the complete copter all together. Propellers (6 on-board) are 9.4×5.0 inch and weight 13 grams [7]. The Motor mounted on board has a stator size of 23×12 mm, KV os 960 rpm/V and weight 57 grams [7]. These motors can be purchased online or retail electronic stores for about \$100 AUD. These motors draw a combined current of about 60 Amps while taking off the ground. Hexacopter of UWA weighs around 2.5 kilograms after all major changes and advancements made, which had a significant impact of the flight time performance.

2) Flight Controller



Figure 3: Pixhawk flight controller. Copied from (Pixhawk, 2015)

Flight controller system on board the flying system is Pixhawk that uses Aurdipilot (software). It can be purchased from online 3DR robotics store for \$199 US. This system is responsible for converting flight instructions such as move up, down, left right, rotate etc., received from the user remote into the signals for motors to execute maneuvers while airborne. This change was a recent upgrade from the NAZA controller, which was installed at handover.

Table 1

Important key features [8]
- 168 MHZ 252 MIPS Cortex-M4F
- 14PWM/Servo outputs
- Integrated backup system for in-flight recovery and manual override.
- Redundant power supply input and automatic failover.

3) Flight Computer



Figure 4: Raspberry Pi 2. Copied from (Next Media, 2015)

Figure 4 shown above, is an image of the flight computer that is responsible for controlling the Hexacopter in an autonomous mode. This computer (Raspberry Pi 2), has a mini SD card installed which contains all pre-installed programs required, as well as saves data as it is collected during flight. All codes written are saved on this copter and executed in autonomous mode through webserver.

4) GPS (Global Positioning system)



Figure 5: GPS system. Copied from (Hobby King, 2015)

A GPS module is used to calculate and record the current location of the hexacopter for various purposes in this project. Position recorded by this unit is used to predict the next way point while navigating around the object. It is essential to receive accurate data from this GPS unit for maintenance of a strict radius around the object allowing formation of a 3D model. Failure to do so could cause sudden change in flight path and might crash into trees, building or humans if unable to recover from it. This GPS is accurate up to ± 5 meters.

5) Remote



Figure 6: Remote. Copied from (Futaba, 2015)

Futaba remote controller is a 2.4Ghz fastest technology for two-ways telemetry communication with 8 channels [9]. This controller is used for manual control of the Hexacopter and also used to place the copter in the autonomous mode by tripping a single switch.

6) Distance Sensor



Figure 7: Lidar (Distance sensor). Copied from (Ocean Control, 2015)

LIDAR lite is small compact optical distance sensor for approximately \$114.95 AUD, which can be easily mounted on Hexacopters, robots, UAV systems [10]. This is reasonably accurate up to 40 meters and has a 3° of FOV (Field of view) and 14mm optic receiver [10]. It can record up to 500 readings/seconds, which is relatively efficient for scanning purposes [11]. The LIDAR lite is designed to be used for measuring the distance from the ground but for this project it is used to calculate the distance between the copter and object in front of the LIDAR.

Table 2

Features [11]	
Range (laser emitter)	0 – 40 m
Accuracy up to	: $\pm 0.025m$
Current absorbed	<100mA
Frequency	1 – 500Hz

7) Camera



Figure 8: PiCam. Copied from (RMS, 2015)

PiCam shown in Figure 8 mounted on the Hexacopter system is a 5-megapixel mono-focal rolling shutter camera. It is used to take still pictures and record videos during flight. In this project the PiCam has been an essential component and played an integral role, as it is used primarily to take still images for offline software to create environmental maps and 3D models of object. The PiCam is connected to the Raspberry pi 2 (Flight computer) and all images or videos taken are saved on the memory card installed in the raspberry pi.

Table 3

Specifications [12]	
Size	25×20×9
Weight	3g
Still resolution	5MP
Video modes	1080p30, 720p60 & 640×480p60/90
Fixed focus	1 m to infinity
Focal length	3.60 mm ± 0.01

Horizontal view	53.50 ± 0.13 degrees
Vertical view	41.41 ± 0.11 degrees

B. Software

1) Raspbian

Raspbian is a Linux based operating system installed on this Raspberry Pi 2. This software ultimately makes the Raspberry pi a mini-computer and creates a suitable platform to test all the code written on this project. Having this operating system allowed the task of programming to be made easier and as a result faster, this is due to Linux being regarded as one of the best operating systems to program on. Details regarding this OS can be found on the Raspberry pi foundation website [13].

2) Platform

This platform shown below at Figure 9, is similar to the one created in 2014, except there is a new LIDAR sensor addition. "Pixcoptex" is the software inclusive of all the programs written in C++ to have the Hexacopter system perform intelligent operations. In order to have all the sensor information updated in the background, threading is quite extensively utilised in the system.

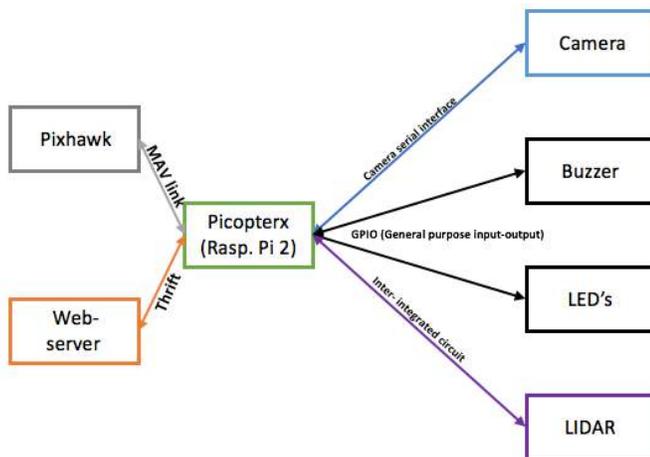


Figure 9: Raspberry pi and inter connections

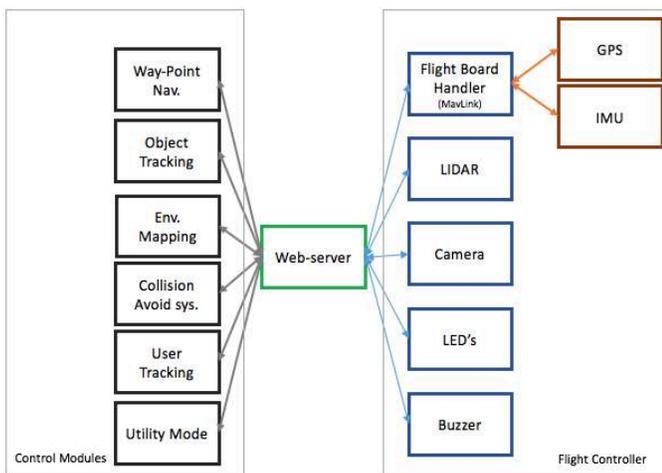


Figure 10: Autonomous software (within "picoptex")

As shown above in Figure 10, webserver is the primary connection between the flight controller and the control modules. All commands from the webserver is sent through the thrift handler, which then enables control modules such as Collision Avoidance system or Environmental mapping allowing for prompt execution. To reduce compatibility issues, CMake is used, which is compiled on Linux based operating systems and on the Raspberry pi 2.

3) AutoStitch

Autostitch is a 2D image stitching software used in this project for stitching images acquired during 2D mapping of the ground. This is a freeware commercially available for imagery stitching purposes which is downloadable from the Autostich official website. It uses the SIFT (Scale Invariant Feature Transform) and is a well known technique used for tiling images together. This software has made 2D stitching very simple. Following Table 4 shows a brief steps that the software takes to tile the images together.

Table 4

Detect and match feature points [14]
1. Find feature points
2. Match the feature points among images
3. Determine overlapping images
4. Solve hymnographies among entire images library.
5. Blend it

4) PhotoScan

This 3D modelling software is primarily used in this project to create textured 3D models from the images acquired of the mapped objects. This is a commercial product, which is available for purchase at \$79 AUD in a professional version student pack. This pack be used on both windows and OSX systems. To compute large projects, it is recommended to use 64-bit operating system computer or laptops, since 32-bit system struggles to process data [15].

Table 5: 3D constructions

Process to constructing a 3D model [16]
1. Align the captured images while matching common points and approximate position of the camera.
2. Build geometry. Based on the approximate camera position from the object, it creates 3D polygon mesh.
3. Algorithmic method SIFT is used to create 3D mesh generation.
4. After geometry is built, a textured model is created

C. Web Interface

Web interface shown in Figure 11 was designed and implemented by UWA student Alexander Mazur (*hexacopter thesis team*) in 2014 at The University of Western Australia. The interface uses thrift to communicate between the

controllers and user. It has a user-friendly environment and functions are integrated on it. This interface underwent several changes and advancements to ease 2015 project requirements.



Figure 11: Web interface at handover

Latest web interface design is shown in Figure 12. Web interface is one of the most interactive ways to communicate with the Hexacopter and get it to perform actions and mission as desired. Through the webserver, the user is able to view camera streaming, see the current location of the hexacopter on the map and able to send commands through for execution of pre-written modules such as environmental mapping for 3D modelling and 2D mapping in lawn mower pattern.

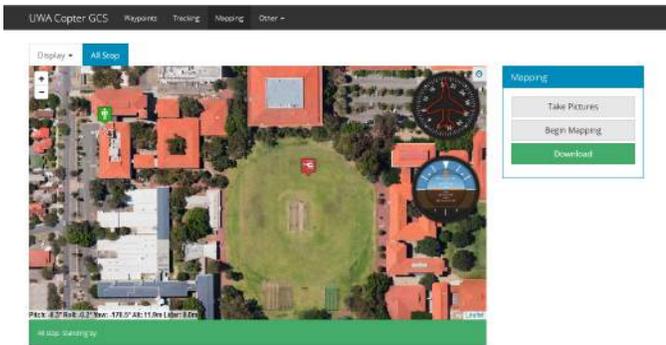


Figure 12: Web interface used to communicate with the Hexacopter

One of the major benefits of having this server run and being able to view live operations, is that the user is able to make informed decisions in a timely manner and thus can start and abort any mission at anytime during flight. This gives safety controls to the user and is able to test new written codes or program without having to worry about losing control.

III. EARLY MODIFICATIONS ANF IMPROVEMENT

A. Pixhawk replaced NAZA controller

Last year's Hexacopter system used a NAZA flight controller and after performing several tests and flying demonstrations, the team realised that the on board controller had limitations to modifications required and was also not sustainable throughout the project. The NAZA flight controller is

restricted to autonomous modifications, which limited our team from progressing with new ideas. The Flight controller was updated to 3DR Pixhawk controller, which allowed for greater benefits. Refer to appendix for in-depth further details.

B. Raspberry pi 2 replaced Raspberry B+

Raspberry pi 2 is a more powerful and advanced version of Pi B+ model. Upon upgrading to this model, we were able to gain more image processing speed with using the previously used code on the old Pi B+. Processing speed increased between 14-20 frames per seconds.

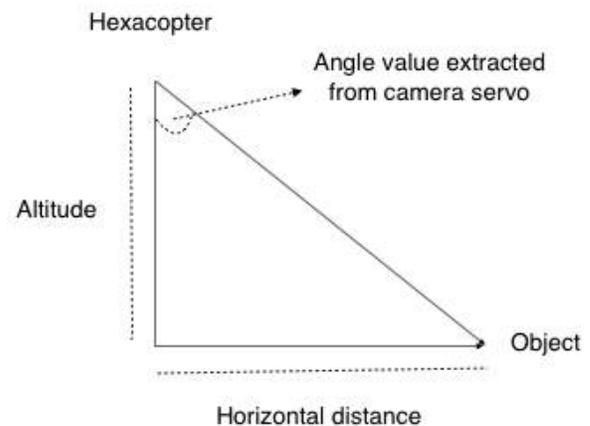
Table 6

Raspberry Controller	Resolution	FPS
Model Pi 2	160X120	23
Model Pi B+	160X121	14
Model Pi B	160X122	14.5

C. Object detection and camera tracking

Currently, the Hexacopter is tracking images by identifying pixel position of the targeted object. This method is failing to make critical adjustments due to the changing horizontal distance upon changing attitude.

A new method is being implemented to eliminate/reduce this error by calculating horizontal distances using the small angle approximation. Pixhawk has a reliable barometer and from research, it has been shown that it gives accurate height measurements on a larger range of altitudes. Testing and calibration will be performed and appropriate adjustments will be done to acquire reliable results. Figure 12 shows the actual representation for the method.



Let h = Altitude

h = Altitude data extracted

θ = Angle value extracted from the camera servo

d = Horizontal distance

$$d = h \times \tan^{-1}(\theta)$$

This method is believed to reduce error and give more consistent results.

D. User interface

Web-interface was design in 2014 at UWA to ease communication between user and the copter. It's an extraordinary design but flaws were found while using it. Object tracking function was malfunctioning and was rectified by Jeremy Tan (*team member, Hexacopter*) by rewriting it.

Another issue was the inconvenience of not having many options to configure systems on board. Alterations were made to make functioning more convenient for the user. More testing and improvements are to be made for user's convenience and future modifications as well as projects.

E. Fail-safe mode fixed

At the start of the project, the team identified that the fail-safe feature of the copter wasn't operating. Configuration on the transmitter was done incorrectly and upon performing diagnostics and testing, team was able to resolve the issue.

F. Landing gear replaced

Unsealed MDF landing gears were installed to have the Hexacopter mounted in order for the battery to be installed underneath the frame. Being made out of cardboard material, they were prone to liquid damage and wear & tear over time. It also increased torque upon hard landing in unfavorable weather conditions. Landing gears were damaged several times during testing leading to a decision to replace it with plastic landing gear (compatible with DJI F550 frame).



Figure 13: Old landing gear (left) and new landing gear (right)

These new landing gear shown above in Figure 13 (right) have proven to be quite reliable after surviving several harsh landing.

IV. FLIGHT PLANNING

A. For 2D Mapping

This is achieved by acquiring still images of the ground while moving in a certain direction and orientation. There are many mapping patterns used such as but not limited to; lawn mover pattern, spiral pattern, etc. This project has focused on using lawn mover pattern shown in Figure 14, as it the the most efficient and easiest way to cover ground. Pre-determine path is set by the user on the web interface by setting waypoint on the viewable map on the interface.



Figure 14: 2D mapping using lawn mower pattern

These images are taken and stored in the memory card inside the raspberry pi 2. Later, these images as shown in Figure 17 can be extracted and processed further for stitching purposes. While performing this operation, few points needs to be considered, which are listed in Table 7 for the user.

Table 7

Avoid
- Shadow on the field
- Shiny surface
- Very high altitude to eliminate any possibilities of poor pixilated images

B. For 3D Mapping

3-Dimensional mapping is slightly more challenging than 2-Dimensional mapping, as it required to hold a fix radius while rotating around the object. Hexacopter is equipped with a single mono lens camera and therefore the only way 3D modelling can only be achieved is by taking multiple shots of the object covering 360 degrees' field of view and model it is using an offline software. Distance sensor on the flying system (Lidar) calculates the distance from the POI or the object. This radius is then maintained through out the rotation. Figure 15 shows the process chain to to enable the Hexacopter to take images at various angles and altitude.

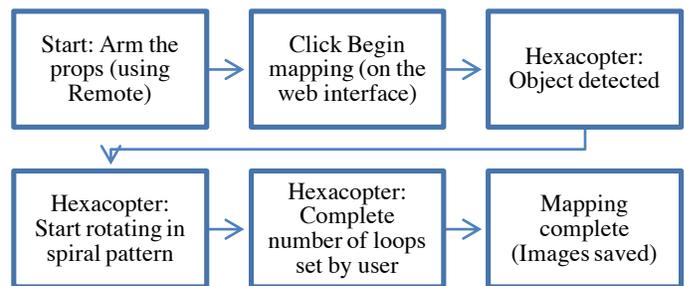


Figure 15: 3D mapping flow chart

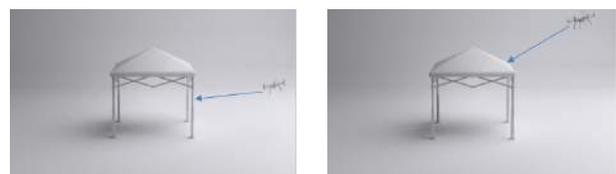


Figure 16: 3D mapping around the desired object.

As mentioned before while rotating around the object, it is essential to maintain a fix radius in other words, the blue line shown in Figure 16 must be kept constant through out the rotation. After completing one full rotation around an object, the Hexacopter rises in altitude by 1 meter and continues taking images as shown above in Figure 16. LIDAR on board, is fed with values corresponding to live radius and makes sure it maintains a safe distance from the object. Data collect from the LIDAR is also used to keep the radius fixed by subtracting the errors caused by the GPS, which is accurate up to ± 5 meters. Following points noted in Table 8 needs to be avoided while 3D mapping.

Table 8

Mainly avoid [16]
<ul style="list-style-type: none"> - Transparent and shiny bodies - Crop or transform images - Not textured and flat bodies or POI

V. MAPPING & IMAGE PROCESSING

A. 2D Stitching (offline)

After extracting the images as shown in Figure 17 from the memory card of the Raspberry pi, we are able to upload it on to Autostitch, which then creates a panoramic image (stitched image). It would be a good practice to delete any images that are majorly overlapping. This would decrease the processing time.

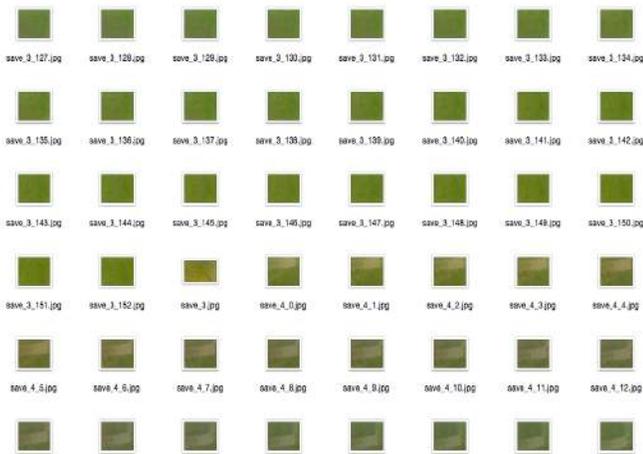


Figure 17: Collected 2D images of James Oval (The University of Western Australia)

These images are stitched using RANSAC (Random Sample Consensus) algorithm, which detects all the true matches, deletes non-matching points and compute homography [17].

Table 9

Homography estimation algorithm (RANSAC) [17]
<ul style="list-style-type: none"> - Choose 4 random images - Calculate homography H - Compute $SSD(p'_i, H_{p_i} (< \epsilon))$- inliers. - Hold the greatest set on inliers. - Re-calculate least-squares H using all inliers.

This is then verified by a probabilistic model to confirm the image match.

Table 10

Probabilistic model [17]
<ul style="list-style-type: none"> - Match probability of RANSAC inliers/outliers was created by true/false matching. - Images matched if $p(m = 1 f^{1:n_f}) > p_{min}$, where $n_f = \#$ (overlap area), $n_i = \#$ (inliers), $m = r.$, $m=1$ and $f^i \in \{0,1\}$. - Compute with the use of Bayes' rules, and the image is matched if $n_i > 5.9 + 0.22n_f$ is satisfied.

This process could have been achieved online (Stitching images live while airborne) but due to processing limitations it is was not implemented.

B. 3D Modelling (offline)

1) How to begin Environmental mapping through web server?

3D modelling is done under autonomous conditions. Basically, the Hexacopter is placed directly in front the object (POI) that is selected for 3D modeling. The user then arm's the props by using hand held controller and click "begin mapping" as shown in the Figure 18 below.

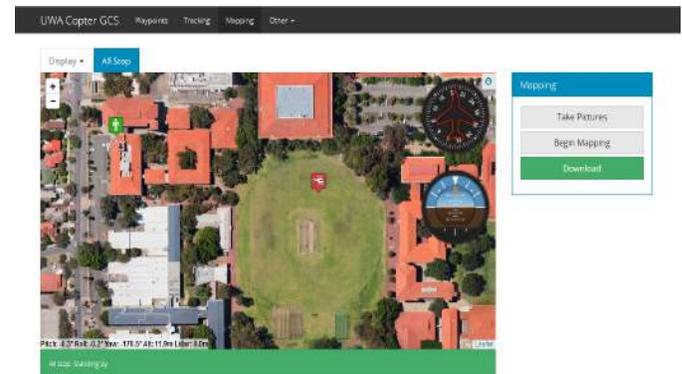


Figure 18: Environmental mapping interface

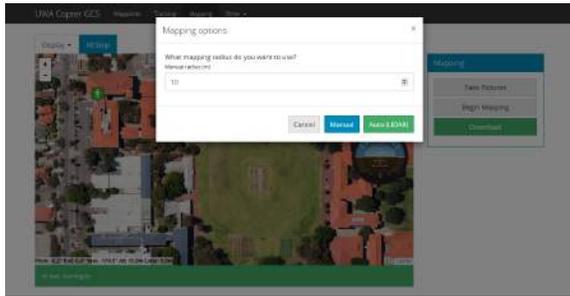


Figure 19: Env. Mapping requesting radius input

Before it takes off the ground, the user is requested to enter a desired or approximately calculated radius as shown above in Figure 19 that would be used to rotate around the object. Otherwise, the user is also able to point the LIDAR lite at the object to see the actual distance from it, which could then be used by clicking the Auto (LIDAR) icon shown in Figure 19 to lock the radius.

The Hexacopter will take off the ground to an altitude of 5 meters and start by rotating around the object whilst taking pictures. After every rotation is complete, it adds a meter to its current altitude. Once it has completed user desired rotations around the object, all the images can be viewed by clicking on the button **download** as shown in Figure 20 below.

Index of /pics/				
Name	Last Modified	Size	Type	Access
img_2018-05-01-11-20-01.jpg	2018-05-01 11:20:01	11.0K	image/jpeg	
img_2018-05-01-11-20-02.jpg	2018-05-01 11:20:02	11.0K	image/jpeg	
img_2018-05-01-11-20-03.jpg	2018-05-01 11:20:03	11.0K	image/jpeg	
img_2018-05-01-11-20-04.jpg	2018-05-01 11:20:04	11.0K	image/jpeg	
img_2018-05-01-11-20-05.jpg	2018-05-01 11:20:05	11.0K	image/jpeg	
img_2018-05-01-11-20-06.jpg	2018-05-01 11:20:06	11.0K	image/jpeg	
img_2018-05-01-11-20-07.jpg	2018-05-01 11:20:07	11.0K	image/jpeg	
img_2018-05-01-11-20-08.jpg	2018-05-01 11:20:08	11.0K	image/jpeg	
img_2018-05-01-11-20-09.jpg	2018-05-01 11:20:09	11.0K	image/jpeg	
img_2018-05-01-11-20-10.jpg	2018-05-01 11:20:10	11.0K	image/jpeg	
img_2018-05-01-11-20-11.jpg	2018-05-01 11:20:11	11.0K	image/jpeg	
img_2018-05-01-11-20-12.jpg	2018-05-01 11:20:12	11.0K	image/jpeg	
img_2018-05-01-11-20-13.jpg	2018-05-01 11:20:13	11.0K	image/jpeg	
img_2018-05-01-11-20-14.jpg	2018-05-01 11:20:14	11.0K	image/jpeg	
img_2018-05-01-11-20-15.jpg	2018-05-01 11:20:15	11.0K	image/jpeg	
img_2018-05-01-11-20-16.jpg	2018-05-01 11:20:16	11.0K	image/jpeg	
img_2018-05-01-11-20-17.jpg	2018-05-01 11:20:17	11.0K	image/jpeg	
img_2018-05-01-11-20-18.jpg	2018-05-01 11:20:18	11.0K	image/jpeg	
img_2018-05-01-11-20-19.jpg	2018-05-01 11:20:19	11.0K	image/jpeg	
img_2018-05-01-11-20-20.jpg	2018-05-01 11:20:20	11.0K	image/jpeg	
img_2018-05-01-11-20-21.jpg	2018-05-01 11:20:21	11.0K	image/jpeg	
img_2018-05-01-11-20-22.jpg	2018-05-01 11:20:22	11.0K	image/jpeg	
img_2018-05-01-11-20-23.jpg	2018-05-01 11:20:23	11.0K	image/jpeg	
img_2018-05-01-11-20-24.jpg	2018-05-01 11:20:24	11.0K	image/jpeg	
img_2018-05-01-11-20-25.jpg	2018-05-01 11:20:25	11.0K	image/jpeg	
img_2018-05-01-11-20-26.jpg	2018-05-01 11:20:26	11.0K	image/jpeg	
img_2018-05-01-11-20-27.jpg	2018-05-01 11:20:27	11.0K	image/jpeg	
img_2018-05-01-11-20-28.jpg	2018-05-01 11:20:28	11.0K	image/jpeg	
img_2018-05-01-11-20-29.jpg	2018-05-01 11:20:29	11.0K	image/jpeg	
img_2018-05-01-11-20-30.jpg	2018-05-01 11:20:30	11.0K	image/jpeg	
img_2018-05-01-11-20-31.jpg	2018-05-01 11:20:31	11.0K	image/jpeg	
img_2018-05-01-11-20-32.jpg	2018-05-01 11:20:32	11.0K	image/jpeg	
img_2018-05-01-11-20-33.jpg	2018-05-01 11:20:33	11.0K	image/jpeg	
img_2018-05-01-11-20-34.jpg	2018-05-01 11:20:34	11.0K	image/jpeg	
img_2018-05-01-11-20-35.jpg	2018-05-01 11:20:35	11.0K	image/jpeg	
img_2018-05-01-11-20-36.jpg	2018-05-01 11:20:36	11.0K	image/jpeg	
img_2018-05-01-11-20-37.jpg	2018-05-01 11:20:37	11.0K	image/jpeg	
img_2018-05-01-11-20-38.jpg	2018-05-01 11:20:38	11.0K	image/jpeg	
img_2018-05-01-11-20-39.jpg	2018-05-01 11:20:39	11.0K	image/jpeg	
img_2018-05-01-11-20-40.jpg	2018-05-01 11:20:40	11.0K	image/jpeg	
img_2018-05-01-11-20-41.jpg	2018-05-01 11:20:41	11.0K	image/jpeg	
img_2018-05-01-11-20-42.jpg	2018-05-01 11:20:42	11.0K	image/jpeg	
img_2018-05-01-11-20-43.jpg	2018-05-01 11:20:43	11.0K	image/jpeg	
img_2018-05-01-11-20-44.jpg	2018-05-01 11:20:44	11.0K	image/jpeg	
img_2018-05-01-11-20-45.jpg	2018-05-01 11:20:45	11.0K	image/jpeg	
img_2018-05-01-11-20-46.jpg	2018-05-01 11:20:46	11.0K	image/jpeg	
img_2018-05-01-11-20-47.jpg	2018-05-01 11:20:47	11.0K	image/jpeg	
img_2018-05-01-11-20-48.jpg	2018-05-01 11:20:48	11.0K	image/jpeg	
img_2018-05-01-11-20-49.jpg	2018-05-01 11:20:49	11.0K	image/jpeg	
img_2018-05-01-11-20-50.jpg	2018-05-01 11:20:50	11.0K	image/jpeg	

Figure 20: Images stored from Env. Mapping on Raspberry pi 2

These images can be downloaded for further processing. It is saved in the Home/pics folder.

2) 3D modelling done using PhotoScan

After extracting images from the memory card of the Raspberry Pi 2, these can be uploaded onto PhotoScan for modelling.

Table 11

Steps to create a 3D model [16]

- Remove unnecessary images from the chunk.
- Click Workflow/Align Photos
- Click Workflow/ Build Dense Cloud
- Click Workflow/ Mesh (for viewing the mesh)
- Finally click Workflow/ Build Texture.

After the texture is build, user is able to view the 3D model on the screen.

C. 3D printing

During the whole project, many 3D models were created on the computer during which writer made decision to print a 3D model using one of the 3D printers available at the University of Western Australia. It was important to see as part of environmental mapping project, if accurate models of object were achievable and if they are able to print.

From one of the models created, a '.stl' file was generated on PhotoScan, which was then used in a 3D printer to print an actual 3D model (solid model). The materials used by the printer to create the model was ABS plastic, this is a fairly common material used worldwide in many 3D printers.

Model shown in Figure 21 (Left) is later discussed in the results sections.

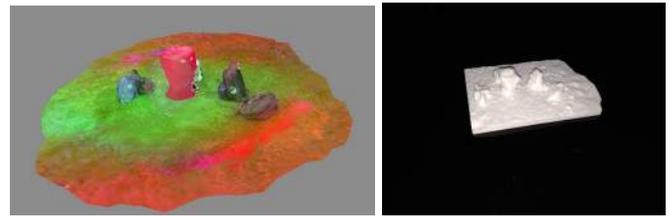


Figure 21: Model used to print (left) and printed 3D of left image (right)

VI. RESULTS

A. Results from 2D stitching

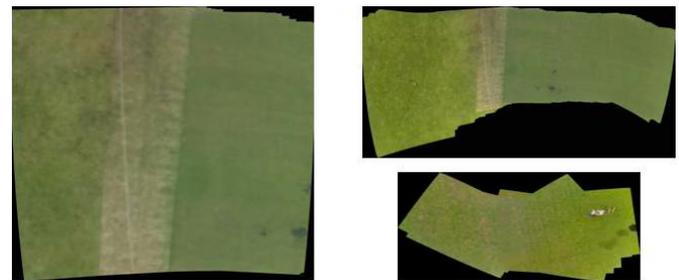


Figure 22: Stitched image of James Oval (UWA)

The figure shown above is an output of several images stitched together with the use of the AutoStitch Software. Images that overlapped with each other or did not present with any matched points were rejected promptly by the software. This is a clear example of easy 2D stitching in which results can be obtained within several minutes of uploading photos to the software.

B. Several 3D modelling test results

1) Test 1

The following image is of tent with three red bins placed under the tent for 3D mapping. Bins were deliberately placed underneath to be able to see how 3D modeling software performs when it encounters with a symmetrical environment scenario.

Images below are the screenshots taken from the actual 3D model created by the PhotoScan.

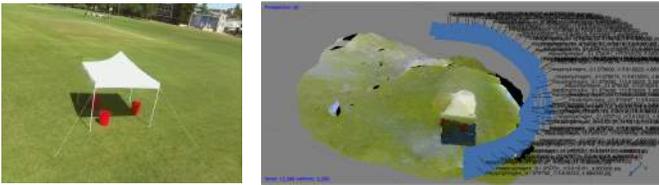


Figure 23: Test 1 actual setup (left) and image locations (right)

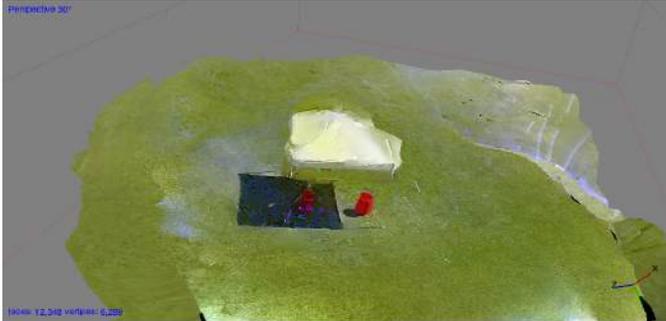


Figure 24: Test 1 model screenshot II

This model contains many broken images and uneven ground. This is due to bad 3D stitching, which was caused by poor image quality and received insufficient number of images as many images saved got rejected due to not being in the frame as shown in Figure 23 (right). It was also realised that most images that passed were taken at a high altitude (camera facing down), which caused the model to look good from the top but broken from middle. Table 12 presents the data from collected from the PhotoScan software.

Table 12: Test 1 data

Modelling data	
Cameras	66/142 aligned
Tie Points	12,292 points
Dense Cloud	1,338,156 points
3D Model	12,348 faces

2) Test 2

Using PhotoScan

This test was performed after it was realised that it is essential to take images from lower altitude. Bags and bins were used to see the performance/detail modeling of the PhotoScan and view it ability add details to its models.

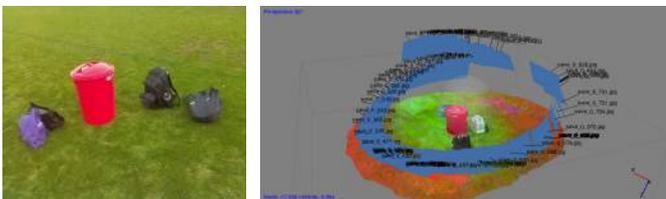


Figure 25: Test 2 model actual setup (left) and images locations (right), PhotoScan(right).

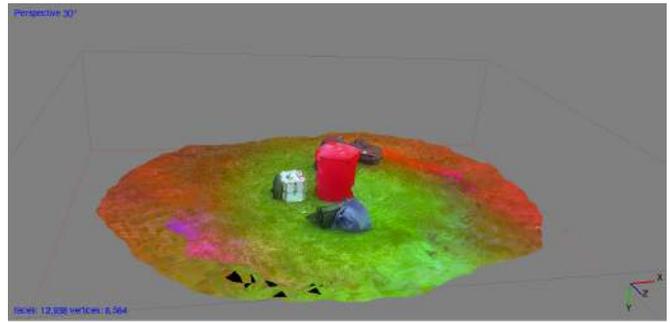


Figure 26: Test 2 3D model (PhotoScan)

Table 13: Test 2 Data

Modelling data	
Cameras	90/91 aligned
Tie Points	15,011 points
Dense Cloud	2,056,693 points
3D Model	12,938 faces

This was one of the better models created as a result of capturing images at lower altitude and smaller radius from the object. From all the pictures acquired 90/91 (98.9%) images aligned together, which was one of the best results ever achieved during testing period. This is the model used in 3D printer to create a 3D model as shown in Figure 21 (right).

Using 123 Autodesk

Another 3D modelling software known as 123 Autodesk was used to create a 3D model.



Figure 27: Test 2 model (images locations) 123 Autodesk (Left), 3D wired model (Right)

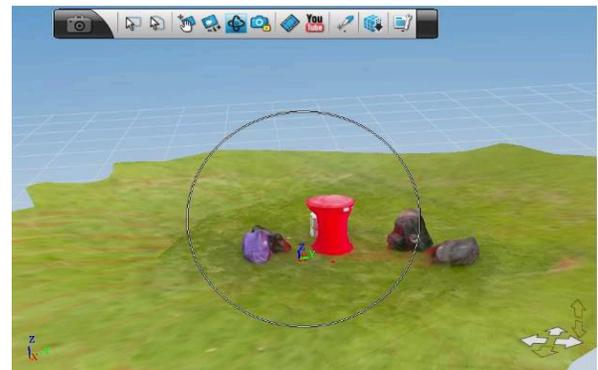


Figure 28: Test 2 3D model (123 Autodesk)

As shown in Figure 28, a smooth 3D model is created after approximately one hours of processing time.

C. Comparison

In this session, two 3D models were generated by two different software for their accuracy, details visible on the objects along with exposure.

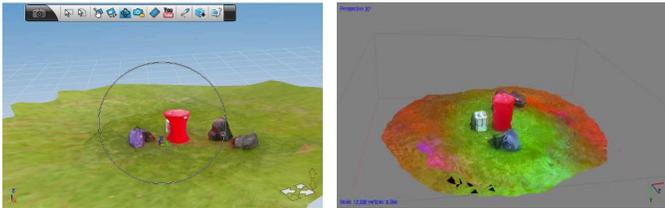


Figure 29: 3D model created by 123 Autodesk (left) and 3D model created by PhotoScan (right)

Both the software was presented with the same number of images used for modeling. It can clearly be identified that the model created by Photoscan is lot more detailed than the one created by 123 Autodesk. Fine lines and texture is visible on the PhotoScan model whereas a smooth texture is found on 123 Autodesk. Model created by the Autodesk shows better greenery region than the one created by PhotoScan, which is better for printing on paper purposes.

Point to note that 123 Autodesk software was a free ware used for this modelling purpose.

D. Final Results

After many tests and failures, following results are obtained from the final environmental mapping session conducted at UWA. Many aspects such as weather conditions; locations were considered before mapping the tent. It was important to consider all the factors that had affected previous testing sessions to obtain the best results possible. Image shown in Figure 31 (left) and (right) shows the image's locations in the model. It is clearly evident that most of the images that were captured during the test session were used in the model as there are no empty regions shown in Figure 23.



Figure 30: Tent used for 3D modelling test

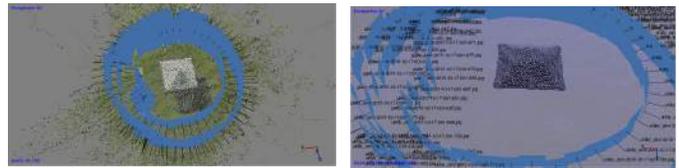


Figure 31: Dense cloud (left) and solid frame with all the images aligned (right)

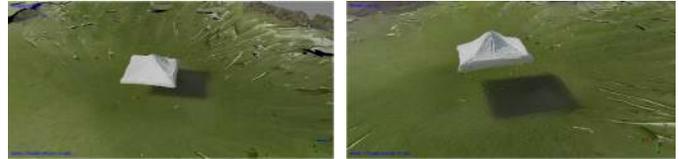


Figure 32: 3D textured model top view (left and right)



Figure 33: 3D textured model including environment

Table 14: Final results data

Modelling data	
Cameras	224/226 aligned
Tie Points	29,159 points
Dense Cloud	94,159 points
3D Model	179,999 faces

Images acquired during the final testing were much more accurate compared to any other testing sessions conducted. It was due to the perfect weather and lighting conditions. During the creation of the 3D model, PhotoScan **aligned 99.12%** of the images together and found 29,159 points, which was highest points ever recorded of all the testing done in past.

All the images acquired from this testing session were used for 3D modeling in other words non of the images taken were rejected due to not meeting minimum standards. Also during the test, many images were taken from the same altitude to have maximum over laps which can be seen in the Figure 31.

VII. ISSUES FACED

A. Wind turbulence

Framing

It is sometimes quite challenging to capture the object in the camera frame. While rotating around the object, it sometimes missed the POI or the object in which case those images are rejected.

Pitch and roll stability

While compensating for the wind drifts, it is difficult for the Hexacopter to stay focused on the object while rotating around it. There is also a delay between the information regarding the

error caused due to the strong winds and correction sent through the feedback loop to the flight controller to correct it.

Gimbal compensations

Gimbal on board is also responsible to keep the object in the frame but due to delay in receiving the correction signal it sometimes missed the object.



Figure 34: Rejected images (Object not captured not within the frame)

B. Poor image quality

Vibrations caused due to the high speed revolving motors has caused significant decrease in the quality of the pictures taken. These vibrations can not be eliminated from the system and having a rolling shutter camera makes it more challenging to capture good quality images.



Figure 35: Stretched image

C. Camera quality

As the PiCam is a 5-megapixel inexpensive rolling shutter camera, it limits image quality significantly. Its is a single lens mono-focal camera doesn't let to focus on the object but image quality within the reasonable range is satisfactory. This limitation has caused difficulties while creating a 3D model. Rolling shutter is a method of capturing images by scanning across the either horizontally or vertically instead of taking a snap shot of the complete scene, given the camera is placed or held still. This procedure caused distortion in the images while rotating around the object and taking pictures.

D. GPS inaccuracies

There are many reasons for the GPS to output an inaccurate position. It can can simply caused to the environment such as buildings, trees or it could be ionosphere or troposphere as the

signals gets refracted. Due to this inaccuracy, it is quite difficulty to hold a constant radius from the POI or object being mapped as indicated in the image below. A distance sensor was installed to maintain the radius while rotating around it.

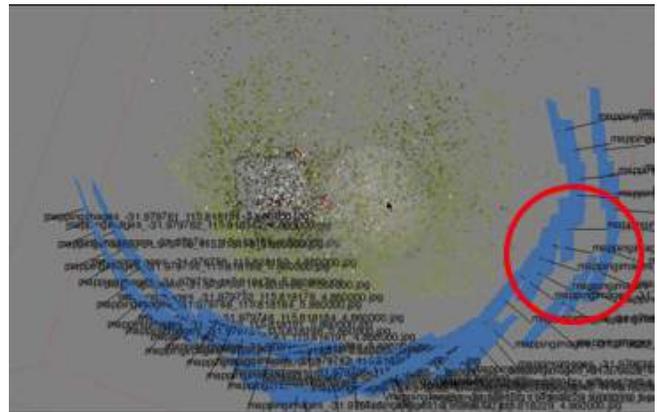


Figure 36: GPS inaccuracies

As visible from the figure above, two images at the same bearing at different rotation has failed to maintain fix radius. This creates holes, bents and broken parts in the 3D model.

E. Wireless connection unsteadiness

The wireless dongle connected to the Raspberry Pi 2 is quite unstable causing the image streaming quality (frame rate) to drop significantly. Perhaps, recent modification to the Pi protective shell stabilised the physical connection between the dongle and the pi on board. This design improvement has increased connection stability and has relatively solved the issue.

F. Gimbal issues

Camera gimbal plays a majoring role in image tracking functionality. Not having control over pan (yaw) of the camera caused difficulties in having it to face (point at the object) the targeted object. The user had to manually hover it to a location with clear view of the target. It was difficult to lock the position in windy conditions.

G. Battery limitation

Currently, 3 batteries are capable of taking the copter flying. They are limited to ~ 15 minutes of flying provided no aggressive maneuvers are made during the flight. During testing, it was evident that multiple landing and take offs using the same battery gave about approximately 10 minutes of flying.

H. Drag

During testing, it is evident that the copter sometimes banks in one direction causing it to drift. It is understood that it is due to the non-uniform frame design and further testing will be conducted to pin point the issue or design will be improved to have the center of gravity satisfied.

VIII. CONCLUSION

This study shows how easily anyone can acquire images and creates 2D maps and 3D models of any object or POI. It has also explored on ways to capture better images quality from mapping around the area or objects and factors that affect the images acquisition process.

System is able now able to take images from ground using the way points placed by the user and create 2D maps using offline stitching software. It is also now able to take off the ground autonomously and use either the radius set by user or use the distance sensor to estimate radius to rotate around the object to take images. These images are then process in 3D modeling software to create 3 dimension models.

While great 3D models were created during this project, there is still room for improvement such as the choice of camera used in the project. Increasing the camera quality would increase the details in every pictures taken, which makes it easier for modeling software to create smoother models with more in depth details in it. It is also shown how weather conditions affect 3D mapping and decreases model quality as many images get rejected due to its poor quality.

IX. APPLICATIONS

There are many applications to this advancement to make lives easier such as for people working in the mining industry; they are able to produce lives maps and 3D models of pits or under constructions regularly. They will no longer have to travel and take images on portable or carry able cameras.

Maps and models developed can also be added to google maps. This way people can volunteer to update maps and no longer have to depend on unreliable and outdated maps, which has not updated for a extended period of time. With this new facility, people will no longer just be able to see the roads on google street view, they will able to view buildings, statues or significant monuments in 3 dimension.

Models created from the software are also printable using 3D printer. This would be an easier way to print small model of people or buildings, by just flying the UAV system around it.

X. FUTURE WORK

This project can be further extended to creating usable maps, which can be added to the webserver currently being used. This is can be achieved by embedding coordinates to maps created and then adding it to the webserver of the Hexacopter. This way user is able to have most up to date map for use. This would be very useful for remote area or mining industry, where lives updating maps would be quite useful.

The Hexacopter team of 2015 tried to combine all the four projects together, which would have made the system detect the interested object while flying around randomly and then map the object whilst making sure it does not collide or crash with anything in the environment. Process is shown in Figure 37 below.

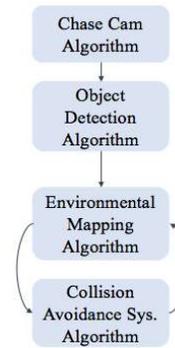


Figure 37: Combined all the theses of 2015

Next year team should also look into adding another Raspberry Pi to the flying system. This improvement would give them more processing power along with an ability to process 2D images on board to create panoramic images. They could also look into sending live images from the Raspberry Pi to the ground computer and construct 3D models simultaneously. This would reduce the overall time for the while procedure.

XI. REFERENCES

- [1] **Radek Baránek and František Šolc, "Modelling and Control of a Hexa-copter," 2012.**
- [2] Alexander Mazur, *Autonomous operation and control of a Multirotor Unmanned Aerial Vehicle through 4G LTE using onboard GPS and image processing*. Perth, WA, Australia: The University of Western Australia.
- [3] Baranek Radek and Solc Frantisek, "Modelling and Control of a Hexa-copter," vol. 978-1-4577-1868-7/12, October 2015.
- [4] Warren R. Young, *The Helicopters The Epic of Flight.*: Time-life Books.
- [5] Vincenzo Cani, "Image Stitching for UAV remote sensing application ," Computer Engineering , School of Castelldefels of Universitat Politècnica de Catalunya , Thesis.
- [6] V. Artale , C.L.R. Milazzo , and A. Ricciardello , "Mathematical Modeling of Hexacopter," vol. 7, 2013.
- [7] (2015, October) DJI Innovation. [Online]. http://download.dji-innovations.com/downloads/flamewheel/en/F550_User_Manual_v2.0_en.pdf
- [8] 3DR Robotics. [Online]. <https://pixhawk.org/modules/pixhawk>
- [9] (2015 , October) Hobby King. [Online]. http://www.hobbyking.com/hobbyking/store/_46049_futaba_14sg_14_channel_2_4ghz_computer_radio_system_2_4ghz_fasstest_set_mode_1_.html
- [10] (2015, October) LIDAR-Lite V2. [Online]. <https://www.sparkfun.com/products/13680>
- [11] Spark Fun. [Online]. <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/lidarlite2DS.pdf>
- [12] PiCam. [Online]. <https://www.raspberrypi.org/documentation/hardware/camera.md>
- [13] Downloads. [Online]. <https://www.raspberrypi.org/downloads/>
- [14] University of Wisconsin-Madison. [Online]. <http://pages.cs.wisc.edu/~dyer/cs534-spring10/slides/09-mosaics2.pdf>
- [15] F. Neitzel and J. Klonowski , "MOBILE 3D MAPPING WITH A LOW-COST UAV SYSTEM," 2011, Conference on Unmanned Aerial Vehicle in Geomatics, Zurich, Switzerland.
- [16] Agisoft. [Online]. http://downloads.agisoft.ru/pdf/photoscan-pro_0_9_0_en.pdf
- [17] AutoStitch. [Online]. <http://pages.cs.wisc.edu/~dyer/cs534-spring10/slides/09-mosaics2.pdf>
- [18] Pixhawk. (2015, October) Pixhawk. [Online]. https://pixhawk.org/_media/modules/pixhawk-logo-view.jpg?w=400&tok=331821
- [19] (2015, October) Next Media. [Online]. <http://i.nextmedia.com.au/News/Raspberry%20Pi%2020Model%20B.jpg>
- [20] (2015, October) Hobby King. [Online]. <http://www.hobbyking.com/hobbyking/store/catalog/76928.jpg>
- [21] (2015, October) Futaba. [Online]. <http://www.futabarc.com/systems/futk9410-14sg/futk9410-main-1g.jpg>
- [22] (2015, October) Ocean Control. [Online]. <https://oceancontrols.com.au/images/D/13680-04.jpg>
- [23] (2015, October) RMS. [Online]. http://regmedia.co.uk/2013/10/08/picam_big.jpg
- [24] (2015, October) Hobby King. [Online]. <http://hobbyking.com/hobbyking/store/catalog/17296.jpg>
- [25] Radek Baránek and František Šolc , *Modelling and Control of a Hexa-copter*. Brno , Czech Republic : IEEE , 2012, vol. 978-1-4577-1868-7/12.
- [26] Simon T.Y. Suen , Y. Edmund Lam, , and K.Y. Kenneth Wong , "Photographic stitching with optimized object and color matching based on image derivatives," vol. 15, pp. 7689 – 7696 , 2007.
- [27] O. Targhagh, "Hexacopter navigation ," *Autonomous Navigation of Unmanned Aerial Vehicles* , 2014.
- [28] NAZA, "User Manual," DJI Innovations , 2012.
- [29] 3DR. [Online]. <https://store.3drobotics.com/products/3dr-pixhawk>
- [30] Futaba, "Instruction Manual (14 SG)," Manual.
- [31] Element14. [Online]. <http://au.element14.com/raspberry-pi-accessories>
- [32] Alexander Mazur, "Autonomous operation and control of a Multirotor Unmanned Aerial Vehicle through 4G LTE using onboard GPS and image processing ," Faculty of Engineering, Computing and Mathematics , The University of Western Australia , Thesis 2014.
- [33] Raspberry Pi Foundation. Camera. [Online]. <https://www.raspberrypi.org/documentation/hardware/camera.md>
- [34] Oscar Soler Cubero, "Image Stitching," Telecommunications Engineering , University of Limerick, Thesis.
- [35] Richard Szeliski, "Image Alignment and Stitching ," 2005.
- [36] Ting Hsu1 Chiou and A. Beuker Rob , "In Visual Communications and Image Processing ," *Multiresolution Feature-Based Image Registration* , June 2000.
- [37] Krishna Paudel, "Stitching of X-ray Images ," in *Stitching of X-ray Images.*: UPPSALA UNIVERSITET, 2012.
- [38] Camera. [Online]. <https://www.raspberrypi.org/documentation/hardware/camera.md>
- [39] Chris Harris and Mike Stephens, "A Combined Corner and Edge Detector," 1988.
- [40] MATTHEW BROWN and DAVID G. LOWE , "Automatic Panoramic Image Stitching using Invariant Features," December 2006, Department of Computer Science, University of British Columbia, Vancouver, Canada.
- [41] Herbert Bay , Tinne Tuytelaars, and Luc Van Gool, "SURF: Speeded Up Robust Features".
- [42] Herbert Bay , Andreas Ess, Tinne Tuytelaars, and Luc Van Gool , "Speeded-Up Robust Features (SURF)," Sept 2008.
- [43] Infrared Proximity Sensor - Sharp GP2Y0A21YK. [Online]. <https://www.sparkfun.com/products/242>
- [44] PX4 Hardware. [Online]. <https://pixhawk.org/modules/start>
- [45] NAZA-M. [Online]. <http://www.dji.com/product/naza-m>
- [46] Futaba-rc. [Online]. <http://www.futaba-rc.com>
- [47] Raspberrypi. [Online]. <https://www.raspberrypi.org>
- [48] Raspberry Pi Camera Board. [Online]. <http://ledsandchips.com/store/233>
- [49] Warren R. Young, "The Helicopters The Epic of Flight ," October 2015.

XII. APPENDIX

A. Environmental Mapping Code

```

/**
 * 3D modelling with spiral pattern
 * Detect an object and set initial radius.
 * Set an initial radius.
 */

/*
 Final functions structure:

-->>> After pressing the env mapping.
Actions
(1) Lift of the GND (3 meters)
(2) Rotate to locate object (colour set by user)
(3) When object found, stop rotating and move towards the object (at a slow speed).
(4) Stop when the object is fitting the whole screen
(5) Rotate around the POI whilst taking pic and increasng alt. (Height increment after
completing every loop.
(6) If the object lost return to action (2)

End after completing the loops set by user
*/

#include "common.h"
#include "env_mapping.h"
#include "utility.h"
#define _USE_MATH_DEFINES
#include <cmath>

using namespace picopter;
using namespace picopter::navigation;

EnvironmentalMapping::EnvironmentalMapping(Options *opts, int radius)
: m_finished{false}
, m_radius(radius)
{
}

EnvironmentalMapping::EnvironmentalMapping(int radius)
: EnvironmentalMapping(NULL, radius) {}

EnvironmentalMapping::~EnvironmentalMapping() {
}

void EnvironmentalMapping::GotoLocation(FlightController *fc, Coord3D l, Coord3D roi, bool
relative_alt) {
    GPSData d;
    double wp_distance, wp_alt_delta;
    fc->fb->SetGuidedWaypoint(0, 3, 0, l, relative_alt);
    fc->fb->SetRegionOfInterest(roi);

    do {
        if (fc->lidar) {
            float v = fc->lidar->GetLatest() / 100;
            if (v > 0.1 && v < 2) {
                //STAHF
                fc->fb->Stop();
                break;
            }
        }
        fc->gps->GetLatest(&d);
        wp_distance = CoordDistance(d.fix, l);

```

```

        wp_alt_delta = l.alt != 0 ?
            std::fabs((d.fix.alt-d.fix.groundalt)-l.alt) : 0;
        fc->Sleep(100);
    } while ((wp_distance > 2 || wp_alt_delta > 0.2) && !fc->CheckForStop());
}

/*
void EnvironmentalMapping::SearchingMapping(FlightController *fc) {
    //Rotate to search
    while (!fc->CheckForStop()) {
        fc->fb->SetYaw(5, true);
        fc->Sleep(100);

        fc->cam->GetDetectedObjects(&objects);
        if (objects.size() > 0) {
            //We have a detected object
            break;
        }
    }

    //Move forward slowly
    Vec3D msp{0, 0.2, 0};
    while (!fc->CheckForStop()) {
        fc->fb->SetBodyVel(msp);
        fc->cam->GetDetectedObjects(&objects);
        if (objects.size() > 0) {
            ObjectInfo o = objects.front();
            if (o.bounds.area() > 0.3*(o.image_width * o.image_height)) {
                fc->fb->Stop();
                break;
            }
        }
    }
}
*/

void EnvironmentalMapping::Run(FlightController *fc, void *opts) {
    Log(LOG_INFO, "Environmental mapping initiated; awaiting authorisation...");
    SetCurrentState(fc, STATE_AWAITING_AUTH);

    if (!fc->WaitForAuth()) {
        Log(LOG_INFO, "All stop acknowledged; quitting!");
        return;
    }

    Log(LOG_INFO, "Authorisation acknowledged.");
    if (!fc->gps->WaitForFix(200)) {
        Log(LOG_WARNING, "No GPS fix; quitting.");
        return;
    }

    //Takeoff if we're not in the air
    if (!fc->fb->IsInAir()) {
        UtilityModule module(UtilityModule::UTILITY_TAKEOFF);
        module.Run(fc, reinterpret_cast<void*>(3));
    }

    SetCurrentState(fc, STATE_ENV_MAPPING);
    std::vector<ObjectInfo> objects;

    //while (!fc->CheckForStop()){
    //    o.bounds.area > 0.1*(o.image_width * o.image_height)

    //double start_radius = fc->lidar->GetLatest();
    //double alt = fc->gps->GetLatestRelAlt();
    GPSTData d;
    fc->gps->GetLatest(&d);

```

```

Coord3D centre = {d.fix.lat, d.fix.lon, d.fix.alt - d.fix.groundalt};
centre = CoordAddOffset(centre, 7, 90-fc->imu->GetLatestYaw()); //10m in front of copter
//centre = CoordAddOffset(centre, Vec3D{-7, 0, 0}); //5m To west of copter.

//Ah, magic offsets
float initial_yaw = 270-fc->imu->GetLatestYaw();
fc->fb->SetRegionOfInterest(centre);
for (int j=1; j<4; j+=1) {
    for (int i=0; i<360 && !fc->CheckForStop(); i=i+5) {
        Coord3D location = CoordAddOffset(centre, m_radius /*7 start_radius*/,
i+initial_yaw);
        GotoLocation(fc, location, centre, false);
        Log(LOG_DEBUG, "%d", i);

        fc->cam->GetDetectedObjects(&objects);
        if (objects.size() > 0) {
            //Log(LOG_DEBUG, "%.7f, %.7f", centre.lat, centre.lon);
            std::string path = std::string(PICOPTER_HOME_LOCATION "/pics/mappingimages") + "_"
+
            std::to_string(location.lat) + "_" +
            std::to_string(location.lon) + "_" +
            std::to_string(location.alt) +
            std::string(".jpg");

            fc->cam->TakePhoto(path);
        }
        fc->Sleep(100);
    }
    centre.alt += j;
    Log(LOG_DEBUG, "NEW LOOP");
}
//}

fc->fb->UnsetRegionOfInterest();
fc->fb->Stop();
m_finished = true;
}

bool EnvironmentalMapping::Finished() {
    return m_finished;
}

```

B. Graphical representation of difference between flight computer and remote

Manual mode: Signal sent from the remote to the flight controller.

Autonomous mode: Signals sent from Raspberry pi

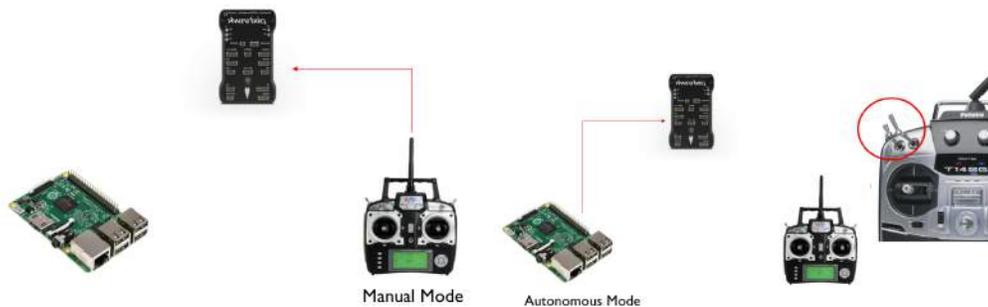


Figure 38: Manual mode (left) and Autonomous mode (right)