

**Integrated System Design for Intelligent Subterranean Exploration using  
Experimental UAVs**

by

Shane Joseph Allan

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Mechanical Engineering  
University of Alberta

© Shane Joseph Allan, 2023

# Abstract

Unique challenges arise when attempting to navigate unknown GPS-denied subterranean environments. To successfully explore these environments a vehicle must be able to reconcile the nonlinear dynamics of motion, accommodate for diverse terrain, adjust to static and moving obstacles, combat degraded sensor conditions, and transmit high quality information. Each of these fields have been investigated independently, however within the scope of this research these challenges are addressed collectively by a prototype unmanned aerial vehicle (UAV) which utilizes a sophisticated motion planner, customized hardware, and artificial intelligence-based image enhancement.

A motion planner suitable for the described challenges addresses a four-part problem. Generation of a high-fidelity map, exploratory motivation, a global navigational strategy, and an optimized and adaptable local trajectory planner. Each of these sub-tasks are addressed by implementing the Fast Autonomous UAV Exploration (FUEL) algorithm into the prototype platform.

Limitations to hardware systems present unique boundaries to efficient flight performance. Consumers typically address these issues through costly performance upgrades of existing components. However, when the flight terrain is predetermined, slight geometric modifications to the airframe can be implemented to optimize efficiency. Within this research, the vehicle uses custom-designed twisted rotor arms. This strategy is used to redirect a portion of the vertical thrust to produce greater yaw authority.

Arguably, the main value of a UAV is measured by the information it transmits to

remote operators. Motivated by this, five convolutional neural networks tasked with enhancing degraded low light images are implemented onboard the vehicle. Each of these networks are tested in real-time experiments to quantify their performance and their net benefits to the operator.

The ultimate goal of this research is to experimentally validate all presented design selections, specifically to understand the limitations and real-world performance of the systems and algorithms used.

*Dedicated to my beloved parents and sister,*

***Miles & Lilly & Sarah***

*to my beautiful girlfriend,*

***Megan***

*and to my loving grandparents,*

***Alex & Joanne & Lorraine & Jim***

*for their endless love, selfless encouragement, and gracious support*

# Acknowledgements

This challenging and rewarding experience could only have been achieved from a culmination of gracious support. Firstly, I would like to express my sincerest gratitude to my supervisor, Dr. Martin Barczyk, for your boundless mentorship, humbling intellect, lighthearted humour, and generous support. I cannot thank you enough for the inspiration and motivation you fostered throughout this process.

I would like to thank my incredible parents Miles Allan and Lilly Toker for your endless love and support. I am so proud to call you my parents and I can attribute much of who I am to your caring guidance. I will endlessly cherish the time we spend together - love you both, always.

To my sister Sarah who is my most dependable support. Thank you for always providing your grounded advice, loving encouragement, and enlightening humour. Your personality always radiates more joy in my life.

To my dearest girlfriend Megan for your unbounded love, encouragement, patience, and compassion. You continually inspire me to grow and battle through adversity. There is no one else I could imagine being by my side throughout this journey and my life.

Thank you to each of my companions in the Mechatronic Systems Lab. To one of my closest friends Charanjot, for your honesty and sincerity - it has been an absolute joy to have a friend such as yourself. To Younes for your inspiration and brilliance, I appreciate all the time you took to mentor me through this experience. To Amir and Ivy for all the humour and joy you brought to the lab each day. I will miss each of you as we continue to pursue our careers independently.

To my grandparents Alex and Joanne Allan, and Lorraine and Jim Hawkins. I will always revere your intellect and love. Your accomplishments have inspired me to take this journey and I cannot thank you enough for your curiosity, encouragement, and generous support throughout my life and academic career.

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	2
1.2	Thesis Objectives . . . . .	3
1.3	Thesis Outline . . . . .	3
<b>2</b>	<b>Background</b>	<b>4</b>
2.1	Quadcopter Model . . . . .	4
2.2	Depth Image Formation . . . . .	6
2.3	Convolutional Neural Networks . . . . .	9
2.3.1	Autoencoder . . . . .	9
2.3.2	Feedforward . . . . .	10
2.3.3	RetinexNet . . . . .	10
2.3.4	U-Net . . . . .	11
2.4	Motion Planning . . . . .	11
<b>3</b>	<b>Hardware</b>	<b>15</b>
3.0.1	Environmental Perception . . . . .	17
3.0.2	Hexacopter Environment Perception . . . . .	19
3.0.3	Quadcopter Environment Perception . . . . .	20
<b>4</b>	<b>Low Light Video and Image Enhancement</b>	<b>22</b>
4.1	Introduction . . . . .	22
4.2	Database Generation . . . . .	23
4.3	Methods . . . . .	24
4.3.1	Autoencoder . . . . .	24
4.3.2	Modified Autoencoder . . . . .	24
4.3.3	Feedforward . . . . .	25
4.3.4	RetinexNet . . . . .	26
4.3.5	U-Net . . . . .	27

4.4	Results . . . . .	31
4.5	Summary of Results . . . . .	36
<b>5</b>	<b>Evolution and Evaluation of Motion Planners</b>	<b>38</b>
5.1	Introduction . . . . .	38
5.2	Software Implementation . . . . .	39
5.2.1	Hexacopter Navigation . . . . .	40
5.2.2	Quadcopter Navigation . . . . .	42
5.3	Experimental Flight Testing . . . . .	44
5.3.1	Real-time Flight Test Results for the Hexacopter Vehicle . . .	44
5.3.2	Real-time Flight Test Results for the Quadcopter Vehicle . . .	49
5.3.3	Rotor Arm Redesign . . . . .	50
5.4	Summary . . . . .	56
<b>6</b>	<b>Conclusion</b>	<b>59</b>
6.1	Future Work . . . . .	60
	<b>Bibliography</b>	<b>62</b>

# List of Tables

3.1	Primary hardware components of the hexacopter and quadcopter designs.	17
4.1	Hyperparameters for training CNN models. . . . .	29
4.2	Model runtime evaluation. . . . .	33
4.3	Model accuracy evaluation. . . . .	35

# List of Figures

2.1	Quadcopter free body diagram. . . . .	4
2.2	Epipolar geometry for a calibrated stereo camera. . . . .	7
3.1	Side-view of prototype hexacopter (left) and quadcopter (right) designs.	15
3.2	Top-view of prototype hexacopter (left) and quadcopter (right) designs.	16
3.3	Hardware assembly of the hexacopter (left) and quadcopter (right) systems. . . . .	18
3.4	Hexacopter and quadcopter "X" air frame configurations. . . . .	18
3.5	Standard and depth images produced by the ZED 2 camera. . . . .	20
3.6	Quadcopter drift after performing a forward pitch maneuver in HITL testing. . . . .	21
4.1	Autoencoder architecture. . . . .	25
4.2	Modified Autoencoder architecture. . . . .	26
4.3	Feedforward architecture. . . . .	27
4.4	RetinexNet architecture. . . . .	28
4.5	U-Net architecture. . . . .	28
4.6	Training loss of the Autoencoder model. . . . .	31
4.7	Training loss of the modified Autoencoder model. . . . .	32
4.8	Training loss of the Feedforward model. . . . .	33
4.9	Training loss of the RetinexNet model. . . . .	34
4.10	Training loss of the U-Net model. . . . .	35
4.11	The images are A) Test image, B) Ground truth, C) Autoencoder, D) Modified Autoencoder, E) Feedforward, F) RetinexNet, G) U-Net. . .	37
5.1	System architecture and information transmission. . . . .	40
5.2	Hexacopter global motion planner architecture. . . . .	41
5.3	Quadcopter global motion planner architecture. . . . .	45

5.4	Wide view of hexacopter navigation (top), reference trajectory to set-point (bottom left), 3D voxel map construction after experimental test flight (bottom right). . . . .	46
5.5	Top perspective of the tracked trajectories by the hexacopter system in the underground space using our presented global motion planner. . . . .	47
5.6	Flight test of the hexacopter employing the global motion planner in an outdoor environment with a 15 km/h wind speed. . . . .	48
5.7	Frontier clusters (left) and local trajectory generation (right). . . . .	50
5.8	A) Standard view of quadcopter trajectory. B) Wide view of quadcopter trajectory. C) Real-time voxel grid mapping. D) Onboard quadcopter camera POV. . . . .	51
5.9	Total estimated volume travelled by the quadcopter with a 3 meter tall ceiling height. . . . .	51
5.10	Unexplored identified frontiers during experimental flight test. . . . .	52
5.11	3D perspective of the tracked trajectories by the quadcopter system in the underground space using the adapted exploration algorithm. . . . .	52
5.12	Tilted quadcopter arms. . . . .	53
5.13	Free body diagram of tilted quadcopter arms. . . . .	54
5.14	Experimental design for force and torque measurements. . . . .	56
5.15	Thrust and torque generation as a result of the arm tilt angle. . . . .	57
5.16	Force and torque percent change as a result of the quadcopter arm tilt angle. . . . .	58

# Abbreviations

**AV** Autonomous vehicle.

**CNN** Convolutional Neural Network.

**DARPA** Defence Advanced Research Projects Agency.

**DOF** Degree of Freedom.

**EKF** Extended Kalman Filter.

**ESC** Electronic Speed Controller.

**EV** Electric Vehicle.

**FBD** Free Body Diagram.

**FCS** Flight Control System.

**FOV** Field of View.

**FPS** Frames Per Second.

**FUEL** Fast UAV Exploration.

**GNSS** Global Navigation and Satellite System.

**GPS** Global Positioning System.

**IMU** Inertial Measurement Unit.

**INS** Inertial Navigational System.

**LiDAR** Light Detection and Ranging.

**LLIE** Low Light Image Enhancement.

**ML** Machine Learning.

**MS-SSIM** Multi-Scale Structural Similarity Index.

**MSE** Mean Squared Error.

**NCC** Normalized Cross Correlation.

**NMPH** Nonlinear Model Predictive Horizon.

**PSNR** Peak Signal to Noise Ratio.

**SSD** Sum of Squared Differences.

**SSIM** Structural Similarity Index Measure.

**ToF** Time of Flight.

**UAV** Unmanned Aerial Vehicle.

# Chapter 1

## Introduction

Exploratory robotics is a competitive field that is continuously innovating worldwide within academia. Research is motivated through many prestigious robotics competitions such as the DARPA Robotics Challenge, NASA Space Robotics Challenge, Technoxian, and VEX Robotics Challenge [1–4]. Each of these competitions require multidisciplinary collaboration to achieve success. Some of the most impactful research groups in robotics are the ETH Zurich Institute of Robotics and Intelligent Systems, Dyson Robotics Lab at the Imperial College London, Computer Science and Artificial Intelligence Lab at MIT, and the Oxford Robotics Institute.

ETH Zurich is currently pursuing research in robust land based learning and adaptive robotics [5–7] in addition to the aid of exoskeleton applications in rehabilitation medicine [8, 9]. Imperial College London is developing algorithms for scene and object perception [10–12] and collaborative robotics [13]. MIT is exploring complex and non-tradition control strategies for robotics such as brain-controlled and gesture based [14, 15]. The robotics team at Oxford is establishing various reinforcement learning strategies in addition to motion planning frameworks [16–19]. This is only a few of the prominent institutions and their current research interests with many more building off previous successes or independently developing new technologies. Locally, the University of Alberta houses world-class robotics research in conjunction with industry leading professors. Some of the prominent areas of research are ad-

ditive manufacturing, AI/ML in robotics, advanced robotic construction, nonlinear controls, exploratory aerial and ground vehicles, and biomechanical robotics.

## 1.1 Motivation

Industrial demands for increased safety, efficiency, and operability have accelerated the need for autonomous systems. These directives have motivated the design of a variety of vehicle platforms capable of performing autonomous tasks. However, lacking in current research is an experimentally validated and optimized vehicle platform capable of navigating through complex environments. Addressing this need is becoming standard across various industries such as infrastructure, resource extraction, and recreation.

Exemplifying this industrial trend is one of Canada's largest resource extraction companies, Suncor. The company deployed a fleet of autonomous haulage vehicles for product transportation within their mining operations [20]. This quickly encouraged other producers such as CNRL and Imperial Oil to adapt this technology in order to stay competitive [21, 22]. This is not only a local trend as companies such as Glencore and Rio Tinto have extensive global investments in autonomous systems [23, 24]. Driving modern research is the complexity of the environments these systems operate in. Highly remote, dynamic, and sensory deprived underground mining operations are the apex of difficulty for autonomous systems. These environments lack GPS signals which inhibits the attainment of an absolute positioning reference. The difficulty of this challenge has motivated government bodies such as the Defence Advanced Research Projects Agency (DARPA) to offer substantial monetary rewards for the top performers in the Subterranean Robotics Challenge [1]. This motivation for advancement continues as companies such as NASA and SpaceX push to explore other astronomical bodies such as neighboring planets, asteroids, and exoplanets where GPS is unavailable.

## 1.2 Thesis Objectives

The goal of the research presented in this thesis is to build a robust and efficient quadcopter capable of autonomous GPS-denied flight. We present an evolution of design decisions which highlight the problems faced and the lessons learned. Within the scope of this research are extensive hardware and software considerations which include off-the-shelf component selections, mounting and interfacing, sensor evaluation, and software integration/modification. This research evaluates all the presented design choices and experimental flight tests to ensure their validity and performance.

Building on the previous work in [25–28] this thesis aims at improving the efficiency of the UAV through the elimination of the Velodyne LiDAR sensor. These sensors are common in underground applications due to their high-fidelity mapping ability. However, this sensor is expensive, computationally intensive, and the mass of the device restricts the agility of the aerial vehicle. Through the removal of this device the efficiency was improved while reducing capital costs.

## 1.3 Thesis Outline

This research progresses chronologically to highlight the lessons learned during design. Therefore, the remainder of this thesis will be structured as follows: Chapter 2 reviews comparable research and methodologies to understand the current trends and challenges. Chapter 3 presents the hardware and software infrastructure used onboard the custom UAVs. Chapter 4 covers convolutional neural networks used for artificial low light enhancement. Chapter 5 describes the experimental flight tests of our vehicle and onboard path planning algorithms. Chapter 6 summarizes the thesis and suggests future work directions.

# Chapter 2

## Background

### 2.1 Quadcopter Model

The 6-DoF rigid body dynamics of a quadcopter are derived from the free body diagram (FBD) in Figure 2.1.

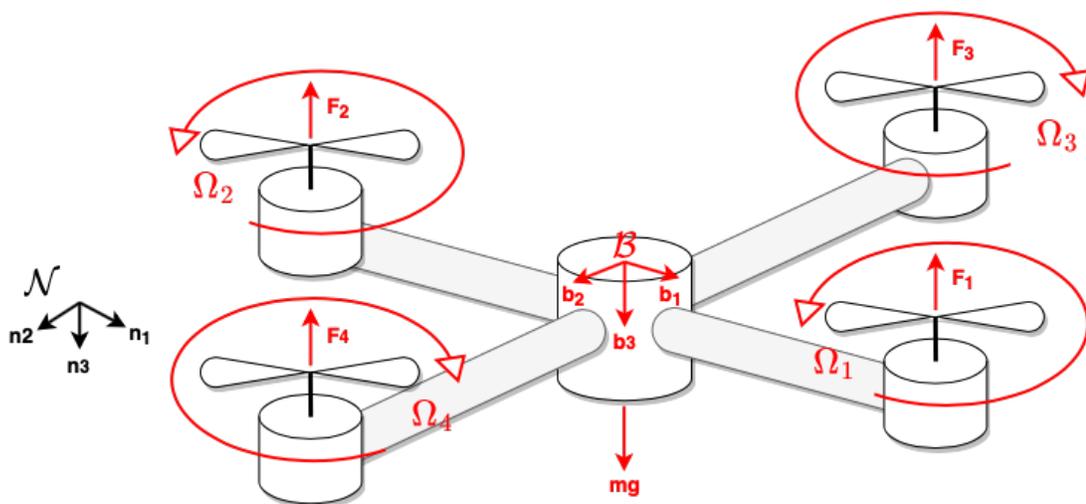


Figure 2.1: Quadcopter free body diagram.

Figure 2.1 shows the ground-fixed NED (north-east-down) inertial frame of reference  $\mathcal{N}$  with  $\Omega$  denoting the rotational speed of the propellers, opposite to the torque generated. A non-inertial reference frame  $\mathcal{B}$  is fixed to the body of the quadcopter at its centre of mass (CM) with gravity force  $mg$ . The onboard electronic speed controllers (ESCs) directly control the four rotor speeds  $\Omega_1, \dots, \Omega_4$  which are proportional to the propeller thrusts  $F_1, \dots, F_4$  and counter-torques  $\tau_1, \dots, \tau_4$ . In this way

the controls generate the wrench  $F_b$  in Equation (2.1).

$$F_B = \left[ \sum_{i=1}^4 f_{B,i} \quad \sum_{i=1}^4 \tau_{B,i} \right] + F_{B,g} \quad (2.1)$$

where:

$$\sum_{i=1}^4 f_{B,i} = \begin{bmatrix} 0 \\ 0 \\ -f_1 - f_2 - f_3 - f_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -k_L(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \end{bmatrix} \quad (2.2)$$

and

$$\sum_{i=1}^4 \tau_{B,i} = \begin{bmatrix} df_4 - df_2 \\ df_1 - df_3 \\ \tau_1 + \tau_3 - \tau_2 - \tau_4 \end{bmatrix} = \begin{bmatrix} dk_L(\omega_4^2 - \omega_2^2) \\ dk_L(\omega_1^2 - \omega_3^2) \\ k_D(\omega_1^2 + \omega_3^2 - \omega_2^2 - \omega_4^2) \end{bmatrix} \quad (2.3)$$

In Equation (2.3),  $d$  is the length from the center of mass to the point of thrust generation with  $k_L$  and  $k_D$  being the lift and drag coefficients inherent to the propeller geometry. Additionally,  $f$  is the thrust and  $\omega$  is the angular velocity vector.

The gravity wrench applied to the body at the centre of mass with the  $\mathcal{N}$  and  $B$  frames aligned is Equation (2.4).

$$F_{B,g} = \begin{bmatrix} R^T & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & mg & 0 & 0 & 0 \end{bmatrix}^T \quad (2.4)$$

In equation (2.4),  $R \in SO(3)$  is the rotation matrix of frame  $\mathcal{B}$  with respect to frame  $\mathcal{N}$ , such that  $R^T$  transforms a vector in frame  $\mathcal{N}$  basis to frame  $\mathcal{B}$  basis.

Applying the net wrench in (2.1) to the Newton-Euler equations governing rigid-body dynamics [29] results in (2.5)

$$\begin{bmatrix} \sum_{i=1}^4 f_{B,i} \\ \sum_{i=1}^4 \tau_{B,i} \end{bmatrix} = \begin{bmatrix} m\dot{v} \\ \mathcal{I}_B \dot{\omega}_B \end{bmatrix} + \begin{bmatrix} \omega_B \times mv \\ \omega_B \times \mathcal{I}_B \omega_B \end{bmatrix} \quad (2.5)$$

The dynamics of the attitude matrix  $R \in SO(3)$  and position of the body's CM with respect to the  $\mathcal{N}$  frame origin are governed by [29]

$$\dot{R} = RS(\omega_B) \quad (2.6)$$

$$\dot{p}_W = Rv_B \quad (2.7)$$

where  $S$  is an operator mapping  $\mathbb{R}^3$  vectors to  $3 \times 3$  skew-symmetric matrices such that  $S(x)y = x \times y$ ,  $x, y \in \mathbb{R}^3$ . The matrix  $R \in SO(3)$  can be represented as the roll-pitch-yaw Euler angle sequence  $(\phi, \theta, \psi) \in \mathbb{R}^3$  through (2.8) [29]

$$R = \begin{bmatrix} c_\psi c_\theta & -s_\psi c_\phi + c_\psi s_\theta s_\phi & s_\psi s_\phi + c_\psi s_\theta c_\phi \\ s_\psi c_\theta & c_\psi c_\phi + s_\psi s_\theta s_\phi & -c_\psi s_\phi + s_\psi s_\theta c_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \quad (2.8)$$

Each of Equations (2.6) and (2.7) define the kinematic equations which govern the dynamics of a +-frame quadcopter as illustrated in Figure 2.1 [29]. In this research the hardware UAV employs an X-frame configuration where the longitudinal axis is at  $45^\circ$  between the two front arms. The transformation required to change the quadcopter to this configuration will be provided in Chapter 5.

## 2.2 Depth Image Formation

Sensor perception is achieved through a variety of devices such as LiDAR, rangefinders, ultrasonic, thermal, and stereo cameras [30]. Each of these sensors are advantageous under specific operating conditions and can provide standalone localisation solutions for robotic applications. When considering commercial value and availability stereo cameras can provide effective solutions. To properly localise a vehicle in 3D-space using optical methods, calibration must be performed. Equation (2.9) establishes the calibration matrix which is manipulated by the focal length, scaling factors, and centering offsets [31].

$$K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.9)$$

Assuming the camera is calibrated the principles of epipolar geometry can be applied to estimate the 3D coordinates of a point as seen in Figure 2.2.

This is done through trigonometric relations between two 2D points on the individual image planes across the epipolar scan lines.

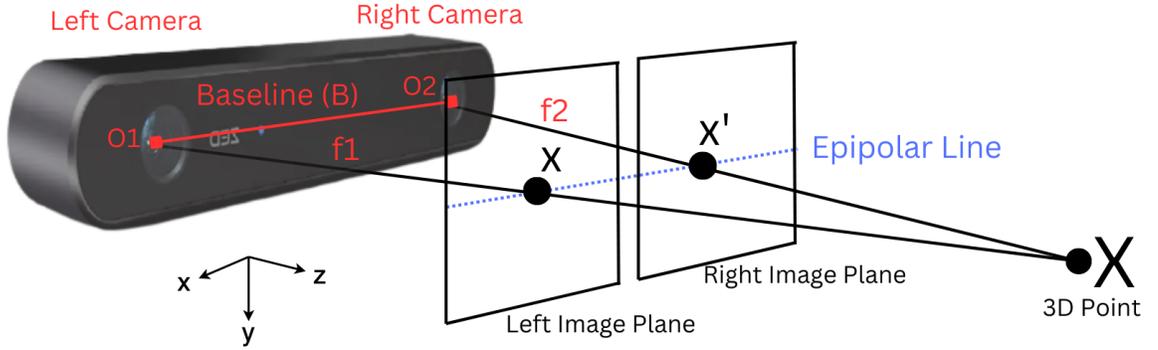


Figure 2.2: Epipolar geometry for a calibrated stereo camera.

Suppose that  $x$  and  $x'$  are the locations of the 3D point in the left and right image plane, written in homogeneous coordinates as

$$x = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad \text{and} \quad x' = \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \quad (2.10)$$

The relations are exemplified in Equations (2.11) and (2.12).

$$\hat{x} = K_1^{-1}x \quad (2.11)$$

$$\hat{x}' = K_2^{-1}x' \quad (2.12)$$

which, after the independent calibration matrices are applied results in the following relation between the two points in Equation (2.13).

$$\hat{x}' = R_c \hat{x} + T_c \quad (2.13)$$

Now, by exploiting the coplanar relationship between them we can arrive at the essential matrix in Equation (2.14).

$$E = S(T_c)R_c \quad (2.14)$$

This essential matrix allows for the coordinates of a point within one camera's reference to be related to another. Imposing the epipolar constraint where the epipoles are located at infinity along the horizontal axis, the images are rectified, and the epipolar lines are aligned evenly as in Equation (2.15).

$$\hat{x}^T E \hat{x}' = 0 \quad (2.15)$$

Using this and substituting (2.14) into (2.15) produces Equation (2.16).

$$\hat{x}^T \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -B \\ 0 & B & 0 \end{bmatrix} \hat{x}' = \begin{bmatrix} u & v & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -B \\ Bv' \end{bmatrix} = 0 \quad (2.16)$$

In which case  $\hat{x}$  is a point within one calibrated camera's image plane,  $\hat{x}'$  is a second point in the image plane of a second calibrated camera, and  $B$  is the baseline between camera centers.

Simplifying (2.16), results in Equation (2.17)

$$Bv = Bv' \quad (2.17)$$

These relations allow for the understanding that only a horizontal translation is required to derive depth of a point on the image plane from two images of a cali-

brated and rectified stereo camera. Therefore, the disparity can be calculated through Equation (2.18).

$$\text{Disparity} = \hat{x} - \hat{x}' = \frac{Bf}{z} \quad (2.18)$$

The match between  $\hat{x}$  and  $\hat{x}'$  is found along the epipolar line directly or by using methods such as sum of squared differences (SSD), normalized cross correlation (NCC), or structural similarity index measure (SSIM). Finally, the depth which is inversely proportional to the disparity can be calculated [31].

## 2.3 Convolutional Neural Networks

The importance of information rich visual data is becoming increasingly important as it becomes more heavily relied upon for critical tasks. From exoplanet exploration [32], subterranean navigation [33], and under water observation [34], visual data is continually being used to drive human based decision making in challenging environments. Machine learning, particularly CNNs are leading frameworks for processing and enhancing visual data.

### 2.3.1 Autoencoder

Autoencoders are two-part functions constructed of an encoder and a decoder. The encoder, composed of parameterized layers, looks to replicate the features extracted from the input data [35]. Conversely, the decoder looks to reconstruct the initial images by decoding the extracted features following the identification [36]. Under-complete autoencoders are identifiable through a bottleneck layer in which the output dimension is smaller than the input; constraining the features extracted by the model [37]. These constraints ensure the most representative features are extracted in the learning stage through the inclusion of a loss function to minimize reconstruction error [38]. Utilizing these principles, denoising autoencoders enhance images by learning significant features of corrupted images and abstracting noise in the bottleneck [39].

Autoencoders enhanced low light images in the studies of [40], [41], [42] as standalone and supplementary solutions.

### **2.3.2 Feedforward**

Feedforward convolutional neural networks combine convolutional layers with fully connected layers in a linear fashion to learn the features of the input data [43]. The addition of max pooling between convolutional layers reduces spatial complexity of feature maps and decreases computational complexity, but can restrict fine detail during extraction [44] [45]. Combining these processes is the study performed by Li et al. and the introduction of LightenNet [46]. This study successfully enhances the visibility of low light images using four convolutional layers as a backbone for the model. Similarly, Schwartz et al. proposed DeepISP - a two stage network for low, local and high, global level feature extraction [47]. This network performed image enhancement for both well-lit and low light images to effectively improve feature perception.

### **2.3.3 RetinexNet**

Land et al. [48] proposes the deconstruction of an image into the components of illuminance and reflectance. Known as Retinex theory, a primitive interdependence of these image properties is established and is later refined in the works of [49], [50], [51], [52] to abstain from degrading image artifacts such as the halo effect, color distortion, and inadequate edge sharpening [53]. Combining deep learning frameworks with Retinex theory is RetinexNet proposed by Wei et al. [54]. This model is constructed of three modules: deconstruction, adjustment, and reconstruction. The deconstruction module processes the images through a Decom-Net which is a variation of convolutional layers to extract the illuminance and reflectance representations of the initial image. These are then processed through an Enhance-Net which, again, utilizes a variation of convolutional layers while manipulating the spacial resolution of the feature maps.

Combining these networks into a pipeline renders quality qualitative results.

### 2.3.4 U-Net

U-Net proposed by Ronneberger et al. [55] introduces a sophisticated CNN structure for refined feature extraction. This symmetric network uses the competing operations of max pooling and up-sampling to compress and expand the dimensionality of the feature maps in addition to convolutional layers to extract prominent features [56]. Skip connections are used to retain high dimension details within low dimensional maps across the network [57]. This framework has been modified in the study performed by Ai and Kwon [58] and coined Attention U-Net. This network integrates attention gates within the skip connections. This occurs before concatenation to filter irrelevant and noisy information input to the feature maps during forward and backward propagation to achieve improved image enhancement [59]. Alternative modifications for low light enhancement to the U-Net structure are presented by Cai and U [60]. In this implementation recurrent residual convolutional layers replace traditional convolutions and the spacial resolution of the bottleneck is greatly increased in size from the use of multiways dilated convolution blocks [60]. These modifications produced substantial improvements visually and quantitatively.

Overall, each of the models chosen for comparison achieved previous successes in LLIE. The exact structures used in this article are not identical to those reviewed in this section, but follow the foundational frameworks to establish a baseline for comparison.

## 2.4 Motion Planning

UAV flight has become increasingly popular for a variety of applications including recreation [61], commercial [62], industrial [63], and military sectors [64]. This variety forces the diversification of UAV design to adapt to user-specific applications. At the lowest level UAV hardware consists of the aerial platform, propulsion system, and

flight control system (FCS) [65]. The platform architecture is situational, but common variations include fixed wing, single rotor, multi-rotor, and hybrid systems [66]. Typical propulsion systems are either combustion based or electrical with some specialty variations [67]. The FCS is modified based on mission analytics. Examples include full autonomy, semi-autonomous, teleoperated, or remote controlled [68]. To achieve full autonomy the embedded FCS must perform independent mission planning and adaptable flight operation without human interference. Environmental perception is a critical step in autonomy which relies on sensory inputs from external devices. This includes sensors such as Global Navigation and Satellite System (GNSS), barometers, Inertial Measurement Units (IMUs), compasses, Light Detection and Ranging (LiDARs), stereo and monocular cameras, ultrasonic, RADAR, and Time of Flight sensors (ToF) [69]. These sensors provide estimation of the vehicle states such as position, orientation, and speed to the FCS. Independently, some of these sensors suffer from inaccuracies and drift over time [69]. To mitigate this problem, Integrated Navigation Systems (INS) are proposed to introduce sensor fusion typically using a Kalman filter to incorporate redundancy through additional sensors for more robust navigational information [70]. GNSS systems are one of the most popular sensors providing time-synchronized positional and velocity data [67]. However, in subterranean environments GNSS information is either unreliable or unavailable in which vision-based navigation [71] or INS is relied upon [72].

Navigational tactics deployed onboard UAVs are traditionally classified as deliberative (global planning), sensor-based (local planning), or hybrid [65].

Following the establishment of the vehicle's states and environmental map, there are many algorithms proposed to allow a UAV to navigate efficiently within a local space. Dijkstra's algorithm [73] is proposed as an efficient algorithm which determines the shortest route to an end goal. This is done through a weighted graph where every node within the graphical map is iteratively updated to provide the shortest distance between all nodes. This methodology guarantees the shortest path

between two chosen nodes can always be attained, assuming the map contains all positive edge weights. However, this methodology requires computational complexity particularly when the map size increases dramatically. Resolving this constraint is the A\* algorithm. Utilized by [74–76] for autonomous navigation the A\* algorithm is based upon a heuristic function which estimates the distance to an objective node. This strategy evaluates the cost function only to the goal node as opposed to the entire search space, greatly reducing computational demand. Rapidly exploring random trees (RRT) and probabilistic roadmaps (PRM) are two sampling based techniques which extract points within the configuration space to be grouped into nodes for local traverse [77]. RRT commonly uses random sampling techniques to navigate the environment resulting in suboptimal solutions. The standard RRT algorithm may vary to include auxiliary paths which contain environmental details regarding the end goal known as informed RRT [78] and further optimized and informed algorithms such as RRT\* which use nearest-n node grouping strategies [79]. PRM and the success of implementation changes based on the users chosen sampling method and node grouping techniques. Evaluated and tested in [80, 81] PRM can use sampling techniques such as random, grid, halton, etc., in combination with different grouping methods such as nearest-n, component, visibility, etc.. Artificial potential fields (APF) originally proposed by [82] is an algorithm that establishes attractive and repulsive forces around the target and obstacles, respectively. The navigation through this field is computed iteratively through a gradient descent algorithm to locate optimal pathways [83]. However, this algorithm does not guarantee the global optima is reached and can get trapped in a local minimum. The works of [84, 85] suggest modifications to APF to ensure satisfactory trajectory planning is achieved. [84] presents a simulated annealing method which uses nearest neighbour sampling and energy derived probabilistic associations to encourage departure from the local minimum. Secondly, [85] implemented a Regular Hexagon-Guided method to encourage control actions within constrained geometry to escape the local minima. In recent years machine

learning (ML) algorithms, particularly that of reinforcement learning (RL) and deep reinforcement learning (DRL) are being used for path planning [86]. These strategies have now extended to collaborative multi-vehicle platforms such as UAV swarms in [87] and advanced to task-specific swarm objectives in [88]. Most complex DRL algorithms use the deep deterministic policy gradients (DDPG) proposed by [89]. In the context of path planning DDPG establishes the environments as a continuous state space where a subset of this environment is actionable by the vehicle. Within the DDPG there are two neural networks. Simplification of the algorithm describes the first network as the actor which suggests actions to the second network, the critic based on the current state. The critic evaluates these actions and provides feedback refining the policy to achieve a desired action [89].

Many variants of the above algorithms have been proposed to increase efficiency, optimality, and convergence. Such algorithms are RJA-Star [90], G-RRT\* [91], RRT\*-SMART [92]. RL has been adapted and integrated with traditional path planners to introduce hybrid solutions such as PRM-RL [93] and APF-RL [94]. Each of these methodologies show improvements in specific areas over the original algorithms. FUEL, a variant of the A\* search algorithm which introduces a heuristic function in addition to B-spline optimization proposed by Zhou et al. eliminates the need for computationally intense sensors such as LiDAR. This compact algorithm only requires a depth image from a stereo camera to initiate frontier-based exploration and global/local tour generation. Due to these advantages this algorithm was selected to be the end-to-end motion planner of the proposed quadcopter.

# Chapter 3

## Hardware

Achieving autonomous navigation of a UAV is perceived primarily as a complex software problem however, in task based autonomous flight this is only one objective. For a UAV to accomplish mission critical objectives designers must consider the restrictions of payload, sensors, computational time, flight time, maneuverability, cost, and the environment. Each of these considerations can be further refined to the examination of material selection, durability, accuracy, error tolerance, aerodynamics, operability ranges/temperatures, power consumption, and safety.

The two UAVs discussed in this thesis have been specifically designed for their deployed environments. Targeting operations in underground mines, light-rail train (LRT) tunnels and caves, the chosen design features presented in Figure 3.1 and 3.2 reflect the exposure to these environments.



Figure 3.1: Side-view of prototype hexacopter (left) and quadcopter (right) designs.



Figure 3.2: Top-view of prototype hexacopter (left) and quadcopter (right) designs.

Initially, the X-Frame hexacopter design was chosen for the task of autonomous flight. Uncertainty in sensor requirements, payload, and flight time established the need for a 6-propeller design which could adapt to an ambiguous flight-stack. However, once achieved it was observed that the design could be simplified. This resulted in the production of the X-Frame quadcopter.

The main hardware components of these systems are presented in Table 3.1.

The hardware commonalities between the two UAVs are the result of their shared objective and prior knowledge of efficiency. This resulted in both prototypes utilizing the Nvidia Jetson NX computing modules, Pixhawk 5x flight controllers, and DJI airframes. The primary differences are with the sensors used for environmental perception. A high level overview of the hardware construction is found in Figure 3.3.

The airframes are the widely available F550 and F450 Flamewheel kits from DJI. These come standard with PA66+GF30 fibreglass reinforced polyamide plastic components such as 5" tall landing gears, 8" arms, and 9" propellers. The kit also contains DJI 430 Lite electronic speed controllers (ESCs), top and bottom mounting

Table 3.1: Primary hardware components of the hexacopter and quadcopter designs.

Hexacopter	Quadcopter
DJI F550 Flamewheel Frame	DJI F450 Flamewheel Frame
Velodyne Puck LITE LiDAR	Garmin LiDAR-LITE v3
Pixhawk 5x Flight Controller	Pixhawk 5x Flight Controller
Nvidia Realsense T265	Stereolabs ZED 2 Camera
Jetson Xavier NX	Jetson Xavier NX
DJI 430 Lite ESC	DJI 430 Lite ESC
DJI 2312E/960 KV Motor	DJI 2312E/960 KV Motor
NVMe 970 EVO Plus SSD	NVMe 970 EVO Plus SSD
AW-CB375NF Wifi Module	AW-CB375NF Wifi Module
9" DJI Propellers	9" DJI Propellers
1-MaxAmps 5450 120c/14.8V LiPo Battery	1-Venom 5000 50c/14.8V LiPo Battery
2-Venom 2200 50c/11.1V LiPo Batteries	1-Venom 2200 50c/11.1V LiPo Battery

plates with an integrated PCB for electronic transmission to the ESC's. Altogether these UAVs are a lightweight 3.7kg and 2.3kg for the hexacopter and quadcopter, respectively. Both the platforms are configured in the  $X$  orientation with the vehicles forward heading placed at  $30^\circ$  for the hexacopter and  $45^\circ$  for the quadcopter between the front arms. These orientations are displayed in Figure 3.4.

### 3.0.1 Environmental Perception

Active perception is essential for accurate and precise mission planning. Responsible for the localization of sensory data is the Pixhawk 5X flight controller. This module is deployed with the FMUv5X open standard firmware running on a 32-Bit Arm Cortex-M7 at 216MHz with 2MB of flash memory and 512KB RAM flight management system alongside a 32-Bit Arm Cortex-M3 running at 24MHz with a 8KB SRAM IO processor for external sensor processing. This device also contains thermally isolated and vibration dampened triple redundant IMUs and double redundant

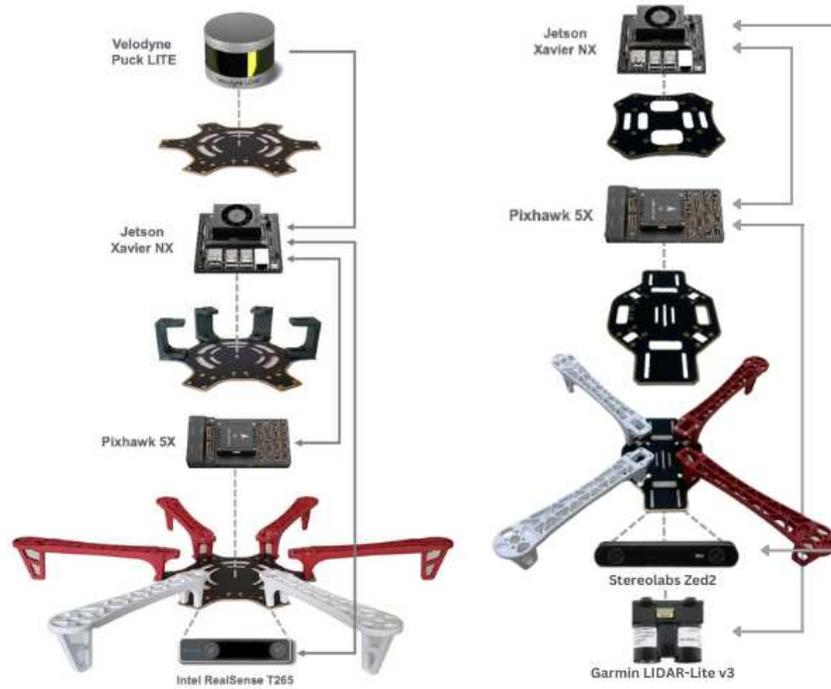


Figure 3.3: Hardware assembly of the hexacopter (left) and quadcopter (right) systems.

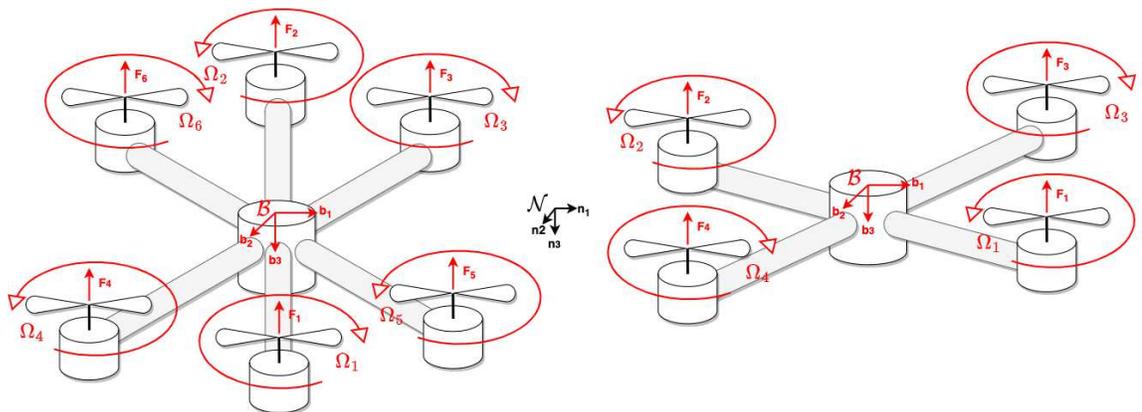


Figure 3.4: Hexacopter and quadcopter "X" air frame configurations.

barometers, ensuring fail safe sensory acquisition. With the Ardupilot v4.3.0 software platform and the QGroundControl ground station interface, users are able to control all elements of the flight parameters. The integrated IMUs provide vehicle position, orientation, inertial, and altitude information. However, as a standalone sensor the IMU suffers from inaccuracies due to noise and drift. A critical feature of the Pixhawk platform is the availability to perform sensor fusion of external devices through the use of a pre-programmed Extended Kalman Filter (EKF). This filter allows for seamless integration of an external IMU to improve the accuracy of estimation.

### **3.0.2 Hexacopter Environment Perception**

The hexacopter introduces sensor fusion techniques using the Pixhawk 5X flight controller for data accumulation and processing. To accurately localize the UAV within its environment two independent approaches were investigated. Firstly, a Velodyne Puck LITE LiDAR device is introduced for state estimation. This 16-channel 3-D laser scanner renders accurate point cloud data at 10 Hz for a range up to 100m. This cylindrical scan has a FOV of 360° accumulating 300,000 points/second. This information is passed to a Fast Lidar Odometry and Mapping (FLOAM) algorithm which performs the localization and mapping. The second solution utilizes the decentralized processing power onboard the Intel Realsense T265 stereo camera. This device has a 173° FOV with a 848x800 resolution in addition to an embedded visual SLAM algorithm for position and velocity estimates at rates of 200Hz. Independently these algorithms offer accurate solutions for environment perception. However, due to the advantages of a 360° FOV and higher precision within its operating range the Velodyne LiDAR was used as the primary method for perception and state estimation.

The Intel Realsense T265 camera is used as a redundant state estimator which provides IMU data for estimation for the Pixhawk 5X.

### 3.0.3 Quadcopter Environment Perception

The quadcopter introduces redundant IMU data into the flight controller through a secondary sensor embedded within the ZED 2 stereo camera from StereoLabs. This camera provides inertial sensor data at a rate of 400 Hz. In addition to the IMU, the ZED 2 camera provides stereo depth estimation as a 360x640 pixel depth map for 120° field of view at distances of up to 20m with accuracy of >95% up to 15m. This depth image is produced through triangulation of matched pixels across stereo image pairs. The depth map produced by the ZED 2 camera is shown in Figure 3.5.



Figure 3.5: Standard and depth images produced by the ZED 2 camera.

The combination of these sensors with the manufacturers built in stereo SLAM algorithm, allows for the UAV to perform 6-DoF positional tracking through visual inertial odometry (VIO). Post-integration, Hardware in The Loop (HITL) testing was performed to validate the design selection. The prototype UAV suffered from inaccuracies during specific axial maneuvers. When the UAV would pitch forward to travel linearly along the X-axis as defined in Figure 3.4, the VIO reported the UAV drifted upward in the negative Z-axis when completing the trajectory. This inability to accurately track the desired trajectory is shown in Figure 3.6.

Similarly, this result was mirrored during opposing pitch maneuvers where the UAV would drift downward in the positive Z-axis after the final position was reached. Investigation revealed that as the UAV pitches the VIO indicates the drone is descending as the heading begins to point toward the ground. As a result, the UAV compensates for this perceived loss of elevation by ascending after the trajectory is

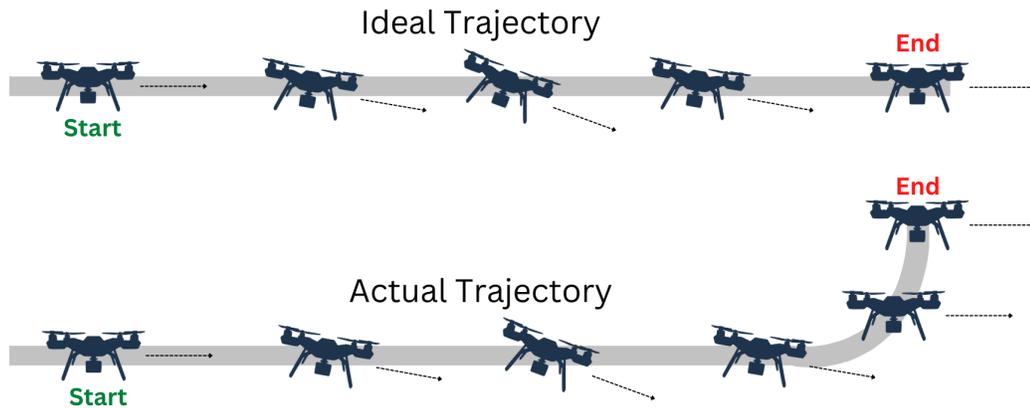


Figure 3.6: Quadcopter drift after performing a forward pitch maneuver in HITL testing.

complete. The altitude information derived from VIO fusion of the ZED 2 camera and flight controller were unable to reconcile accurate altitude information even after the inclusion of barometric data from the Pixhawk. Therefore, a Garmin LiDAR-Lite v3, a laser-based altitude sensor, was integrated into the design. This sensor pulses at 500 Hz to obtain accurate altitude information to a precision of  $\pm 2.5\text{cm}$  and is used as a replacement for altitude estimates of the VIO, correcting the drift effects.

# Chapter 4

## Low Light Video and Image Enhancement

### 4.1 Introduction

Low light image enhancement (LLIE) is a procedure in which poorly illuminated and occluded images are processed to improve the perception of the observed scene [95]. This process is important for sub-tasks such as surveillance [96], navigation [97], image classification [98], and intelligent transportation [99]. Each of these applications look to use an enhanced image for further analysis, however frequently neglected is the operational advantage the enhanced image provides as-is. Navigating low light environments for inspection-based tasks such as exploration of caves, rail tunnels, and mines is a challenging feat for locally controlled UAV operators. However, real-time enhanced image feedback provides the user with the option to overcome degraded sensor conditions to fly in these mission specific environments.

LLIE is classically approached through methodologies such as Histogram Equalization [100], Gamma Correction [101], and Retinex theory [102] or a combination thereof [103]. However, recent advancements in machine learning and computer vision can be used as standalone solutions or combined with classical methodologies to provide robust and optimized alternatives. Deep learning can be characterized through three primary branches, that of supervised, semi-supervised and unsupervised learning [104]. For the purposes of this evaluation, we will be contrasting

supervised learning techniques against one another to establish a baseline for future UAV research applications. Specifically, the methodologies exploiting networks architectures of convolutional neural networks (CNNs). To establish this baseline a comparison is drawn between the CNN structures of an Autoencoder, Modified Autoencoder, Feedforward, RetinexNet, and U-Net for the task of LLIE [105]. Each of these architectures have proven to be effective in different applications, especially when computational power is unrestricted [106] and training data sets are highly filtered and precise [107]. However, in reality the images encountered are substandard and circumstances of limited processing capabilities generate unforeseen challenges during real-time implementation. The results of these complications can cause models to become inoperable [108], lagged [109], or suffer from degraded performance [110].

## 4.2 Database Generation

In part, the fundamental success of any machine learning model is derived from the data it is trained on. Size, diversity, and quality of the database directly impacts the accuracy of results [111]. There are many well established datasets for low light such as the LOw-Light dataset (LOL) [54], See in the Dark dataset (SID) [112], and ExDark dataset [113]. Each of these image databases provide a diverse range of visual information. The LOL and SID databases are ideal for supervised training due to their ground truth image pairs. Whereas the ExDark database is ideal for unsupervised training due to the lack of a ground truth. Either the LOL or SID databases could have been used in this application, however to ensure translatable performance in real-time a database produced from the onboard ZED 2 camera was used. A total of 250 RGB images in low/high light were gathered. By generating a custom dataset many parameters can be controlled including the data precision within the image pixels, the resolution, image scenes, and lighting saturation. The ZED 2 stereo camera has a resolution of 1344x376. However, this is obtained by concatenating the field of view

of the two 672x376 lenses. For simplicity and accuracy within implementation and training, 672x376 images were collected for testing using a single lens of the camera. These additional considerations ensure repeatable performance can be obtained.

## 4.3 Methods

Suffering from large amounts of noise and occlusions, the stereo camera underperforms in low lighting conditions. To reconcile these deficiencies, we propose four CNN network architectures to enhance raw 3-channel RGB images. Of the 250 images collected, 80% were reserved for training and 20% for testing. Prior to training, the images were normalized between  $[0,1]$  while retaining the resolutions of 360x640. These models were trained and tested on a 14 Core Intel i9-10940X CPU with an Nvidia GeForce RTX 3090 processing 24 GB of G6X memory. Additional testing was performed on a Jetson Xavier NX with a 6 core Nvidia Caramel ARM CPU and 384 core Nvidia Volta GPU. The frameworks of the model are as follows.

### 4.3.1 Autoencoder

The structure of the Autoencoder implementation can be seen in Figure 4.1.

The Autoencoder consists of two modules, the encoder and decoder each mirroring the other. This model learns and encodes different features from the input images through convolutional layers which are passed through ReLU activation functions. This simultaneously reduces the resolution using max pooling while increasing the number of feature maps between convolutional layers to a maximum of 32. Using this process, the model is able to learn higher level features while decreasing computational requirements. The decoder module uses the opposing operation of upsampling to reconstruct the enhanced 360x640 RGB image.

### 4.3.2 Modified Autoencoder

The Modified Autoencoder structure is presented in Figure 4.2

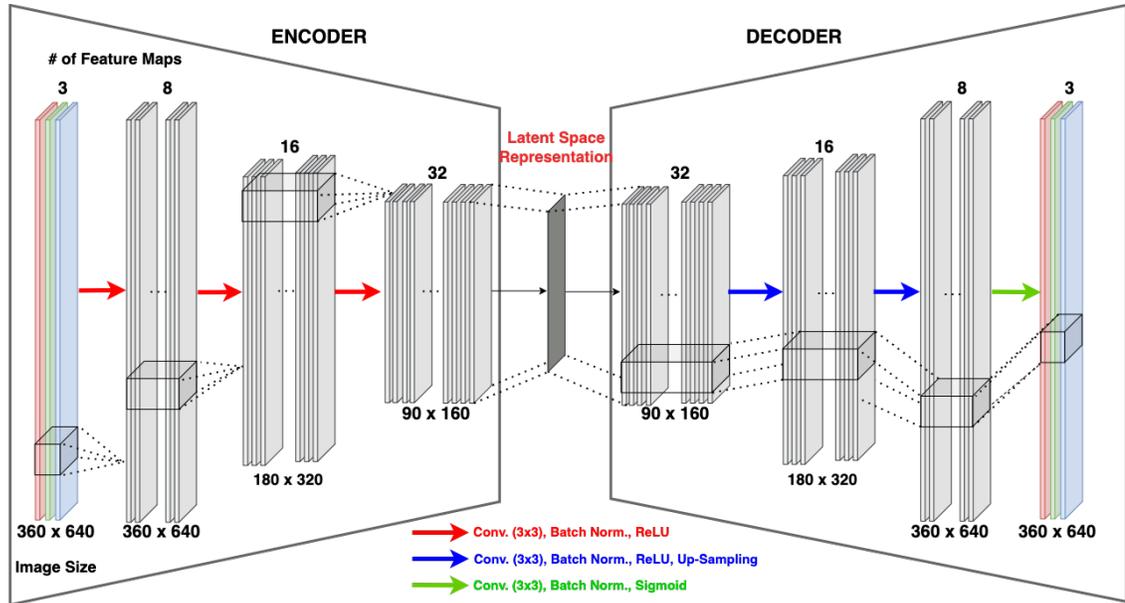


Figure 4.1: Autoencoder architecture.

This model followed the same design as the Autoencoder, but maintains the dimensionality of the feature maps. This variation is introduced to eliminate fine detail loss in resolution reduction before reconstructing the enhanced image. Although computationally more expensive, the extent of this trade-off will be investigated.

### 4.3.3 Feedforward

Next, a Feedforward neural network containing 6 convolutional layers is designed in Figure 4.3.

This model works to extract features from the training data through a series of convolutional layers. Maintaining the resolution between layers, the feature maps are increased from 3 to a maximum of 512. This model operates linearly with all information passing from one convolution to the next with ReLU activation functions between layers. Although a decrease from 512 to 3 channels is a substantial reduction and some fine detail extraction is lost, this model is still effective in eliminating degrading image artifacts.

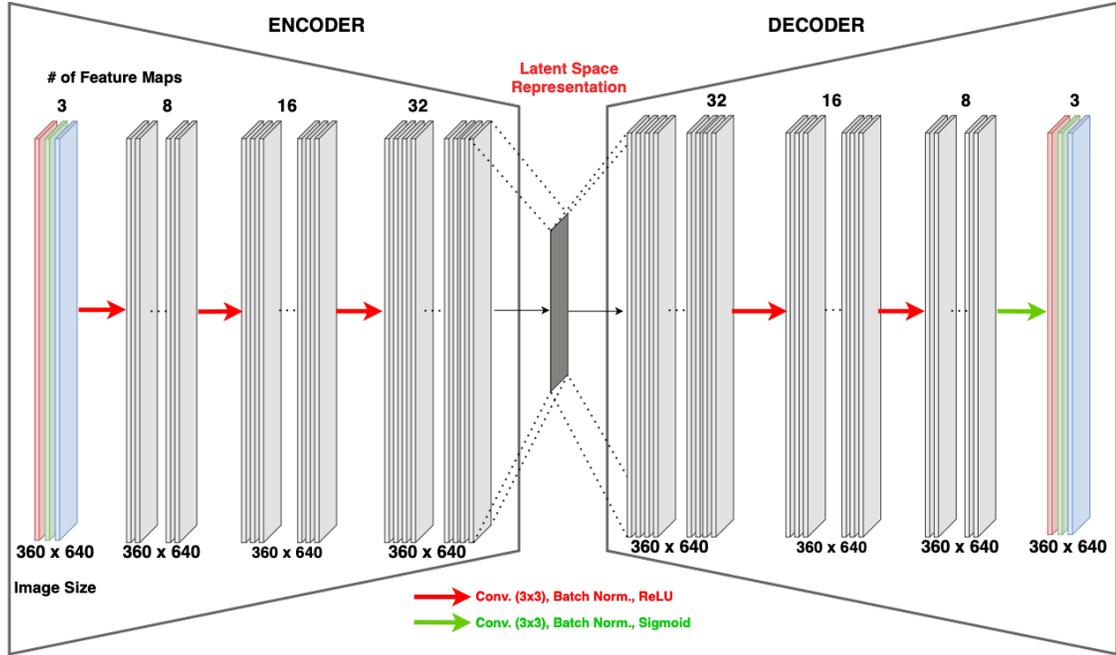


Figure 4.2: Modified Autoencoder architecture.

#### 4.3.4 RetinexNet

The structure of the RetinexNet can be seen in Figure 4.4.

The RetinexNet is a two stage architecture composed of extraction layers, followed by enhancement. In the first module, a Decom-Net is used to extract the image components of the illumination and reflectance. To achieve this decomposition the input RGB image is passed through 5 convolutional layers to produce at a maximum 64 feature maps. Each layer is passed through a ReLU activation function except for the output layer being passed through a sigmoid function. The first output of this model is the 3-channel reflectance tensor which helps identify features related to the color and textures of the scene. The second output is a single channel illumination tensor which can learn features that relate specifically to the brightness and contrast. These tensors are passed to the second module, Enhance-Net. This network concatenates the reflectance and illumination components as a 4-channel tensor before passing it through a series of convolutional layers. Opposing operations of strided convolutions and transposed convolutions work to compress and expand the spacial resolution of

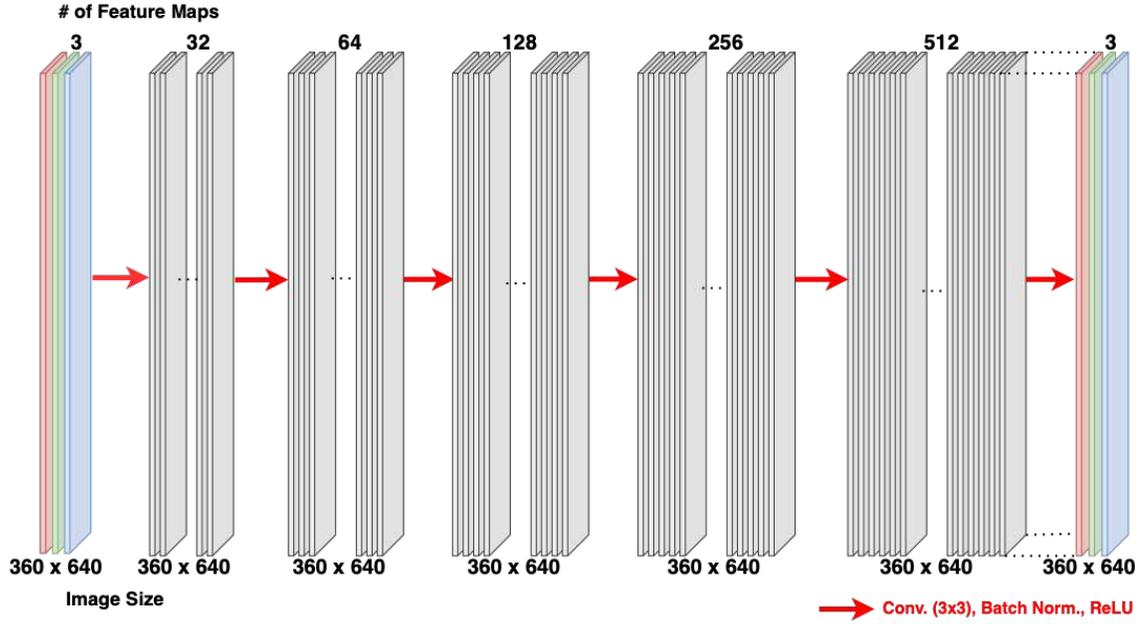


Figure 4.3: Feedforward architecture.

the feature maps to a maximum of 64. The output of this model is a single channel illumination image. This output is then concatenated with each convolutional layer upsampled to maintain a resolution of 360x640. These newly assembled feature maps pass through another convolutional layer to reduce the total from 361 to 64. This layer analyzes different levels of abstraction created through spacial manipulation of the feature maps before passing its output to a final convolutional layer and sigmoid function resulting in a single 360x640 image. This model recombines the reflectance and illumination components as follows to output an enhanced 3-channel RGB image.

$$\text{Image}_{\text{out}}(x, y) = I(x, y) * R(x, y) \quad (4.1)$$

### 4.3.5 U-Net

Finally, the U-net structure is similar to that of the Autoencoder, containing an encoder-decoder couplet. The visualization of this network can be seen in Figure 4.5.

This network is composed of four blocks which work cohesively to manipulate the spatial dimension of the image while fluctuating the number of feature maps. After the feature map resolution is reduced by two-thirds through max pooling operations,



Table 4.1: Hyperparameters for training CNN models.

Parameter	Value
Batch Size	5
Learning Rate	0.001
Epochs	30
Dropout Rate	0.5

the mean squared error (MSE) was used as the loss function and minimized by using the Adam optimization algorithm [114].

Addressing the implementation of the above algorithm,  $\theta$  is the set of model parameters to be optimized with  $y_i$  and  $\hat{y}_i$  being the ground truth and enhanced images at iteration  $i$ , respectively. As the values of  $\theta$  are minimized using Adam’s optimization procedure of moment and bias-corrected moment estimates. At each iteration  $i$ , the gradients of the MSE are calculated which are then used to update the values of  $\theta$ .

To assess the quality of the image enhancement four metrics were used which included Peak Signal to Noise Ratio (PSNR), Structural Similarity Index Metric (SSIM), Multi-Scale Structural Similarity Index (MS-SSIM), and Mean Squared Error.

PSNR is used to compare the strength of a signal before and after noise degradation [115]:

$$\text{PSNR} = 20 \cdot \log_{10} \left( \frac{\text{MAX}^2}{\sqrt{\text{MSE}}} \right) \quad (4.2)$$

where MSE is the mean squared error and MAX is the maximum value of a signal.

SSIM is a metric that aims to assess the similarity of two images. This measure this a combination of the independent analysis of luminance, contrast, and structure [116]:

Luminance:

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad (4.3)$$

Contrast:

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad (4.4)$$

Structural:

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \quad (4.5)$$

These metrics can be combined into the singular valuation of SSIM to evaluate the quality of similarity [116]:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (4.6)$$

where  $\mu_x$ ,  $\sigma_x^2$ ,  $\sigma_{xy}$ , and  $C_1$  are the mean, variance, covariance of  $x$  and  $y$ , and constant of image  $x$ , respectively. These parameters hold true for  $y$  which is an image of equal size to  $x$ .

Multi-Scale Structural Similarity Index (MS-SSIM) is an expanded form of SSIM where the influence of luminance, contrast, and structure can be controlled independently through the exponents of  $\alpha_j$ ,  $\beta_j$ , and  $\gamma_j$ :

$$\text{MS-SSIM}(x, y) = [l_m(x, y)]^{\alpha_M} \prod_{j=1}^M [c_j(x, y)]^{\beta_j} [s_j(x, y)]^{\gamma_j} \quad (4.7)$$

where  $j$  is the current scale and  $M$  is the total number of scales. The scales are defined through the application of a low pass filter to remove high frequency noise in addition to downsampling the image by a factor of two to aid in feature extraction. The scales begin at  $M = 1$  denoting the original image [116].

Finally, the MSE is a pixel to pixel evaluation method which analyzes the average squared difference across all of the pixels in two images to quantify the differences between them.

## 4.4 Results

The performance of the models was evaluated based on the convergence during model training, image quality, PSNR, SSIM, MS-SSIM, MSE, and model run times both onboard and offboard the UAV.

The training time allocated to each of the models was set based on the plateau of the MSE value during training. This indicates the model's inability to extract more information from the training data and converge further. For the Autoencoder, Modified Autoencoder, Feedforward, and RetinexNet models this was observed at 30 epochs. However, the U-Net which is a larger and more sophisticated model, and was allotted 100 epochs before convergence was observed. The results of the training are found in Figures 4.6, 4.7, 4.8, 4.9, and 4.10.

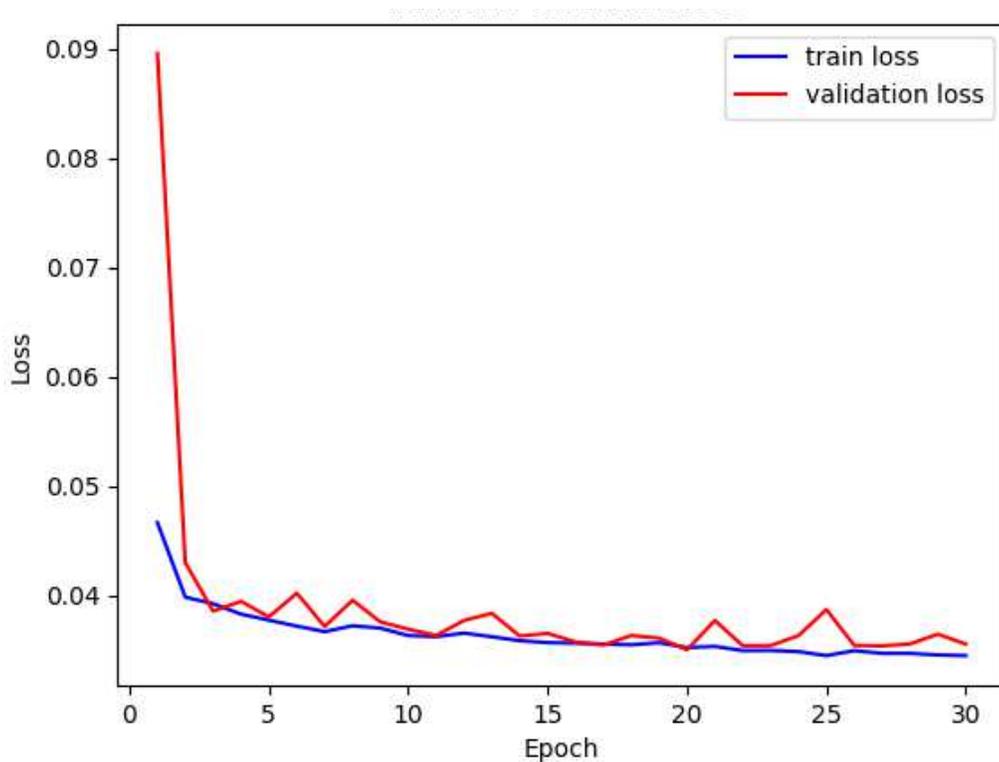


Figure 4.6: Training loss of the Autoencoder model.

The results of this training can be visualized in 4.11 which display a low light test image, its ground truth counterpart, and each of the models outputs.

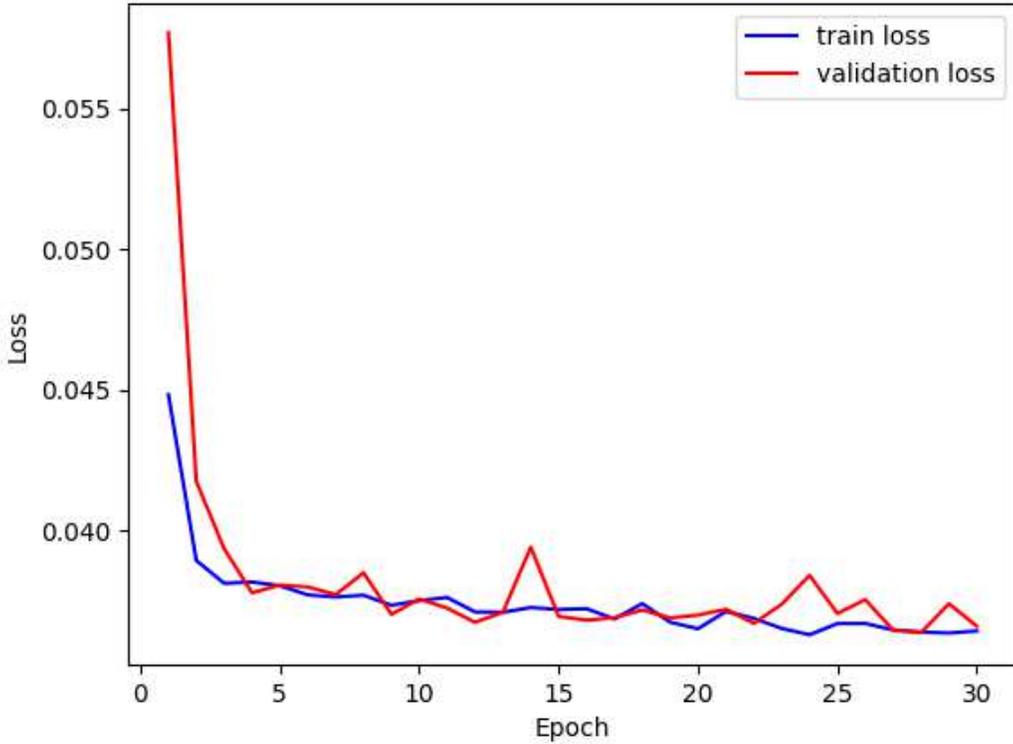


Figure 4.7: Training loss of the modified Autoencoder model.

Examining this result, it is seen that as the complexity of the model increases the more accurate colour information is extracted. In the order of quality, the U-Net, Feedforward, and RetinexNet models are able to extract the most color information from the dark images. The Autoencoder and Modified Autoencoder were ineffective in rendering accurate colour information. Although it is not observed in the still frames of Figure 4.11, the Autoencoder and U-Net architectures suffered from extensive noise and blur when processing video feed. In contrast, the Modified Autoencoder, Feedforward, and RetinexNet frameworks produced sharp and accurate images, particularly around object boundaries.

With the motivation for real-time UAV flight the computational demand of the models was investigated. Running in parallel to the flight stack, the models were clocked during flight maneuvers. The mean time over 500 cycles was recorded to obtain an accurate estimate. These results can be found in Table 4.2.

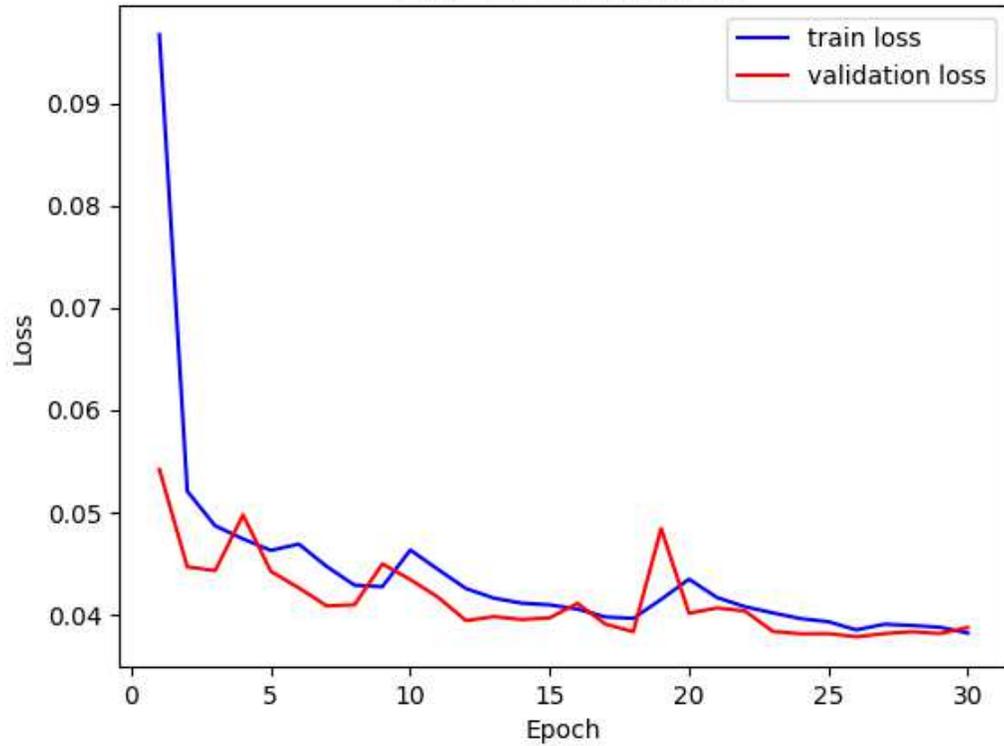


Figure 4.8: Training loss of the Feedforward model.

Table 4.2: Model runtime evaluation.

Model	Flight Computer		Desktop	
	(s)	(FPS)	(s)	(FPS)
Autoencoder	0.021	48	0.007	143
Mod. Auto.	0.056	18	0.006	167
Feedforward	1.518	0.66	0.034	29
RetinexNet	0.236	4.2	0.012	83
U-Net	0.356	2.8	0.014	71

As expected, the least sophisticated model obtained the fastest run time. The Autoencoder followed by the Modified Autoencoder, RetinexNet, U-Net, and Feedforward architectures is the order of operational speed. Table 4.2 highlights the drastic increase in processing time from the desktop to the onboard computer. The increase was 2x, 8.33x, 43.7x, 19.6x, 24.4x, for the Autoencoder, Modified Autoencoder, Feed-

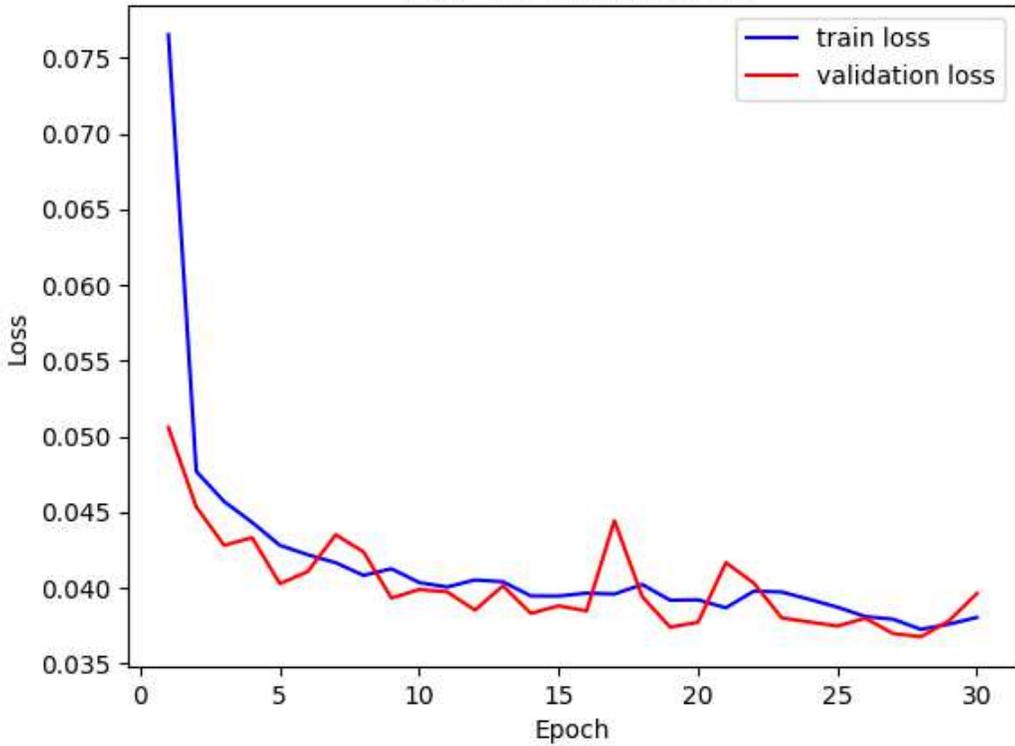


Figure 4.9: Training loss of the RetinexNet model.

forward, RetinexNet, and U-Net models, respectively. This reduction in performance results in large latency during video feed which directly impacts the operator’s ability to fly and visualize the environment. To establish a comparison cinematography publishes video feed at a rate of 24 fps [117]. Although cinematic performance is not required the models which mitigate the observed latency to approach or achieve this baseline are the Autoencoder and Modified Autoencoder.

Next, the models were evaluated by quantitatively comparing the output image with that of the ground truth. The results of this comparison are found in Table 4.3.

The model which produces the most accurate enhancement is U-Net. This model can reconstruct colour information which is sparsely available in the training images. Containing notable advantages this model is the deepest containing 512 feature maps at its maximum while preserving high dimensional details across low dimensional maps through skip connections. The remaining four models produced similar results

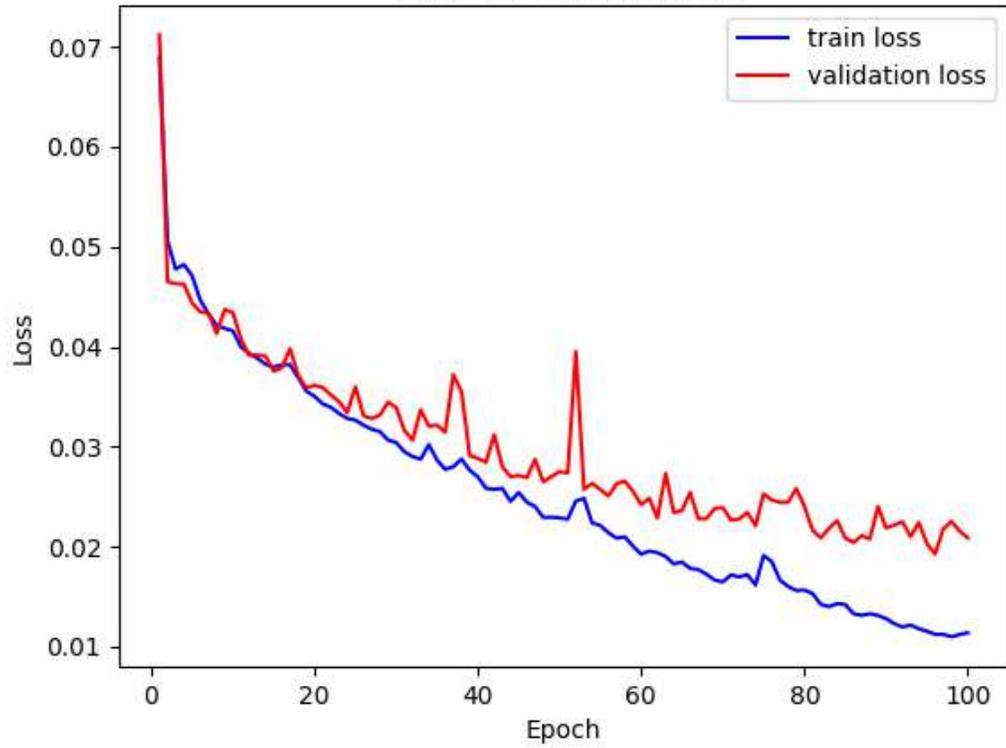


Figure 4.10: Training loss of the U-Net model.

Table 4.3: Model accuracy evaluation.

Model	PSNR	SSIM	MS-SSIM	MSE
Autoencoder	14.544	0.482	0.582	0.036
Mod. Auto.	14.393	0.539	0.573	0.037
Feedforward	14.165	0.535	0.584	0.039
RetinexNet	14.058	0.543	0.566	0.040
U-Net	<b>16.833</b>	<b>0.559</b>	<b>0.714</b>	<b>0.021</b>

to one another despite having noticeable visual differences. These models contained a maximum of 3.4% difference between the PSNR values. Similarly, a maximum of 11.9% , 3.1% and 10.5% for SSIM, MS-SSIM, and MSE, respectively.

## 4.5 Summary of Results

This Chapter aimed to evaluate CNN architectures for the task of low light image enhancement. The study addressed five variations of networks: Autoencoder, Modified Autoencoder, Feedforward, RetinexNet, and U-Net. These models were trained using a custom 250 image database with an 80/20 training-testing split. These models were assessed on their visual similarity to the ground truth, accuracy of enhancement through the metrics of PSNR, SSIM, MS-SSIM, and MSE, and by their computational demand. For enhancement of static images, the best performing model was the U-Net which achieved PSNR, SSIM, MS-SSIM, and MSE values of 16.8 dB, 0.559, 0.714, and 0.021, respectively. For visual quality during streaming the best performing model was the Modified Autoencoder. This is because the other models suffered from deficiencies. Blur and noise were prominent in the outputs of U-Net and Autoencoder and latency in the frame rates of the Feedforward, RetinexNet, and U-Net models. Computationally, the fastest executing models deployed on the Jetson Xavier NX were the Autoencoder (0.021s), Modified Autoencoder (0.056s), RetinexNet (0.236s), U-Net (0.356s), and Feedforward (1.512s). However, when deployed onboard the flight computer the only computationally feasible models were the Autoencoder and Modified Autoencoder. This assessment was performed during manual flight tests to ensure accurate readings when nominal computational loads for standard flight were applied. Notably, the two models which suffered heavily from blur during implementation were the two models which reduced the resolutions of the feature maps. Therefore, for fast and accurate low light image enhancement in UAV applications a Modified Autoencoder structure with consistent spacial resolution between layers is recommended for future considerations.

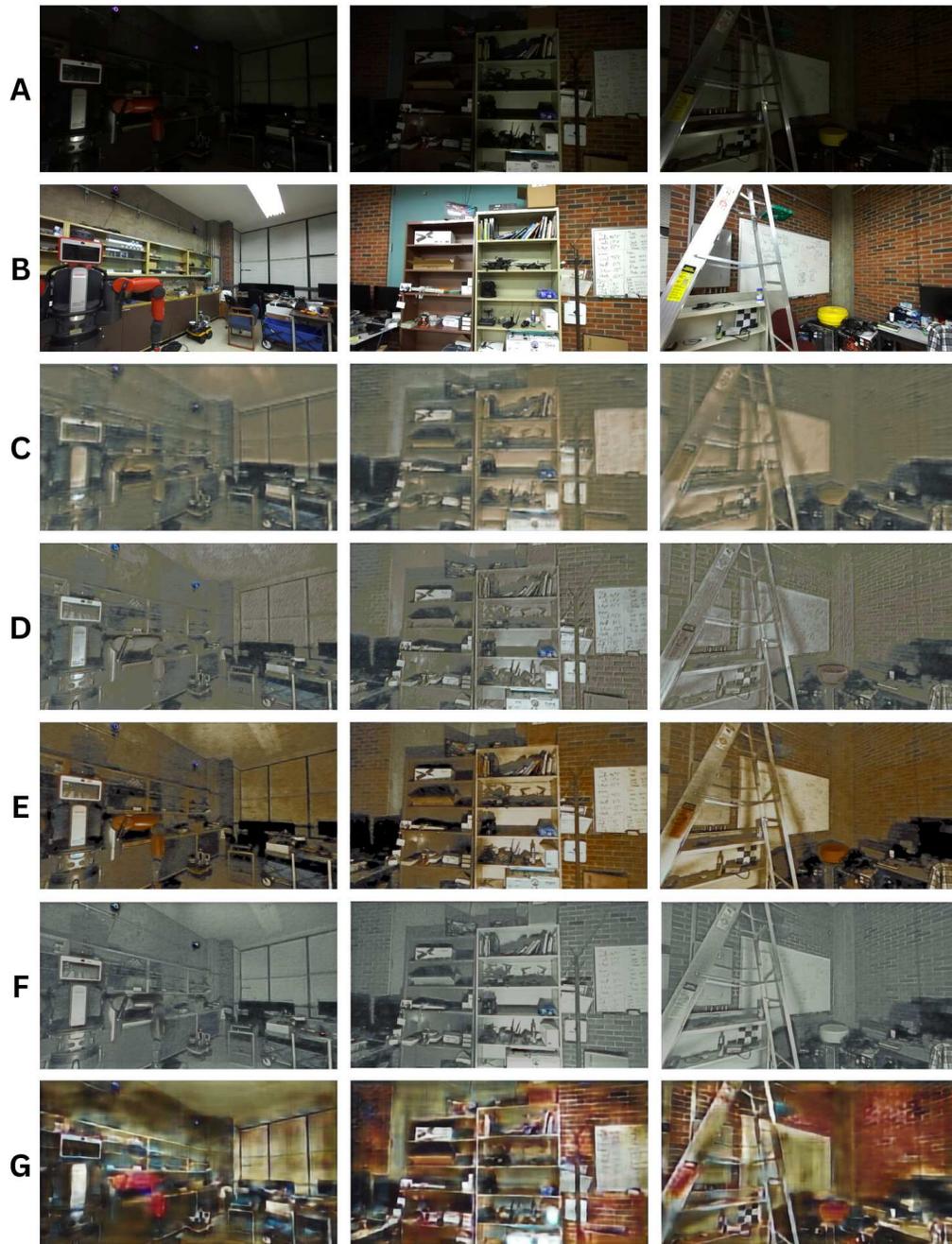


Figure 4.11: The images are A) Test image, B) Ground truth, C) Autoencoder, D) Modified Autoencoder, E) Feedforward, F) RetinexNet, G) U-Net.

# Chapter 5

## Evolution and Evaluation of Motion Planners

### 5.1 Introduction

Unmanned aerial vehicles (UAVs) provide support for tasks such as search and rescue [118], surveillance [119], delivery [120], and inspection [121]. Traditionally these tasks have been dependent on human operation, but as the performance of commercial and consumer UAVs accelerates supervision is becoming less of a requirement. One obstacle to autonomy is the reliance on global positioning systems (GPS) for navigation and control [122]. This dependency becomes problematic when the GPS signal is interrupted or blocked due to an artificial or natural obstruction. These inhibitions occur in urban environments such as buildings, tunnels, and mines and in rural environments such as caves, canyons, mountains, and other topology [123], [124], [125]. When the GPS signal is impeded, UAVs can lose crucial mission dependent information such as their position and velocity [126]. This loss of kinematic data inhibits localization producing significant challenges for the safe operation of UAVs in sensory constrained situations.

Subterranean environments sustain GPS-denied conditions making exploration a complex task. This geography is characterized through uneven terrain, spacial restrictions, darkness, and reduced abiotic variability [127]. Proposed as a universal solution for autonomous exploration and navigation are two prototype UAVs. These

evolutionary designs use differing sensor-fusion techniques to accurately accumulate spacial visual information in these environments. Along with differentiation in environmental perception these UAVs present different strategies for local and global path planning in combination with object detection and avoidance. The hexacopter platform achieved successes in autonomous underground flight through high performing sensors and a custom software stack. An optimized local path planner introduced as Non-linear Model Predictive Horizon (NMPH) is an adaptive learning architecture based on Deep Reinforcement Learning for computationally efficient trajectory generation, dynamic obstacle avoidance, and smooth optimized flight paths [25]. Proposed as a global motion planner is the fusion of the NMPH framework with a volumetric graph-based planner for adaptation to perceptual changes.

A quadcopter design for GPS-denied subterranean exploration followed the hexacopter prototype. This iteration used lower cost sensors in addition to a custom software stack. This system made use of the Fast UAV Exploration (FUEL) framework as an end-to-end system solution for autonomous navigation [128]. As a global planner this architecture exploits a frontier-based exploration methodology through real-time voxel-grid mapping [128]. The local path planner is designed using a hybrid A\* search algorithm optimized through B-splines to generate feasible kinodynamic pathways [129].

Each of these methods provide different, yet effective solutions for subterranean navigation capable of handling the non-linear dynamics of UAV flight. In addition to the system designs we explore the challenges encountered and the lessons learned.

The evolution of our work will be presented chronologically as we advanced through this challenge.

## 5.2 Software Implementation

As the prototype quadcopter was designed, it retained many hardware commonalities as the objective of autonomous GPS-denied flight remained a common goal. However,

where the two evolutions differed is in their path planning strategies. The software architecture can be viewed in Figure 5.1

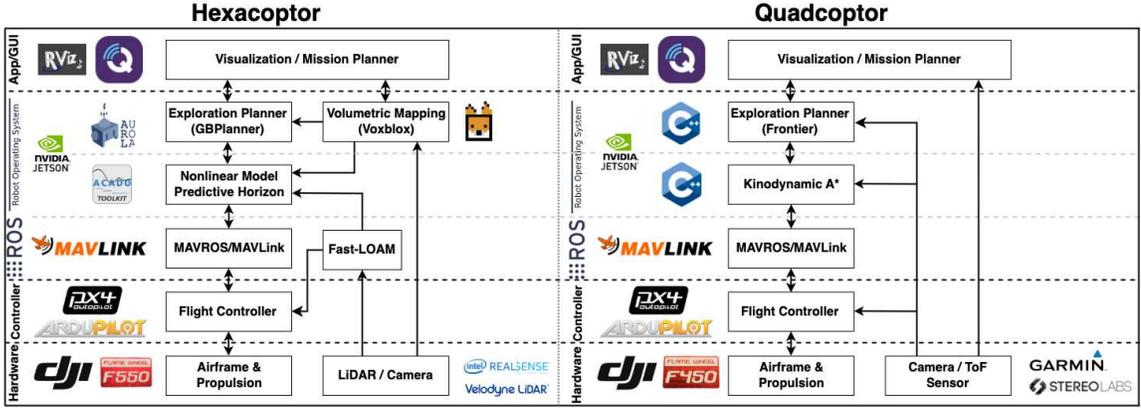


Figure 5.1: System architecture and information transmission.

The hexacopter implements the FLOAM algorithm for localization and mapping. Utilized in both local and global path planning this information is processed by the NMPH and GBPlanner nodes for mission planning. Streamlining this process is the quadcopter subsystems which uses VIO and depth images to optimize its global and local path planner embedded within the FUEL frameworks.

## 5.2.1 Hexacopter Navigation

In [26], we developed and implemented a modular global motion planning framework for the hexacopter, which aims to generate optimal paths in unexplored, dynamic, and GPS-denied environments. Our approach integrates a graph-based exploration technique [130] with our proposed optimization-based local path planning algorithm, which is called Nonlinear Model Predictive Horizon (NMPH) [25, 27, 28]. NMPH considers the vehicle’s dynamics and accounts for dynamic obstacles. This integration yields optimal and robust paths that enable the hexacopter to navigate through complex terrains with ease. Figure 5.2 illustrates the overall architecture of the global motion planner implemented on the hexacopter vehicle.

The global motion planner requires a map that contains the physical representation

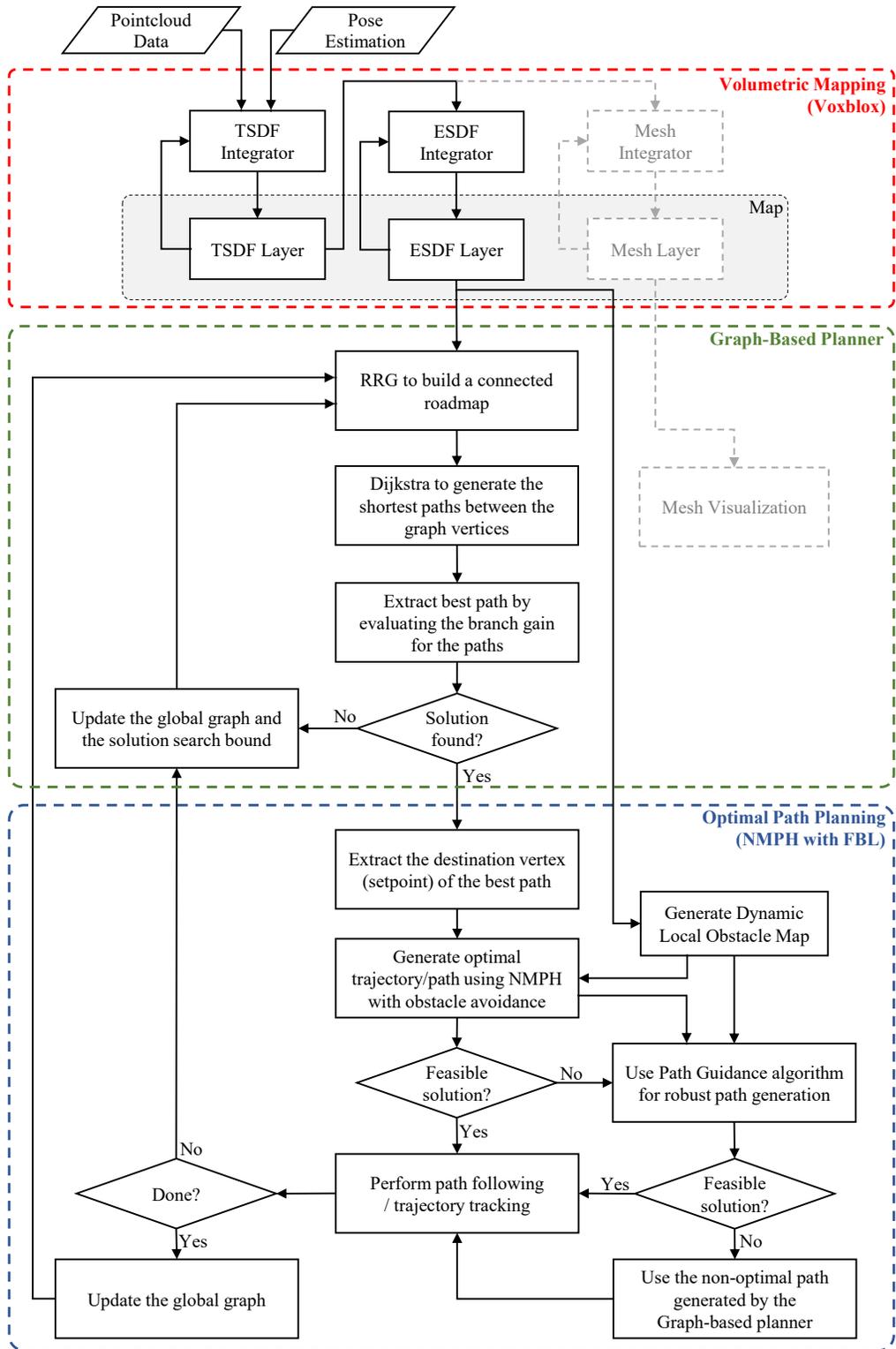


Figure 5.2: Hexacopter global motion planner architecture.

of the environment, which includes both static and dynamic objects. In our work, we employ volumetric mapping for this purpose, due to its computational efficiency, ease of visualization, ability to be incrementally constructed, and provision of the voxel grid structure necessary for planning.

The graph-based planning approach uses the Rapidly-exploring Random Graph (RRG) algorithm to construct a connected roadmap graph and utilizes the Dijkstra search algorithm to determine the best path within the graph.

The primary objective of the graph-based approach is to direct the vehicle towards unexplored areas in the environment and provide endpoints for the local path planner. Subsequently, the NMPH local path planning method generates optimal paths to these endpoints while avoiding both static and moving obstacles. The different scenarios for finding a viable path are depicted in Figure 5.2.

The reference trajectory computed by NMPH is sent to the vehicle’s controller for tracking purposes. As the drone moves, the NMPH continuously updates the reference trajectory based on the system and environment states. The exploration mission continues until either the environment is fully explored, or the operator interrupts the mission.

Details about the NMPH algorithm and the modular global motion planner can be found in our recent works [26–28]. These publications provide comprehensive insights into the methodologies and implementations utilized in our research.

## 5.2.2 Quadcopter Navigation

To achieve fully autonomous, GPS-denied navigation onboard the quadcopter a sophisticated motion planner was adapted and integrated within the UAV’s architecture. The FUEL algorithm proposed by Zhou et al. includes both frontier driven exploration and local optimized trajectory generation produced in the authors earlier work [129]. The framework of this algorithm is presented in Figure 5.3. Represented as a voxel grid map the spacial visual information of the environment is encoded to the

virtual map using a Euclidean Signed Distance Field which can derive the distance to objects both in relation to one another and to the vehicles position. Through this deployment voxels which are virtual cubic occupants of 3D space are stacked as opaque clusters for occupied areas, translucent clusters for frontiers, and are transparent for unoccupied space.

Motivating the UAV to investigate the unknown boundaries/space of the environment is the frontier-based exploration [131]. This module marks frontiers as groups of unexplored voxels next to those previously explored. After grouping these segments of unknown space, a centroid is traditionally calculated as a navigational waypoint. However, within FUEL this suboptimal solution is addressed by uniformly sampling viewpoints in an attempt to efficiently span the unexplored region by traversing subsections within the frontier space.

The global tour to the identified frontier clusters is formulated as a modified asymmetric travelling salesman problem [132]. By removing the homing requirement and the penalties incurred during movement through previously explored spaces to new clusters the traditionally closed-loop problem can be reconfigured as an asymmetric open-loop. This global tour continually motivates the vehicle to navigate a collision free path to the areas that result in the highest coverage. However, penalties integrated within the cost function are applied to maneuvers resulting in rapid yaw and velocity changes during transitions. This regulated global tour is continually updated as new frontiers are discovered resulting in unbounded exploration until all clusters are covered.

Efficiently navigating static and dynamic obstacles within a local area requires a precise representation of the virtual environment. Within the local environment motion primitives are generated to establish elementary movements and trajectories the quadcopter can follow. Each of the motion primitives are assessed for their ability to avoid obstacles and maintain dynamic feasibility. These primitives are updated as the map expands and evolves through the UAV exploration. A single primitive

is generated through discretization of control inputs over a constant time duration containing information of position and velocity. In total three independent 1-D polynomials are established. Using these primitives more sophisticated trajectories can be assembled to move from the current state to the goal state. To assemble these pathways a hybrid A\* search algorithm is used. An initial cost function which locates the optimal pathway to the end-goal is established in addition to a heuristic function to account for the current and end states position and velocity to minimize the time to the end goal. By balancing these functions, the optimal path which takes the shortest time is found.

Since the connection of the primitives by means of the hybrid A\* algorithm are not inherently smooth the concept of B-Splines is introduced. The control points the UAV is directed to follow can be adjusted to ensure smooth continuity between one another. This encourages dynamic stability and optimized trajectory transition. With the addition of the convex hull property of B-Splines the safety of the proposed trajectory can be evaluated. Proposed as a safeguard to constrain the velocity and acceleration of the control points in addition to establishing a buffer to occupied voxels the convex hull can ensure dynamically feasible pathways. If an unacceptable pathway is detected the algorithm can replan and produce a secondary trajectory. However, if no additional pathways are available the algorithm will terminate leaving the quadcopter in the last achieved state.

## **5.3 Experimental Flight Testing**

### **5.3.1 Real-time Flight Test Results for the Hexacopter Vehicle**

We utilized a DJI FlameWheel F550 hexacopter that is constructed and equipped with specialized instruments. The primary objective is to explore unknown environments by employing the global motion planner presented in Section 5.2.1. The detailed hardware setup and for the hexacopter vehicle is described in detail in Chapter 3.

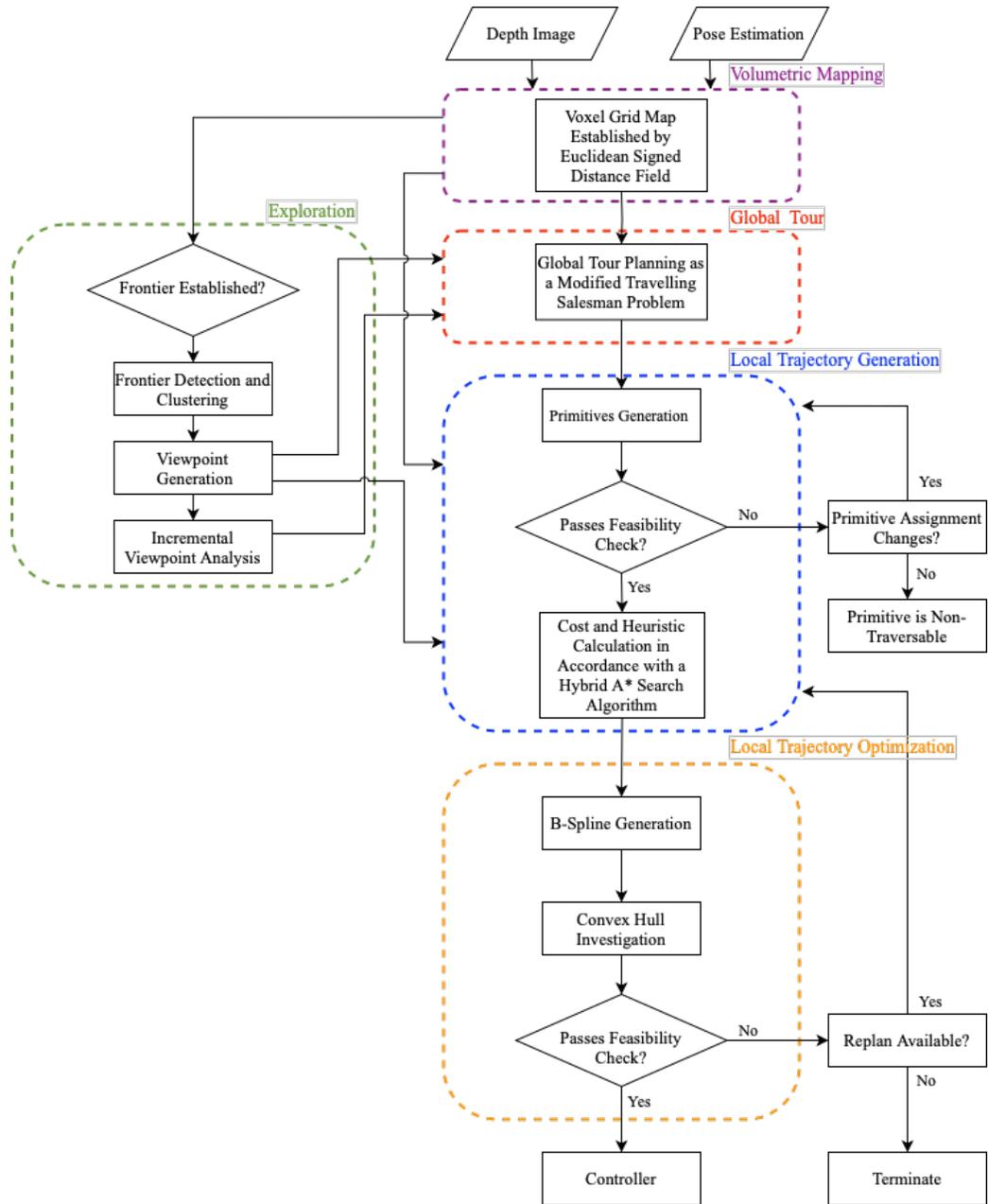


Figure 5.3: Quadcopter global motion planner architecture.

The flight test aims to evaluate the exploration performance of the global motion planner’s capabilities and the functionality of the NMPH local trajectory planning within the global planner. The test is conducted within an underground parking area, as illustrated in upper image in Figure 5.7. The exploration performance of the system was tested as the global motion planner generated several terminal setpoints to

survey an unexplored area. Our proposed NMPH algorithm provided local reference trajectories between the current vehicle location and the next terminal setpoint at a rate of 200 Hz to ensure a smooth flight.

The start and the end (home position) of the exploration mission are further depicted in Figure 5.7 in the bottom left and right images, respectively.

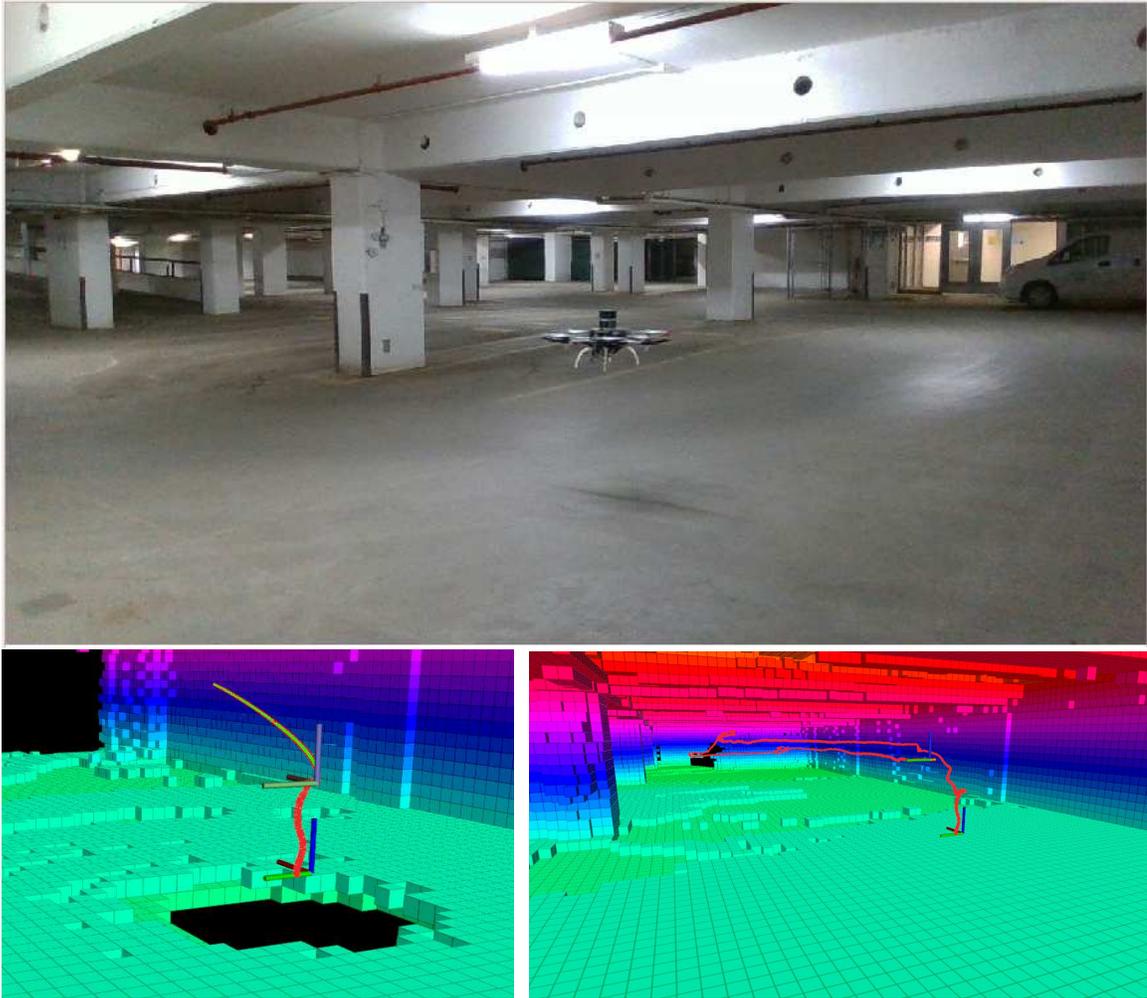


Figure 5.4: Wide view of hexacopter navigation (top), reference trajectory to setpoint (bottom left), 3D voxel map construction after experimental test flight (bottom right).

It is worth noting that the NMPH planning algorithm consistently generated a continuous trajectory toward the setpoints computed by the graph-based planner. Also, the vehicle showed very good tracking performance in following the generated trajectories. Moreover, Figure 5.5 presents various perspectives showing the tracked

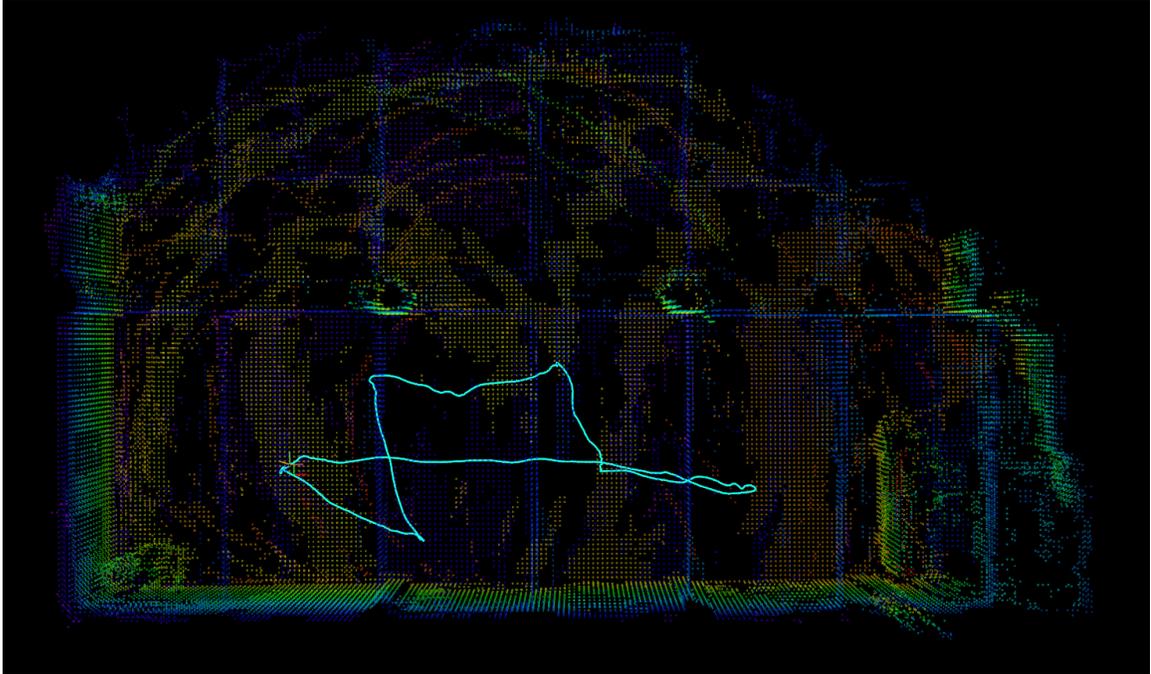


Figure 5.5: Top perspective of the tracked trajectories by the hexacopter system in the underground space using our presented global motion planner.

trajectories of the hexacopter vehicle upon successful completion of the exploration mission.

The volume traversed by the vehicle spans approximately  $486\text{m}^3$ , with a total flight duration of 3 minutes. Following the generation of 8 setpoints by the global motion planner, the drone was commanded to return to its home position.

Overall, the navigation capabilities and reliability of our system, enhanced with the proposed approaches, were thoroughly tested and validated in an unknown and GPS-denied environment. The results demonstrated that the system can achieve autonomous navigation and exploration even in confined spaces.

To evaluate the trajectory planning performance in the presence of wind disturbance (approximately  $15\text{ km/h}$ ), a hardware flight test was conducted in an outdoor environment. The results of the outdoor flight test can be seen in Figure 5.6. Despite the presence of the wind disturbance, the vehicle exhibited seamless navigation capabilities between the generated setpoints, similar to the earlier tests.

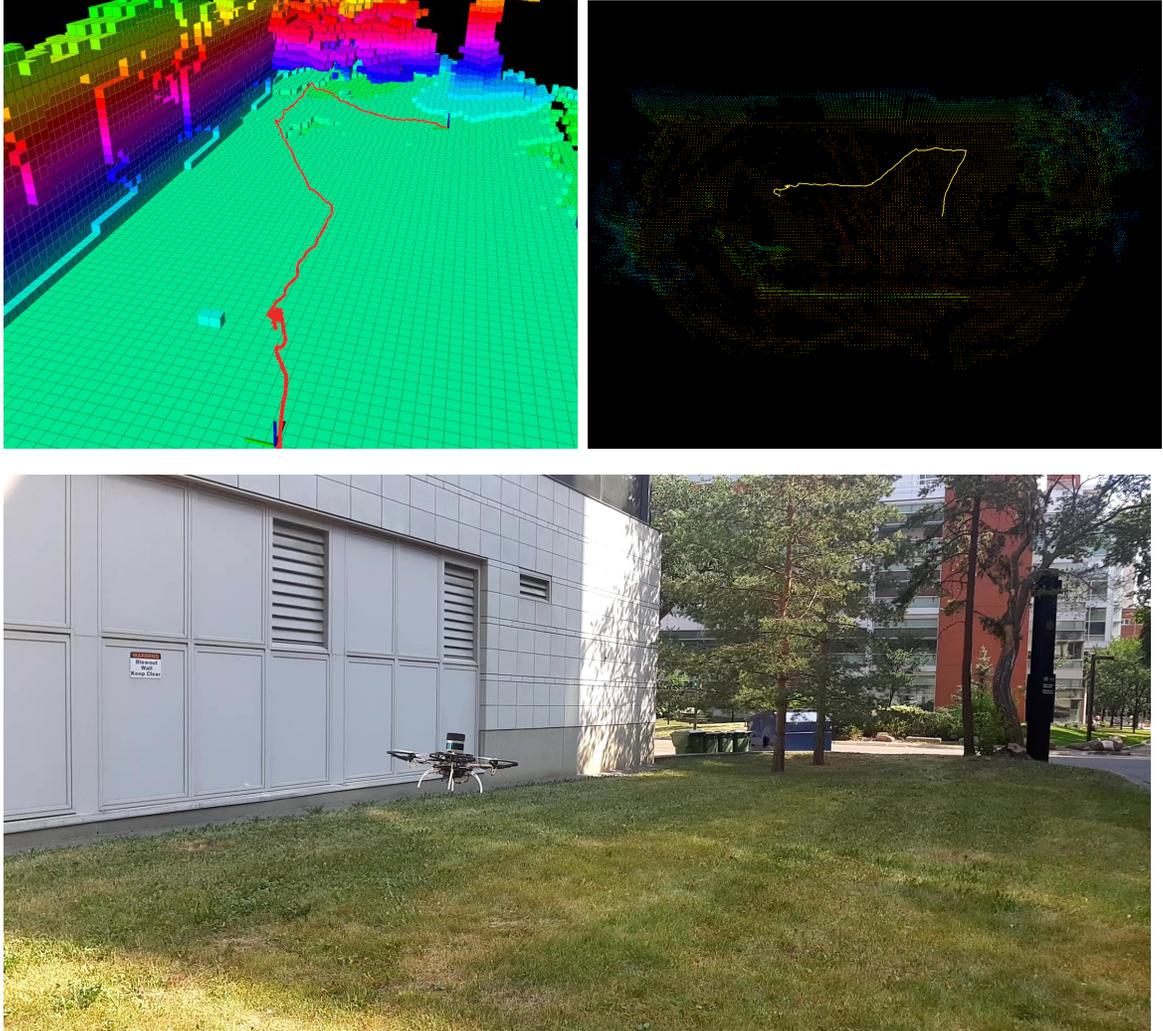


Figure 5.6: Flight test of the hexacopter employing the global motion planner in an outdoor environment with a 15 km/h wind speed.

### 5.3.2 Real-time Flight Test Results for the Quadcopter Vehicle

Validating our work and demonstrating an effective integration of FUEL into our quadcopter are the underground experimental flight tests performed. From the left image in Figure 5.7 we can see the initial frontiers which are generated during start-up. Because little information is known of the perceived surroundings large frontiers are visible. The blue, red, and purple frontiers are one large unexplored region. However, this region is divided into colorized subsections using a principal component analysis to guide the exploration more efficiently. The right image of Figure 5.7 is a depiction of the initial local trajectory toward an unexplored frontier. Noticeably smooth and uniform the A\* pathway, smoothed by the B-spline, can avoid the imminent obstruction to establish an efficient path toward the goal.

Figure 5.7 demonstrates an underground flight test performed in the second underground level of a parkade. This concrete encapsulated environment eliminates any GPS transmission demonstrating a practical test environment for the proposed methods.

This figure presents multiple viewpoints of the UAV in addition to the real-time mapping performed as it traverses the unknown environment. Hard constraints on the velocity and acceleration were set at 0.3 m/s and 0.5 m/s<sup>2</sup>. This restriction was implemented due to the bottleneck in the depth image processing. Limited to a maximum of 8 FPS using the neural depth functionality the environmental perception was slower than the processing abilities of the navigational algorithms introduced. Lowering the depth image quality was infeasible as the information transmitted suffered substantially, particularly when faced with monochrome and textureless surfaces. This flight test demonstrates our vehicle’s ability to autonomously map and navigate an unknown environment in real-time under GPS-denied conditions.

Travelling for a total time of 3 minutes and 39 seconds, Figure 5.8 exemplifies the quadcopter’s ability to perform successful exploratory missions. During this time the

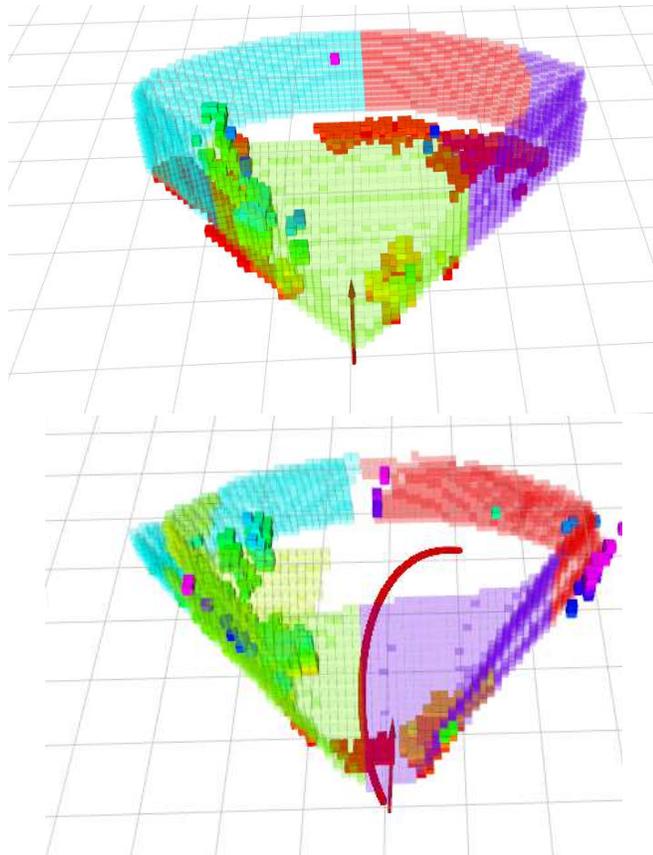


Figure 5.7: Frontier clusters (left) and local trajectory generation (right).

vehicle traversed through open regions, around corners, and narrow passages for a total explored volume of  $518\text{m}^3$  which was approximated in figure 5.9.

However, this flight test was not the limit of the quadcopter as seen in Figure 5.10 where many unexplored frontiers were identified by the aerial vehicle that were not yet traversed.

The UAV performance was extensively tested in both constrained and open environments for a maximum exploration time frame of 11 minutes with the current hardware configuration.

### 5.3.3 Rotor Arm Redesign

By investigating the path traversed by the quadcopter in Figure 5.11 it is understood the FUEL algorithm encourages aggressive yaw action to accommodate for the limited

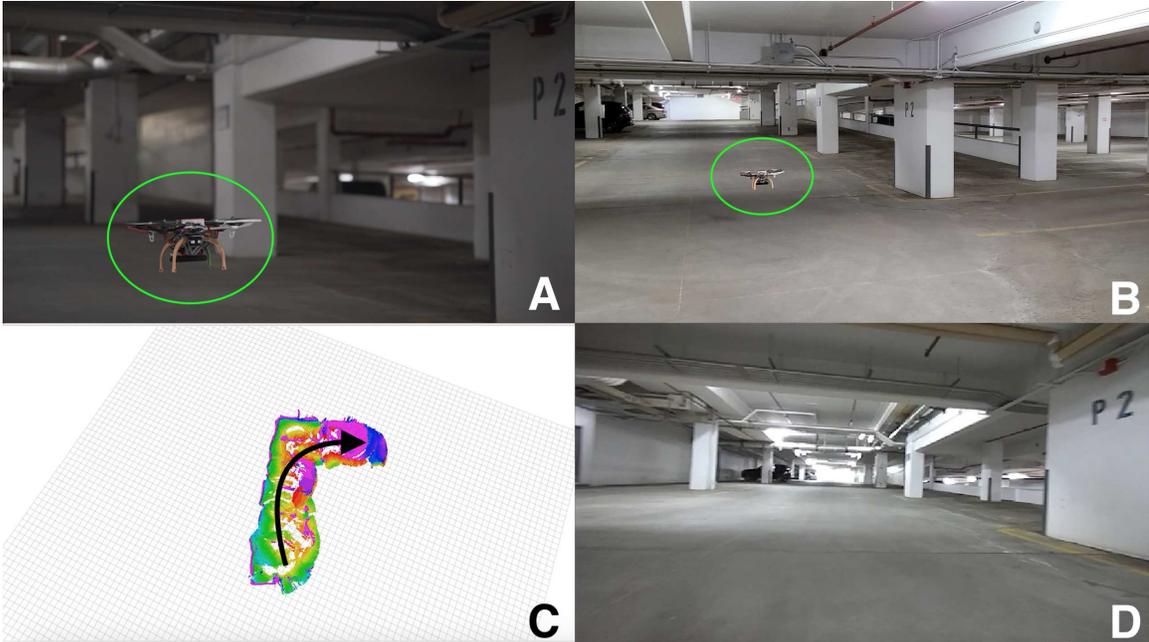


Figure 5.8: A) Standard view of quadcopter trajectory. B) Wide view of quadcopter trajectory. C) Real-time voxel grid mapping. D) Onboard quadcopter camera POV.

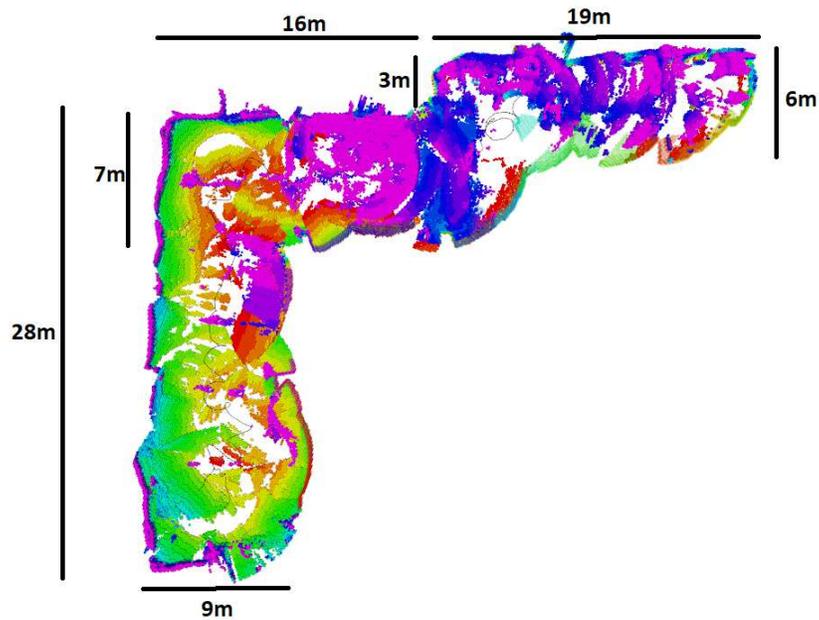


Figure 5.9: Total estimated volume travelled by the quadcopter with a 3 meter tall ceiling height.

FOV of the stereo camera.

A yaw maneuver requires an aggressive counter-torque imbalance. To achieve this

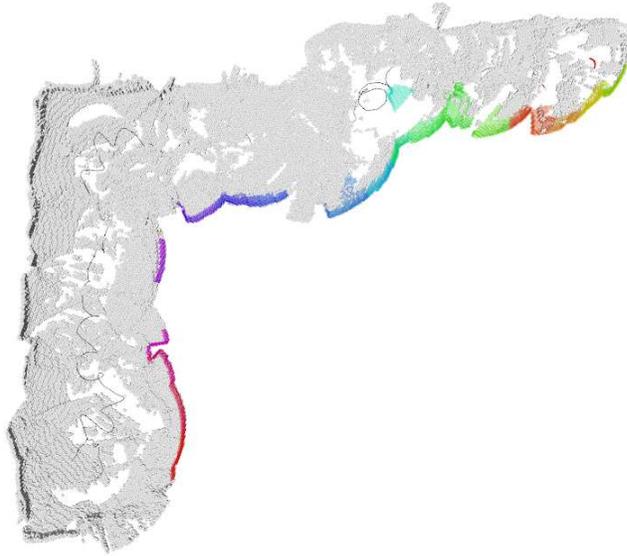


Figure 5.10: Unexplored identified frontiers during experimental flight test.



Figure 5.11: 3D perspective of the tracked trajectories by the quadcopter system in the underground space using the adapted exploration algorithm.

the angular velocity of two opposing motors are slowed to induce a rotation. However, when attempting to yaw the quadcopter would experience a drop in altitude indicating the vehicle could not produce enough thrust to perform this maneuver. Through

experimental testing the thrust to weight ratio was determined to be 1.7:1; 15% less than the recommended ratio of 2:1 [133]. This ratio demonstrates the limitations of the current design. As a solution the geometry of the UAV was modified to reallocate the available thrust. A twist was introduced into each of the arms of the quadcopter as seen in the prototypes of Figures 5.12a and 5.12b. This twist redistributes vertical thrust disproportionately to increase the torque of the vehicle with only a slight decrease in available thrust.



(a) Tilted mounts at 5 10 15 and 20 top view.



(b) Tilted mounts at 5 10 15 and 20 front view.

Figure 5.12: Tilted quadcopter arms.

The twist of each arm can be calculated as a matrix representing the total effects on the body. The free-body diagram in Figure 5.13 represents the orientation of the designed UAV. In this diagram the positive  $X$  direction is at  $45^\circ$  from the front two motors of the X-frame quadcopter.

For simplicity, we can first examine the force and torques applied to the rigid body in a +-orientation where the body frame axis can be aligned along the arms to derive Equation 5.1.

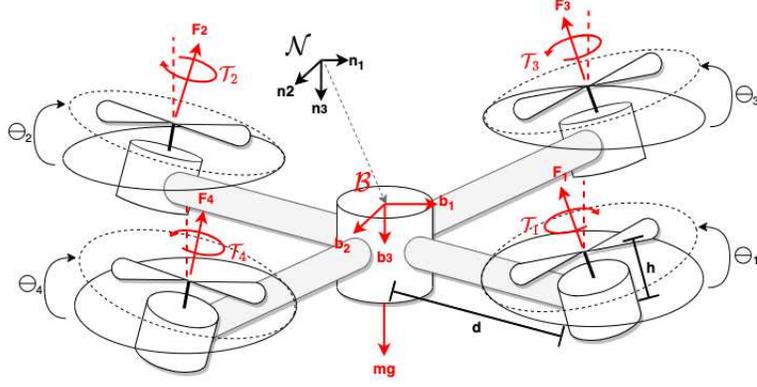


Figure 5.13: Free body diagram of tilted quadcopter arms.

$$\sum F_{b,+} = \begin{bmatrix} 0 \\ F_1 \sin(\alpha) \\ -F_1 \cos(\alpha) \\ F_1 \sin(\alpha)h \\ F_1 \cos(\alpha)d - \tau_1 \sin(\alpha) \\ F_1 \sin(\alpha)d + \tau_1 \cos(\alpha) \end{bmatrix} + \begin{bmatrix} 0 \\ -F_2 \sin(\alpha) \\ -F_2 \cos(\alpha) \\ -F_2 \sin(\alpha)h \\ -F_2 \cos(\alpha)d - \tau_2 \sin(\alpha) \\ F_2 \sin(\alpha)d - \tau_2 \cos(\alpha) \end{bmatrix} + \begin{bmatrix} -F_3 \sin(\alpha) \\ 0 \\ -F_3 \cos(\alpha) \\ F_3 \cos(\alpha)d + \tau_3 \sin(\alpha) \\ F_3 \sin(\alpha)h \\ -F_3 \sin(\alpha)d - \tau_3 \cos(\alpha) \end{bmatrix} + \begin{bmatrix} F_4 \sin(\alpha) \\ 0 \\ -F_4 \cos(\alpha) \\ -F_4 \cos(\alpha)d - \tau_4 \sin(\alpha) \\ -F_4 \sin(\alpha)h \\ -F_4 \sin(\alpha)d + \tau_4 \cos(\alpha) \end{bmatrix} + \begin{bmatrix} R^T & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ mg \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (5.1)$$

where  $h$  is the height of the propeller from the base of the mounts and  $d$  is the distance from the center of gravity of the point of thrust generation. However, since  $h \ll d$  it can be presumed to produce negligible torque resulting, in Equation 5.2.

$$\sum F_{b,+} = \begin{bmatrix} 0 \\ F_1 \sin(\alpha) \\ -F_1 \cos(\alpha) \\ 0 \\ F_1 \cos(\alpha)d - \tau_1 \sin(\alpha) \\ F_1 \sin(\alpha)d + \tau_1 \cos(\alpha) \end{bmatrix} + \begin{bmatrix} 0 \\ -F_2 \sin(\alpha) \\ -F_2 \cos(\alpha) \\ 0 \\ -F_2 \cos(\alpha)d - \tau_2 \sin(\alpha) \\ F_2 \sin(\alpha)d - \tau_2 \cos(\alpha) \end{bmatrix} + \begin{bmatrix} -F_3 \sin(\alpha) \\ 0 \\ -F_3 \cos(\alpha) \\ F_3 \cos(\alpha)d + \tau_3 \sin(\alpha) \\ 0 \\ -F_3 \sin(\alpha)d - \tau_3 \cos(\alpha) \end{bmatrix} + \begin{bmatrix} F_4 \sin(\alpha) \\ 0 \\ -F_4 \cos(\alpha) \\ -F_4 \cos(\alpha)d - \tau_4 \sin(\alpha) \\ 0 \\ -F_4 \sin(\alpha)d + \tau_4 \cos(\alpha) \end{bmatrix} + \begin{bmatrix} R^T & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ mg \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (5.2)$$

Because the force and torque equations were formulated with a +-airframe configuration, a  $45^\circ$  rotation about the Z-axis must be applied to the rigid body. This rotation matrix is shown in Equation 5.3.

$$R_z = \begin{bmatrix} \cos(\pi/4) & -\sin(\pi/4) & 0 \\ \sin(\pi/4) & \cos(\pi/4) & 0 \\ 0 & 0 & 1 \end{bmatrix} \longrightarrow R_{b,z} = \begin{bmatrix} R_z & 0 \\ 0 & R_z \end{bmatrix} \quad (5.3)$$

Equation 5.4 applies this rotation to the forces and torques produced by the X-frame quadcopter.

$$\sum F_{b,X} = R_{b,z} F_{b,+} \quad (5.4)$$

The resulting forces and torques experienced by the quadcopter are summarized in Equation 5.5.

$$\sum F_{b,X} = \begin{bmatrix} -\cos(\beta)(F_3 \sin(\alpha) - F_4 \sin(\alpha)) - \sin(\beta)(F_1 \sin(\alpha) - F_2 \sin(\alpha)) \\ \cos(\beta)(F_1 \sin(\alpha) - F_2 \sin(\alpha)) - \sin(\beta)(F_3 \sin(\alpha) - F_4 \sin(\alpha)) \\ -F_1 \cos(\alpha) - F_2 \cos(\alpha) - F_3 \cos(\alpha) - F_4 \cos(\alpha) + mg \\ \cos(\beta)(\tau_3 \sin(\alpha) + \tau_4 \sin(\alpha) + dF_3 \cos(\alpha) - dF_4 \cos(\alpha)) + \sin(\beta)(\tau_1 \sin(\alpha) + \tau_2 \sin(\alpha) - dF_1 \cos(\alpha) + dF_2 \cos(\alpha)) \\ \sin(\beta)(\tau_3 \sin(\alpha) + \tau_4 \sin(\alpha) + dF_3 \cos(\alpha) - dF_4 \cos(\alpha)) - \cos(\beta)(\tau_1 \sin(\alpha) + \tau_2 \sin(\alpha) - dF_1 \cos(\alpha) + dF_2 \cos(\alpha)) \\ \tau_1 \cos(\alpha) - \tau_2 \cos(\alpha) - \tau_3 \cos(\alpha) + \tau_4 \cos(\alpha) + dF_1 \sin(\alpha) + dF_2 \sin(\alpha) - dF_3 \sin(\alpha) - dF_4 \sin(\alpha) \end{bmatrix} \quad (5.5)$$

The forces and torques were measured experimentally using the force-torque sensor embedded within the end-effector of the Baxter robot by Rethink Robotics. The experimental setup is presented in Figure 5.14.

Where the quadcopter is rigidly secured to the base of the end-effector; level in the X-Y plane. Commanded to 100% of the input the UAV attempted to increase altitude and yaw in two independent tests during which three trials were averaged. The force feedback of the Baxter robot resisted the UAV maneuvers, maintaining a static position during testing. For each of the arm tilt angles three trials were performed measuring the forces and torques in the Z-axis with an average of the three readings during steady state being presented in Figure 5.15

As the tilt angle is increased the force available to generate vertical thrust in the Z-axis decrease. This force is transferred to the X and Y-axis, subsequently increasing the vehicle's ability to yaw. However, examining Figure 5.16 it is understood that the trade-off between thrust reduction and increased yaw authority has a large proportionate disparity. This differentiation provides application specific advantages, particularly for exploration.



Figure 5.14: Experimental design for force and torque measurements.

The exploration algorithm deployed onboard the quadcopter is dependent on the ZED 2 stereo camera. With a limited field of view, the algorithm encourages aggressive yaw maneuvering to expand the explorable frontiers and refine local trajectory planning. By reducing the thrust by 10.7% the torque can be increased by 53.7% between  $0^\circ$  and  $10^\circ$ . This trade-off is substantial for stereo vision based underground exploration where aggressive altitude changes are limited, but yaw agility is required.

## 5.4 Summary

Fully autonomous GPS-denied navigation presents extensive challenges for subterranean vehicle development. Presented as an iterative approach we have experimentally validated both the hexacopter and quadcopter vehicles in the target environ-

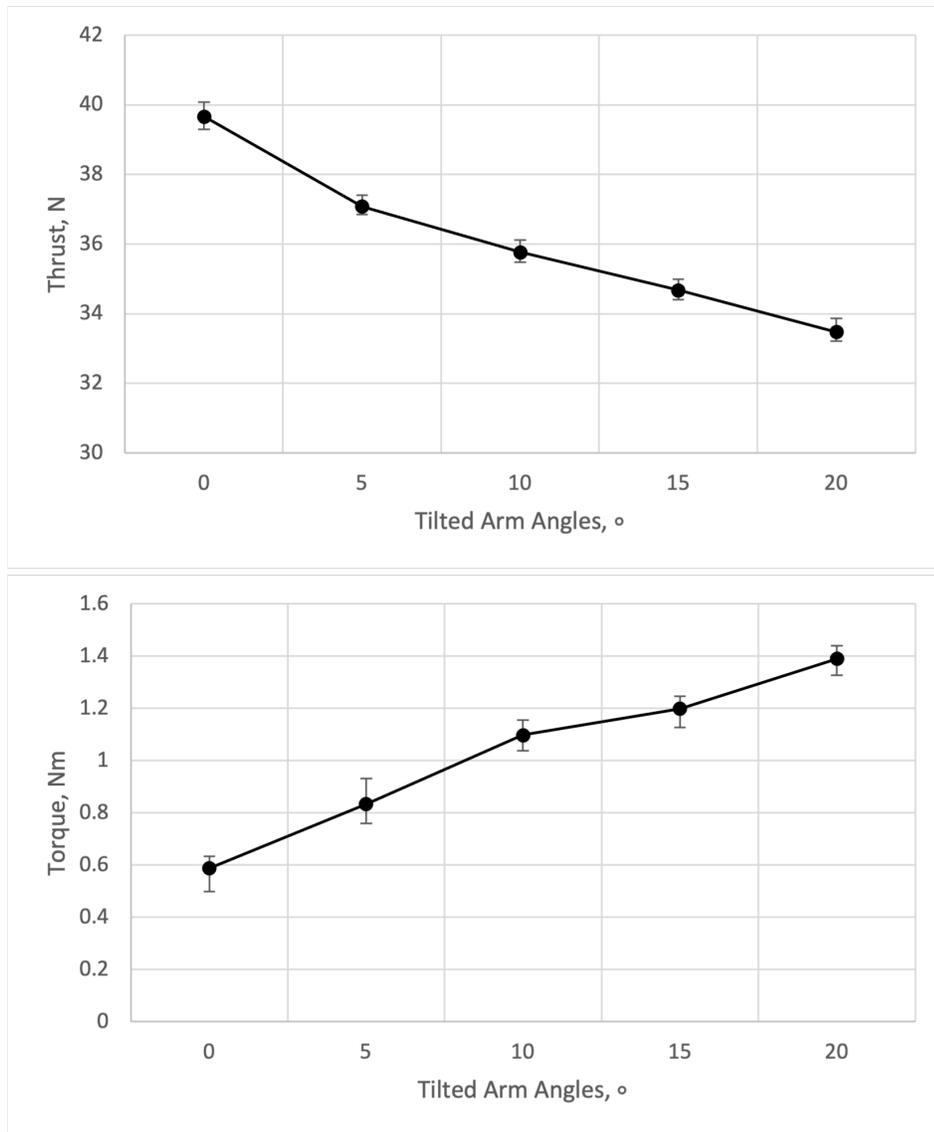


Figure 5.15: Thrust and torque generation as a result of the arm tilt angle.

ment. Considerations we evaluated in our designs were flight platforms, sensor stacks, propulsion systems, navigation algorithms, and agility. The hexacopter, possessing a higher payload threshold, carries a larger more capable sensor stack. Utilizing a Nonlinear Model Predictive Control Horizon optimized through reinforcement learning the hexacopter can plan efficient local trajectories. Using GBPlanner the hexacopter can fuse pointcloud data acquired from a 3D LiDAR with the voxel based software to recreate a virtual representation of the perceived scene. From the lessons learned of the hexacopter a more agile and cost-efficient quadcopter was developed.

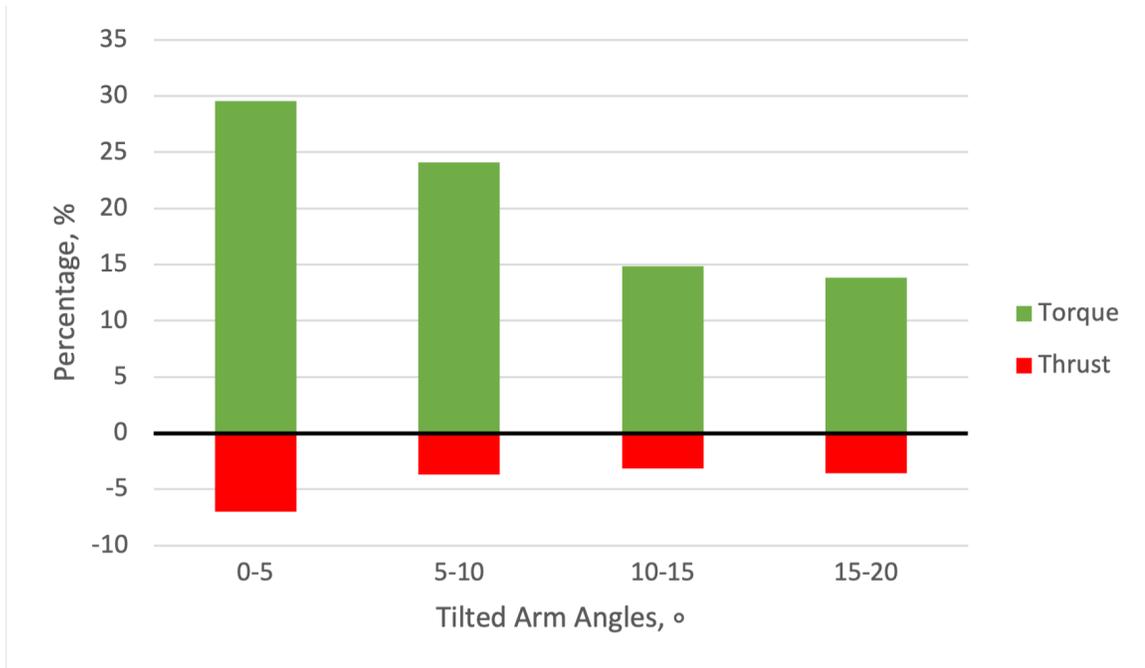


Figure 5.16: Force and torque percent change as a result of the quadcopter arm tilt angle.

This platform integrated the Fast UAV Exploration algorithm which produces global and local trajectories from stereo depth images. Similar to the hexacopter, a voxel-based map is reconstructed in a virtual environment to be used by the navigation algorithms. A modified asymmetric travelling salesman problem encapsulating the theory of frontier-based exploration is responsible for global waypoint establishment. A hybrid A\* approach is optimized using B-splines to efficiently create local trajectories within the environment.

Evaluated as progressive iterations of one another we have been able to achieve two independent fully autonomous GPS-denied vehicle platforms for underground exploration. It is our motivation that the design solutions presented provide insight to future UAV researchers in the field of subterranean exploration.

# Chapter 6

## Conclusion

Fully autonomous unmanned aerial vehicle (UAV) flight in GPS-denied environments is a complex task involving challenges in both hardware and software. However, with the motivation to increase safety, reduce human error, and optimize productivity this research area is gaining momentum. This thesis presented an evolution of two UAV system designs capable of fully autonomous UAV flight in GPS-denied environments.

Vision data is required for robotic systems performing tasks such as visual autonomous navigation or environmental imaging. However, the quality of this data is degraded in poorly lit environments. To overcome this issue, machine learning-based algorithms can be used to enhance low-light image frames. In this work the convolutional neural network architectures of Autoencoder, Modified Autoencoder, Feedforward, RetinexNet, and U-Net were implemented to enhance low-light images in real-time onboard a prototype aerial drone. The networks were run on the drone's single-board computer and fed video data from real flight maneuvers to realistically measure system performance. The network which accomplished the best image enhancement of static images was U-Net. However, full frame-rate performance was reduced to 10% with this model. The Modified Autoencoder was found to be the best performing model capable of 60% of full frame-rate performance. This model was found capable of producing colourless, but sharper and brighter images than U-Net.

The design considerations included an evolution of the flight platform, propulsion systems, sensor stack, and navigational path planners. The hexacopter, initially designed for ambiguous sensor requirements and payload introduces the optimized local trajectory planner of Nonlinear Model Predictive Horizon using deep learning directives. Global awareness is achieved through the graph-based planner GBPlanner. This design incorporated LiDAR sensing technology for accurate environmental perception. Succeeding the hexacopter is the lightweight and more versatile quadcopter. The quadcopter builds off the lessons learned from the previous platform to achieve the objective of autonomous flight. This design incorporates the Fast UAV Exploration (FUEL) algorithm to accomplish stable navigation with the use of a stereo depth camera for environment sensing. The presented systems and methodologies are verified through experimental flight tests in GPS-denied environments. By presenting two iterations of design this research aims to provide a benchmark for future research in the field of exploratory aerial robotics.

## 6.1 Future Work

The use of autonomous UAVs provides an opportunity to accomplish previously restricted and dangerous tasks found in industry and academia. With the advancements made in this research some of these limitations are solved. However, there are still prominent roadblocks that limit an optimized solution. One of the largest is the onboard computational power. This restriction is evident in the processing of sophisticated motion planners and dense sensory data collection. Therefore, it would be beneficial to utilize sensors with independent processing, more efficient algorithms, or upgrade to a more capable onboard computer.

Presented as a low-cost alternative to current industrial standards, the airframe and mechanical hardware of the proposed UAV continues to require robust safety considerations. To protect the propellers a ducted design should be investigated to reduce the potential for damage in the event of a collision. Similarly, a caged

design could be considered to protect the air frame. However, the shape, weight, and geometry must be optimized to ensure efficient flight performance.

Finally, a low-battery safeguard should be introduced into the FUEL algorithm. This would evaluate the length of the path traversed and the battery remaining. The determination of an exploration threshold would allow the quadcopter to safely return to the home position. Each of these design considerations would expand the capabilities of autonomous GPS-denied UAV flight and allow industry and researchers to further progress into novel applications for autonomous exploration.

# Bibliography

- [1] M. Tranzatto *et al.*, “CERBERUS: Autonomous Legged and Aerial Robotic Exploration in the Tunnel and Urban Circuits of the DARPA Subterranean Challenge,” *Field Robotics*, vol. 2, no. 1, pp. 274–324, Mar. 2022. DOI: 10.55417/FR.2022011.
- [2] J. Harbaugh, “Space Robotics Challenge,” 2016.
- [3] *Technoxian - World Robotics Championship*, 2014.
- [4] *VEX Robotics - Competition*, 2007.
- [5] P. Arm *et al.*, “Scientific exploration of challenging planetary analog environments with a team of legged robots,” *Science Robotics*, vol. 8, no. 80, Jul. 2023, ISSN: 2470-9476. DOI: 10.1126/SCIROBOTICS.ADE9548.
- [6] L. Wellhausen and M. Hutter, “ArtPlanner: Robust Legged Robot Navigation in the Field,” *Field Robotics*, vol. 3, no. 1, pp. 413–434, Mar. 2023. DOI: 10.55417/fr.2023013.
- [7] F. Bjelonic *et al.*, “Learning-based Design and Control for Quadrupedal Robots with Parallel-Elastic Actuators,” *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1611–1618, Jan. 2023. DOI: 10.1109/LRA.2023.3234809.
- [8] Y. Zimmermann, M. Sommerhalder, P. Wolf, R. Riener, and M. Hutter, “ANYexo 2.0: A Fully Actuated Upper-Limb Exoskeleton for Manipulation and Joint-Oriented Training in All Stages of Rehabilitation,” *IEEE Transactions on Robotics*, 2023, ISSN: 19410468. DOI: 10.1109/TRO.2022.3226890.
- [9] Y. Zimmermann *et al.*, “Human-Robot Attachment System for Exoskeletons: Design and Performance Analysis,” *IEEE Transactions on Robotics*, 2023, ISSN: 19410468. DOI: 10.1109/TRO.2023.3268587.
- [10] X. Kong, S. Liu, M. Taher, and A. J. Davison, “vMAP: Vectorised Object Mapping for Neural Field SLAM,” Feb. 2023.
- [11] K. Mazur, E. Sucar, and A. J. Davison, “Feature-Realistic Neural Fusion for Real-Time, Open Set Scene Understanding,” Oct. 2022.
- [12] T. Y. Tang, D. De Martini, and P. Newman, “Point-based metric and topological localisation between lidar and overhead imagery,” *Autonomous Robots*, vol. 47, no. 5, pp. 595–615, Jun. 2023, ISSN: 15737527. DOI: 10.1007/S10514-023-10085-W/FIGURES/19.

- [13] A. Patwardhan, R. Murai, and A. J. Davison, “Distributing Collaborative Multi-Robot Planning with Gaussian Belief Propagation,” *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 552–559, Mar. 2022. DOI: 10.1109/LRA.2022.3227858.
- [14] J. DelPreto, A. F. Salazar-Gomez, S. Gil, R. Hasani, F. H. Guenther, and D. Rus, “Plug-and-play supervisory control using muscle and brain signals for real-time gesture and error detection,” *Autonomous Robots*, vol. 44, no. 7, pp. 1303–1322, Sep. 2020, ISSN: 15737527. DOI: 10.1007/S10514-020-09916-X/FIGURES/13.
- [15] J. DelPreto and D. Rus, “Plug-and-play gesture control using muscle and motion sensors,” *ACM/IEEE International Conference on Human-Robot Interaction*, pp. 439–448, Mar. 2020, ISSN: 21672148. DOI: 10.1145/3319502.3374823.
- [16] M. Rigter, B. Lacerda, and N. Hawes, “RAMBO-RL: Robust Adversarial Model-Based Offline Reinforcement Learning,” Apr. 2022, ISSN: 10495258.
- [17] B. DeMoss, P. Duckworth, N. Hawes, and I. Posner, “DITTO: Offline Imitation Learning with World Models,” Feb. 2023.
- [18] M. N. Finean, L. Petrović, W. Merkt, I. Marković, and I. Havoutis, “Motion Planning in Dynamic Environments Using Context-Aware Human Trajectory Prediction,” *Robotics and Autonomous Systems*, vol. 166, Jan. 2022, ISSN: 09218890. DOI: 10.1016/j.robot.2023.104450.
- [19] A. Meduri, P. Shah, J. Viereck, M. Khadiv, I. Havoutis, and L. Righetti, “BiConMP: A Nonlinear Model Predictive Control Framework for Whole Body Motion Planning,” *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 905–922, Jan. 2022, ISSN: 19410468. DOI: 10.1109/TRO.2022.3228390.
- [20] Suncor Energy, “Suncor Energy implements first commercial fleet of autonomous haul trucks in the oil sands,” 2018.
- [21] V. McDermott, *Imperial deploying seven automated trucks, CNRL starting pilot in 2019 — Fort McMurray Today*, 2018.
- [22] P. Moore, *Imperial Oil has now converted 22 Cat 797 trucks at Kearl to autonomous operation - International Mining*, 2020.
- [23] Glencore, *Using advanced autonomous technology to keep our employees safe*, 2019.
- [24] Rio Tinto, *Scania and Rio Tinto agree to develop autonomous haulage solutions supporting a pathway to lower emissions mining*, 2022.
- [25] Y. Al Younes and M. Barczyk, “Adaptive Nonlinear Model Predictive Horizon Using Deep Reinforcement Learning for Optimal Trajectory Planning,” *Drones*, vol. 6, no. 11, p. 323, Nov. 2022, ISSN: 2504446X. DOI: 10.3390/DRONES6110323/S1.

- [26] Y. Al Younes and M. Barczyk, “Optimal Motion Planning in GPS-Denied Environments Using Nonlinear Model Predictive Horizon,” *Sensors 2021, Vol. 21, Page 5547*, vol. 21, no. 16, p. 5547, Aug. 2021, ISSN: 1424-8220. DOI: 10.3390/S21165547.
- [27] Y. A. Younes, M. B. Robotics, and u. 2021, “Nonlinear model predictive horizon for optimal trajectory generation,” *mdpi.com*, 2021. DOI: 10.3390/robotics10030090.
- [28] Y. A. Younes, M. B. Robotics, and u. 2022, “A backstepping approach to non-linear model predictive horizon for optimal trajectory planning,” *mdpi.com*, 2022. DOI: 10.3390/robotics11050087.
- [29] R. M. Murray, Z. Li, H. Kong, and S. S. Sastry, “A Mathematical Introduction to Robotic Manipulation,” 1994.
- [30] Y. Zhang, A. Carballo, H. Yang, and K. Takeda, “ISPRS Journal of Photogrammetry and Remote Sensing Perception and sensing for autonomous vehicles under adverse weather conditions: A survey,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 196, pp. 146–177, 2023. DOI: 10.1016/j.isprsjprs.2022.12.021.
- [31] Y. Ma, J. Košecká, S. Soatto, and S. Sastry, “An Invitation to 3-D Vision From Images to Models,” 2001.
- [32] W. J. Borucki, “KEPLER Mission: development and overview,” *Reports on Progress in Physics*, vol. 79, no. 3, p. 036 901, Feb. 2016, ISSN: 0034-4885. DOI: 10.1088/0034-4885/79/3/036901.
- [33] T. Dang, M. Tranzatto, S. Khattak, F. Mascari, K. Alexis, and M. Hutter, “Graph-based subterranean exploration path planning using aerial and legged robots,” *Journal of Field Robotics*, vol. 37, no. 8, pp. 1363–1388, Dec. 2020, ISSN: 15564967. DOI: 10.1002/ROB.21993.
- [34] B. Liu *et al.*, “Underwater Hyperspectral Imaging Technology and Its Applications for Detecting and Mapping the Seafloor: A Review,” *Sensors (Basel, Switzerland)*, vol. 20, no. 17, pp. 1–21, Sep. 2020, ISSN: 14248220. DOI: 10.3390/S20174962.
- [35] T. Larrue, Y. Li, X. Meng, and C.-M. Han, “Denoising Videos with Convolutional Autoencoders A Comparison of Autoencoder Architectures,” 2018. DOI: 10.1145/nnnnnnn.nnnnnnn.
- [36] H. Lei and Y. Yang, “CDAE: A Cascade of Denoising Autoencoders for Noise Reduction in the Clustering of Single-Particle Cryo-EM Images,” *Frontiers in Genetics*, vol. 11, p. 627 746, Jan. 2021, ISSN: 16648021. DOI: 10.3389/FGENE.2020.627746/FULL.
- [37] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, pp. 494–495.

- [38] T. Koike-Akino and Y. Wang, “Stochastic Bottleneck: Rateless Auto-Encoder for Flexible Dimensionality Reduction,” *IEEE International Symposium on Information Theory - Proceedings*, vol. 2020-June, pp. 2735–2740, Jun. 2020, ISSN: 21578095. DOI: 10.1109/ISIT44484.2020.9174523.
- [39] L. Chen and W. Y. Deng, “Instance-wise denoising autoencoder for high dimensional data,” *Mathematical Problems in Engineering*, vol. 2016, 2016, ISSN: 15635147. DOI: 10.1155/2016/4365372.
- [40] K. G. Lore, A. Akintayo, and S. Sarkar, “LLNet: A Deep Autoencoder Approach to Natural Low-light Image Enhancement,” *Pattern Recognition*, vol. 61, pp. 650–662, Nov. 2015, ISSN: 00313203. DOI: 10.1016/j.patcog.2016.06.008.
- [41] P. A. A. Bennur, M. Gaggar, and S. S. S, “LCA-Net: Light Convolutional Autoencoder for Image Dehazing,” Aug. 2020.
- [42] S. Park, S. Yu, M. Kim, K. Park, and J. Paik, “Dual Autoencoder Network for Retinex-Based Low-Light Image Enhancement,” *IEEE Access*, pp. 22 084–22 093, 2018. DOI: 10.1109/ACCESS.2018.2812809.
- [43] R. Nirthika, S. Manivannan, A. Ramanan, and R. Wang, “Pooling in convolutional neural networks for medical image analysis: a survey and an empirical study,” *Neural Computing and Applications*, vol. 34, no. 7, pp. 5321–5347, Apr. 2022, ISSN: 14333058. DOI: 10.1007/S00521-022-06953-8.
- [44] R. P. de Lima and K. Marfurt, “Convolutional Neural Network for Remote-Sensing Scene Classification: Transfer Learning Analysis,” *Remote Sensing 2020, Vol. 12, Page 86*, vol. 12, no. 1, p. 86, Dec. 2019, ISSN: 2072-4292. DOI: 10.3390/RS12010086.
- [45] R. Nirthika, S. Manivannan, A. Ramanan, and R. Wang, “Pooling in convolutional neural networks for medical image analysis: a survey and an empirical study,” *Neural Computing and Applications*, vol. 34, no. 7, pp. 5321–5347, Apr. 2022, ISSN: 14333058. DOI: 10.1007/S00521-022-06953-8/TABLES/14.
- [46] C. Li, J. Guo, F. Porikli, and Y. Pang, “LightenNet: A Convolutional Neural Network for weakly illuminated image enhancement,” *Pattern Recognition Letters*, vol. 104, pp. 15–22, 2018. DOI: 10.1016/j.patrec.2018.01.010.
- [47] E. Schwartz, R. Giryes, and A. M. Bronstein, “DeepISP: Towards Learning an End-to-End Image Processing Pipeline,” *IEEE Transactions on Image Processing*, vol. 28, no. 2, pp. 912–923, Jan. 2018. DOI: 10.1109/TIP.2018.2872858.
- [48] E. H. Land and J. J. McCann, “Lightness and Retinex Theory,” *JOSA, Vol. 61, Issue 1, pp. 1-11*, vol. 61, no. 1, pp. 1–11, Jan. 1971, ISSN: 00303941. DOI: 10.1364/JOSA.61.000001.
- [49] Z.-U. Rahman, G. A. Woodell, and D. J. Jobson, *A Comparison of the Multi-scale Retinex With Other Image Enhancement Techniques*, 1997.
- [50] X. Liu, Q. Xie, Q. Zhao, H. Wang, and D. Meng, “Low-light Image Enhancement by Retinex Based Algorithm Unrolling and Adjustment,” Feb. 2022.

- [51] H. Keerthi and S. B., “An Improved Retinex Method for Low Light Image Enhancement,” *International Journal of Intelligent Systems and Applications in Engineering*, vol. 10, no. 1s, pp. 418–428, Oct. 2022.
- [52] Z. Lei, C. Tang, M. Xu, B. Liang, T. Wu, and R. Han, “Enhancement method with naturalness preservation and artifact suppression based on an improved Retinex variational model for color retinal images,” *JOSA A, Vol. 40, Issue 1, pp. 155-164*, vol. 40, no. 1, pp. 155–164, Jan. 2023, ISSN: 1520-8532. DOI: 10.1364/JOSAA.474020.
- [53] H. Tang, H. Zhu, H. Tao, and C. Xie, “An Improved Algorithm for Low-Light Image Enhancement Based on RetinexNet,” *Applied Sciences (Switzerland)*, vol. 12, no. 14, Jul. 2022, ISSN: 20763417. DOI: 10.3390/APP12147268.
- [54] C. Wei, W. Wang, W. Yang, and J. Liu, “Deep Retinex Decomposition for Low-Light Enhancement,” *British Machine Vision Conference 2018, BMVC 2018*, Aug. 2018.
- [55] W. Weng and X. Zhu, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” *IEEE Access*, vol. 9, pp. 16 591–16 603, May 2015, ISSN: 21693536. DOI: 10.1109/ACCESS.2021.3053408.
- [56] L. Zhao, K. Wang, J. Zhang, A. Wang, and H. Bai, “Learning deep texture-structure decomposition for low-light image restoration and enhancement,” *Neurocomputing*, vol. 524, pp. 126–141, Mar. 2023, ISSN: 0925-2312. DOI: 10.1016/J.NEUCOM.2022.12.043.
- [57] W. Xu *et al.*, “High-Resolution U-Net: Preserving Image Details for Cultivated Land Extraction,” *Sensors (Basel, Switzerland)*, vol. 20, no. 15, pp. 1–23, Aug. 2020, ISSN: 14248220. DOI: 10.3390/S20154064.
- [58] S. Ai and J. Kwon, “Extreme low-light image enhancement for surveillance cameras using attention U-net,” *Sensors (Switzerland)*, vol. 20, no. 2, Jan. 2020, ISSN: 14248220. DOI: 10.3390/S20020495.
- [59] J. Schlemper *et al.*, “Attention gated networks: Learning to leverage salient regions in medical images,” *Medical Image Analysis*, vol. 53, pp. 197–207, Apr. 2019, ISSN: 1361-8415. DOI: 10.1016/J.MEDIA.2019.01.012.
- [60] Y. Cai and U. Kintak, “Low-Light Image Enhancement Based on Modified U-Net,” *International Conference on Wavelet Analysis and Pattern Recognition*, vol. 2019-July, Jul. 2019, ISSN: 21585709. DOI: 10.1109/ICWAPR48189.2019.8946456.
- [61] P. Foehn *et al.*, “AlphaPilot: autonomous drone racing,” *Autonomous Robots*, vol. 46, no. 1, pp. 307–320, Jan. 2022, ISSN: 15737527. DOI: 10.1007/S10514-021-10011-Y/TABLES/3.

- [62] F. Outay, H. A. Mengash, and M. Adnan, “Applications of unmanned aerial vehicle (UAV) in road safety, traffic and highway infrastructure management: Recent advances and challenges,” *Transportation Research. Part A, Policy and Practice*, vol. 141, p. 116, Nov. 2020, ISSN: 09658564. DOI: 10.1016/J.TRA.2020.09.018.
- [63] T. R. Wanasinghe, R. G. Gosine, O. De Silva, G. K. Mann, L. A. James, and P. Warriar, “Unmanned aerial systems for the oil and gas industry: Overview, applications, and challenges,” *IEEE Access*, vol. 8, pp. 166 980–166 997, 2020, ISSN: 21693536. DOI: 10.1109/ACCESS.2020.3020593.
- [64] G. Udeanu, A. Dobrescu, and M. Oltean, “Unmanned Aerial Vehicle in Military Operations,” *Scientific Research and Education in the Air Force*, vol. 18, no. 1, pp. 199–205, 2016. DOI: 10.19062/2247-3173.2016.18.1.26.
- [65] T. Elmokadem and A. V. Savkin, “Towards Fully Autonomous UAVs: A Survey,” *Sensors (Basel, Switzerland)*, vol. 21, no. 18, Sep. 2021, ISSN: 14248220. DOI: 10.3390/S21186223.
- [66] A. Tahir, J. Boling, M. H. Haghbayan, H. T. Toivonen, and J. Plosila, “Swarms of Unmanned Aerial Vehicles — A Survey,” *Journal of Industrial Information Integration*, vol. 16, p. 100 106, Dec. 2019, ISSN: 2452-414X. DOI: 10.1016/J.JII.2019.100106.
- [67] D. Giordan *et al.*, “The use of unmanned aerial vehicles (UAVs) for engineering geology applications,” *Bulletin of Engineering Geology and the Environment*, vol. 79, no. 7, pp. 3437–3481, Sep. 2020, ISSN: 14359537. DOI: 10.1007/S10064-020-01766-2/TABLES/15.
- [68] H.-M. Huang, “Autonomy Levels for Unmanned Systems (ALFUS) Framework Volume I: Terminology Version 2.0,” 2004. DOI: 10.6028/NIST.SP.1011-I-2.0.
- [69] E. Balestrieri, P. Daponte, L. De Vito, and F. Lamonaca, “Sensors and Measurements for Unmanned Systems: An Overview,” *Sensors*, vol. 21, no. 4, pp. 1–27, Feb. 2021, ISSN: 14248220. DOI: 10.3390/S21041518.
- [70] M. Zhang, K. Li, B. Hu, and C. Meng, “Comparison of Kalman Filters for Inertial Integrated Navigation,” *Sensors*, vol. 19, no. 6, Mar. 2019, ISSN: 14248220. DOI: 10.3390/S19061426.
- [71] B. Sinopoli, M. Micheli, G. Donato, and T. J. Koo, “Vision based navigation for an unmanned aerial vehicle,” *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1757–1764, 2001, ISSN: 10504729. DOI: 10.1109/ROBOT.2001.932864.
- [72] G. T. Schmidt, “Navigation sensors and systems in GNSS degraded and denied environments,” *Chinese Journal of Aeronautics*, vol. 28, no. 1, pp. 1–10, Feb. 2015, ISSN: 1000-9361. DOI: 10.1016/J.CJA.2014.12.001.
- [73] E. Dhulkefl, A. Durdu, and H. Terzioglu, “Dijkstra Algorithm Using UAV Path Planning,” *Selcuk University Journal of Engineering Science and Technology*, pp. 92–105, Dec. 2020, ISSN: 2147-9364. DOI: 10.36306/KONJES.822225.

- [74] A. Karoline dos Santos Poli *et al.*, “Fast Path Planning for Firefighting UAV Based on A-Star algorithm,” *Journal of Physics: Conference Series*, vol. 2029, no. 1, p. 012103, Sep. 2021, ISSN: 1742-6596. DOI: 10.1088/1742-6596/2029/1/012103.
- [75] D. Mandloi, R. Arya, and A. K. Verma, “Unmanned aerial vehicle path planning based on A\* algorithm and its variants in 3d environment,” *International Journal of System Assurance Engineering and Management*, vol. 12, no. 5, pp. 990–1000, Oct. 2021, ISSN: 09764348. DOI: 10.1007/S13198-021-01186-9.
- [76] J. Kwak and Y. Sung, “Autonomous UAV Flight Control for GPS-Based Navigation,” *IEEE Access*, vol. 6, pp. 37 947–37 955, Jul. 2018, ISSN: 21693536. DOI: 10.1109/ACCESS.2018.2854712.
- [77] D. Guastella, M. Labbadi, G. Gugan, and A. Haque, “Path Planning for Autonomous Drones: Challenges and Future Directions,” *Drones 2023, Vol. 7, Page 169*, vol. 7, no. 3, p. 169, Feb. 2023, ISSN: 2504-446X. DOI: 10.3390/DRONES7030169.
- [78] V. Vonasek, J. Faigl, T. Krajnik, and L. Preucil, “RRT-path - A guided rapidly exploring random tree,” *Lecture Notes in Control and Information Sciences*, vol. 396, pp. 307–316, 2009, ISSN: 01708643. DOI: 10.1007/978-1-84882-985-5{\\_}28/COVER.
- [79] B. B. Maseko, C. E. van Daalen, and J. Treurnicht, “Optimised Informed RRTs for Mobile Robot Path Planning,” *IFAC-PapersOnLine*, vol. 54, no. 21, pp. 157–162, Jan. 2021, ISSN: 2405-8963. DOI: 10.1016/J.IFACOL.2021.12.027.
- [80] R. Geraerts and M. H. Overmars, “A comparative study of probabilistic roadmap planners,” *Springer Tracts in Advanced Robotics*, vol. 7 STAR, pp. 43–57, 2004, ISSN: 16107438. DOI: 10.1007/978-3-540-45058-0{\\_}4/COVER.
- [81] Z. Xu, D. Deng, and K. Shimada, “Autonomous UAV Exploration of Dynamic Environments Via Incremental Sampling and Probabilistic Roadmap,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2729–2736, Apr. 2021, ISSN: 23773766. DOI: 10.1109/LRA.2021.3062008.
- [82] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 500–505, 1985, ISSN: 10504729. DOI: 10.1109/ROBOT.1985.1087247.
- [83] F. Bounini, D. Gingras, H. Pollart, and D. Gruyer, “Modified Artificial Potential Field Method for Online Path Planning Applications. IEEE Intelligent Vehicle symposium,” pp. 180–185, 2017. DOI: 10.1109/IVS.2017.7995717{\\_}i}.
- [84] M. Park, J. Jeon, and M. Lee, “Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing,” *IEEE International Symposium on Industrial Electronics*, vol. 3, pp. 1530–1535, 2001. DOI: 10.1109/ISIE.2001.931933.

- [85] X. Fan, Y. Guo, H. Liu, B. Wei, and W. Lyu, "Improved Artificial Potential Field Method Applied for AUV Path Planning," *Mathematical Problems in Engineering*, vol. 2020, 2020, ISSN: 15635147. DOI: 10.1155/2020/6523158.
- [86] Y. Li, S. Zhang, F. Ye, T. Jiang, and Y. Li, "A UAV Path Planning Method Based on Deep Reinforcement Learning," *2020 USNC-URSI Radio Science Meeting (Joint with AP-S Symposium), USNC/URSI 2020 - Proceedings*, pp. 93–94, Jul. 2020. DOI: 10.23919/USNC/URSI49741.2020.9321625.
- [87] J. Westheider, J. Ruckin, and M. Popovic, "Multi-UAV Adaptive Path Planning Using Deep Reinforcement Learning," Mar. 2023.
- [88] B. Liu, S. Wang, Q. Li, X. Zhao, Y. Pan, and C. Wang, "Task Assignment of UAV Swarms Based on Deep Reinforcement Learning," *Drones 2023, Vol. 7, Page 297*, vol. 7, no. 5, p. 297, Apr. 2023, ISSN: 2504-446X. DOI: 10.3390/DRONES7050297.
- [89] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, Sep. 2015.
- [90] J. Li, W. Zhang, Y. Hu, S. Fu, C. Liao, and W. Yu, "RJA-Star Algorithm for UAV Path Planning Based on Improved R5DOS Model," *Applied Sciences 2023, Vol. 13, Page 1105*, vol. 13, no. 2, p. 1105, Jan. 2023, ISSN: 2076-3417. DOI: 10.3390/APP13021105.
- [91] S. Ganesan, S. K. Natarajan, and A. Thondiyath, "G-RRT\*: Goal-oriented sampling-based RRT\* path planning Algorithm for mobile robot navigation with improved convergence rate," *ACM International Conference Proceeding Series*, Jun. 2021. DOI: 10.1145/3478586.3478588.
- [92] J. Nasir *et al.*, "RRT\*-SMART: A rapid convergence implementation of RRT\*," *International Journal of Advanced Robotic Systems*, vol. 10, Jul. 2013, ISSN: 17298806. DOI: 10.5772/56718/ASSET/IMAGES/LARGE/10.5772{\\_}56718-FIG12.JPEG.
- [93] A. Faust *et al.*, "PRM-RL: Long-range Robotic Navigation Tasks by Combining Reinforcement Learning and Sampling-based Planning," 2018.
- [94] H. Jiang, S. Li, J. Zhang, Y. Zhu, X. Xu, and D. Liu, "Efficient state representation with artificial potential fields for reinforcement learning," *Complex and Intelligent Systems*, pp. 1–12, Feb. 2023, ISSN: 21986053. DOI: 10.1007/S40747-023-00995-8/FIGURES/7.
- [95] C. Li *et al.*, "Low-Light Image and Video Enhancement Using Deep Learning: A Survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 12, pp. 9396–9416, Dec. 2022, ISSN: 1939-3539. DOI: 10.1109/TPAMI.2021.3126387.

- [96] M. F. Che Aminudin and S. A. Suandi, "Video surveillance image enhancement via a convolutional neural network and stacked denoising autoencoder," *Neural Computing and Applications*, vol. 34, no. 4, pp. 3079–3095, Feb. 2022, ISSN: 14333058. DOI: 10.1007/S00521-021-06551-0/FIGURES/9.
- [97] Y. Long, M. Du, X. Luo, S. Li, and Y. Liu, "Research on Navigation and Positioning of Electric Inspection Robot Based on Improved CNN Algorithm," *Journal of Sensors*, vol. 2022, 2022, ISSN: 16877268. DOI: 10.1155/2022/9369543.
- [98] Y. L. Chung and J. J. Yang, "Application of a Mask R-CNN-Based Deep Learning Model Combined with the Retinex Image Enhancement Algorithm for Detecting Rockfall and Potholes on Hill Roads," *IEEE International Conference on Consumer Electronics - Berlin, ICCE-Berlin*, vol. 2021-November, 2021, ISSN: 21666822. DOI: 10.1109/ICCE-BERLIN53567.2021.9720001.
- [99] G. T. S. Ho, Y. P. Tsang, C. H. Wu, W. H. Wong, and K. L. Choy, "A Computer Vision-Based Roadside Occupation Surveillance System for Intelligent Transport in Smart Cities," *Sensors 2019, Vol. 19, Page 1796*, vol. 19, no. 8, p. 1796, Apr. 2019, ISSN: 1424-8220. DOI: 10.3390/S19081796.
- [100] H. Zhu *et al.*, "A Review of Histogram Equalization Techniques in Image Enhancement Application," *Journal of Physics: Conference Series*, vol. 1019, no. 1, p. 012026, Jun. 2018, ISSN: 1742-6596. DOI: 10.1088/1742-6596/1019/1/012026.
- [101] S. Rahman, M. M. Rahman, M. Abdullah-Al-Wadud, G. D. Al-Quaderi, and M. Shoyaib, "An adaptive gamma correction for image enhancement," *Eurasip Journal on Image and Video Processing*, vol. 2016, no. 1, pp. 1–13, Dec. 2016, ISSN: 16875281. DOI: 10.1186/S13640-016-0138-1/TABLES/4.
- [102] J. Yang, J. Wang, L. L. Dong, S. Y. Chen, H. Wu, and Y. W. Zhong, "Optimization algorithm for low-light image enhancement based on Retinex theory," *IET Image Processing*, vol. 17, no. 2, pp. 505–517, Feb. 2023, ISSN: 1751-9667. DOI: 10.1049/IPR2.12650.
- [103] M. I. Ashiba, M. S. Tolba, A. S. El-Fishawy, and F. E. Abd El-Samie, "Gamma correction enhancement of infrared night vision images using histogram processing," *Multimedia Tools and Applications*, vol. 78, no. 19, pp. 27771–27783, Oct. 2019, ISSN: 15737721. DOI: 10.1007/S11042-018-7086-Y/FIGURES/6.
- [104] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data 2021 8:1*, vol. 8, no. 1, pp. 1–74, Mar. 2021, ISSN: 2196-1115. DOI: 10.1186/S40537-021-00444-8.
- [105] Y. Wang, W. Xie, and H. Liu, "Low-light image enhancement based on deep learning: a survey," *Optical Engineering*, vol. 61, no. 04, Apr. 2022, ISSN: 15602303. DOI: 10.1117/1.OE.61.4.040901.

- [106] H. Qi, E. R. Sparks, and A. Talwalkar, “PALEO: A PERFORMANCE MODEL FOR DEEP NEURAL NETWORKS,” *International Conference on Learning Representations*, 2017.
- [107] M. Decuyper, M. Stockhoff, S. Vandenberghe, a., and X. Ying, “An Overview of Overfitting and its Solutions,” *Journal of Physics: Conference Series*, vol. 1168, no. 2, p. 022 022, Feb. 2019, ISSN: 1742-6596. DOI: 10.1088/1742-6596/1168/2/022022.
- [108] N. C. Thompson, K. Greenewald, K. Lee, G. F. Manso, and A. I. Lab, “THE COMPUTATIONAL LIMITS OF DEEP LEARNING,”
- [109] J. Hanhirona, T. Kamarainen, S. Seppala, M. Siekkinen, V. Hirvisalo, and A. Yla-Jaaski, “Latency and Throughput Characterization of Convolutional Neural Networks for Mobile Computer Vision,” vol. 18, 2018.
- [110] A. K, A. M. Jain, and A. K. R, “Automated Headlight Intensity Control and Obstacle Alerting System,” *International Journal of Engineering Research & Technology*, vol. 6, no. 13, Apr. 2018, ISSN: 2278-0181. DOI: 10.17577/IJERTCONV6IS13058.
- [111] W. Zheng *et al.*, “The impact factors on the performance of machine learning-based vulnerability detection: A comparative study,” *Journal of Systems and Software*, vol. 168, p. 110 659, Oct. 2020, ISSN: 0164-1212. DOI: 10.1016/J.JSS.2020.110659.
- [112] C. Chen, Q. Chen, J. Xu, and V. Koltun, “Learning to See in the Dark,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3291–3300, May 2018, ISSN: 10636919. DOI: 10.1109/CVPR.2018.00347.
- [113] Y. P. Loh and C. S. Chan, “Getting to know low-light images with the Exclusively Dark dataset,” *Computer Vision and Image Understanding*, vol. 178, pp. 30–42, Jan. 2019, ISSN: 1077-3142. DOI: 10.1016/J.CVIU.2018.10.010.
- [114] D. P. Kingma and J. L. Ba, “Adam: A Method for Stochastic Optimization,” *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, Dec. 2014.
- [115] F. A. Fardo, V. H. Conforto, F. C. de Oliveira, and P. S. Rodrigues, “A Formal Evaluation of PSNR as Quality Measurement Parameter for Image Segmentation Algorithms,” May 2016.
- [116] Z. Wang, E. P. Simoncelli, and A. C. Bovik, “Multi-scale structural similarity for image quality assessment,” *Conference Record of the Asilomar Conference on Signals, Systems and Computers*, vol. 2, pp. 1398–1402, 2003, ISSN: 10586393. DOI: 10.1109/ACSSC.2003.1292216.
- [117] T. J. Nyman, E. P. A. Karlsson, and J. Antfolk, “As time passes by: Observed motion-speed and psychological time during video playback,” *PLoS ONE*, vol. 12, no. 6, Jun. 2017, ISSN: 19326203. DOI: 10.1371/JOURNAL.PONE.0177855.

- [118] M. Polka, S. Ptak, and L. Kuziora, “The Use of UAV’s for Search and Rescue Operations,” *Procedia Engineering*, vol. 192, pp. 748–752, Jan. 2017, ISSN: 1877-7058. DOI: 10.1016/J.PROENG.2017.06.129.
- [119] S. Hayat, E. Yanmaz, and R. Muzaffar, “Survey on Unmanned Aerial Vehicle Networks for Civil Applications: A Communications Viewpoint,” *IEEE Communications Surveys and Tutorials*, vol. 18, no. 4, pp. 2624–2661, Oct. 2016, ISSN: 1553877X. DOI: 10.1109/COMST.2016.2560343.
- [120] J. P. Aurambout, K. Gkoumas, and B. Ciuffo, “Last mile delivery by drones: an estimation of viable market potential and access to citizens across European cities,” *European Transport Research Review*, vol. 11, no. 1, pp. 1–21, Dec. 2019, ISSN: 18668887. DOI: 10.1186/S12544-019-0368-2/FIGURES/20.
- [121] M. Mandirola, C. Casarotti, S. Peloso, I. Lanese, E. Brunesi, and I. Senaldi, “Use of UAS for damage inspection and assessment of bridge infrastructures,” *International Journal of Disaster Risk Reduction*, vol. 72, p. 102 824, Apr. 2022, ISSN: 2212-4209. DOI: 10.1016/J.IJDRR.2022.102824.
- [122] Y. Lu, Z. Xue, G. S. Xia, and L. Zhang, “A survey on vision-based UAV navigation,” *Geo-spacial Information Science*, vol. 21, no. 1, pp. 21–32, Jan. 2018, ISSN: 10095020. DOI: 10.1080/10095020.2017.1420509.
- [123] H. Ko, B. Kim, and S. H. Kong, “GNSS multipath-resistant cooperative navigation in urban vehicular networks,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 12, pp. 5450–5463, Dec. 2015, ISSN: 00189545. DOI: 10.1109/TVT.2015.2481509.
- [124] J. Martz, W. Al-Sabban, and R. N. Smith, “Survey of unmanned subterranean exploration, navigation, and localisation,” *IET Cyber-Systems and Robotics*, vol. 2, no. 1, pp. 1–13, Mar. 2020, ISSN: 2631-6315. DOI: 10.1049/IET-CSR.2019.0043.
- [125] S. W. Forrest, M. R. Recio, and P. J. Seddon, “Moving wildlife tracking forward under forested conditions with the SWIFT GPS algorithm,” *Animal Biotelemetry*, vol. 10, no. 1, pp. 1–11, Dec. 2022, ISSN: 20503385. DOI: 10.1186/S40317-022-00289-9/TABLES/6.
- [126] E. S. Abdolkarimi and M. R. Mosavi, “A low-cost integrated MEMS-based INS/GPS vehicle navigation system with challenging conditions based on an optimized IT2FNN in occluded environments,” *GPS Solutions*, vol. 24, no. 4, pp. 1–19, Oct. 2020, ISSN: 15211886. DOI: 10.1007/S10291-020-01023-9/TABLES/7.
- [127] W. B. White and D. C. Culver, *Encyclopedia of caves*, 2nd ed. Elsevier Inc., 2012, pp. 1–9, ISBN: 9780123838322. DOI: 10.1016/C2010-0-64793-6.
- [128] B. Zhou, Y. Zhang, X. Chen, and S. Shen, “FUEL: Fast UAV Exploration using Incremental Frontier Structure and Hierarchical Planning,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 779–786, Oct. 2020, ISSN: 23773766. DOI: 10.48550/arxiv.2010.11561.

- [129] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, “Robust and Efficient Quadrotor Trajectory Generation for Fast Autonomous Flight,” *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3529–3536, Oct. 2019, ISSN: 23773766. DOI: 10.1109/LRA.2019.2927938.
- [130] T. Dang, F. Mascarich, S. Khattak, C. Papachristos, and K. Alexis, “Graph-based Path Planning for Autonomous Robotic Exploration in Subterranean Environments,” *IEEE International Conference on Intelligent Robots and Systems*, pp. 3105–3112, 2019, ISSN: 21530866. DOI: 10.1109/IROS40897.2019.8968151.
- [131] B. Yamauchi, “A Frontier-Based Approach for Autonomous Exploration,” *Navy Center for Applied Research in Artificial Intelligence*, 1997.
- [132] M. Fischetti, A. Lodi, and P. Toth, “Exact Methods for the Asymmetric Traveling Salesman Problem,” pp. 169–205, 2007. DOI: 10.1007/0-306-48213-4{-}4.
- [133] U. F. Ukaegbu, L. K. Tartibu, M. O. Okwu, and I. O. Olayode, “Development of a Light-Weight Unmanned Aerial Vehicle for Precision Agriculture,” *Sensors (Basel, Switzerland)*, vol. 21, no. 13, Jul. 2021, ISSN: 14248220. DOI: 10.3390/S21134417.