EAAE: Energy-Aware Autonomous Exploration for UAVs in Unknown 3D Environments

Jacob Elskamp

Delft University of Technology

TU Delft

EAAE: Energy-Aware Autonomous Exploration for UAVs in Unknown 3D Environments

by

Jacob Elskamp

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Wednesday July 9, 2025 at 13:00.

Student number:	5053390	
Project duration:	November 10, 2024 – July 9, 2025	
Thesis committee:	Dr. Ir. C. de Wagter,	TU Delft, Chair
	Dr. ir. M. Popović,	TU Delft, Supervisor
	Dr. Ir. C. Pek,	TU Delft, External examiner
Supervisors:	Dr. ir. M. Popović,	TU Delft
	Ir. L. Bauersfeld,	UZH Zurich
	Ir. M. Shi,	TU Delft

Cover: Generated using ChatGPT, personal prompt

TU Delft

Preface

This thesis marks the completion of my Master's degree in Aerospace Engineering at Delft University of Technology. To me, Delft will always represent a place where smart, ambitious minds come together to tackle the challenges of tomorrow. This research reflects my interests in autonomous systems and cutting-edge innovation.

I would like to extend my gratitude to my supervisors, Marija Popović, Leonard Bauersfeld, and Moji Shi, for their invaluable guidance, constructive feedback, and unwavering support throughout this project. Their expertise has greatly enriched the quality of this work, and it has been a true honour to learn from them.

Finally, I would like to thank my parents, partner, and friends for their continued support throughout my studies and during this thesis project. Even if the technical details made little sense to them, they were always there to listen.

Here's to autonomy.

Jacob Elskamp
Delft, June 2025

Contents

Preface i

Nomenclature iv

I Preliminary 1

Introduction 2

Research Questions 2

II Scientific Article 4

1 Introduction 5

2 Related Works 6
2.1 Frontier-based Exploration 6
2.2 Sampling-based Exploration 6
2.3 Alternative Exploration Methods 6
2.4 Current Energy-aware Exploration Methods 7

3 Background 7
3.1 Problem Formulation 7
3.2 Theoretical foundation 7
3.3 Assumptions 8

4 Approach 8
4.1 System Overview 8
4.2 Perception and Mapping 9
4.3 Exploration Strategy 9
4.3.1 Frontier Detection and Clustering 9
4.3.2 Energy-awareness Module 10
4.4 Planning 11
4.4.1 Global Planning 12
4.4.2 Local Reactive Planning 12
4.5 Control and Simulation Framework 12
4.6 Summary 12

5 Experiment Setup 13
5.1 Simulation 13
5.2 Baselines 15
5.3 Measurements 15

6 Results 16
6.1 Exploration Rate 16
6.2 Map Information Entropy 17
6.3 Energy Consumption 18
6.4 Computational Efficiency 19
6.5 Summary 19

7 Discussion 20

8 Conclusion 20

References 21

III Closure 23

Conclusion 24

A Additional Results 25
A.1 Power Consumption over Time . . . 25
A.2 Visualization Exploration Mission . . . 25

B Literature Review 27

References 42

C Project Planning 45
C.1 Planning . . . 47

Nomenclature

Abbreviation	Definition
ATSP	Asymmetric Travelling Salesman Problem
EAAE	Energy-Aware Autonomous Exploration
FoV	Field of View
NBV	Next-Best View
RGB-D	Red-Green-Blue and Depth (camera)
RH-NBV	Receding-Horizon Next-Best View
RL	Reinforcement Learning
RRT*	Rapidly-exploring Random Tree Star
SAR	Search and Rescue
SDE	Stochastic Differential Equation
SL	Supervised Learning
SLAM	Simultaneous Localization and Mapping
TSP	Travelling Salesman Problem
UAV	Unmanned Aerial Vehicle

Part I

Preliminary

Introduction

Exploration has been a key driver of human progress throughout history. From the earliest days of humanity, people have explored unknown environments in pursuit of food, shelter, and knowledge. In the modern age, autonomous exploration represents the intersection of this timeless quest for discovery and technological innovation. Autonomous exploration nowadays provides a safer and more efficient approach to missions, providing humans with additional information about the environment without endangering humans lives. Examples of these missions are search and rescue operations, infrastructure inspections, environmental and wildlife monitoring, planetary exploration, and more.

Energy-awareness is a pivotal challenge in the domain of autonomous exploration, especially for unmanned aerial vehicles (UAVs), where energy limitations significantly constrain their potential. One of the most captivating aspects of UAVs is their compact size, which enables them to navigate environments that would otherwise be inaccessible. However, this very advantage also brings the critical drawback of limited energy supply, restricting mission duration and capabilities. Addressing energy efficiency for UAVs during missions is one of the most pressing steps required to ensure the reliable application of autonomous exploration systems in real-world scenarios. By incorporating energy awareness into exploration algorithms, these systems can become not only more effective but also capable of sustaining longer and more impactful missions.

Exploration algorithms for autonomous systems are well-established; however, they generally lack consideration for energy-awareness, which is critical for maximising efficiency in real-world applications. Addressing this gap is essential to advancing the capabilities of UAVs in challenging scenarios. The potential impact of energy-aware autonomous exploration is immense. A key driver in the motivation for this thesis project is the application in disaster response, such as search and rescue operations. These scenarios where time and precisions are of the essence, UAVs can make a significant difference, quickly location individuals in need while minimizing risks to humans responders. Beyond disaster response, these technologies are exceptionally promising in fields as industrial inspections, enabling safer and more thorough assessments of critical infrastructure. Another area that could be greatly impacted is in the field of planetary exploration. These missions have very limiting energy budgets whilst still having the aim to explore large unknown environments on the surface and possibly in caves. These real-world applications underline the importance of energy-efficient exploration as a foundation for achieving impactful outcomes in diverse and vital missions.

Research Questions

This section aims to provide the formulation of the main research objective and corresponding research questions for this thesis project. These research questions are based on the knowledge gap identified and described in chapter 8.

Research Objective The primary objective of this research project is to develop an autonomous exploration algorithm for UAVs that is informed by an integrated energy consumption model whilst maximizing exploration coverage and the rate of exploration and minimizing energy consumption.

Research Question 1 How to develop an autonomous exploration algorithm for a UAV that can be integrated with an energy-consumption model?

Research Question 2 How can energy consumption of an UAV be modelled based on a given trajectory that is planned by an exploration algorithm?

Research Question 3 What are the differences in exploration coverage and exploration rate with respect to the exploration algorithm that does not account for the energy consumption?

The link of these research questions to the current knowledge gap as described in Equation B is explained as follows. First of all, the first research question, captures the lack of exploration algorithms that are integrable with an energy consumption model. Furthermore, the second research question, aims to provide an energy consumption model that is able to output information regarding the consumed and required energy, such that the exploration planner is able to make informed decisions. Finally, the third research question suggest an evaluation of the effects of incorporating energy-awareness into an exploration algorithm.

Part II

Scientific Article

EAAE: Energy-Aware Autonomous Exploration for UAVs in Unknown 3D Environments

Jacob Elskamp

Abstract — **Autonomous exploration enables unmanned aerial vehicles (UAVs) to map unknown environments without human intervention. While state-of-the-art algorithms primarily focus on maximizing coverage or minimizing exploration time, they often overlook energy efficiency, a critical constraint for battery-powered UAVs. This thesis introduces EAAE, a modular Energy-Aware Autonomous Exploration framework that explicitly incorporates energy consumption into the exploration decision-making process. By combining frontier detection, a K-means divisive clustering algorithm, energy-aware target selection, and dual-layer planning, the framework balances information gain with energy cost. The algorithm is evaluated in simulation using the Agilicious control stack, with a physically realistic power model based on rotor speeds. Two 3D environments of varying complexity are used to compare EAAE against two baselines, an information-gain-based frontier method and distance-based frontier methods. Results show that EAAE consistently achieves the lowest total energy consumption while maintaining competitive or lower exploration times. Moreover, it demonstrates similar performance in terms of information entropy and reduced power variability, contributing to more efficient and robust mapping. This work highlights the importance of integrating energy-awareness into exploration pipelines and provides a foundation for further research into long-endurance autonomous aerial missions.**

Keywords: UAV exploration, Energy-Aware Planning, Autonomous Exploration, Frontier-Based Exploration, 3D Mapping, Energy Consumption Model

1 Introduction

Autonomous exploration has become a crucial component in modern robotics, enabling unmanned aerial vehicles (UAVs) to navigate and map unknown environments efficiently. This capability is particularly valuable in applications such as search and rescue [1], industrial inspection [2], and planetary exploration [3], where human intervention is either dangerous or impractical. However, one of the primary challenges in UAV-based exploration is energy limitation, which directly impacts mission duration and overall feasibility. Energy-aware exploration seeks to maximize exploration efficiency by integrating energy consumption information into the planning and decision-making steps of the algorithm. Despite significant advancements in autonomous exploration algorithms, the majority of approaches do not account for energy constraints.

Recent studies have demonstrated that UAVs exhibit an optimal flight speed that maximizes range or endurance, offering insights into the relationship between velocity and energy consumption [4]. Moreover, exploration algorithms that promote higher average flight speeds have been shown to reduce exploration time [5], a proxy often associated with improved energy efficiency. However, this assumed correlation remains largely unvalidated for actual exploration algorithms, which typically neglect the nonlinear and flight-state-dependent nature of UAV energy consumption. This thesis addresses this gap by integrating an energy estimation model into the planning loop, enabling informed trajectory selection based on estimated energy cost rather than simple proxies such as time or distance.

This gap limits the practical applicability of existing solutions in real-world scenarios and, consequently, is the main motivation for this research. To address this gap, this work aims to find a answer to the following research question:
What are the effects on exploration rate and mapping quality of an autonomous exploration algorithm that is informed by an integrated energy consumption model?

By addressing this research question and research gap, the primary contribution of this thesis can be summarized as follows:

- Development and evaluation of an autonomous exploration algorithm for UAVs in unknown environments that is informed by an integrated energy consumption model.

Traditional exploration frameworks typically select the next waypoint based on spatial heuristics such as proximity or the information gain. However, these heuristics often neglect the underlying energy cost of maneuvering a UAV through 3D space. This thesis introduces an energy-aware selection mechanism that leverages full trajectory generation and offline energy estimation to inform decision-making. Specifically, the algorithm computes dynamically feasible trajectories to the most promising frontier clusters using a global kino-dynamic planner. These trajectories are then offline-simulated in the Agilicious framework to estimate energy consumption based on rotor-level power profiles. By selecting the frontier that minimizes predicted energy expenditure, rather than simply maximizing information gain or minimizing distance. This contribution demonstrates that such an energy-informed approach can lead to lower total energy usage while maintaining or even slightly improving exploration performance.

This paper is outlined as follows: in chapter 2 the current state-of-the-art of autonomous exploration is introduced and the research gap is further highlighted. In chapter 3 the background information is provided as well as the formal problem statement. Thirdly, in chapter 4 the approach is described, including a system overview and elaboration of the various modules used in the algorithm. Fourthly, in chapter 5, the experimental setup is outlined including model details and explanation of the used baselines. In chapter 6 the results from the comparative study are presented. Chapter 7 provides a discussion on these results. Finally, in chapter 8 a conclusion and suggestions for future work are presented.

2 Related Works

Autonomous exploration algorithms aim to map unknown environments fast and efficient. Various studies in recent years have addressed autonomous exploration using UAVs. These studies can be divided into three subcategories: frontier-based methods, sampling-based methods, and alternative approaches. The last section describes relevant work on the existing energy-aware exploration methods for UAVs, which form the foundation for incorporating full energy-awareness into an autonomous exploration algorithm.

2.1. Frontier-based Exploration

The concept of using frontiers for autonomous exploration was first introduced by B. Yamauchi in [6]. This method categorizes the environment into one of the following: free space, occupied space, or unknown space. Subsequently, it defines the boundary region between the free space and the unknown space as a frontier region. From these frontier locations, the next best waypoint is selected. This method still forms the foundation for many studies in autonomous exploration [7]. The first exploration strategies using a frontier-based approach focused on minimizing the physical distance between the current robot position and the location of the frontier cluster centroid [6], [8], [7], [9]. Gao et al. propose considering the frontier size and the cost of turning the robot as well [10]. Another novel contribution was introduced in [5]. Here, the authors propose to select the frontier that minimizes the velocity change to maintain a high speed during exploration. The majority of the works related to exploration can be categorized as greedy algorithms. This means that the decision-making involves selecting the next best waypoint regardless of future steps. A non-greedy method is described in [11], here the authors propose to find a frontier sequence by using the traveling salesman problem formulation. More recent contributions include FUEL, which introduced incremental frontier updates and hierarchical planning for efficient global path coverage [12]; FAEP, extending FUEL by incorporating frontier-level and adaptive yaw-planning to reduce inefficient back-and-forth movements [13]; and LAEA, proposing an Environmental Information Gain (EIG) strategy combined with LiDAR data that prioritizes small and isolated frontier clusters, further minimizing redundant exploration paths [14].

2.2. Sampling-based Exploration

Sampling-based methods utilize the sampling of viewpoints to determine the next best position for exploration. While frontier-based approaches are well-suited for large environments due to the ability to detect unexplored areas in the global map, next-best-view (NBV) based approaches perform well in cluttered spaces [15]. NBV-based approaches rely on the NBV theory first proposed by Connolly in 1985 [16]. This method determines a sequence of viewpoints aiming to maximize the visibility of an object while minimizing travel costs. In contrast to frontier-based methods, NBV approaches rely on randomly sampling viewpoints within the known or partially-known environment and subsequently selecting the optimal viewpoint based on potential information gain and distance. Early implementations, such as the one described in [17], evaluate candidate viewpoints using a visibility gain function and penalize distance. Bircher et al. [18] later introduced the receding-horizon next-best-view (RH-NBV) method, which utilizes rapidly exploring random trees (RRT and RRT*) to iteratively expand and evaluate candidate viewpoints based on their expected visibility gains. Hybrid strategies utilize the frontier-based and NBV-based advantages for exploration as described in [19] and [20].

2.3. Alternative Exploration Methods

There are also other ways to approach the exploration problem. A novel contribution was made in [21] in which the authors propose a stochastic differential equation-based approach to simulate gas particles in the free space and explore based on the expansion pattern of these particles. More recently, various learning-based approaches were introduced. These methods utilize supervised learning (SL) and reinforcement learning (RL) to enhance the scalability of the exploration algorithms to larger unknown environments. One such approach is presented in [22], where the authors propose Active Neural SLAM, a reinforcement learning-based framework that jointly learns mapping, localization, and exploration policy. The system trains an agent to maximize area coverage using deep neural networks that predict utility from partial observations, demonstrating strong generalization to novel environments. Other works, such as [23], demonstrate vision-based exploration using deep networks that learn to navigate towards semantic goals.

2.4. Current Energy-aware Exploration Methods

Current autonomous exploration methods have shown promising results in various environments; however, these approaches typically neglect energy constraints. To effectively use UAVs in real-world exploration missions, integrating energy-awareness into the decision-making process of exploration algorithms is crucial. While some methods partially address energy efficiency by striving for near-zero acceleration flight [5], [24], other approaches consider fixed, energy-based penalties for distinct flight phases [25], [26], [27]. Despite these advancements, incorporating energy-awareness into autonomous exploration remains an open challenge.

3 Background

This section establishes the theoretical framework and contextual background of the study, outlining the specific problem formulation as well as the underlying assumptions related to the algorithm and energy modelling.

3.1. Problem Formulation

This research addresses the problem of autonomous exploration in unknown three-dimensional (3D) environments using a single unmanned aerial vehicle (UAV). The primary objective is to generate a complete map of the environment by classifying all accessible space as either free or occupied, using onboard sensing. A second objective in this research, in contrast to other works, lies in the focus on the integration of energy-awareness into the planning cycle. The goal is not only to reduce mission duration but also to minimize the total energy consumed (in joules) throughout the complete exploration process. This dual-objective exploration of efficiency in both time and energy is especially relevant for real-world scenarios where flight endurance is limited by battery capacity.

The specific problem addressed in this thesis can be formulated as follows: given an enclosed and initially unknown 3D environment, and a UAV equipped with onboard sensing and mapping capabilities, design an exploration strategy that incrementally selects goal waypoints in a way that leads to complete map coverage with minimal energy expenditure, without compromising on exploration time, mapping quality or mapping completeness.

Exploration strategy

In this work, energy-awareness is introduced by evaluating the energy cost $E(\pi_i)$ of a planned trajectory π_i to a frontier cluster c_i using offline execution of a globally planned, dynamically feasible path. To balance exploration performance and computational load, the full energy estimation process is applied only to a filtered subset of candidate clusters $C' \subset C$, consisting of the N clusters with the highest information gain $IG(c_i)$, defined as the number of unknown voxels in the cluster. For each cluster $c_i \in C'$, a corresponding trajectory π_i is generated, forming the subset of candidate trajectories $\Pi' \subset \Pi$. The final exploration target is then selected by choosing the trajectory with the lowest predicted energy cost:

$$\pi^* = \arg\min_{\pi_i \in \Pi'} E(\pi_i) \tag{3.1}$$

This approach ensures that exploration proceeds towards informative regions of the environment while avoiding unnecessary energy expenditure due to inefficient or dynamically aggressive trajectories. Compared to traditional frontier-based methods that rely solely on spatial heuristics such as distance or information gain, this formulation incorporates UAV-specific energy dynamics directly into the planning loop.

3.2. Theoretical foundation

The method proposed in this research builds on several theoretical principles that together motivate the research question regarding energy efficiency in exploration missions. In particular, it considers aerodynamic effects specific to quadrotor UAVs and their influence on energy consumption during autonomous exploration.

Quadrotor Aerodynamics

UAVs, particularly multirotors, rely primarily on electric propulsion systems for flight. The majority of their energy consumption arises from the power required to generate sufficient lift to counteract gravity, with additional energy used for overcoming aerodynamic drag, executing dynamic maneuvers, and powering onboard electronics. Of these, the propulsion-related energy dominates the total power budget [4].

Traditional models often approximate UAV power consumption using a quadratic formulation where thrust and torque are proportional to the square of the propeller's rotational speed. While this model performs adequately for near-hover conditions, it becomes inaccurate in forward flight due to unmodeled aerodynamic effects, such as dynamic lift and induced drag [4]. These effects play a crucial role in determining the true energy efficiency of a trajectory.

Dynamic lift occurs when the airflow over the rotor blades increases due to forward motion, thereby reducing the required propeller speed and associated power consumption. Conversely, linear rotor drag (or induced drag) increases with forward speed and can significantly contribute to the overall aerodynamic load. These opposing phenomena result in a non-trivial relationship between flight speed and power consumption. Experimental and simulation studies confirm that there exists an optimal flight speed at which a multirotor minimizes energy consumed per distance traveled [4].

Energy-Time Trade-off

The insight explained above is particularly relevant in the context of energy-aware autonomous exploration. Although faster trajectories can reduce total mission time, they may require disproportionately more energy due to higher aerodynamic loads from aggressive flight maneuvers. On the other hand, slower, more energy-efficient flight paths result in longer missions, increasing the total time the UAV remains airborne and thus incurring a greater baseline energy cost for hovering and electronics operation.

Therefore, a trade-off emerges between flight speed and energy efficiency: minimizing exploration time does not necessarily minimize energy usage. This motivates the integration of a flight-state-dependent energy estimation model into the exploration planning process. Rather than selecting the next best waypoint based solely on information gain or distance, the proposed method evaluates the energy cost of candidate trajectories, enabling a decision-making process that balances exploration efficiency against energy consumption.

3.3. Assumptions

The method proposed in this research is built upon several assumptions regarding the simulation and UAV model. These assumptions are as follows:

- **Environment:** The simulation environment is enclosed and bounded, containing only static elements or being empty. No dynamic obstacles or environmental changes occur during flight.
- **External disturbances:** The simulation assumes calm conditions with no wind, precipitation, or other external disturbances affecting flight dynamics.
- **Sensor:** The UAV has perfect sensing capabilities with zero measurement noise.
- **Pose estimation:** The UAV's pose is assumed to be perfectly known at all times, without drift or noise in position or orientation estimation.
- **Planning consistency:** It is assumed that the trajectory planned by the global planner for offline energy evaluation is sufficiently similar, in terms of power requirements, to the trajectory executed that is planned by the local planner during real-time flight.
- **Energy model:** The total power consumption of the UAV is modelled as the sum of mechanical power and an electrical loss term, based on the formulation and parameters described in the work by Bauersfeld et al. [4]. Furthermore, the motor voltage is assumed to be constant.

4 Approach

As introduced in chapter 1, the main goal of this research is to explore a bounded, unknown 3D environment $V_{total} \subset \mathbb{R}^3$. We propose the Energy-Aware Autonomous Explorer (EAAE), an autonomous exploration framework that is informed by an energy component during the decision-making step. A system overview and further elaboration on the various modules are presented in this chapter.

4.1. System Overview

The exploration algorithm is built using existing open-source frameworks supplemented by our own contributions. The system comprises four main modules, i.e., perception and mapping, exploration, path planning, and the control and simulation module. Testing is performed using *ROS*, *RVIZ*, and simulated *Gazebo* environments. *RVIZ* is used for mapping visualizations and *Gazebo* for retrieving depth camera data from the virtual environment. The perception and mapping module uses point cloud data generated by a depth camera sensor and detects frontiers continuously throughout the mission. Once the exploration is initialized, an iterative process starts. First, the algorithm uses the detected frontiers and clusters and filters them based on reachability. Subsequently, for the three clusters with the largest information gain, a global trajectory and the corresponding required energy are calculated. Once the target is selected, a reactive local planner generates a collision-free dynamic trajectory towards the goal. Once the UAV reaches the target, the cycle starts again. This process stops when the environment

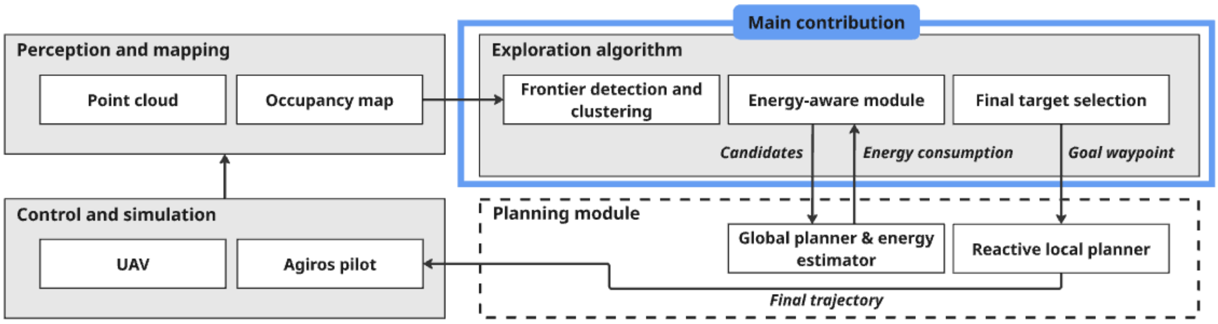


Figure 4.1: Simplified system overview of exploration algorithm.

is successfully explored. A simplified illustration of the system overview, including the four main modules, is included below in Figure 4.1.

4.2. Perception and Mapping

The environment is mapped as a 3D voxel grid representation, where every voxel v has dimension r^3, where r is the resolution of the map. The resolution of the voxel grid reflects a trade-off between mapping precision and memory usage. The environment is scanned using an onboard, forward-facing, depth camera sensor that generates point cloud data. This point cloud data is used by the *OctoMap* package [28] to generate an occupancy grid. This occupancy grid mapping offers an efficient, structured, and probabilistic representation of the environment, dividing continuous 3D space into discrete units (voxels). Each voxel stores a probabilistic estimate of being free, occupied, or unknown, based on sensor input. The combined set of voxels forms the complete environment representation.

$$V_{total} \equiv V_{free} \cup V_{occupied} \cup V_{unknown} \tag{4.1}$$

The *OctoMap* is updated continuously during flight according to the incoming point cloud data. Initially, the complete space is unknown and as the exploration progresses the amount of unknown space decreases and the amount of free and occupied space increases. The exploration is successful when $V_{free} \cup V_{occupied} \equiv V_{total}/V_{residual}$, where $V_{residual}$ is the residual space that is inaccessible by the sensor. This representation enables efficient spatial reasoning and supports both motion planning and frontier detection by explicitly identifying boundaries between explored and unexplored space. The occupancy information is also used for evaluating the exploration progress and the mapping quality.

4.3. Exploration Strategy

The exploration module is responsible for identifying and selecting navigation targets that lead the UAV toward unknown areas of the environment. In this work, we adopt a frontier-based exploration strategy, where the boundaries between known and unknown space are used as regions of interest to guide the motion of the UAV. This module consists of two main phases: frontier detection and clustering, and energy-informed candidate selection.

4.3.1. Frontier Detection and Clustering

A second crucial step for this frontier-based exploration algorithm is the detection and clustering of frontiers. A frontier voxel $f \in \mathcal{F}$ is defined as a free voxel that has a neighboring unknown voxel [6]. These locations are points of interest for exploration purposes as these points are by definition accessible through free space and provide information about unknown space because of the unknown neighboring area. The proposed approach loops through all the voxels and checks using *OctoMap* utilities if the voxel is free and has an unknown neighbor. If so, the voxel is added to the list of frontiers.

As the complete set of frontiers is rather large for a small map resolution, a clustering algorithm is applied. This is done to reduce the number of potential candidates but preserve the information regarding exploration potential in terms of the number of frontier points in a frontier cluster. The clustering algorithm that is used in this method is the divisive K-means clustering algorithm. This method splits the initial set into two groups and recursively clusters the points in these groups based on Euclidean distance till all the clusters meet the size requirement.

$$r_{max} \leq \tan(\frac{\mathrm{FoV}_{hor}}{2})d_{max} \tag{4.2}$$

Here, r_{max} represents the maximum distance between a frontier point and the centroid of the cluster to which that frontier point belongs. Furthermore, d_{max} is defined as the maximum range of the depth camera. FoV_{hor} stands for the horizontal field of view of the depth camera. The divisive K-means clustering parameters are visualized in Figure 4.2. This constraint ensures that the entire cluster can be observed from a single visit, reducing the need for redundant repositioning and ensuring efficient exploration.

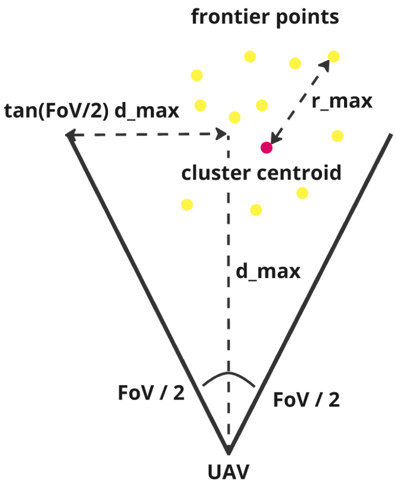


Figure 4.2: Illustration of divisive K-means clustering parameters. Frontier points are divisively clustered till the cluster can be explored in one field of view.

This clustering method provides two main advantages: because the divisive clustering algorithm clusters top-down, there is no need to determine a K-value for the final number of clusters. This is especially useful in an exploration scenario, as the total set size is unknown and thus finding an optimal number of clusters is impractical. Secondly, this provides a way to reduce the number of potential waypoints but preserves the information regarding exploration potential. This is because both the centroid of the cluster as well as the cluster size, that is, the number of points within that cluster, are included in the data structure of the clusters. This list of candidate clusters is then filtered on reachability by checking the direct neighboring voxel of the candidate cluster centroid and assuring no known occupied voxels exist within space. This prevents selecting goals that would result in infeasible or collision-prone trajectories.

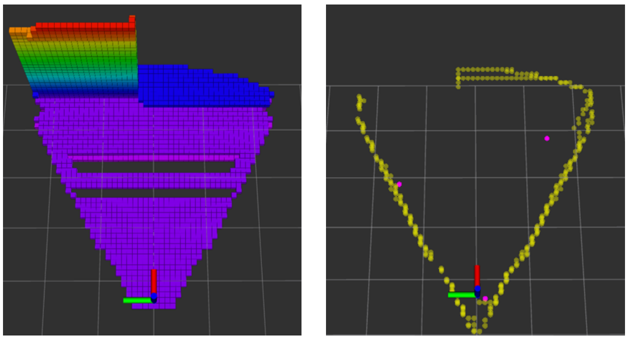
To support trajectory generation in the global planning module (subsection 4.4.1), multiple candidate viewpoints are sampled around each candidate cluster centroid, only viewpoints that lie in free space are considered. This increases the likelihood of finding a collision-free path toward the cluster. For each sampled viewpoint, a target yaw angle is computed that orients the UAV toward the cluster centroid. This ensures that, upon arrival, the UAV is oriented to observe the entire frontier region associated with the cluster. A visualization of this process is shown in Figure 4.4.

Since each candidate requires a corresponding trajectory for energy evaluation, the total number of clusters is further reduced. This is necessary to limit the computational load of the global planner, which would otherwise be required to generate full trajectories for all clusters which would be too time-consuming. The proposed method selects the three clusters with the largest information gain. This information gain is calculated by:

$$IG(c_i) = \text{COUNT}(f) \tag{4.3}$$

where f represents a frontier voxels within cluster c_i. This ensures that smaller clusters that would not provide a significant amount of new information are omitted. This final list of feasible candidates clusters is the input for the energy-aware module.

The process of mapping the environment, and detecting and clustering frontiers using *OctoMap*, is visualized in Figure 4.3a and Figure 4.3b, respectively.



(a) Visualization of the 3D environment representation using OctoMap, occupied space colored based on height.

(b) Frontier detection based on OctoMap data in yellow and corresponding frontier clusters in pink.

Figure 4.3: 3D environment representation and frontier clustering.

4.3.2. Energy-awareness Module

After frontiers are detected, clustered, and filtered, the exploration cycle starts. First, for all candidate clusters, a global trajectory is planned using the offline, global planner described in subsection 4.4.1. This planner generates full trajectories from the current UAV position towards the candidate clusters' positions. These are later evaluated using the *Agilicious* framework. The steps as well as the rationale behind this approach are discussed in this subsection.

Offline planning refers to the computation of a complete trajectory from an initial state to a target state before execution, without any feedback or re-planning during execution. This approach is different from online planning, where the trajectory is generated incrementally during flight based on sensor input and environmental changes, as used in the local reactive planner (subsection 4.4.2). Offline trajectory planning allows for reasoning based on a trajectory before it is executed. As discussed by Shiller [29], offline planners incorporate the full system dynamics and actuator limits to generate optimal trajectories using numerical optimization methods. Details on the global offline planner used in this work are further described in subsection 4.4.1 and section 5.1.

As discussed in chapter 2, various approaches have been proposed to incorporate energy-awareness into autonomous UAV planning. These methods differ primarily in how energy consumption is estimated. The most common strategies include approximating energy based on geometric metrics such as distance, estimating energy from flight-state or velocity-dependent models. Each of these methods offers different advantages and disadvantages.

To motivate the use of the offline planning and execution approach adopted in this work, a comparison is presented in Table 4.1. This table summarizes the key advantages and disadvantages of alternative methods relative to the proposed solution.

Table 4.1: Comparison of energy-aware methods relative to the proposed offline execution approach.

Method	**Advantages**	**Disadvantages**
Empirical flight-state-based estimation [26]	Fast; Simple implementation; No full trajectory required	Needs trajectory segmentation; Inaccurate; Requires empirical data;
Velocity-based energy-awareness [5]	Relatively easy to implement; Generalizable; Low computational load	Planning not informed by energy component; Not suitable for cluttered environments
Distance-based estimation [25]	Very simple estimation; Generalizable; Low computational load	Ignores UAV dynamics; Assumes constant power draw; Does not distinguish between aggressive vs. smooth maneuvers
Offline execution (This work)	Accurate; No proxy approximation; Generalizable	Slow for many candidates; Requires offline-planning;

While each method has its advantages, their respective disadvantages highlight important limitations in the context of energy-aware exploration. The empirical flight-state-based approach as described in [26] is attractive due to its simplicity and fast execution; however, its dependence on pre-segmented trajectories and platform-specific data makes it less accurate and harder to generalize across UAV types or flight conditions.

Velocity-based awareness methods, as in [5], offer a lightweight and generalizable solution, but they make indirect assumptions that the UAV is able to reach the optimal velocity which may not hold on complex or cluttered environments.

Distance-based methods, as in [25] provide an extremely simple solution to energy-awareness incorporation. However, estimating energy only based on euclidean distance results in inaccurate estimates for scenarios involving acceleration, deceleration, or sharp turns. These models inherently ignore flight dynamics and treat energy cost as a static, geometry-only metric.

In contrast, the offline execution approach used in this work provides an accurate estimate of the actual energy needed to reach each candidate for all trajectories and flight phases. It accounts for UAV dynamics, without relying on simplified motion assumptions. Although more computationally expensive, it allows for reliable, platform-independent energy estimation. This is required for making informed planning decisions.

Once the candidate trajectories are generated, they are passed to the energy estimation module for evaluation. This module executes each trajectory offline using the *Agisim* physics simulator from the *Agilicious* framework [30]. For every setpoint along the trajectory, the geometric controller computes a control command. These commands are then applied iteratively in the simulator at fixed time steps, updating the UAV state accordingly. During this virtual execution, the rotor speeds are logged, allowing the computation of instantaneous power and the total energy required to execute the trajectory. This process yields an accurate energy estimate for each candidate path without requiring real-world flight trials.

4.4. Planning

As previously stated, the proposed exploration strategy utilizes two distinct planning modules: a global planner for offline trajectory generation and energy estimation, and a local reactive planner for real-time obstacle avoidance. This dual-planner architecture addresses the limitations of the reactive planner in terms of full path planning. Specifically, the local planner is designed to compute short-range, dynamically feasible trajectories based on the current sensor horizon, approximately 1.5 times the range of the depth camera. While effective for collision avoidance, this limited planning horizon is insufficient for evaluating the required energy for visiting a candidate cluster. To overcome this, a global planner is integrated to generate full, obstacle-free trajectories to each candidate cluster centroid. These globally planned trajectories enable energy estimation and informed target selection before execution of the trajectory.

Preferably, a single planner capable of both global trajectory generation for energy estimation and real-time local obstacle avoidance would be employed. However, to the best of the author's knowledge, such an integrated solution was not available at the time of writing. This limitation is acknowledged and discussed in chapter 7, and proposed as a direction for future work in chapter 8. In the absence of such a solution, the separation into two specialized planning modules enables the system to leverage the strengths of each.

4.4.1. Global Planning

To support energy-aware decision-making, a global planner is integrated into the exploration pipeline. This module generates dynamically feasible trajectories from the UAV's current position to each candidate cluster, ensuring obstacle avoidance and adherence to flight constraints in terms of maximum velocity and acceleration. These trajectories span the full path to a target, making them suitable for energy evaluation. By providing complete and feasible paths, this global planning component enables informed target selection based not only on spatial characteristics but also on anticipated energy consumption.

4.4.2. Local Reactive Planning

Once a goal waypoint is selected, a trajectory is generated using the local reactive planner, ensuring safe flight. The local reactive planner that is implemented generates a trajectory to the selected target frontier while ensuring collision avoidance using real-time depth data from the camera sensor. The planner uses odometry data from the flight control module. The final trajectory is formatted according to the Agilicious trajectory message type, which includes the time and the position, velocity, and acceleration components in all three dimensions.

Unlike the global planner, the local planner operates with a distance horizon. It does not attempt to plan over the full global map but instead ensures safety within the UAV's immediate environment. This reactive behavior is useful for future implementation in environments with uncertainty or dynamic obstacles.

4.5. Control and Simulation Framework

For control and simulation, the *Agilicious* framework is utilized [30]. This is an open-source and open-hardware agile quadrotor platform designed for high-performance flight control. Agilicious provides a modular architecture that supports various UAV configurations, making it suitable for generalizable energy-aware planning and control research.

The framework integrates a real-time control stack with a simulation module, making it capable of executing computationally intensive tasks, such as energy estimation using the offline-planned trajectory. Simulation is performed using a combination of *Gazebo*, which provides the sensor models and simulated environment (e.g., RGB-D data), and *Agisim*, *Agilicious*'s built-in physics simulator, which accurately models quadrotor dynamics and responses.

A key component in the control architecture is the controller. In this study, the geometric controller is used. This controller provides fast and accurate trajectory tracking and has been demonstrated to support aggressive flight maneuvers with minimal computational latency [30].

Importantly, Agilicious supports both geometric and model predictive controllers (MPC), offering flexibility depending on the performance requirements or hardware constraints of future implementations. This modularity not only enhances the applicability of this method but also ensures that both the energy-aware module and control stack can be used for different UAV platforms.

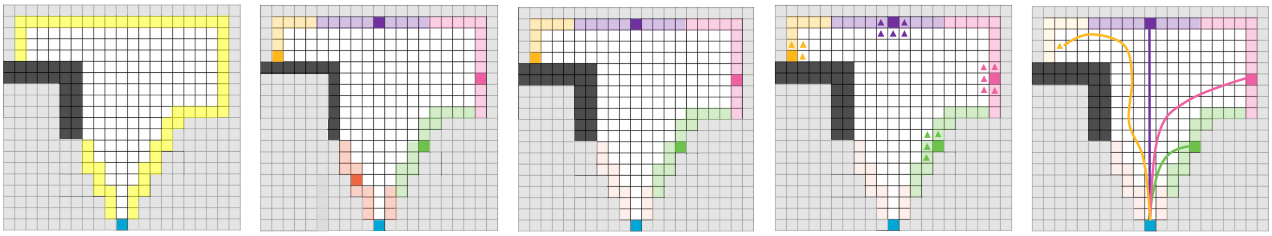
4.6. Summary

This chapter presented the Energy-Aware Autonomous Explorer (EAAE), a modular framework for autonomous UAV exploration in unknown 3D environments. The approach integrates perception, exploration, planning, and control into a single system that incorporates energy-awareness into the decision-making process.

Perception and mapping are achieved using an RGB-D camera and an occupancy grid representation, enabling continuous identification of frontiers. These frontiers are clustered using a divisive K-means algorithm, reducing computational load while preserving exploration-relevant information. Clusters are filtered for reachability, and viewpoint sampling is applied to support trajectory generation.

To enable energy-informed target selection, a global planner generates full, dynamically feasible trajectories to candidate clusters. These trajectories are virtually executed using the *Agilicious* framework to compute accurate energy estimates. The candidate requiring the least energy is selected as the next exploration target. For trajectory execution, a local reactive planner ensures safe navigation using real-time sensor data.

The overall cycle—from frontier detection to goal selection and execution—is summarized in Figure 4.4.



(a) (b) (c) (d) (e)

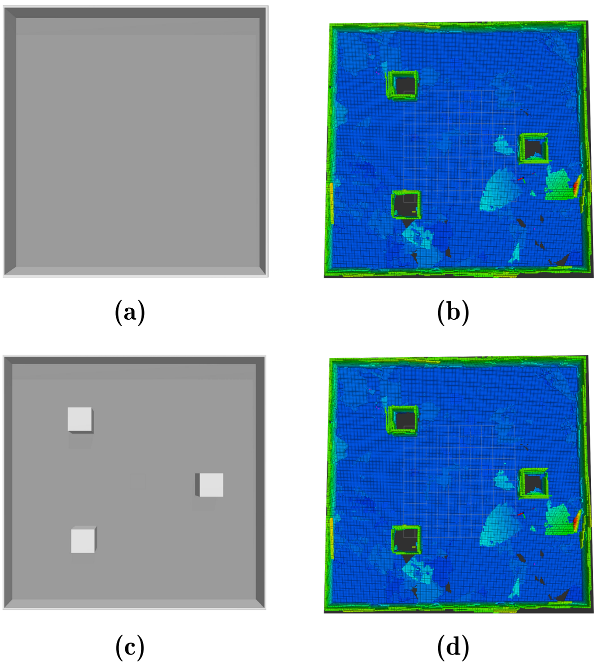
Figure 4.4: Overview of the frontier-based exploration cycle. (a) Represents the frontier detection visualized in yellow, being the boundary between unknown (grey) and known free (white) areas. (b) Frontier clustering. (c) Filtering of inaccessible or too-near clusters. (d) Sampling of viewpoints around the cluster centroid. (e) Global path planned towards the waypoint.

This pipeline enables energy-efficient and collision-free exploration in unknown environments.

5 Experiment Setup

In order to evaluate the performance of the energy-aware exploration algorithm, various experiments are performed in a simulated environment using the *Agisim* simulator from the *Agilicious* framework and *Gazebo*.

The virtual environments are shown in Figure 5.1. These worlds are referred to as the *Simple* and *Pillars* environments, respectively.



(a) (b) (c) (d)

Figure 5.1: (a) Simulated *Simple* environment in *Gazebo* with dimensions 20x20x2.5 [m]. (b) Fully explored *Simple* environment using *Octomap*. (c) Simulated *Pillars* environment in *Gazebo* with dimensions 22x22x2.5 [m]. (d) Fully explored *Pillars* environment using *Octomap*.

5.1. Simulation

For reproduction purposes, this section elaborates on the details of the approach as explained in chapter 4. Detailed simulation parameters, including UAV model specifications, depth camera parameters, simulation parameters, and energy estimation settings, are summarized in Table 5.1. The method proposed in this research is built, and evaluated on a laptop with an Inter Core i7-13700H@5.0GHz processor, 16GB of RAM and Ubuntu 20.04 LTS as the operating system. The majority of the code is written in C++.

Perception and Mapping

The simulated platform is a quadrotor UAV equipped with a forward-facing RGB-D camera, which serves as the primary perception sensor. The camera provides depth and color information, enabling dense point cloud generation used for mapping and real-time obstacle avoidance. These sensing capabilities are important for both frontier detection and safe trajectory planning.

For 3D environment representation and mapping, an occupancy grid is generated using the *OctoMap* framework [28]. *OctoMap* models the space as a hierarchical tree of cubic voxels (octrees), where each voxel contains a probabilistic estimate of occupancy based on incoming depth measurements. This structure offers a memory-efficient and scalable solution for real-time mapping in virtual 3D environments. The map is continuously updated during flight using incoming point cloud data, with voxels probabilistically classified as *free*, *occupied*, or *unknown*. This occupancy information serves as the foundation for frontier detection by identifying free voxels that border unknown space. These frontiers are used to guide exploration towards unmapped regions of the environment. The mapping resolution is set at 0.1 m. This resulted in sufficient detail for exploration whilst keeping computational load manageable. For larger map sizes, it is advised to lower the map resolution.

Exploration Strategy

As discussed in chapter 4, frontiers are detected based on the *OctoMap* data as a boundary between known free and unknown space. As planning a trajectory towards all frontier points is computationally infeasible, a clustering algorithm is applied. This implementation uses a divisive K-means clustering approach that recursively splits the frontier set based on Euclidean distance. Clustering continues until a spatial constraint is met: all points within a cluster must lie within the field of view of the onboard camera from a single observation point. The proposed method uses a depth camera range of 5.0 m and a horizontal field of view of one radial, so this results in a maximum cluster radius of 2.73 m using Equation 4.2.

For each cluster, additional feasibility filtering is applied. A candidate cluster is rejected if occupied voxels are present in the immediate surroundings (within a 0.1 m radius in the x–y plane), ensuring the UAV is not sent towards hard-to-reach goals. Such an example scenario is visualized in Figure 5.2.

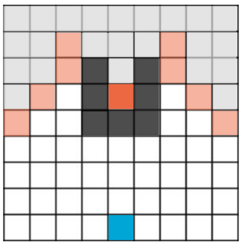


Figure 5.2: Illustration of a possible hard-to-reach cluster which would be detected and removed from the set of candidate clusters.

To further support path planning and visibility optimization, multiple viewpoints are sampled around each candidate cluster centroid. These viewpoints are sampled at 0.5 m from the centroid in all three dimensions.

Once the frontier clusters are identified, the energy-aware module uses the global trajectories for these candidates to calculate the amount of energy it would take to reach them. This is done using the global planner as described in **??**. Energy is estimated via offline execution of the trajectory in *Agilicious*, using rotor speed outputs and the following model:

$$E = \sum_{i=1}^{n} \int P_i(t)\, dt \tag{5.1}$$

The power $P_i(t)$ of each rotor is modeled as a nonlinear function of its angular velocity $\omega_i(t)$, according to the following empirical relation:

$$\begin{aligned} P_i(t) = 6.088 \times 10^{-3} \cdot \omega_i(t) + 1.875 \times 10^{-8} \cdot \omega_i^3(t) \\ + 7.700 \times 10^{-20} \cdot \omega_i^6(t) \end{aligned} \tag{5.2}$$

Here, n denotes the number of rotors (in this case, $n = 4$), $P_i(t)$ is the instantaneous power output of rotor i at time t, and $\omega_i(t)$ is the corresponding angular velocity expressed in radians per second (rad/s). This follows the empirical relation derived by Bauersfeld and Scaramuzza [4], which accounts for mechanical power and an electrical loss term for brushless motors. The proposed method selects the candidate that requires the least amount of energy to be reached.

Planning

To enable global trajectory planning and facilitate energy estimation, we adopt the risk-aware kinodynamic A* algorithm and trajectory optimization framework from Chen et al. [31]. This planner first creates a topological path through the free space using a modified RRT* algorithm that considers map connectivity and obstacle clearance. The resulting path is then inflated into a series of convex polyhedra, forming a safe corridor that encapsulates feasible free space. A time allocation strategy is employed across the corridor segments to enforce dynamic feasibility. Within this corridor, a minimum-snap trajectory is optimized using B-splines, subject to corridor, velocity, and acceleration constraints. The smoothness and completeness of the generated trajectory make it suitable for offline execution, as a full feasible trajectory is required for energy estimation.

This planner first attempts to generate a trajectory directly to the centroid of a candidate frontier cluster. If this attempt fails, it proceeds to evaluate the set of sampled viewpoints associated with that cluster. A cluster is discarded if no feasible trajectory can be found to either its centroid or any of its viewpoints. In practice, such failures may arise due to the absence of a collision-free path within the currently explored free space or because the trajectory optimizer is unable to compute a solution that satisfies the imposed dynamic and geometric constraints.

Once the goal is selected, the local planner generates a reactive trajectory towards to goal cluster. For this module the the open-source planner proposed in [32] is used, called *EGO-planner*. The odometry data from the UAV's control module as well as the global point cloud data from the perception and mapping module, are utilized as inputs. Using these inputs, *EGO-planner* generates a trajectory that is translated into an *Agilicious*-conform type using a translator node to ensure compatibility between the planner and the control module.

Control and Simulation Framework

For control and simulation the open-source, open-hardware flight control and simulation framework *Agilicious* is utilized. This framework is in further detail described in [30]. Because of its highly modular design, *Agilicious* can be used for various real-life and simulation scenarios. The proposed method uses a geometric controller and the UAV is modeled in the *Agilicious*'s own physics simulator, *Agisim*. The UAV is spawned as a static object in *Gazebo* to retrieve continuous point cloud data of the *Gazebo* environment.

Table 5.1: Experimental and simulation parameters for energy-aware autonomous exploration.

Category	**Parameter**	**Value**
UAV	Rotor Count	4
	Max Velocity [m/s]	5.0
	Max Acceleration [m/s^2]	4.0
	Mass [kg]	0.752
Depth camera	Image Resolution [px]	640 × 480
	Depth Range [m]	0.5 – 5.0
	Frame Rate [Hz]	30
	Horizontal FoV [rad]	1.0
Simulation	Octomap Resolution [m]	0.1
	Map Size: Simple [m]	20 × 20 × 2.5
	Map Size: Pillars [m]	22 × 22 × 2.5
	Sampling & control frequency [Hz]	300
Energy Estimation	Power Model Coefficients	Equation 5.2
	Timestep energy calculation [s]	0.02
	Integration Method	Midpoint
Hardware	Intel Core i7-13700H@5.0GHz, 16GB memory	
ROS Version	Noetic	

The proposed framework is publicly available at: `https://github.com/jelskamp/EAAE`, to support future research.

5.2. Baselines

To evaluate the impact of incorporating energy-awareness into autonomous exploration, the proposed method is compared against two widely adopted frontier-based exploration methods. Both baseline methods are implemented by the author, based on established techniques from the literature, to ensure consistent integration and comparability within the simulation framework.

Classic Frontier Method

The first baseline is a classical frontier-based exploration strategy in which the UAV always selects the frontier cluster with the largest number of frontier voxels. This method follows the same frontier detection and divisive K-means clustering procedure as described in the proposed method (chapter 4), but uses a simpler selection criterion. Specifically, it ranks all reachable clusters based on their frontier voxel count and selects the cluster $c^\star$ according to:

$$c^\star = \arg\max_{c_i \in \mathcal{C}} \mathrm{COUNT}(f \in c_i) \tag{5.3}$$

where $\mathcal{C}$ is the set of all feasible frontier clusters, and f denotes a frontier voxel. This approach implicitly assumes that the largest visible frontier is the most informative region to explore next.

Advantages of this method include its simplicity, computational efficiency, and the potential to rapidly gather information regarding the environment in early stages of this exploration mission. However, it does not consider the cost of reaching the selected frontier. As such, it may favor distant frontiers that require long or energy-intensive trajectories, which is suboptimal for resource-constrained systems such as UAVs.

Nearest Frontier Method

The second baseline uses a greedy nearest-frontier selection strategy, in which the UAV always selects the closest feasible frontier cluster. After frontier detection and clustering, the distance d_i from the UAV's current position $\mathbf{p}_{\text{uav}}$ to the centroid of each cluster c_i is computed. The next goal $c^\star$ is selected as:

$$c^\star = \arg\min_{c_i \in \mathcal{C}} \|\mathbf{p}_{c_i} - \mathbf{p}_{\text{uav}}\|_2 \tag{5.4}$$

where $\mathbf{p}_{c_i}$ is the centroid position of cluster c_i and $\|\cdot\|_2$ denotes the Euclidean norm.

Advantages include fast decision-making and low computational load. This method prioritizes frontiers that are immediately accessible and minimizes the travel distance per exploration step. It is particularly suitable for cluttered environments where long-range planning is often obstructed. However, as this method seeks out the nearest goal, it means it may fail to prioritize highly informative regions or make globally efficient exploration decisions. This can result in suboptimal coverage efficiency and increased mission duration in large or sparse environments.

Due to the method's incremental nature, this method explores the environment gradually. If a nearby region is unintentionally missed, it may only be revisited much later, potentially when the UAV is already far away, leading to increased total mission time.

5.3. Measurements

To evaluate the performance of the proposed energy-aware exploration algorithm, three key metrics are used: exploration rate, map information entropy, and total energy consumption. Each method is evaluated over five independent simulation runs in both the

Simple and *Pillars* environments. The results presented in chapter 6 represent the performance across these runs to ensure robustness.

Exploration Rate

An important performance metric to assess autonomous exploration algorithms is the exploration rate. This metric is defined as the percentage of the map that is explored over time. An effective algorithm rapidly achieves a high percentage of area coverage, approaching complete coverage (100%) within a short time frame. The formal expression is:

$$\text{Exploration Rate}(t) = \frac{|V_{free}(t) \cup V_{occupied}(t)|}{|V_{total}|} \times 100\%$$

where $V_{free}(t)$ and $V_{occupied}(t)$ are the sets of free and occupied voxels at time t, respectively, and V_{total} denotes the total number of voxels in the environment.

Information Entropy

A second performance metric used in this research is the information entropy of the explored map, also known as Shannon's entropy [33]. Information entropy is an indicator of uncertainty and knowledge acquisition. The information entropy for the i^{th} cell is defined as the expectation of the logarithm of the probability distribution $\mathbb{P}_i(x_i)$:

$$H_i \equiv -\mathbb{E}[\log \mathbb{P}_i(x_i)] = -\sum_{x_i \in X_i} \mathbb{P}_i(x_i) \log \mathbb{P}_i(x_i) \quad (5.5)$$

For this experiment, two possible states exist: occupied (with probability $p = \mathbb{P}_i(\text{occupied})$) and known free (with probability $1 - p$). This allows the information entropy for a cell to be expressed as:

$$H_i[x_i] = -p \log(p) - (1 - p) \log(1 - p) \quad (5.6)$$

As stated above, the information entropy is a measure of uncertainty. For illustration of the meaning of this metric, consider the following example. In the event of a 'perfect' exploration mission, all cells are labeled with 100% confidence. This means all cells have a probability to be either definitely occupied ($p = 1.0$) or definitely free ($p = 0.0$). This would result in zero total information entropy ($H = 0$ bits)). In a worst-case scenario exploration mission, all cells have a probability of $p = 0.5$ to be occupied. This results in the highest possible information entropy of $H = 1$ bit per cell.

Energy Consumption

Lastly, the consumed energy for the method is evaluated in both environments, calculated using rotor speed data retrieved from the UAV's state in simulation. The underlying model used for energy computation is detailed in chapter 4, specifically in Equation 5.1 and Equation 5.2. At each control timestep $\Delta t = 0.02$ s, the instantaneous power $P_i(t)$ of rotor i is computed using the empirical model from Equation 5.2. The total energy is then given by:

$$E = \sum_{i=1}^{4} \sum_{t=0}^{T} P_i(t) \cdot \Delta t \quad (5.7)$$

This cumulative energy value aims to accurately represent the actual energy consumption if the UAV were to fly this trajectory. As described in the problem statement in chapter 3, one of the objectives of this research is to minimize this value whilst also keeping the exploration time at a minimum.

6 Results

This chapter presents the results of the proposed EAAE algorithm in comparison with two baseline methods: Classic Frontier and Nearest Frontier exploration these baselines are discussed in section 5.2. All methods are evaluated using identical initial conditions, with the UAV starting at position $(0, 0, 1)$ in both the *Simple* and *Pillars* environments. The results are based on five independent runs per method per environment, ensuring statistical robustness. An overview of the key performance metrics, including exploration time, total energy consumption, and map entropy, is summarized in Table Table 6.3. A detailed analysis is provided in the following sections, supported by visualizations and additional tables. The goal of this chapter is to evaluate the impact of integrating energy-awareness into the exploration pipeline in terms of efficiency, mapping quality, and computational performance.

6.1. Exploration Rate

Exploration rate is a critical performance metric in autonomous exploration, indicating how quickly the algorithm can uncover previously unknown regions of the environment. Figure 6.1 and Figure 6.2 present the percentage of explored voxels over time for each method in the *Simple* and *Pillars* environments, respectively. Each result is averaged over five trials, with the UAV always initialized at $(0, 0, 1)$.

***Simple* environment:** in this environment, our proposed method, EAAE, achieves the fastest total exploration time, completing the task in an average of

160.0 seconds. This corresponds to a 10.3% and 12.5% reduction in total time compared to the Classic and Nearest baseline methods, which complete the mission in 178.3 and 182.8 seconds, respectively. The Classic method, which greedily selects the cluster with the highest information gain (see section 5.2), initially exhibits the steepest exploration rate due to its prioritization of high-yield areas in the open space. However, as shown in the curve convergence near 90% map completion (around $t = 110s$), this initial efficiency diminishes as the method must revisit overlooked regions. This behavior aligns with the known drawback of greedy strategies: they may maximize short-term gain while neglecting coverage completeness.

EAAE, by contrast, balances between information gain and energy cost. It initially progresses more cautiously but maintains steady performance throughout the mission, avoiding expensive corrective maneuvers in the final phase. This allows it to overtake the Classic method in the final quarter of the mission. The Nearest Frontier method, which always selects the closest reachable cluster, lags in early performance but gradually converges to full coverage. Its locally optimal yet globally myopic decisions result in inefficient backtracking and delayed completion.

***Pillars* environment:** In the more complex *Pillars* environment, the ranking reverses: the Nearest method completes exploration fastest (321.7 s), followed by EAAE (353.0 s) and Classic (366.6 s). This result illustrates the environment-dependent nature of frontier strategies. The Nearest method, while globally inefficient in open spaces, benefits from its conservative behavior in cluttered scenarios. Consequently, it avoids extensive long-range replanning, a drawback that becomes more apparent for the Classic and EAAE methods in this setting.

Nevertheless, it is worth noting that although the Classic and EAAE methods achieve higher exploration rates earlier in the mission, both slow down after reaching approximately 75% coverage. This drop-off is likely due to remaining unexplored regions being scattered and harder to reach, a result of the long-range decisions that may skip over small occluded regions. EAAE, however, still outperforms the Classic method by nearly 4%, suggesting that its energy-awareness helps mitigate—but not entirely eliminate—the inefficiencies of long-horizon planning in dense environments.

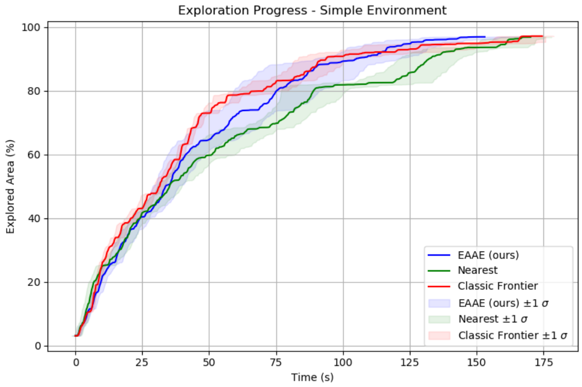


Figure 6.1: Exploration progress for all three algorithms in the *Simple* environment.

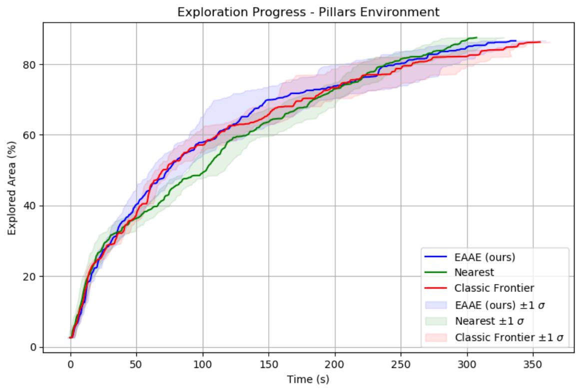


Figure 6.2: Exploration progress for all three algorithms in the *Pillars* environment.

6.2. Map Information Entropy

A second performance metric used to assess exploration quality is the information entropy of the occupancy map. As defined in section 5.1, entropy quantifies the uncertainty of voxel classifications based on the probabilistic values generated by *OctoMap*. Even though all methods rely on the same depth sensor configuration (see Table 5.1), the resulting entropy values can vary significantly. This is because *OctoMap* does not store binary free/occupied labels per voxel, but rather updates voxel occupancy as a probabilistic belief using recursive Bayesian integration [28]. Each point cloud update changes the log-odds occupancy value for the corresponding voxels, depending on the number, direction, and quality of observations.

In particular, the order and frequency of observations, as well as the viewpoint angles, determine whether a voxel becomes confidently classified or remains in an uncertain state. Voxels that are observed only once or from oblique angles may fail to meet the clamping thresholds used in OctoMap and thus remain close to $p = 0.5$, contributing to higher entropy.

***Simple* environment:** In this environment, the Classic method achieves the lowest average map entropy (0.719 bits/cell), followed closely by the proposed method, EAAE (0.725), and the Nearest method (0.735). This result aligns with the initial high exploration rate of the Classic method, as discussed in section 6.1. Since Classic prioritizes clusters with the largest frontier count, it aggressively scans large open areas early in the mission, leading to many direct and repetitive observations from different angles, resulting in a more certain occupancy belief. This enables confident voxel classification and thus lower entropy.

***Pillars* environment:** In the more cluttered Pillars environment, EAAE achieves the lowest average entropy (0.754 bits/cell), followed by Classic (0.767) and Nearest (0.784). The Classic method's preference for high-information clusters can become a disadvantage in cluttered environments, where navigating toward these areas may result in fewer revisits and more occlusions. Consequently, parts of the environment may be viewed only once or from oblique angles, leaving more voxels in an uncertain state. In contrast, the EAAE method's balanced trade-off between information gain and energy cost leads to smoother trajectories. This likely results in more effective visits in terms of map certainty, contributing to a lower entropy.

The Nearest method performs worst in both environments, likely due to its reactive behavior. By prioritizing short, local movements, it rarely revisits areas from different angles. This limited number of viewpoints leads to less confident classifications and higher overall map entropy. A statistical overview of the entropy results is included in Table 6.3

6.3. Energy Consumption

This section evaluates the total energy and average power consumption of the proposed method compared to the Nearest and Classic Frontier baseline methods. Table 6.3 presents the summary statistics for both environments, and Figure 6.3 and 6.4 visualize the energy distribution per method over five runs.

Energy

In both the *Simple* and *Pillars* environments, the proposed EAAE method consistently demonstrates the lowest total energy consumption. In the *Simple* environment, EAAE consumes on average 21.2 kJ, compared to 30.3 kJ for Nearest and 22.2 kJ for Classic (Table 6.3). This difference is particularly pronounced when compared to Nearest, with EAAE requiring approximately 30% less energy to complete the mission. However, the difference with Classic is smaller, only 1.0 kJ on average, despite EAAE completing the exploration task about 10.3% faster (see section 6.1).

This small difference in energy use can be attributed to the integral relationship $E = \int P(t)\,dt$. In the *Simple* environment, without obstacles, long trajectories can be executed without frequent replanning and stopping. This allows the UAV in the Classic method to plan relatively long, straight paths that require less power on average, thus resulting in a relatively low total consumed energy even though the exploration time is 10.3% longer.

In the more cluttered *Pillars* environment, the differences become larger. EAAE consumes 45.0 kJ on average, compared to 57.0 kJ for Nearest and 51.4 kJ for Classic, representing energy savings of 21% and 12.5%, respectively. This further supports the effectiveness of energy-aware trajectory selection in environments where longer, straight trajectories are less common.

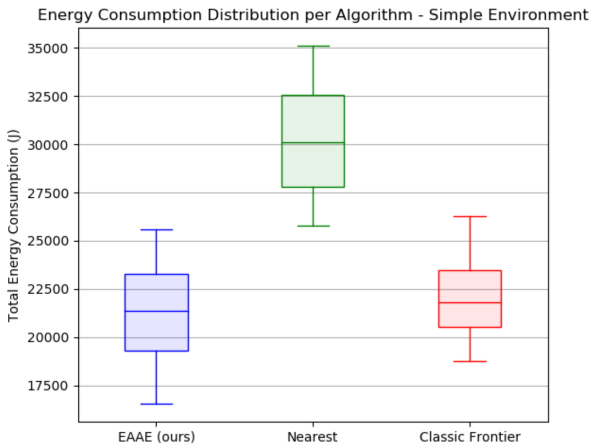


Figure 6.3: Energy consumption distribution for all three algorithms in the *Simple* environment.

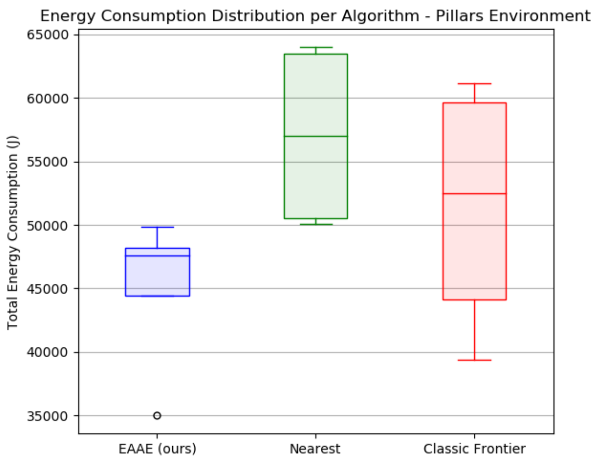


Figure 6.4: Energy consumption distribution for all three algorithms in the *Pillars* environment.

Power

To better understand the relationship between mission duration and energy usage, we examine the average power draw per method (Table 6.1). In the *Simple* environment, Classic Frontier exhibits the lowest average power draw at 136.3 W, followed closely by EAAE (139.2 W). Nearest has the highest average power draw at 167.2 W, reflecting its more aggressive maneuvering style and inefficient back-and-forth movements.

In the *Pillars* environment, EAAE exhibits the lowest average power draw (133.7 W), versus 134.8 W and 168.4 W, from the Classic Frontier and Nearest methods, respectively. It is important to note that average power consumption is not a direct indicator of energy efficiency unless paired with mission duration. The proposed method demonstrates both a favorable power profile and shorter mission durations in the *Simple* environment, leading to overall energy efficiency. In contrast, the Nearest method, while completing the *Pillars* environment fastest, suffers from a significantly higher average power draw and thus has the highest total energy consumption. This trade-off between exploration speed and energy-efficient paths is further discussed in **??**

Table 6.1: Average power consumption per method in the *Simple* and *Pillars* environments.

Scene	Method	Power (W) Avg	Power (W) Std
Simple	EAAE (ours)	139.24	16.65
	Nearest	167.23	32.36
	Classic Frontier	**136.33**	**15.31**
Pillars	EAAE (ours)	**133.70**	**14.80**
	Nearest	168.36	46.75
	Classic Frontier	134.80	16.86

6.4. Computational Efficiency

In addition to exploration performance and energy consumption, this section presents the computational characteristics of the proposed method to assess its feasibility for possible onboard deployment. Table 6.2 summarizes the average computation time per module, measured across all runs in both environments.

The results indicate that the majority of computational resources are consumed by the trajectory generation module. In both environments, this module accounts for approximately 98% of the total computation time, with an average duration of 3331.6 ms per step in the Simple environment and 1887.0 ms in the Pillars environment. In contrast, other modules such as clustering and energy estimation require significantly less computation time, averaging below 50 ms. Modules like candidate filtering, viewpoint sampling, and target selection were consistently below 1 ms per execution and are therefore not included in the table.

These results highlight that global trajectory generation is the primary computational bottleneck in the EAAE pipeline, while the remaining modules are computationally relatively lightweight and suitable for real-time onboard use.

Table 6.2: Average computation time per module for each environment for the proposed method.

Scene	Average computation time (ms) Clustering Sect. 4.3.1	Trajectory Gen. Sect. 4.4.1	Energy Est. Sect. 4.3.2
Simple	17.8	3331.6	45.2
Pillars	10.0	1887.0	34.4

6.5. Summary

This chapter presented a comparative evaluation of the proposed Energy-Aware Autonomous Explorer (EAAE) against two baseline frontier-based exploration methods: Classic Frontier and Nearest Frontier. Experiments were conducted in two environments, *Simple* and *Pillars*, with five independent runs per method per environment. Performance was assessed across four dimensions: exploration rate, map entropy, energy consumption, and computational efficiency.

In terms of exploration rate, EAAE achieved the shortest completion time in the *Simple* environment, outperforming both baselines. In the more cluttered *Pillars* environment, the Nearest method completed exploration fastest. The Classic method performed competitively in early stages of both environments but slowed down due to inefficient backtracking to missed areas in later stages.

With respect to mapping quality, measured by information entropy, EAAE achieved the lowest entropy in the *Pillars* environment, while Classic achieved the lowest in the *Simple* case. The Nearest method consistently resulted in the highest entropy values, indicating less confident map representations.

Energy consumption analysis showed that EAAE had the lowest total energy usage in both environments, most notably in *Pillars*, where it achieved savings of over 20% compared to Nearest. This advantage is attributed to the integration of energy-aware trajectory selection. Power consumption results showed that while EAAE had the lowest power draw in *Pillars*, the Classic method was marginally more efficient in *Simple*, aligning with its longer but less aggressive flight style.

Finally, computational efficiency measurements revealed that trajectory generation accounted for approximately 98% of total computation time across environments. Other modules, including clustering and energy estimation, required minimal resources, con-

Table 6.3: Performance Statistics per method in the Simple and Pillars environments.

Scene	Method	Exploration time (s)				Energy (kJ)				Entropy (bits/cell)			
		Avg	Std	Max	Min	Avg	Std	Max	Min	Avg	Std	Max	Min
Simple	EAAE (ours)	**160.0**	23.9	**190.1**	**125.8**	**21.2**	3.3	**25.6**	**16.5**	0.725	**0.004**	**0.754**	0.703
	Nearest	182.8	25.3	220	154	30.3	3.5	35.1	25.8	0.735	0.014	0.770	0.687
	Classic	178.3	**22.5**	214	152.4	22.2	**2.7**	26.3	18.8	**0.719**	0.016	0.761	**0.670**
Pillars	EAAE (ours)	353.0	46.7	**389.9**	273.6	**45.0**	**5.8**	**49.8**	**35.0**	**0.754**	0.009	**0.802**	**0.669**
	Nearest	**321.7**	**44.4**	390.2	**266.8**	57.0	6.7	63.9	50.1	0.784	0.011	0.834	0.744
	Classic	366.6	72.6	463.7	289.6	51.4	9.1	61.1	39.4	0.767	**0.004**	0.821	0.741

firming their suitability for real-time onboard execution. A full summary of average and variability metrics is included in Table 6.3.

7 Discussion

This chapter reflects on the findings presented in chapter 6 and evaluates their implications and limitations. While the proposed EAAE algorithm demonstrates improved energy efficiency and competitive exploration performance, several aspects of the experimental setup and methodology may influence the generalizability of the results.

Limited Environmental Diversity: The results presented in this work are based on two simulated test environments: *Simple* and *Pillars*. Although chosen for their different levels of spatial complexity, both still represent a narrow subset of potential exploration scenarios. Both environments are static and bounded, with known scale and no dynamic elements or environmental uncertainty. Consequently, the observed benefits of energy-aware planning may not fully transfer to larger-scale, outdoor, or real-world scenarios involving unstructured terrain, sensor noise, or environmental changes. Further evaluation in more diverse and dynamic settings is required to assess the robustness and generalizability of the proposed approach.

Dependence on Global Planner Efficiency: A core component of the EAAE algorithm is its use of an offline global planner to generate dynamically feasible trajectories toward candidate frontier clusters. As shown in Table 6.2, this step accounts for approximately 98% of the total computation time per decision step. On average, around three seconds are required to compute global trajectories for each new target selection. This adds significant time to the exploration process and affects the comparability of the results, as the baseline methods do not have this step.

Use of Two Distinct Planners: The EAAE framework relies on two separate planners: a global planner used for energy-informed target selection, and a local reactive planner for executing the actual trajectory. While this design enables offline execution of the trajectory and real-time reactivity during flight, it may introduce a discrepancy between the planned trajectory (used for energy estimation) and the executed trajectory (used during flight). As a result, the estimated energy cost used for decision-making may differ from the actual energy consumed.

8 Conclusion

Summary The main goal of this research was to investigate the effect of incorporating energy-awareness into autonomous UAV exploration of unknown 3D environments. The primary contribution was the development of the Energy-Aware Autonomous Explorer (EAAE), a novel exploration framework that selects future navigation goals based on both exploration potential and estimated energy cost. The algorithm combines a global planner for offline energy-informed candidate evaluation with a local reactive planner for real-time trajectory execution.

Evaluating the energy-aware algorithm using the exploration rate and map information entropy showed that energy-informed planning can reduce total energy consumption while maintaining competitive exploration performance. The proposed framework enables more deliberate decision-making and provides a foundation for energy-efficient exploration in practical UAV deployments.

Findings The results demonstrate that EAAE consistently achieves lower energy consumption than baseline methods across environments. This is attributed to its ability to balance high-information regions with low-energy-cost paths. In the Simple environment, this also translated to the shortest exploration time. These outcomes highlight the importance of addressing the trade-off between exploration speed and energy efficiency when designing energy-efficient autonomous exploration systems.

Additionally, both EAAE and the Classic method tended to generate longer, smoother trajectories that align better with the optimal velocity regime for fixed-wing UAV energy efficiency. This contributed to their lower average power consumption compared to the reactive Nearest method. This aligns with aerodynamic theory, where dynamic lift initially reduces the energy cost per unit distance at higher speeds, until induced drag dominates and the cost increases again.

Future Work Following this work, we suggest several directions for future research. First of all, as EAAE is built using two distinct planners, future work should focus on unifying the global offline and the local reactive planner, ensuring offline execution as well as dynamic obstacle avoidance are feasible, and the discrepancy between planned and executed trajectory is kept at a minimum. Also, as the computational time of the global planning step takes the majority of the computational load, future work could aim to optimize this step to enable faster computation, required for real-life deployment.

Furthermore, as current experiments were conducted in relatively small-scale scenarios, the UAVs were less likely to reach maximum velocity values. Therefore, extending evaluations to larger and more complex environments would test the method's generalizability and further explore the effects of the velocity on energy efficiency. Thirdly, as stated in the assumptions, this method relies on an empirical power consumption model based on rotor speeds that only accounts for mechanical power and an electrical loss term, and is formulated using static test environments. Future work could potentially improve the power consumption model to improve generalizability and accuracy. Lastly, we see the potential of the EAAE framework to be adjusted to a non-greedy algorithm by formulating the energy cost and information gain for all candidates as a Traveling Salesman Problem, to evaluate the effects on long-horizon exploration efficiency and total mission energy consumption.

References

[1] Jürgen Scherer et al. "An Autonomous Multi-UAV System for Search and Rescue". In: May 2015, pp. 33–38. DOI: 10.1145/2750675.2750683.

[2] J. Nikolic et al. "A UAV system for inspection of industrial facilities". In: Mar. 2013, pp. 1–8. ISBN: 978-1-4673-1812-9. DOI: 10.1109/AERO.2013.6496959.

[3] Julian Galvez-Serna et al. "UAV4PE: An Open-Source Framework to Plan UAV Autonomous Missions for Planetary Exploration". In: *Drones* 6.12 (2022). ISSN: 2504-446X. DOI: 10.3390/drones6120391. URL: https://www.mdpi.com/2504-446X/6/12/391.

[4] Leonard Bauersfeld and Davide Scaramuzza. "Range, endurance, and optimal speed estimates for multicopters". In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 2953–2960.

[5] Titus Cieslewski et al. "Rapid exploration with multi-rotors: A frontier selection method for high speed flight". In: *IEEE International Conference on Intelligent Robots and Systems*. Vol. 2017-September. 2017. DOI: 10.1109/IROS.2017.8206030.

[6] Brian Yamauchi. "A frontier-based approach for autonomous exploration". In: *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*. IEEE. 1997, pp. 146–151.

[7] Miguel Juliá et al. "A comparison of path planning strategies for autonomous exploration and mapping of unknown environments". In: *Autonomous Robots* 33 (2012), pp. 427–444.

[8] Anirudh Topiwala et al. "Frontier based exploration for autonomous robot". In: *arXiv preprint arXiv:1806.03581* (2018). URL: https://arxiv.org/pdf/1806.03581.

[9] Erkan Uslu et al. "Implementation of frontier-based exploration algorithm for an autonomous robot". In: *2015 International Symposium on Innovations in Intelligent SysTems and Applications (INISTA)* (2015), pp. 1–7. URL: https://api.semanticscholar.org/CorpusID:18068283.

[10] Wenchao Gao et al. "An improved frontier-based approach for autonomous exploration". In: *2018 15th international conference on control, automation, robotics and vision (ICARCV)*. IEEE. 2018, pp. 292–297.

[11] Zehui Meng et al. "A 2-Stage Optimized Next View Planning Framework for 3-D Unknown Environment Exploration and Structural Reconstruction". In: *IEEE Robotics and Automation Letters* PP (Jan. 2017), pp. 1–1. DOI: 10.1109/LRA.2017.2655144.

[12] Boyu Zhou et al. "FUEL: Fast UAV Exploration Using Incremental Frontier Structure and Hierarchical Planning". In: *IEEE Robotics and Automation Letters* 6 (2 2021). ISSN: 23773766. DOI: 10.1109/LRA.2021.3051563.

[13] Yinghao Zhao et al. “Autonomous Exploration Method for Fast Unknown Environment Mapping by Using UAV Equipped With Limited FOV Sensor”. In: *IEEE Transactions on Industrial Electronics* 71 (5 2024). ISSN: 15579948. DOI: 10.1109/TIE.2023.3285921.

[14] Xiaolei Hou et al. “LAEA: A 2D LiDAR-Assisted UAV Exploration Algorithm for Unknown Environments”. In: *Drones* 8.4 (2024), p. 128.

[15] Iván D Changoluisa Caiza et al. “Autonomous Exploration of Unknown 3D Environments Using a Frontier-Based Collector Strategy”. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2024, pp. 13566–13572.

[16] Cl Connolly. “The determination of next best views”. In: *Proceedings. 1985 IEEE international conference on robotics and automation*. Vol. 2. IEEE. 1985, pp. 432–435.

[17] Héctor H. González-Baños and Jean Claude Latombe. “Navigation strategies for exploring indoor environments”. In: *International Journal of Robotics Research*. Vol. 21. 2002. DOI: 10.1177/027836490202101083 4.

[18] Andreas Bircher et al. “Receding horizon next-best-view planner for 3D exploration”. In: *Proceedings - IEEE International Conference on Robotics and Automation*. Vol. 2016-June. Used as baseline in Limited FOV Sensor paper.
As example of a ”sampling-based-method” instead of ”frontier-based”. 2016. DOI: 10.1109/ICRA.2016.7487281.

[19] Magnus Selin et al. “Efficient autonomous exploration planning of large-scale 3-d environments”. In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 1699–1706.

[20] Victor Massagué Respall et al. “Fast Sampling-based Next-Best-View Exploration Algorithm for a MAV”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021, pp. 89–95. DOI: 10.1109/ICRA48506.2021.9562107.

[21] Shaojie Shen et al. “Stochastic differential equation-based exploration algorithm for autonomous indoor 3D exploration with a micro-aerial vehicle”. In: *International Journal of Robotics Research* 31 (12 2012). Used as reference in (among others): Autonmous Expl Method for Fast Unknown Mapping with Limited FOV Sensor. ISSN: 02783649. DOI: 10.1177/0278364912461676.

[22] Devendra Singh Chaplot et al. “Learning to explore using active neural slam”. In: *International Conference on Learning Representations (ICLR)*. 2020.

[23] Yuke Zhu et al. “Target-driven visual navigation in indoor scenes using deep reinforcement learning”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2017.

[24] Akash Patel et al. “Towards energy efficient autonomous exploration of Mars lava tube with a Martian coaxial quadrotor”. In: *Advances in Space Research* 71.9 (2023), pp. 3837–3854.

[25] Zichen Wang et al. “Efficient Autonomous UAV Exploration Framework with Limited FOV Sensors for IoT Applications”. In: *IEEE Internet of Things Journal* (2024).

[26] Carmelo Di Franco and Giorgio Buttazzo. “Energy-aware coverage path planning of UAVs”. In: *Proceedings - 2015 IEEE International Conference on Autonomous Robot Systems and Competitions, ICARSC 2015*. 2015. DOI: 10.1109/ICARSC.2015.17.

[27] Micha Rappaport. “Energy-aware mobile robot exploration with adaptive decision thresholds”. In: *Proceedings of ISR 2016: 47st International Symposium on Robotics*. VDE. 2016, pp. 1–8.

[28] Armin Hornung et al. “OctoMap: An efficient probabilistic 3D mapping framework based on octrees”. In: *Autonomous robots* 34 (2013), pp. 189–206.

[29] Zvi Shiller. “Off-line and on-line trajectory planning”. In: *Motion and operation planning of robotic systems: background and practical approaches* (2015), pp. 29–62.

[30] Philipp Foehn et al. “Agilicious: Open-source and open-hardware agile quadrotor for vision-based flight”. In: *Science robotics* 7.67 (2022), eabl6259.

[31] Gang Chen et al. “RAST: Risk-Aware Spatio-Temporal Safety Corridors for MAV Navigation in Dynamic Uncertain Environments”. In: *IEEE Robotics and Automation Letters* 8.2 (2023), pp. 808–815. DOI: 10.1109/LRA.2022.3231832.

[32] Xin Zhou et al. “Ego-planner: An esdf-free gradient-based local planner for quadrotors”. In: *IEEE Robotics and Automation Letters* 6.2 (2020), pp. 478–485.

[33] Claude Elwood Shannon. “A mathematical theory of communication”. In: *ACM SIGMOBILE mobile computing and communications review* 5.1 (2001), pp. 3–55.

Part III

Closure

Conclusion

This thesis addressed the development of an energy-aware autonomous exploration framework for UAVs operating in unknown 3D environments. While much prior research has focused on maximizing exploration coverage or rate, this work integrates an explicit energy perspective into the exploration decision-making process. By doing so, it aims to increase the practicality and endurance of UAVs in real-world missions where energy is a limiting factor. The work involved the design and implementation of a modular exploration algorithm, integration with a realistic energy model, and systematic comparison with baseline methods across multiple environments in simulation. The following section reflects on the thesis work and aims to answer the research questions as stated in section I

Research Question 1 How to develop an autonomous exploration algorithm for a UAV that can be integrated with an energy-consumption model?

This thesis presents the design and implementation of the Energy-Aware Autonomous Exploration (EAAE) algorithm, which enables UAVs to explore unknown 3D environments while explicitly accounting for energy usage in the decision-making process. The approach builds on existing frontier-based exploration techniques by introducing a two-stage planning strategy: (i) a global planner generates candidate trajectories to multiple frontier clusters, and (ii) energy-aware selection is applied based on a trade-off between information gain and predicted energy cost. The proposed architecture integrates seamlessly with a modular simulation stack, and supports a ROS-based OctoMap mapping system, a geometric controller, and a rotor-based energy estimator node from the Agilicious framework. By structuring the algorithm into decoupled modules for perception, frontier detection, energy-aware planning, and control, it becomes extensible and suitable for integration with realistic UAV dynamics and power models.

Research Question 2 How can energy consumption of an UAV be modeled based on a given trajectory that is planned by an exploration algorithm?

Energy consumption is estimated by simulating the UAV's flight state over candidate trajectories using a full-stack pipeline, which includes state propagation via Agilicious and energy estimation based on rotor dynamics. The model computes the instantaneous power draw based on rotor speeds and integrates this over the planned trajectory to obtain total energy. This allows the exploration algorithm to rank candidate goals not only by their spatial or information-theoretic utility, but also by the anticipated energy required to reach them. Validation of this model was performed in simulation, and the results showed that the estimated energy values are consistent with the actual energy recorded during execution. While this implementation uses an offline trajectory planner and a separate reactive controller for execution, the simulation-based approach allows accurate assessment of energy requirements, and it provides a foundation for further integration of real-time, adaptive energy models.

Research Question 3 What are the differences in exploration coverage and exploration rate with respect to the exploration algorithm that does not account for the energy consumption?

Experimental evaluation in two distinct environments (*Simple* and *Pillars*) showed that the proposed EAAE algorithm consistently achieves competitive or superior performance compared to two common baseline methods: Classic Frontier (information gain-driven) and Nearest Frontier (distance-driven). In particular, EAAE demonstrates a 10–20% reduction in total energy consumption across scenarios while maintaining similar or faster total exploration times. The results highlight that naive methods such as Nearest Frontier are energy-inefficient due to their frequent short-range decisions and backtracking behavior, while Classic methods, although faster in open environments, tend to suffer from late-stage revisits due to greediness. EAAE outperforms these by combining efficient spatial planning with energy-aware decision-making. The analysis further reveals that the relationship between exploration speed and energy usage is non-trivial; faster coverage does not always equate to lower energy use. The results suggest that incorporating energy-awareness leads to more consistent, less erratic power consumption and improves mission-level energy efficiency without compromising mapping performance.

A Additional Results

A.1. Power Consumption over Time

This appendix section shows the power usage over time for all three algorithms in the *Simple* and *Pillars* environments.

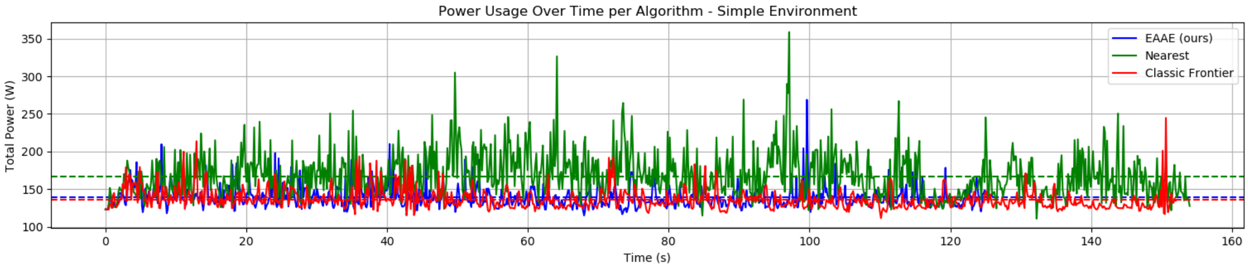


Figure A.1: Power consumption over time for all three methods in the *Simple* environment.

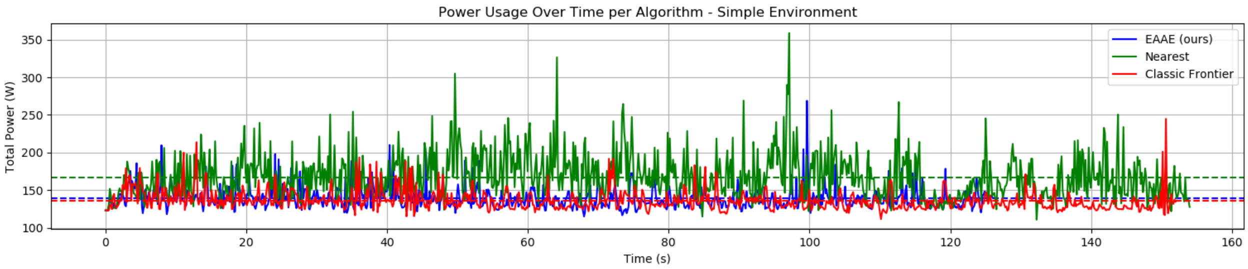


Figure A.2: Power consumption over time for all three methods in the *Pillars* environment.

Figures A.1 and A.2 illustrate the total power consumption over time for all three algorithms in the *Simple* and *Pillars* environments. As observed, the Nearest method consistently exhibits the highest average power usage across both scenarios, which aligns with the findings reported in the main results chapter. Additionally, the Nearest method displays more frequent and larger power fluctuations. These variations can be attributed to its characteristic back-and-forth exploration behavior, resulting in frequent accelerations and decelerations that increase instantaneous power draw.

A.2. Visualization Exploration Mission

This appendix section shows the incremental, local exploration behavior of the Nearest method in the *Pillars* environment.

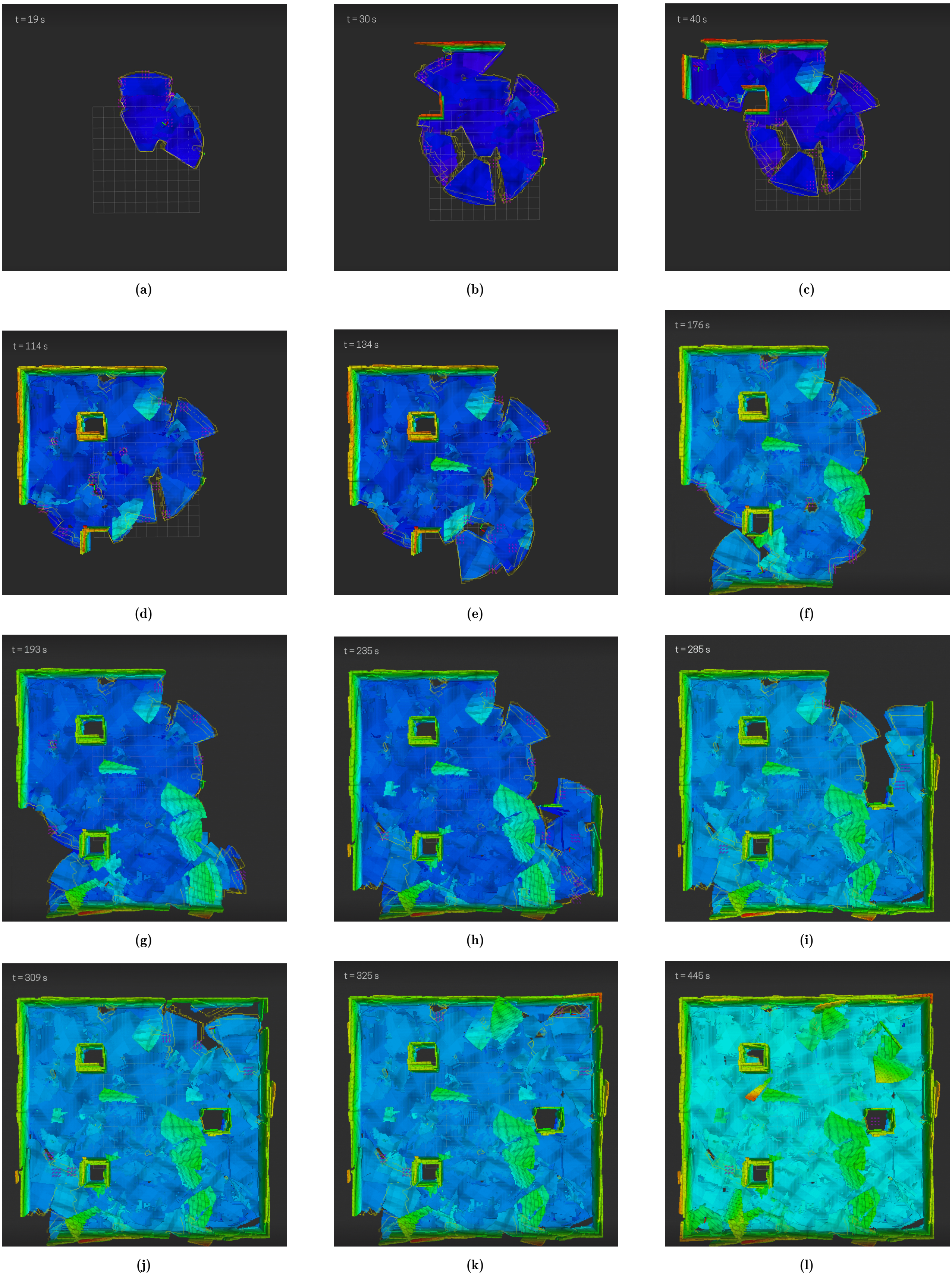


Figure A.3: Snapshots of the Nearest Frontier method illustrating its local, incremental exploration behaviour over time.

B Literature Review

This chapter presents the findings of the literature study that is conducted as part of the first phase of the thesis project. It aims to provide the reader with a clear overview of the literature used in the thesis project as well as providing the reader with the required theoretical knowledge regarding energy-aware autonomous exploration of UAVs in unknown environments. A third aim of this chapter is to highlight the relevance of the discussed literature to this thesis project.

The chapter is organized as follows: Section *State-Of-The-Art Exploration* elaborates on the state-of-the-art (SOTA) algorithms for autonomous exploration. This includes frontier-based approaches, next-best-view (NBV) approaches, and alternative methods such as learning-based and stochastic differential equation-based methods. Section *State-of-the-art Nonlinear Dynamics and Flight-state Dependent Energy Consumption Model* presents the findings regarding the non-linear dynamics and flight-state-dependent energy consumption models. For a better understanding of the current work in the field of energy-aware autonomous exploration, Section *Current Approaches* provides an overview of the current approaches, highlighting some of the main challenges and lessons from current literature. Finally, Section *Knowledge Gap* highlights the knowledge gap that is identified based on the literature study.

State-of-the-art Exploration Algorithms

This section elaborates on the state-of-the-art autonomous exploration algorithms. In this literature study, two main methods are discussed: frontier-based approaches and NBV-based approaches, occasionally named sampling-based approaches. Other novel autonomous exploration algorithms that do not directly base their method on one of the aforementioned theories are described in Section *Alternative Approaches.*

Frontier-based Approaches

This section elaborates on the advancements of frontier-based approaches for autonomous exploration algorithms. The concept of frontiers was first introduced in 1997 by Yamauchi, B.; this work describes the central concept of frontier-based exploration as *"to gain the most new information about the world, move to the boundary of open space and uncharted territory"* [1]. This gave rise to the following terminology: a frontier is the region between unknown space and free space. The method explained in [1] describes the following logic. A map can be divided into three categories: free, occupied, and unknown space. When a robot moves towards the centroid of a frontier-cluster, it can observe the previously unknown space behind it and add it to its exploration map. This results in an increasing share of explored space, and thus greater knowledge of the world around the robot. By navigating to successive frontiers, the robot eventually explores the full map, categorizing all regions as either free or occupied space. This is because all accessible space is contiguous and on the assumption of a perfect sensor and perfect control. This means that enclosed free space cannot be explored as it cannot be reached by the robot. A visualization of the frontier terminology is given in the paper [2] and is shown below.

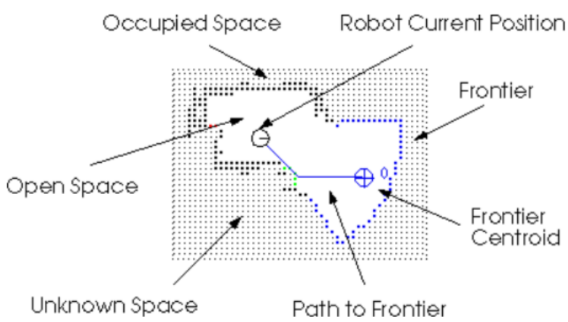


Figure B.1: Frontier terminology. (Source: [2])

The frontier-based theory as described above, even though relatively simple, proved to be extremely useful as it forms the basis for the majority of autonomous exploration research [3]. What remained consistent throughout the literature is the method of defining frontiers. The main improvements regarding performance, i.e the exploration rate, the total amount of explored area et cetera, came from improved environment representation techniques. Another contribution to these improvements came from adjustments in selection criteria for the goal frontier for navigation. A third and more recent factor came from optimized navigation strategies, such as finding a more optimal flight path between goal frontiers or implementing adaptive yaw-manoeuvres.

Below the various methods are discussed and categorized into three main topics that have significantly contributed to the historic successes of autonomous exploration: environment representation mapping, frontier-selection methods, and flight-state and trajectory optimization.

Environment representation mapping

An important contribution to the successes of frontier-based autonomous exploration algorithms throughout the literature is the way the environment is categorized and represented. As described in Appendix B, the map is subdivided into three categories; free, occupied and unknown space. This enables the exploration algorithms with a way of dealing with the probabilistic uncertainties regarding the environment, inherent to real-world sensors such as cameras, laser-range finders, and sonar-based sensors. Early work in frontier-based exploration methods used so-called "evidence grids" [1]. These Cartesian grids are used to represent the environment of the robot using three-dimensional cells that store the probability of occupancy. Initially, all the cells are set to a prior occupancy probability estimate. Although this variable can be tuned for a safer flight plan or more accurate mapping, according to [1] a value of 0.5 proved to be acceptable in experiments.

A significant possibility for improvement for frontier-based exploration algorithms arose because of the introduction of OctoMap in 2013 [4]. The year before, [5] demonstrated vision-based autonomous mapping and exploration using cameras as the main sensor on a quadrotor MAVs, in contrast to previous work relying on laser-range finders or external systems. In 2013, [6] proposed using a front-looking stereo camera to build a global 3D occupancy map using [4]. This is one of the first papers in literature to showcase frontier-based autonomous exploration using an OctoMap representation. The introduction of OctoMap provided a straightforward and accessible platform for future research regarding autonomous mapping and exploration. [4] developed an open-source framework that facilitates the representation of an environment as a 3D grid using voxels that contain the probability of occupancy. The representation of the environment can be updated frequently and thus store information of the explored space. This framework is based on octrees, a hierarchical framework for spatial subdivision in three dimensions [7], [8]. Each node of this octree represents a voxel. OctoMap enables users to select different octree depths, and thus represent the environment using different resolutions to enable computationally-efficient mapping. Visualizations from [4] of the octree structure and of the various OctoMap resolutions are shown in Figure B.2 and Figure B.3, respectively.

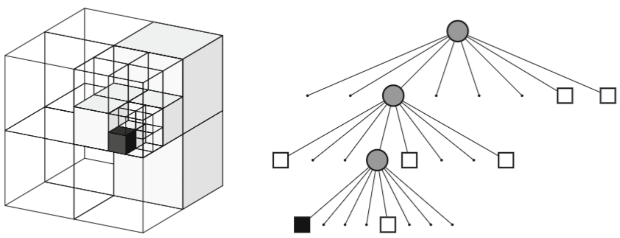


Figure B.2: Octree nodes structure. The volumetric model is shown on the left and the corresponding tree representation on the right. (Source: [4])

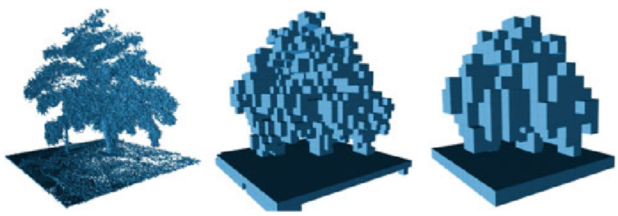


Figure B.3: Different Octomap resolutions. Occupied voxels are displayed in resolutions 0.08, 0.64, and 1.28 m. (Source: [4])

Following the introduction of OctoMap, various methods utilized the OctoMap properties to improve the exploration performance. A promising method that used these OctoMap properties is described in [9]. In this work the authors propose a frontier-based planner for 3D exploration that is applicable for 3D sensors like lidars, which produces large point clouds. According to the authors this method is more scalable as it processes the same data set size more quickly while maintaining a similar exploration time. This is achieved by not using the data from the 3D sensor directly but first, using OctoMap environment representation properties, cluster the frontier points. This allows the algorithm to analyse and plan accordingly using different resolutions. This method is best explained using the visualisation below, Figure B.4. For navigation and planning and detection, a maximal resolution is desired, thus an maximal Octree depth, d_{max}. However, for frontier clustering a more shallow Octree is used, d_{exp}. This implies that the algorithm detects frontiers voxels at the highest resolution (d_{max}) and tries to find the parent nodes of these detected frontiers voxels at the Octree level d_{exp}.

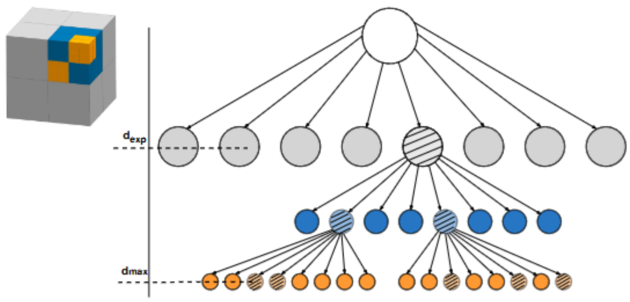


Figure B.4: Overview of the Octree structure used in the multi-resolution frontier method including the indicated Octree depths. (Source: [9])

This trade-off between having a computationally more expensive algorithm or a more detailed map is important to consider when developing an exploration algorithm using voxel-based environment representation structures like OctoMap.

Caiza et al. [10] build upon this multi-resolution frontier-based method by adding a so called collector-strategy. This collector strategy firstly stores all global frontier candidates at each moment of the exploration. The second phase filters out all candidates that are obstructed, too close by, previously visited, or have low information gains. Finally, in the third phase the algorithm selects the closest global candidate as the best frontier and this frontier is send to the path planner. An illustration of this method is given in [10] and shown below.

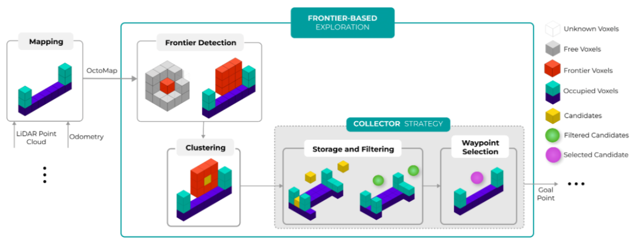


Figure B.5: Overview of the individual phases of the collector-strategy. (Source: [10])

This clearly shows the functionality of OctoMap throughout various works in recent literature. The following section aims to provide the reader with information on the state-of-the-art frontier selection methods and describes how this has changed over the years.

Frontier selection method

Figure B described the importance of accurate environment representation mapping for autonomous exploration algorithms. A second crucial step in exploration algorithms is the high-level planning. This raises the question: *"What is the most optimal frontier point to navigate to next?"*. This section elaborates on the SOTA frontier-selection methods.

The first strategies for frontier-selection focused on minimizing the physical distance between the current robot position and the location of the frontier cluster centroid [1], [2], [3], [11]. This was done using a simple cost function with the objective to minimize the distance as this was given a large cost. Later, the authors of [12] proposed to add more information to the cost function by incorporating not only the distance, but also the frontier size and the cost of turning the robot. Another notable contribution comes from [13], here the authors define information gain as the summation of the expected information enclosed in smaller voxels that are likely to be visible from a particular view. This metric is closely related to the basis of NBV techniques as discussed in Figure B.

Because of this metric, frontier-based approaches could also select goal frontiers based on the expected information gain. This is proposed by the authors of [14]. This method generates candidate points around the frontiers and subsequently filters and clusters these points. Afterwards, the information gain of these candidate points is calculated using the following equation from [15]:

$$G_v = \sum_{\forall r \in R_v} \sum_{\forall x \in X} e(x) \tag{B.1}$$

where G_v is the information gain for candidate point v. This is R_v is the set of rays cast from the candidate point onto the robot's field of view. X is the set of voxels the ray traverses through. $e(x)$ is the mapping uncertainty of each voxel, x [14].

Another important and relevant contribution is described in [16]. In this work, the authors propose a frontier-based method that selects the goal frontier for navigation based upon the required velocity change to reach that specific frontier. This method proved to perform extremely good in terms of exploration rate. Further details on this work are discussed in Equation B and Figure B.

Flight-state and trajectory optimization

A third crucial aspect of exploration algorithms is the navigation. This section describes the advancements made from optimization of navigation algorithms. This includes trajectory optimization as well as optimization of the robot's flight-state.

A major contribution to frontier-based autonomous exploration came from [17]. In this paper, the authors propose an exploration algorithm named Fast UAV Exploration (FUEL). This method maintains relevant information for exploration planners in a so called, frontier information structure (FIS). This infromation structure allows for incremental updates when new space is explored. Secondly, a hierachical planner is proposed. An overview of the contributions of FUEL are visualized in the figure below.

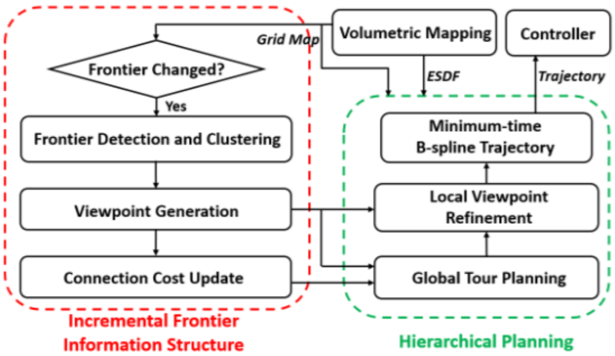


Figure B.6: Overview of FUEL method. (Source: [17])

The hierarchical planner plans the exploration path in three consecutive steps. First, the algorithm finds the global path covering all the frontier clusters that is optimal for environment information accumulation. The second step refines a part of the global path and the third step involves generating a safe, minimum-time trajectory to the first viewpoint of the refined path from the second step. A visualization of this method is given in [17] and presented below.

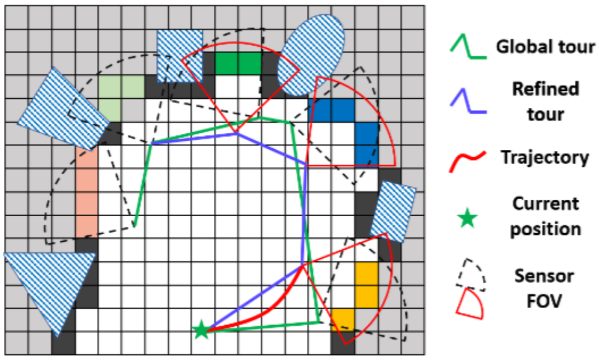


Figure B.7: Path and motion generation using three steps in FUEL method. (Source: [17])

The viewpoints as described in [17] are similar to the the candidate points as described above in Figure B. In FUEL however, viewpoints are generated as follows. Starting from a frontier cluster centroid, viewpoints are uniformly sampled in a cylindrical coordinate system with the origin coinciding with the respective frontier cluster centroid. In order to find a global path covering all frontiers clusters, FUEL states the planning problem as an open-loop Asymmetric Travelling Salesman Problem (ATSP), inspired by [18]. For step two of the hierarchical planner the authors of [17] propose to use Dijkstra's algorithm for finding the shortest path throughout all the viewpoints of all frontiers. Other methods also suggest using path optimization algorithms for exploration. For example, [14] proposes using the A*-algorithm for generating a safe and feasible path. To smoothen out the navigation path, [17] uses B-spline optimization to get continuous trajectories such that the multi-rotor drone can fully use its dynamic capabilities.

The Fast Autonomous Exploration Planner (FAEP), as described in [19], builds upon the FUEL framework. FAEP improves exploration efficiency by reducing back-and-forth manoeuvrers. The authors of FAEP aimed to do this by not only considering frontier-level factors, but also flight-level factors. The frontier-level factors are included in the form of two extra cost items in the ATSP for path planning, similar as in FUEL. The first cost term penalizes frontiers that are farther away from the boundary area (larger d_{min}), whilst still in range of the sensor, r_s. If the boundary frontier is out of range, $D_k \geq r_s$, the frontier is penalized by a positive factor w_d. The boundary area is defined as the area near the edge of the exploration area as determined by the mission. This cost item is described in the following equation from [19]:

$$c_b(k) = \begin{cases} d_{\min}(k) & D_k < r_s \\ d_{\min}(k) \cdot \left(1 + w_d \cdot \frac{D_k - r_s}{r_s}\right) & D_k \geq r_s \end{cases} \tag{B.2}$$

$$d_{\min}(k) = \min\left(d_x^k, d_y^k, d_z^k\right), \quad k \in \{1, 2, \cdots, N_{\text{cls}}\} \tag{B.3}$$

For the second frontier-level cost item the authors propose to use a method called Bottom Ray to detect the range of unknown area behind the frontier. This range is denoted by h_k in the Figure B.8, as can be seen in the equation below area's that are known to have a small unexplored area behind them are prioritized as they have a larger cost, $c_s(k)$ and are thus more likely to cause back-and-forth manoeuvrers.

$$c_s(k) = \frac{h_{max} - h_k}{h_{max}} \tag{B.4}$$

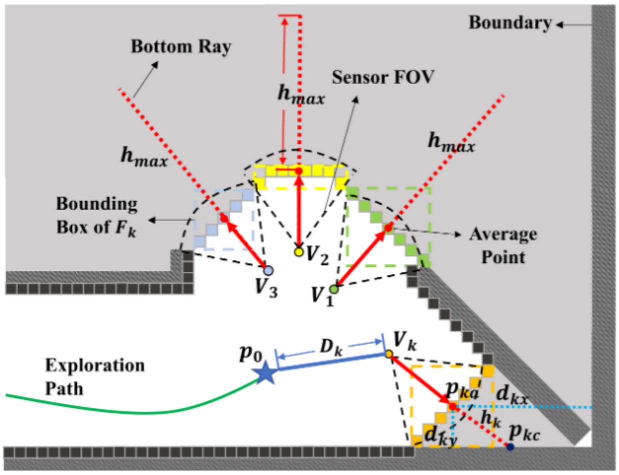


Figure B.8: Diagram to evaluate frontier-level costs in FAEP method. (Source: [19])

This shows that the algorithm prioritizes frontier clusters based on their proximity to the UAV and their potential to provide new information while minimizing repeated visits to previously explored areas to reduce back-and-forth manoeuvres.

As this thesis project aims to investigate the implications of integrating energy-awareness on autonomous exploration algorithms. It is interesting to look into if and how, current SOTA exploration algorithms implement energy-awareness. An insightful novelty in reducing back-and-forth manoeuvres and thus energy-consumption, is developed by Zhao et al. [19] by introducing adaptive yaw-manoeuvres. In this work, the authors propose two-stage yaw planning. Two-stage yaw planning aims to go from the current yaw angle ξ_0, to the required yaw angle at the target viewpoint ξ_e, while passing through a middle yaw ξ_m. This middle yaw is set such that the UAV covers the maximum amount of area during flight. To avoid longer flight times because of this yaw manoeuvre, the authors propose to set a criteria that the yaw-manoeuvre only takes place if the estimated time of the yaw-manoeuvre is less or equal to the flight time to the target viewpoint. An even more recent contribution came from [20] in 2024 with a method called LiDAR-Assisted UAV Exploration Algorithm (LAEA). This method uses LiDAR data with a depth camera to create a hybrid 2D occupancy map. This map allows the algorithm to detect and prioritize small and isolated frontier clusters. By assigning higher weights to these clusters and incorporating an Environmental Information Gain (EIG) optimization strategy, LAEA generates optimized trajectories that cover unexplored regions while minimizing back-and-forth movements. This method proved the be promising as it outperforms both the FUEL and FAEP methods. The relevance of methods like FUEL, FAEP, and LAEA lies in their non-greedy approach, as they plan entire paths ahead rather than selecting only the next best location. This capability is relevant for energy-aware exploration, as an energy-aware algorithm could calculate the energy required for a complete path instead of evaluating single locations, enabling more efficient and informed decision-making.

Next-best-view Approaches

This section reviews the relevant literature on NBV approaches for autonomous exploration. NBV approaches are based on early work by C.I. Connolly in [21]. Here, NBV is first introduced and described as a method to find a sequence of views to obtain a complete model of an object. This is later used as a method for exploration to view an environment as it aims to find the next best viewpoint to obtain as much information about unexplored space as possible, whilst minimizing the cost of travel.

Although there are clear similarities between the NBV-based approaches and the frontier-based approach, as discussed in Appendix B, the main difference is in selecting sampling viewpoints. The NBV-based method, also known as the sampling-based method, relies on random sampling of viewpoints in a certain known, or partially known, space within the sensor's field of view. Frontier-based approaches do not rely in that sense on random sampling as these methods deterministically select frontiers from the partly explored environment. Some frontier-based methods do use sampling for candidate viewpoint generation, as described in [17]. This method, however, uses uniform sampling in a cylindrical coordinate system to sample candidate viewpoints for a certain determined frontier.

This section further elaborates on the advancements made with NBV-based approaches by considering the candidate selection, the use of random trees for NBV methods, and the possibility of combining a NBV-based approach with a frontier-based approach.

Candidate selection

One of the first works that uses the NBV algorithm for autonomous exploration is described in [22]. As briefly described before, NBV methods randomly sample candidate viewpoints in the sensor's field of view, this is shown in Figure B.9(b). Thereafter, the potential gain in visibility outside the explored area is determined. This is indicated by the red area in Figure B.9(c). Finally, the candidate viewpoints q are evaluated based on the value g, in the following expression [22]:

$$g(q) = A(q)exp(-\lambda L(q, q_k)) \tag{B.5}$$

In this equation, $A(q)$ represents the potential visibility gain of a candidate viewpoint q. This visibility gain is the amount of unexplored area visible through the edges of the known area; this is depicted in Figure B.9 as the red area. $L(q, q_k)$ represents the length from the current viewpoint q_k to the candidate viewpoint q. The constant λ can be used to further penalize candidates that are further away from the current position. The above equation is widely used within NBV methods and could be relevant to this thesis topic as a way to possibly incorporate energy-related constraints. An example could be to evaluate candidate viewpoints not just by distance and potential visibility gain, but also the expected amount of energy required to traverse to the candidate viewpoints.

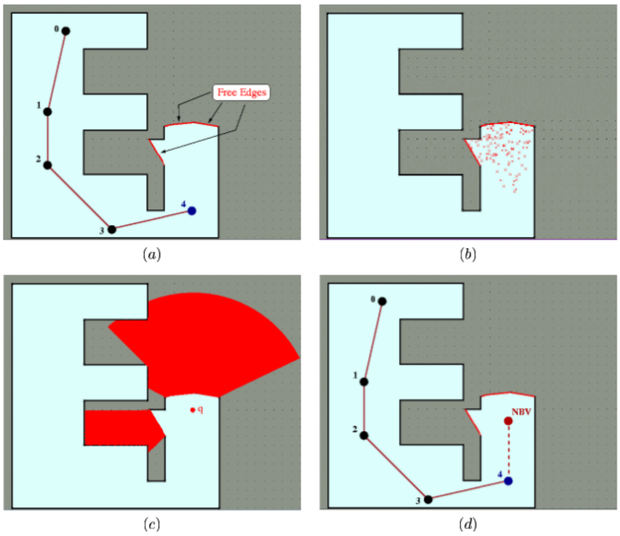


Figure B.9: Overview of the steps in NBV-based methods. (Source: [22])

Random trees for NBV approaches

Another interesting advancement in NBV-based methods is the use of random trees as described in the method: receding-horizon next-best-view (RH-NBV). This method is introduced in the paper: Receding-Horizon Next-Best-View Planner for 3D Exploration [23]. In this paper, the authors propose to use the randomly sampled candidate viewpoints as nodes in a random tree, the edges of which provide a possible path to the viewpoint. In order to grow this random tree, the RH-NBV uses techniques Rapidly Exploring Random Trees (RRT) [24] and RRT* (optimized version of RRT) [25]. Each branch is evaluated based on the amount of unexplored space that can be mapped from this branch. Subsequently, only the first edge is executed, after which the whole process is repeated. This, in fact, explains the *receding horizon* in the name of the RH-NBV method.

Combination of NBV- and frontier-based approaches

A third noteworthy contribution is from the paper Efficient Autonomous Exploration Planning of Large-scale 3D Environments (AEP) [26]. Here, the authors propose to combine a NBV-based approach with a frontier-based approach. The motivation is to overcome the drawback of the back-and-forth manoeuvres in frontier-based methods and the risk for the RH-NBV approach to get stuck when exploring large environments [26]. In AEP a frontier-based approach is used for global planning and a NBV-based approach for local planning. Nodes with a large potential information gain from previous RRTs are cached for later use. Once the robot explores everything in its surroundings, the next frontier will be far away and subsequently get a low score from the RH-NBV planner. In this case, the frontier-based approach will step in as it considers these previously cached nodes. This combination approach showed great performance and introduces an interesting method possibly useful for this thesis project. Another interesting insight is the fact that

Alternative Approaches

To gain a clear overview of all the autonomous exploration algorithms potentially useful for this thesis project, this section aims to cover other relevant works for autonomous exploration, not directly linked to frontier-based or NBV-based methods. First, some advancements in learning-based approaches are discussed. Secondly, a stochastic differential equation-based approach is discussed.

Learning-based approaches

In recent years, learning-based methods have shown significant potential in addressing autonomous exploration challenges, particularly in adaptive informative path planning (AIPP). These approaches leverage supervised learning (SL) and reinforcement learning (RL) to enhance scalability to larger environments.

Supervised learning techniques are often used to improve environment representation and path planning. For example, Gaussian processes have been widely applied to model environmental variables and predict features like temperature or radiation, as seen in [27], enabling robots to gather information with reduced uncertainty. Similarly, neural networks (NNs) are trained on labelled datasets to predict optimal next-best-viewpoints or complete partially observed maps, facilitating efficient exploration.

RL provides an effective framework for autonomous decision-making in exploration tasks. RL allows robots to learn exploration policies by optimizing reward functions linked to information gain and resource constraints. Methods such as Proximal Policy Optimization (PPO) and Deep Q-Networks (DQN) have been employed to maximize map uncertainty reduction or coverage in various domains [27]. This also highlights the relevant possibility of incorporating energy awareness into exploration algorithms. For example in [28], the authors state that their method requires reasoning about both sensing and movement in terms of information gain and energy expended. These advancements not only highlight the potential of learning-based approaches but also the relevance to this thesis project because of the energy constraints. However, an important disadvantage to consider when evaluating the relevance to this project is the issue regarding generalizability to new environments, which is something of great importance in autonomous exploration. Popović et al. stated the following: "However, the lack of realistic labeled training data for AIPP problems remains an open issue restricting the generalizability of existing methods to new environments and domains." [27], this underlines a crucial limitation of learning-based approaches for this thesis project.

Stochastic differential equation-based approach

Another insightful approach is introduced in [29]. In this work the authors propose a stochastic differential equation-based approach. The authors argue that classic frontier-based approaches struggle with exploring 3D spaces because of issues related to occlusion, resolution, and the sensor's field of view. The approach in [29] only represents occupied space. Another key difference with frontier-based approaches is the way of detecting frontiers. This method samples particles in the space contained by the known occupied space. Using a stochastic differential equation the particles are used to simulate a gas using Newtonian dynamics. The location to which the gas expands is determined as the frontier for exploration. Even though this method introduces an interesting new approach, it seems not directly relevant to energy-aware autonomous exploration, as it mostly focuses its efforts to resolve issues like occlusion and resolution often encountered in 3D frontier-based exploration.

An interesting insight from the literature study regarding exploration algorithms that use frontier-based, NBV-based, or other approaches, is that the majority of the before-mentioned works are greedy-algorithms. The term greediness of algorithms is used to distinguish whether an algorithm only thinks about the next best step or also about the future steps. In almost all of the SOTA exploration algorithms mentioned in this section, the decision-making part uses the proximity, velocity change, information gain or a combination of these factors for the decision regarding the next step but not about the sequence of steps. The only methods that decide on an optimal path instead of a next optimal point are FUEL [17] and the works that build upon this method, FAEP [19] and LAEA [20]. This distinction is relevant for this thesis project as the possibility exists that planning ahead in a non-greedy manner improves energy-efficiency once an algorithm is energy-aware in its decision-making.

State-of-the-art Nonlinear Dynamics and Flight-state Dependent Energy Consumption Models

This section elaborates on the state-of-the-art non-linear dynamics and flight-state dependent energy consumption models for UAVs. This is a key research field for energy-aware autonomous exploration as the work related to this field aims to accurately describe an UAVs energy consumption during flight. Works in this field are relevant for this thesis project as this thesis project aims to integrate such an energy consumption model into an exploration algorithm. In this part of the literature study, three important aspects and advancements for accurate energy consumption modelling are described.

Forces and torques modelling

In order for the exploration algorithm to take into account the energy required for a certain frontier/viewpoint, the energy required for a certain trajectory must be calculated. A first important step is calculating the forces and torques generated by the propellers for a certain trajectory. Using these forces and torques, the required power — and subsequently the energy needed — can be determined. A review of the literature on the first step, the forces and torques modelling, is given below.

A relevant model that is used in literature is the quadratic model. The quadratic model states that each propeller generates a force and torque term proportional to the square of its rotational speed. As described by Bauersfeld and Scaramuzza [30], this model holds well for near-hover flight of UAVs but becomes increasingly inaccurate for higher speeds. This is because it ignores important aerodynamic effects as the induced drag and the dynamic lift of the propeller.

Other influential work is based on the blade-element-momentum (BEM) theory. This theory is developed by Froude [31] and extended by Glauert [32] in 1920 and 1935, respectively. BEM theory combines conservation of mass and momentum principles to analyse aerodynamic loads on rotating blades. It calculates forces like thrust and torque by integrating aerodynamic lift and drag on blade sections using two-dimensional airfoil theory and experimental data [33]. BEM theory is important for energy consumption modelling, enabling the estimation of aerodynamic forces and power requirements. Relevant recent work using BEM theory is presented in [30] and [34]. These works are discussed in greater detail below.

Battery modelling

In order to accurately describe the energy consumption of an UAV during a mission, not only the forces and torques generated by the propellers but also the battery model must be estimated. For this thesis project, the most relevant aspect involves estimating the energy required for a given trajectory. This requires developing a method that enables calculations, progressing from trajectory planning to estimated power consumption, then to estimated load, and finally to the required capacity. Battery consumption models are usually categorized as either white box (e.g., an electrochemical model), grey box (e.g., circuit-oriented model) or black box (e.g., NN-based models) models [35]. These categories are used to differentiate between models that use only physical properties of the batteries and the models that purely rely on data without any knowledge on the underlying physical or chemical processes.

Early work on battery modelling is introduced by Peukert, W. [36]. This paper proposes a method to describe the effective capacity under load and has since then became the standard approach for various works because of its simplicity. Even though Peukert's method seems to be an interesting method for effective capacity modelling, it is not applicable to this thesis project. According to the authors of [37], Peukert's model is only applicable to low and medium discharge rates. This makes the model suboptimal for this thesis project as it most likely will not account for the effects of the high transient loads often encountered in UAV flight.

Other widely used methods for battery modelling are based on Thevenin equivalent circuits [38]. Two examples of these methods are the OTC (one time constant) and TTC (two time constant) models, an overview of these methods for Li-ion batteries is given in [39]. Other interesting work that uses the OTC method is described in [30]. Here the authors use a combination of BEM model, an electric motor model, and an OTC model to accurately predict a battery's voltage, even under non-constant discharge rates. This paper brought some relevant insights. First of all, this work describes the decision to use battery voltage models, such as the OTC model, instead of Peukert's battery capacity model. According to the authors: the OTC model is more flexible because it can handle situations where the multicopter has a non-constant power demand, such as in battery-aware path planning for complex missions [30]. However, it is important to note that the OTC model focuses on battery voltage, whereas this thesis project requires the estimation of required power in order to determine the energy required per trajectory. A second contribution is the fact that this method combines a first principles model as BEM together with a body-drag model to calculate the power required at a given speed.

Other work that does model the power consumption of a UAV is described in [40]. In this paper, the authors use real-flight data of a Parrot AR 2.0 Drone Elite Edition to develop an energy-consumption model that can be used in simulations. Based on this measured data, the authors fit a polynomial to the data to obtain an expression for the power based on the angular speed. In order to calculate the required energy E, the following expression is utilized using a distance h, and velocity v, component.

$$E = P * \frac{h}{v} \tag{B.6}$$

In [40] the aim is to find a more accurate energy consumption model to plan mission routes more realistically. The results show a more reliable state-of-charge (SOC) estimation and show that for a certain path planning algorithm that does not use this model, the mission would have stopped early because of an empty battery whereas the suggested method would have predicted this beforehand. Regarding relevance to this thesis project, as this thesis project focuses on exploration rather than path planning, quite the modifications are required. A second important limitation is that this work relies on measured data from one specific drone so does not allow for generalization. However, it is still interesting to see work that incorporates battery- and energy-awareness into an algorithm and the low-complexity mathematics and physics could be of good use to this thesis project. Another example of such a relatively low-complexity energy consumption model based on empirical data is presented in [41].

Learning-based multicopter dynamics

More recent approaches to modelling multicopter dynamics involve learning-based techniques. Two insightful methods using learning-based methods for multicopter dynamics modelling are presented by Bansal et al. and Punjani et al. in [42] and [43], respectively. Punjani et al. propose a method using a Rectified Linear Unit (ReLU) network model to estimate the multicopter's accelerations during a wide range of manoeuvres. Bansal et al. propose a two-layer NN for estimating multicopter accelerations. The goal of the method is to learn parameters $\alpha = (W, w, B, b)$ in order to estimate minimize the root mean square (RMS) error between the observed accelerations and the estimated accelerations. The parameters are defined as follows: W and w are the weight matrices for the hidden layer and output layer, respectively. B and b are the bias vectors for the hidden layer and output layer, respectively. These terms are important for NNs as they define the network´s architecture and are optimised during training. The architecture of this learning-based method is included in Figure B.10 below.

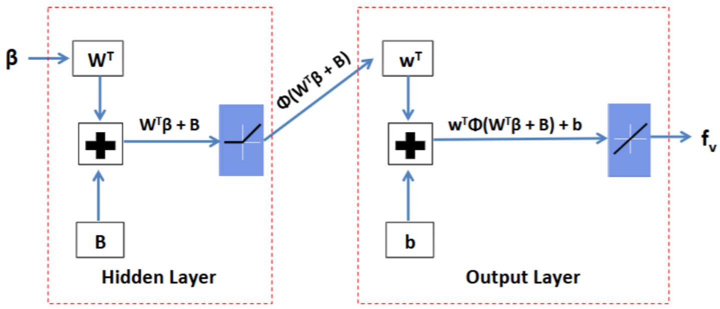


Figure B.10: NN architecture to estimate multicopter dynamics. NN consisting of a hidden ReLU layer and an output layer. Parameters $\alpha = (W, w, B, b)$ are learned. (Source: [42])

While these learning-based methods yield acceptable estimations of multicopter accelerations, recent advancements have further improved the accuracy of multicopter dynamics modelling using a hybrid approach. The authors of NeuroBEM [34] present a state-of-the-art method to estimate multicopter dynamics using a first principles model, BEM, in combination with a deep neural network (DNN) to account for residual forces. The main motivation of this work is to combine the strong generalization capabilities of first principles models with the flexibility of learning-based approaches [34]. The architecture of the proposed method is displayed below in Figure B.11. NeuroBEM proves to be a reliable method for predicting forces and torques, outperforming various baselines that just use first principles models or learning-based methods.

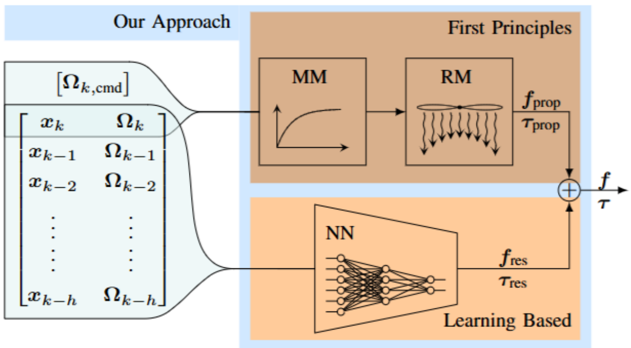


Figure B.11: NeuroBEM architecture consting of motor model (MM), rotor model (RM), and NN for the residual forces. Inputs are the current state x_k, the current motor speeds Ω_k, and the current motor speed commands $\Omega_{k,cmd}$. Outputs are the forces f and torques τ. (Source: [34])

Learning-based methods and hybrid methods have proven to be great approaches for estimating multicopter dynamics. However, as this thesis project does not only focus on energy-consumption modelling but also on autonomous exploration, a more simplistic variant for energy-consumption or multicopter dynamics is preferred.

Current Approaches to Energy-aware Exploration

This section elaborates on the current approaches in the research field of energy-aware autonomous exploration, highlighting the current state-of-the-art and possibilities for future research. It aims to elaborate on methods that overlap the research fields described in Appendix B and Figure B.

Rapid Exploration - A Frontier Method for High Speed Flight

The first relevant method that is described in this section is from the paper: 'Rapid Exploration - A Frontier Method for High Speed Flight', by Cieslewski et al. [16]. This method was briefly discussed in Appendix B. As stated above, this method proposes to select the next best frontier based on the required change in velocity, arguing that a smaller delta results in more high-speed flight and thus a shorter exploration rate. As highlighted in [44] as well, maintaining a consistent flight speed with minimal accelerations leads to a more energy-efficient mission. As this method proves to be of great relevance to this thesis topic in combining a frontier-based exploration approach with an energy-related constraint, it does not seem directly implementable due to the lack of open source code and a general energy consumption model. The principle explained in this method however, could possibly be implemented as a baseline for comparison purposes.

Towards energy efficient autonomous exploration of Mars lava tube with a Martian coaxial quadrotor

Another approach in autonomous exploration is presented in [44]. In this work, the authors claim to have developed an energy-aware model for the exploration of Martian lava tubes using a quadrotor. The paper shows to perform very well in terms of exploration rate and coverage and includes great mapping visualizations. The method uses OctoMap for mapping. Furthermore it uses a frontier-based exploration approach with frontier accessibility classification, risk-aware path planning, and reactive navigation. However, it is not believed that this paper is of great relevance to this thesis project. This is because the lack of implementation of energy-awareness into the algorithm's decision making. The authors of [44] suggest that a path with near-zero acceleration is more energy efficient than other methods with larger accelerations. Though this statement is correct, this is the only part of the method that implies energy-efficiency. For example, the cost function does not account for the expenditure of energy for certain paths. Thus, this paper seems of little relevance to this thesis project.

Energy-Aware Mobile Robot Exploration with Adaptive Decision Thresholds

The method described in [45] proposes an energy-aware exploration algorithm for a mobile robot, the Pioneer 3-DX robot. Though the use of another platform implies that any measured data or numerical values are not directly relevant to this thesis project, the work is still included for the low-complexity calculations and insightful cost function. The mobile robot has to be charged at a docking station once it runs out of energy. This method focuses on reducing the travelled path as this is the largest contributor to the total energy required. The authors opt for a rather simplistic energy consumption model. The power consumption is calculated for two cases, for a maximum speed v_{max} and for a stationary case. This is to ensure that for all possible speeds the power consumption is never underestimated. The power consumption of all components other than locomotion are considered constant. The power consumption of locomotion is calculated by equation (1) of [45]:

$$P_{loc}(v) = 0.29[W] + 7.4\frac{[W]}{[m/s]} \cdot v \tag{B.7}$$

With this equation, the power consumption for the moving and the stationary case can be calculated. This calculation is based on previous work in [46], where the authors studied energy efficiency for the Pioneer 3-DX robot. With the power consumption, the SOC and the energy that is left, E_l, can be calculated. Finally, using equation (5) from [45] the distance the robot is still able to travel, d_l, can be calculated using the current energy level, E_l:

$$d_l = \frac{E_l}{P_m} \cdot v \tag{B.8}$$

Another insightful approach in this method is the proposed cost function f. The cost function as given in [45] is stated below:

$$f = w_1 \cdot d_g + w_2 \cdot d_{gb} + \cdot w_3 \cdot d_{dgbe} + w_4 \cdot \theta_{rel} \tag{B.9}$$

In this equation, w_1, w_2, w_3 and w_4 are weights to adjust the cost function to various environments. d_g is the distance from the robot to the frontier, to make sure the model prefers the shortest path. The second parameter d_{gb} is the distance from the frontier to the docking station. The third parameter d_{gbb}, is either equal to d_{gb} or equal to negative d_{gb}, and this depends on whether the energy level is sufficient. Thus, this ensures that the robot prefers frontiers that are far away in the beginning when the battery level still allows it. The final parameter θ_{rel} refers to the robot's orientation and thus ensures the model prefers frontiers that do not require large turns.

To conclude, this paper shows interesting, relatively simplistic energy consumption calculations. These equations could be relevant to this thesis project for the part of incorporating energy limitations in exploration cost functions. Important to note is that only parts are useful as this model is for a mobile land robot instead of an

UAV and that it focusses on docking stations for recharging. This thesis project aims to develop an algorithm for a single UAV without docking stations, and thus does not aim to find an optimal path considering docking stations but rather aims to explore an unknown area more efficiently and more energy-aware.

Efficient Autonomous UAV Exploration Framework with Limited FOV Sensors for IoT Applications.

Another recent relevant approach is presented in [47], where the authors propose a frontier-based method. This approach is novel as it utilises randomly generated seeds within a specific window. If one of these seeds lies within a known frontier voxel, it is selected as a valid random seed; otherwise, the voxel is added to the list of known voxels. This enables the method to generate candidate frontiers efficiently.

The authors claim that this method is energy-aware, as it incorporates a cost item in the cost function to account for *exploration energy consumption.* However, similar to other works, the only factor considered in energy consumption is the distance to a frontier. This is reflected in the cost function described in equation (8) of [47]:

$$M_{tsp}(0,k) = \lambda_{dis}c_{dis} + \lambda_{dir}c_{dir} + \lambda_{boud}c_{bound} \tag{B.10}$$

In this equation c_{dis} represents the distance costs, given by equation (9) in [47]:

$$c_{dis} = \text{length}(P(P_{uav}, VP_i)) \tag{B.11}$$

The authors explain that; "c_{dis} is used to measure the energy consumption of the UAV for each viewpoint" [47]. The second cost item, c_{dir}, relates to the UAV's orientation, favouring frontiers that require less turns. The third cost item, c_{bound}, ensures that the model prioritises frontiers near the boundary, as these are often skipped and can result in inefficient back-and-forth manoeuvres.

While it is interesting to see a cost function incorporating some consideration of energy, as also observed in [45], this thesis project aims develop a new approach for energy consumption that is not just based on distance.

Energy-Aware Coverage Path Planning of UAVs

Another noteworthy paper is presented by Di Franco and Buttazzo in [48]. In this work, the authors present an energy-aware path planning algorithm. The two main contributions are: an energy model derived from measurements to obtain the power consumption as a function of the UAV dynamics in different operational settings. Another contribution is the energy-aware algorithm that aims to find the speed for a certain path that minimizes energy consumption whilst satisfying coverage and resolution criteria [48]. The latter contribution could be relevant to this thesis project as it calculates the expected required energy for the flight phases: steady flight, accelerating, decelerating, turning, climbing, and descending. An opportunity for this thesis project lies in that fact that the method proposed in [48] does not consider the required energy during path planning. Namely, once the path is planned, the calculated required energy at the optimal flight speed is compared to the total available energy and based on that, the path is pursued or not. This thesis project could possibly incorporate parts of this method by comparing the energy required for all the possible next viewpoint candidates to make an informed decision about where to move to next.

Knowledge Gap

To summarize the findings of the literature study on autonomous exploration, a table is presented below to provide the reader with a clear overview of the current SOTA works in autonomous exploration and their respective main contributions that may be relevant to this thesis project. The works presented in this table are discussed more elaborately in Appendix B and Figure B. As can be seen in Table B.1, the majority of the work in the field of autonomous exploration does not consider energy constraints in the decision-making step of the algorithm. This often leads to algorithms being non-viable for real-world applications where the to-be-explored area is large.

The few works that do address this challenge do not rely on a generalized energy consumption model, but rather make simplifications or rely on empirical data for a specific type of UAV. For example in [16] and [44], the authors argue that the proposed model is energy efficient as the UAVs fly with near-zero accelerations. In [45], the authors propose a method that does calculate power and energy use for a specific robot but it is merely based on known empirical data from previous research on this specific robot. In [47] the authors developed an algorithm that does include an energy-related term in the cost function; however, this cost item only depends on the physical distance and does not account for turns, speed, and other flight dynamics. The work described in [28] includes energy-awareness in the algorithm by comparing the cost of traversing to a next node with the cost of using the UAV's sensors at the current node. Lastly, the work presented in [48] accounts for the energy consumption during coverage path planning. Though the energy consumption calculation seems extensive, it is all based on measured data about the drone's power consumption for various velocities.

All things considered, a clear knowledge gap arises in the field of energy-aware autonomous exploration. More specifically, because of a lack of exploration algorithms that take into account a generally applicable energy consumption model in the decision-making for the best path to take. This thesis project aims to contribute to this part of the field and strives to accomplish this with the research objective and questions described in section I.

Table B.1: Autonomous exploration algorithms literature overview.

	Description	Reference	Mapping	Planning	Navigation	Energy-aware exploration?
Frontier-based	Early-frontier	[1]	-	frontier introduction	-	no
	Rapid	[16]	-	smallest Δv	-	yes (small accelerations)
	FUEL	[17]	frontier clustering	hierarchical planning (FIS) + ATSP	trajectory smoothing (B-splines)	no
	MRF	[9]	frontier clustering + multi-resolution OctoMap	goal frontier optimization	-	no
	Collector	[10]	-	frontier filtering	-	no
	FAEP	[19]	-	multi-level frontier planning	reduced back-and-forth manoeuvrers + adaptive-yaw	no
	LAEA	[20]	hybrid occupancy map	small frontier prioritization + EIG	-	no
	MARS	[44]	-	direct and indirect frontiers	-	yes (small accelerations + minimize yaw + minimize height)
	Adaptive Threshold**	[45]	-	-	-	yes (power consumption calculation + cost function)
	Limited FOV	[47]	-	-	-	yes (cost function)
NBV	Early-NBV	[22]	-	next-best-view introduction + potential gain equation	-	no
	RH-NBV	[23]	-	introduction RRT	-	no
	AEP	[26]	-	global: frontier local: NBV	-	no
Alternative	AIPPMS	[28]	-	adaptive planning	-	yes (decides to spend energy on sensing or on traversing to next node)
	SDE	[49]	-	particle-based frontier detection	-	no
	Energy-aware Coverage*	[48]	-	reduce turns + optimal speed	-	yes (empirical energy calculation specific for one UAV type)

. **this algorithm is a coverage path planning algorithm instead of an exploration algorithm but is included for its energy-aware path planning.*

***the platform that is used in this method is a mobile robot rather than a UAV.*

References

[1] Brian Yamauchi. "A frontier-based approach for autonomous exploration". In: *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*. IEEE. 1997, pp. 146–151.

[2] Anirudh Topiwala et al. "Frontier based exploration for autonomous robot". In: *arXiv preprint arXiv:1806.03581* (2018). URL: https://arxiv.org/pdf/1806.03581.

[3] Miguel Juliá et al. "A comparison of path planning strategies for autonomous exploration and mapping of unknown environments". In: *Autonomous Robots* 33 (2012), pp. 427–444.

[4] Armin Hornung et al. "OctoMap: An efficient probabilistic 3D mapping framework based on octrees". In: *Autonomous robots* 34 (2013), pp. 189–206.

[5] Friedrich Fraundorfer et al. "Vision-based autonomous mapping and exploration using a quadrotor MAV". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 4557–4564.

[6] Korbinian Schmid et al. "Stereo vision based indoor/outdoor navigation for flying robots". In: *2013 IEEE/RSJ international conference on intelligent robots and systems*. IEEE. 2013, pp. 3955–3962.

[7] Donald Meagher. "Geometric modeling using octree encoding". In: *Computer graphics and image processing* 19.2 (1982), pp. 129–147.

[8] Jane Wilhelms and Allen Van Gelder. "Octrees for faster isosurface generation". In: *ACM Transactions on Graphics (TOG)* 11.3 (1992), pp. 201–227.

[9] Ana Batinovic et al. "A Multi-Resolution Frontier-Based Planner for Autonomous 3D Exploration". In: *IEEE Robotics and Automation Letters* 6 (3 2021). MRF: Used as baseline in Autonomous Expl of Uknown Env Using Frontier-based Collector Strategy. ISSN: 23773766. DOI: 10.1109/LRA.2021.3068923.

[10] Iván D Changoluisa Caiza et al. "Autonomous Exploration of Unknown 3D Environments Using a Frontier-Based Collector Strategy". In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2024, pp. 13566–13572.

[11] Erkan Uslu et al. "Implementation of frontier-based exploration algorithm for an autonomous robot". In: *2015 International Symposium on Innovations in Intelligent SysTems and Applications (INISTA)* (2015), pp. 1–7. URL: https://api.semanticscholar.org/CorpusID:18068283.

[12] Wenchao Gao et al. "An improved frontier-based approach for autonomous exploration". In: *2018 15th international conference on control, automation, robotics and vision (ICARCV)*. IEEE. 2018, pp. 292–297.

[13] Stefan Isler et al. "An information gain formulation for active volumetric 3D reconstruction". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2016, pp. 3477–3484.

[14] Liang Lu et al. "Optimal frontier-based autonomous exploration in unconstructed environment using RGB-D sensor". In: *Sensors* 20.22 (2020), p. 6507. URL: https://www.mdpi.com/1424-8220/20/22/6507.

[15] Jeffrey Delmerico et al. "A comparison of volumetric information gain metrics for active 3D object reconstruction". In: *Autonomous Robots* 42.2 (2018), pp. 197–208.

[16] Titus Cieslewski et al. "Rapid exploration with multi-rotors: A frontier selection method for high speed flight". In: *IEEE International Conference on Intelligent Robots and Systems*. Vol. 2017-September. 2017. DOI: 10.1109/IROS.2017.8206030.

[17] Boyu Zhou et al. "FUEL: Fast UAV Exploration Using Incremental Frontier Structure and Hierarchical Planning". In: *IEEE Robotics and Automation Letters* 6 (2 2021). ISSN: 23773766. DOI: 10.1109/LRA.2021.3051563.

[18] Zehui Meng et al. "A Two-Stage Optimized Next-View Planning Framework for 3-D Unknown Environment Exploration, and Structural Reconstruction". In: 2.3 (2017), pp. 1680–1687. DOI: 10.1109/LRA.2017.2655144.

[19] Yinghao Zhao et al. "Autonomous Exploration Method for Fast Unknown Environment Mapping by Using UAV Equipped With Limited FOV Sensor". In: *IEEE Transactions on Industrial Electronics* 71 (5 2024). ISSN: 15579948. DOI: 10.1109/TIE.2023.3285921.

[20] Xiaolei Hou et al. "LAEA: A 2D LiDAR-Assisted UAV Exploration Algorithm for Unknown Environments". In: *Drones* 8.4 (2024), p. 128.

[21] Cl Connolly. "The determination of next best views". In: *Proceedings. 1985 IEEE international conference on robotics and automation*. Vol. 2. IEEE. 1985, pp. 432–435.

[22] Héctor H. González-Baños and Jean Claude Latombe. "Navigation strategies for exploring indoor environments". In: *International Journal of Robotics Research*. Vol. 21. 2002. DOI: 10.1177/0278364902021010834.

[23] Andreas Bircher et al. "Receding horizon next-best-view planner for 3D exploration". In: *Proceedings - IEEE International Conference on Robotics and Automation*. Vol. 2016-June. Used as baseline in Limited FOV Sensor paper.
As example of a "sampling-based-method" instead of "frontier-based". 2016. DOI: 10.1109/ICRA.2016.7487281.

[24] Steven LaValle. "Rapidly-exploring random trees: A new tool for path planning". In: *Research Report 9811* (1998).

[25] Sertac Karaman and Emilio Frazzoli. "Sampling-based algorithms for optimal motion planning". In: *The International Journal of Robotics Research* 30.7 (2011), pp. 846–894. DOI: 10.1177/0278364911406761. eprint: https://doi.org/10.1177/0278364911406761. URL: https://doi.org/10.1177/0278364911406761.

[26] Magnus Selin et al. "Efficient autonomous exploration planning of large-scale 3-d environments". In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 1699–1706.

[27] Marija Popovic et al. "Robotic Learning for Adaptive Informative Path Planning". In: *arXiv preprint arXiv:2404.06940* (2024).

[28] Shushman Choudhury et al. "Adaptive informative path planning with multimodal sensing". In: *Proceedings of the International Conference on Automated Planning and Scheduling*. Vol. 30. 2020, pp. 57–65.

[29] Shaojie Shen et al. "Autonomous indoor 3D exploration with a micro-aerial vehicle". In: *2012 IEEE international conference on robotics and automation*. IEEE. 2012, pp. 9–15.

[30] Leonard Bauersfeld and Davide Scaramuzza. "Range, Endurance, and Optimal Speed Estimates for Multicopters". In: *IEEE Robotics and Automation Letters* 7 (2 2022). ISSN: 23773766. DOI: 10.1109/LRA.2022.3145063.

[31] William Froude. *On the elementary relation between pitch, slip, and propulsive efficiency*. Tech. rep. 1920.

[32] H. Glauert. "Airplane Propellers". In: *Aerodynamic Theory: A General Review of Progress Under a Grant of the Guggenheim Fund for the Promotion of Aeronautics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1935, pp. 169–360. ISBN: 978-3-642-91487-4. DOI: 10.1007/978-3-642-91487-4_3. URL: https://doi.org/10.1007/978-3-642-91487-4_3.

[33] D Zhao et al. "Offshore wind turbine aerodynamics modelling and measurements". In: *Wind Turbines Aerodyn Energy Harvest* (2019), pp. 373–400.

[34] Leonard Bauersfeld et al. "NeuroBEM: Hybrid Aerodynamic Quadrotor Model". In: *Robotics: Science and Systems*. 2021. DOI: 10.15607/RSS.2021.XVII.042.

[35] S Tamilselvi et al. "A review on battery modelling techniques". In: *Sustainability* 13.18 (2021), p. 10042.

[36] W Peukert. "Über die abhängigkeit der kapazität von der entladestromstärke bei bleiakkumulatoren". In: *Elektrotechn. Zeitschr.* (1897).

[37] N Galushkin et al. "A critical review of using the Peukert equation and its generalizations for lithium-ion Cells". In: *Journal of The Electrochemical Society* 167.12 (2020), p. 120516.

[38] Don H Johnson. "Origins of the equivalent circuit concept: the voltage-source equivalent". In: *Proceedings of the IEEE* 91.4 (2003), pp. 636–640.

[39] Xiaoqiang Zhang et al. “A review of li-ion battery equivalent circuit models”. In: *Transactions on Electrical and Electronic Materials* 17.6 (2016), pp. 311–316.

[40] Yukai Chen et al. “A case for a battery-aware model of drone energy consumption”. In: *2018 IEEE international telecommunications energy conference (INTELEC)*. IEEE. 2018, pp. 1–8.

[41] Hasini Viranga Abeywickrama et al. “Empirical power consumption model for UAVs”. In: *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*. IEEE. 2018, pp. 1–5.

[42] Somil Bansal et al. “Learning quadrotor dynamics using neural network for flight control”. In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE. 2016, pp. 4653–4660.

[43] Ali Punjani and Pieter Abbeel. “Deep learning helicopter dynamics models”. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2015, pp. 3223–3230.

[44] Akash Patel et al. “Towards energy efficient autonomous exploration of Mars lava tube with a Martian coaxial quadrotor”. In: *Advances in Space Research* 71.9 (2023), pp. 3837–3854.

[45] Micha Rappaport. “Energy-aware mobile robot exploration with adaptive decision thresholds”. In: *Proceedings of ISR 2016: 47st International Symposium on Robotics*. VDE. 2016, pp. 1–8.

[46] Yongguo Mei et al. “A case study of mobile robot’s energy consumption and conservation techniques”. In: *ICAR’05. Proceedings., 12th International Conference on Advanced Robotics, 2005*. IEEE. 2005, pp. 492–497.

[47] Zichen Wang et al. “Efficient Autonomous UAV Exploration Framework with Limited FOV Sensors for IoT Applications”. In: *IEEE Internet of Things Journal* (2024).

[48] Carmelo Di Franco and Giorgio Buttazzo. “Energy-aware coverage path planning of UAVs”. In: *Proceedings - 2015 IEEE International Conference on Autonomous Robot Systems and Competitions, ICARSC 2015*. 2015. DOI: 10.1109/ICARSC.2015.17.

[49] Shaojie Shen et al. “Stochastic differential equation-based exploration algorithm for autonomous indoor 3D exploration with a micro-aerial vehicle”. In: *International Journal of Robotics Research* 31 (12 2012). Used as reference in (among others): Autonmous Expl Method for Fast Unknown Mapping with Limited FOV Sensor. ISSN: 02783649. DOI: 10.1177/0278364912461676.

C Project Planning

This section provides the project plan for this thesis project. In the first section the methodology is described including the various work packages with corresponding inputs and outputs. Section C.1 presents the thesis project planning through two figures. The first figure outlines the high-level project schedule, including key deadlines and deliverables, while the second figure provides a more detailed view, focusing on specific work packages.

Methodology

In order to develop a fully functional autonomous exploration algorithm that is integrated with an energy consumption model, various crucial tasks have to be completed. These tasks are categorized into a total of 9 work packages. These work packages with the corresponding tasks are presented below. For brevity, the term 'work package' is from here onwards referred to with WP.

Note: regarding the required software systems, programs and packages, at this point it is not decided but the thesis project will most likely be conducted using, among other: ROS, Gazebo, Python, C++, MATLAB, OctoMap.

Note: for every WP; the expected duration, the corresponding margin for potential delay, the inputs, and the outputs are listed. This information is based on current estimates and is subject to change throughout the project. The margins are set based upon expected probability of delay and are thus higher for more technical tasks as developing and implementing code as opposed to less technical tasks as the preparation of a review session.

WP1: Project initialization & literature study

The aim of this WP is to set up agreements with the responsible supervisors and sign all required contracts. Furthermore, this WP contains the literature study which will account for the majority of the time required for this WP.

- Create project description
- Read relevant literature
- Write literature review chapter of report
- Create literature review presentation

Expected duration: 4 weeks
Margin: 3 days
Inputs: -
Outputs: literature chapter, signed thesis contract, documentation of supervision agreements, (understanding of) relevant literature, knowledge gap

WP2: Development of methodology & research objective

The aim of this WP is to develop the methodology and research objective.

- Develop research objective and corresponding research questions
- Develop methodology
- Write project planning chapter of report

Expected duration: 2 weeks
Margin: 2 days
Inputs: knowledge gap
Outputs: methodology chapter, research objective and questions, method

WP3: Implementation of exploration algorithm

The aim of this WP is to successfully implement an autonomous exploration algorithm that is capable of conducting an exploration mission in a simulated environment. Secondly, it should be integrable with an energy-consumption model, most likely by outputting the trajectories for the set of potential target viewpoints.

- Reading and understanding the code

- Setting up the environment
- Run successfully in simulated environment

Expected duration: 6
Margin: 2 weeks
Inputs: relevant algorithms from literature
Outputs: exploration algorithm, trajectory

WP4: Implementation of energy-consumption model

The aim of this WP is to successfully implement an energy-consumption model that is capable of estimating the energy required for a given viewpoint/trajectory. This model should be able to run in parallel with the exploration algorithm.

- Reading and understanding the code and the underlying physics
- Setting up the environment

Expected duration: 5 weeks
Margin: 1 week
Inputs: relevant algorithms from literature, trajectory
Outputs: energy-consumption model, energy required for given trajectory

WP5: Integration of modules

The aim of this WP is to integrate the two modules developed in WP3 and WP4.

- Create overview of input and outputs for both modules
- Design full architecture
- Integrate energy-consumption model into exploration algorithm

Expected duration: 2 weeks
Margin: 1 week
Inputs: exploration algorithm, energy-consumption model
Outputs: complete integrated algorithm, complete code

WP6: Simulation & evaluation

The aim of this WP is to simulate the complete integrated algorithm and evaluate it with respect to baselines. A potential baseline is the exploration algorithm that is energy-UNaware. Other baselines will also be considered depending on available resources.

- Set up simulation environment
- Develop/decide performance indicators
- Run complete, integrated program in simulation environment
- Run energy-UNaware module
- Evaluate and compare results for both algorithms

Expected duration: 3 weeks
Margin: 1 week
Inputs: complete integrated algorithm,
Outputs: simulation results energy-aware algorithm, simulation results energy-UNaware algorithm, performance indicators

WP7: Visualisation & reporting

The aim of this WP is to visualize the findings and report it in the thesis report. Important to note is that reporting will also take place during the research phase, however, in this WP the focus will be on refining.

- Decide on way of visualization, including possible visualization tools
- Finalize report according to scientific writing conventions

Expected duration: 1 week
Margin: 1 week
Inputs: all simulation results, performance indicators
Outputs: (results) visualizations

WP8: Preparation Green-Light review

The aim of this WP is to prepare for the Green-Light (GL) review meeting.

- Submit thesis draft
- Create GL review presentation
- Faculty forms, documents, and other practical matters

Expected duration: 2 weeks
Margin: 2 weeks
Inputs: simulation results, results visualizations, complete code and other report content
Outputs: thesis draft, GL review presentation

WP9: Preparation thesis defence

The aim of this WP is to finalize the thesis report and prepare for presenting and defending the conclusions.

- Finalize thesis report
- Create thesis defence presentation
- Prepare for defence questions
- Faculty forms, documents, and other practical matters

Expected duration: 3 weeks
Margin: 3 days
Inputs: thesis draft, feedback GL review
Outputs: final thesis report, thesis defense presentation

C.1. Planning

The planning is divided into four phases. The first phase is the Literature Review & Research Definition Phase. The goal of this first phase is to develop a clear research objective based on the knowledge gap that is identified during the review of the relevant literature. The second phase is called Research Phase 1. In this phase the first half of the research is conducted. The deliverable of Research Phase 1 is the midterm report. In the third phase, Research Phase 2, the research is completed and the thesis draft is submitted. In the final phase, the Research Dissemination, a Green Light Review is conducted to determine whether the student is ready to proceed to the thesis defense. During this review, a formal go/no-go decision is made regarding the defence. A complete overview of the high-level planning is presented in Figure C.1.

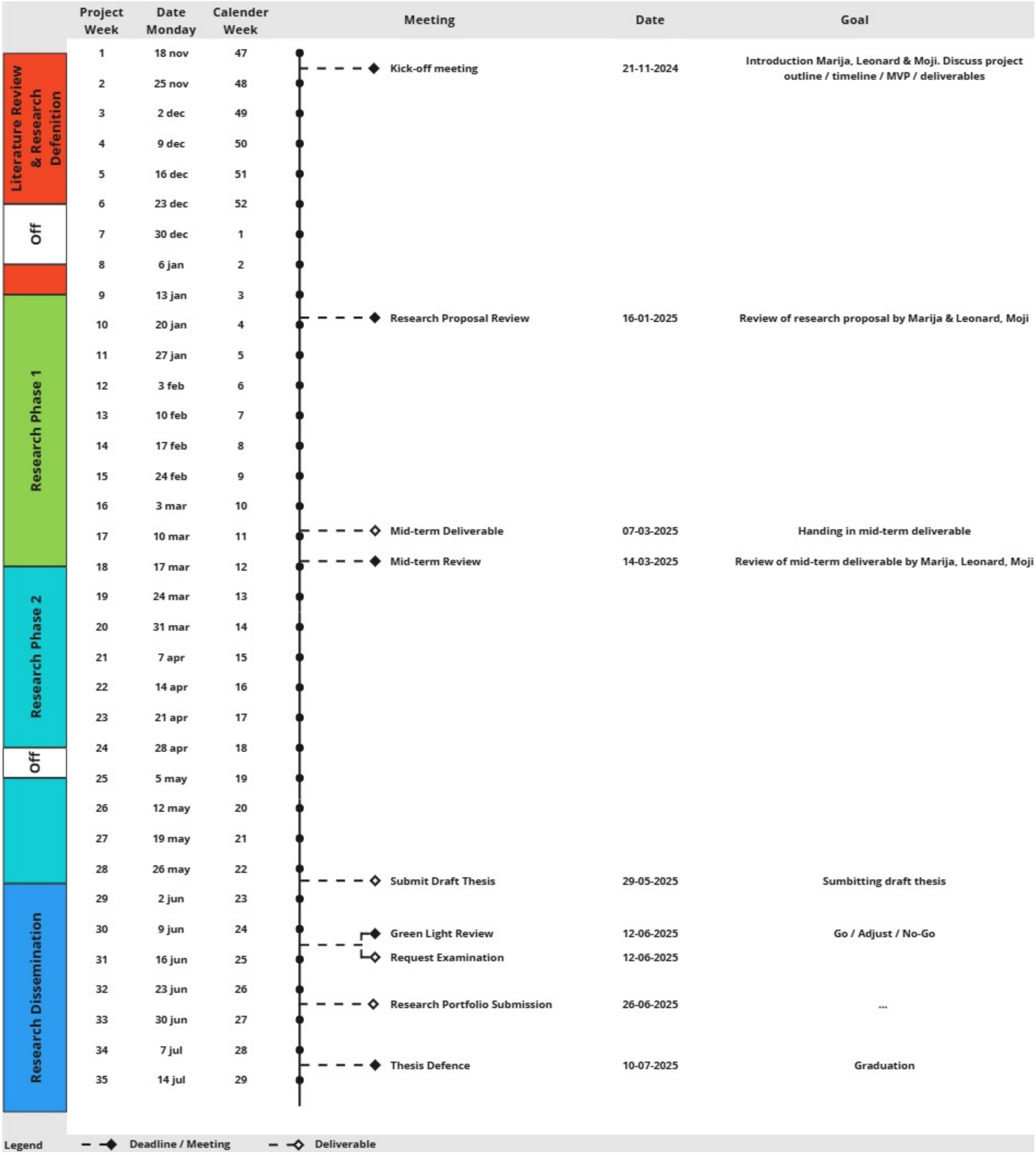


Figure C.1: High-level thesis project planning including (intermediate) deadlines and deliverables.

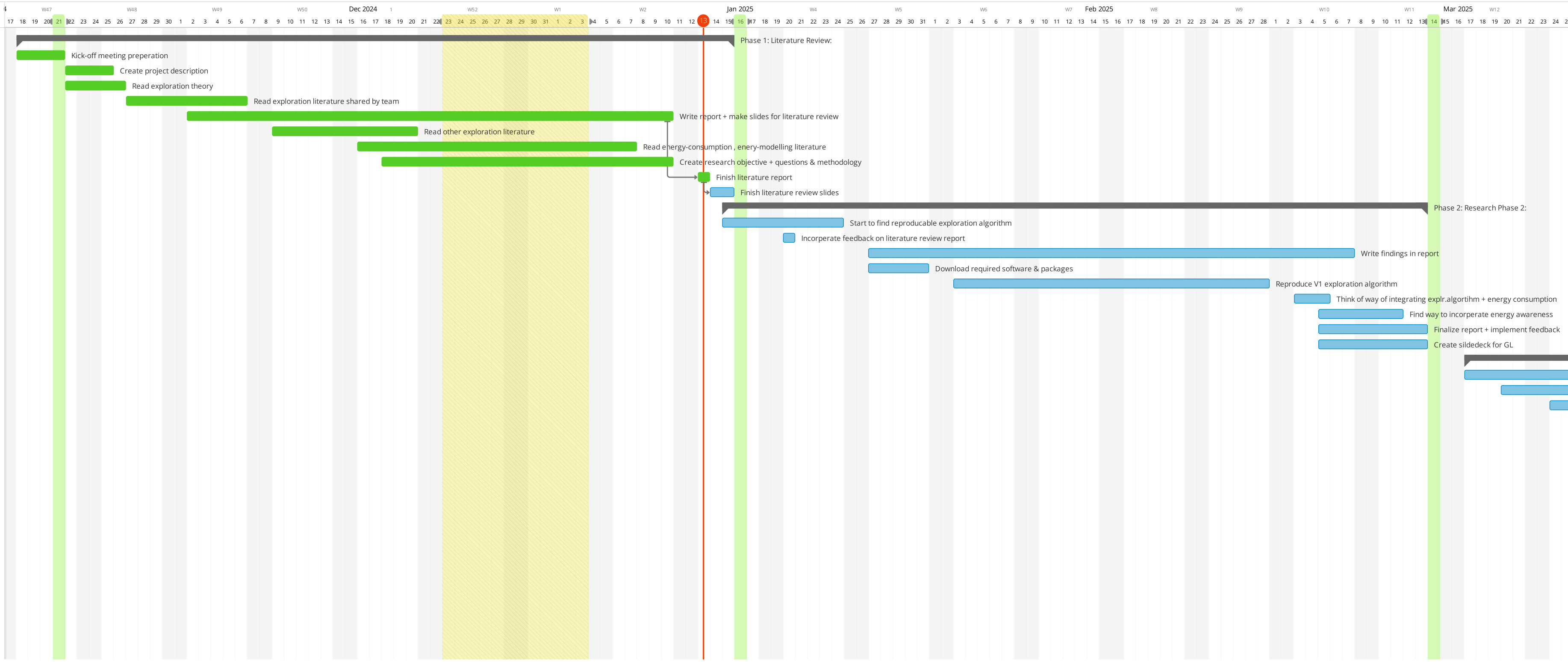
Secondly, a more detailed planning is presented. This second planning focusses on the various work packages as described in Section C.1 and shows their interdependencies and corresponding expected deadlines. This planning is presented in Figure C.3. below.

Figure C.2: Detailed thesis project planning including work packages developed with *InstaGant* (below).

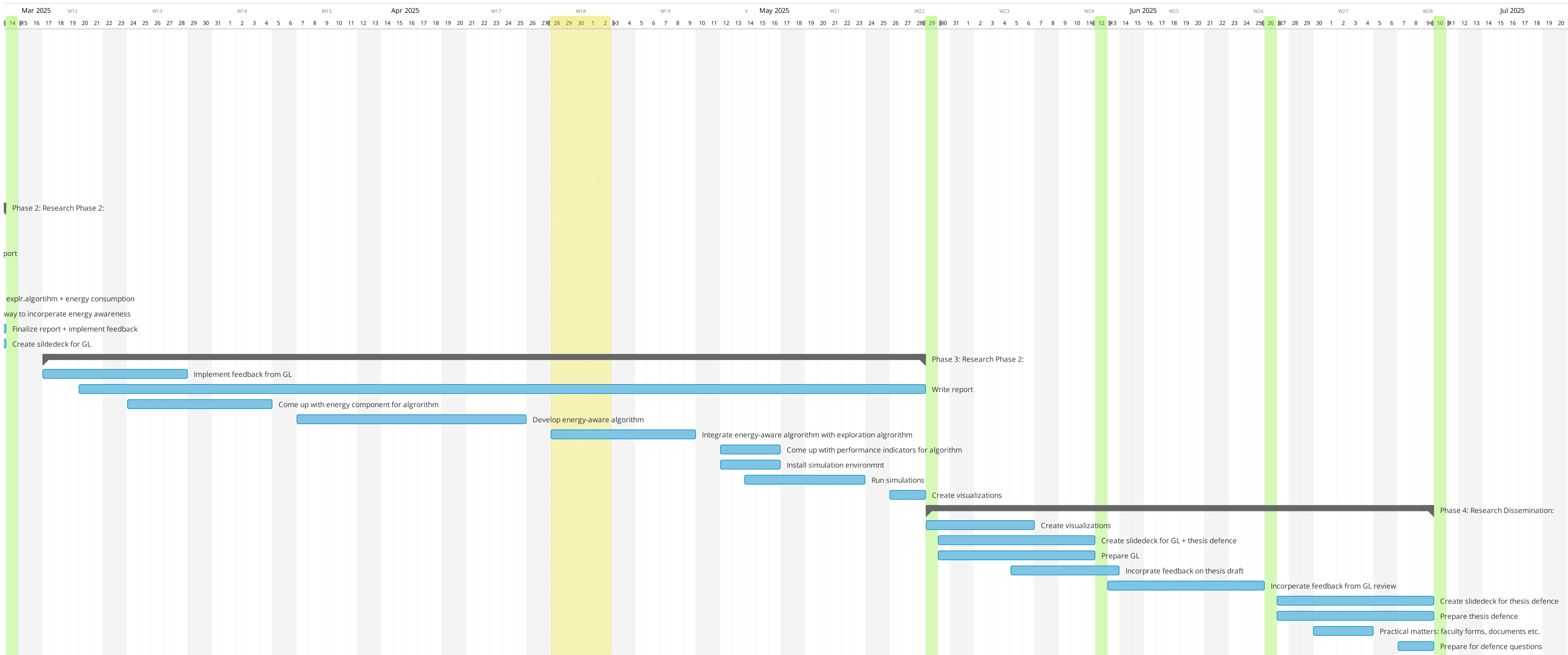
Thesis detailed planning

Read-only view, generated on 13 Jan 2025

	ACTIVITIES	ASSIGNEE	EH	START	DUE	%
	Phase 1: Literature Review:		230h	18/Nov	15/Jan	97%
1	Kick-off meeting preperation	jacob Elskamp	24h	18/Nov	21/Nov	100%
2	Create project description	jacob Elskamp	4h	22/Nov	25/Nov	100%
3	Read exploration theory	jacob Elskamp	20h	22/Nov	26/Nov	100%
4	Read exploration literature share...	jacob Elskamp	40h	27/Nov	06/Dec	100%
5	Write report + make slides for lite...	jacob Elskamp	40h	02/Dec	10/Jan	100%
6	Read other exploration literature	jacob Elskamp	40h	09/Dec	20/Dec	100%
7	Read energy-consumption , enery...	jacob Elskamp	30h	16/Dec	07/Jan	100%
8	Create research objective + quest...	jacob Elskamp	16h	18/Dec	10/Jan	100%
9	Finish literature report	jacob Elskamp	8h	13/Jan	13/Jan	100%
10	Finish literature review slides	jacob Elskamp	8h	14/Jan	15/Jan	0%
	Phase 2: Research Phase 2:		384h	15/Jan	13/Mar	0%
12	Start to find reproducable explora...	jacob Elskamp	48h	15/Jan	24/Jan	0%
13	Incorperate feedback on literatur...	jacob Elskamp	8h	20/Jan	20/Jan	0%
14	Write findings in report	jacob Elskamp	40h	27/Jan	07/Mar	0%
15	Download required software & pa...	jacob Elskamp	32h	27/Jan	31/Jan	0%
16	Reproduce V1 exploration algorith...	jacob Elskamp	160h	03/Feb	28/Feb	0%
17	Think of way of integrating explr.a...	jacob Elskamp	24h	03/Mar	05/Mar	0%
18	Find way to incorperate energy a...	jacob Elskamp	32h	05/Mar	11/Mar	0%
19	Finalize report + implement feedb...	jacob Elskamp	8h	05/Mar	13/Mar	0%
20	Create sildedeck for GL	jacob Elskamp	32h	05/Mar	13/Mar	0%
	Phase 3: Research Phase 2:		342h	17/Mar	28/May	0%
22	Implement feedback from GL	jacob Elskamp	32h	17/Mar	28/Mar	0%
23	Write report	jacob Elskamp	80h	20/Mar	28/May	0%
24	Come up with energy component ...	jacob Elskamp	30h	24/Mar	04/Apr	0%
25	Develop energy-aware algorithm	jacob Elskamp	120h	07/Apr	25/Apr	0%
26	Integrate energy-aware algrorith...	jacob Elskamp	30h	28/Apr	09/May	0%
27	Come up wtith performance indic...	Unassigned	10h	12/May	16/May	0%
28	Install simulation environmnt	jacob Elskamp	10h	12/May	16/May	0%
29	Run simulations	jacob Elskamp	20h	14/May	23/May	0%
30	Create visualizations	jacob Elskamp	10h	26/May	28/May	0%
	Phase 4: Research Dissemination:		230h	29/May	09/Jul	0%
32	Create visualizations	jacob Elskamp	40h	29/May	06/Jun	0%
33	Create slidedeck for GL + thesis d...	jacob Elskamp	40h	30/May	11/Jun	0%
34	Prepare GL	jacob Elskamp	16h	30/May	11/Jun	0%
35	Incorprate feedback on thesis draft	jacob Elskamp	16h	05/Jun	13/Jun	0%
36	Incorperate feedback from GL rev...	jacob Elskamp	20h	13/Jun	25/Jun	0%
37	Create slidedeck for thesis defence	jacob Elskamp	38h	27/Jun	09/Jul	0%
38	Prepare thesis defence	jacob Elskamp	32h	27/Jun	09/Jul	0%
39	Practical matters: faculty forms, d...	jacob Elskamp	4h	30/Jun	04/Jul	0%
40	Prepare for defence questions	jacob Elskamp	24h	07/Jul	09/Jul	0%



Instagantt



Mar 2025
Apr 2025
May 2025
Jun 2025
Jul 2025
Phase 2: Research Phase 2:
port
explr.algortihm + energy consumption
way to incorperate energy awareness
Finalize report + implement feedback
Create sildedeck for GL
Phase 3: Research Phase 2:
Implement feedback from GL
Write report
Come up with energy component for algrorithm
Develop energy-aware algorithm
Integrate energy-aware algrorithm with exploration algrorithm
Come up witth performance indicators for algorithm
Install simulation environmnt
Run simulations
Create visualizations
Phase 4: Research Dissemination:
Create visualizations
Create slidedeck for GL + thesis defence
Prepare GL
Incorprate feedback on thesis draft
Incorperate feedback from GL review
Create slidedeck for thesis defence
Prepare thesis defence
Practical matters: faculty forms, documents etc.
Prepare for defence questions