REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

AUTONOMOUS MULTIPLE GROUND TARGET DETECTION AND TRACKING

Ömür YILDIRIM

MASTER OF SCIENCE THESIS
Department of Electronics and Communications Engineering
Program of Electronic Engineering

Supervisor
Assoc. Prof. Dr. Revna ACAR VURAL

January, 2023

REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

AUTONOMOUS MULTIPLE GROUND TARGET DETECTION AND TRACKING

A thesis submitted by Ömür YILDIRIM in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE** is approved by the committee on 17.01.2023 in Department of Electronics and Communications Engineering, Program of Electronic Engineering.

Assoc. Prof. Dr. Revna ACAR VURAL
Yildiz Technical University
Supervisor

Approved By the Examining Committee

Assoc. Prof. Dr. Revna ACAR VURAL, Supervisor
Yildiz Technical University _____

Prof. Dr. Nurhan TÜRKER TOKAN, Member
Yildiz Technical University _____

Assoc. Prof. Dr. Yiğit Dağhan GÖKDEL, Member
İstanbul Bilgi University _____

I hereby declare that I have obtained the required legal permissions during data collection and exploitation procedures, that I have made the in-text citations and cited the references properly, that I haven't falsified and/or fabricated research data and results of the study and that I have abided by the principles of the scientific research and ethics during my Thesis Study under the title of Autonomous Multiple Ground Target Detection and Tracking supervised by my supervisor, Assoc. Prof. Dr. Revna ACAR VURAL. In the case of a discovery of false statement, I am to acknowledge any legal consequence.

Ömür YILDIRIM

Dedicated to my family

ACKNOWLEDGEMENTS

I want to start by thanking Assoc. Prof. Dr. Revna ACAR VURAL, my advisor and supervisor during this project, for her ideas, support and guidance. Our discussions and review of the field helped to get the idea to start this project. I'm glad to work with her. I want to thank my co-advisor Prof. Dr. Klaus Diepold. During my stay at the Technical University of Munich, our long discussions and ambitious ideas brought another perspective to make this project better. During my studies, I have used several open-source packages to help my work. I would like to acknowledge the Python community for all the great packages and their documentation, and the Robot Locomotion Group of MIT for their cutting-edge visualization tool director. Additionally, thanks to the team of Cacoo tool, with their contribution I was able to draw flawless diagrams. Despite it may not always feel like it, there is a life outside of this project. Without the love and support of my family, it could be way harder for me to find the energy to get through this process. Especially, I want to thank my sister for helping me with our discussions about environmental monitoring. Finally, I would like to thank Maximilian Ulmer for our long discussions about autonomy and his contribution to his part of this project. I hope our work will lead to better projects.

Ömür YILDIRIM

TABLE OF CONTENTS

LIST OF SYMBOLS

α	Angle of given variable
r	Diameter of given parameter
ε	Error rate of given variable
μ	Mean value
π	Pi
σ	Standard deviation
Δt	Travel time
V	Velocity

LIST OF ABBREVIATIONS

2D	Two-dimensional
3D	Three-dimensional
AI	Artificial Intelligence
CCPP	Complete Coverage Path Planning
GPS	Global Positioning System
IMU	Inertial Measurement Unit
LIDAR	Laser Imaging, Detection and Ranging
LiPo	Lithium Polymer Battery
m	meter
m/s	metre per second
mAh	milliampere hour
min	minutes
NN	Neural Networks
s	second
SLAM	Simultaneous Localization and Mapping
TUM	Technische Universität München
UAV	Unmanned Aerial Vehicles
VCD	Vertical Cell Decomposition

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Autonomous Multiple Ground Target Detection and Tracking

Ömür YILDIRIM

Department of Electronics and Communications Engineering
Master of Science Thesis

Supervisor: Assoc. Prof. Dr. Revna ACAR VURAL

Environmental monitoring has a key role to reduce human damage to nature and wildlife. Due to intense tracking and observation tasks, it's an expensive solution. Autonomous observation is an open discussion to raise the efficiency of environmental monitoring and to reduce the cost of operation. Drones are a possible candidate for environmental monitoring with their proven success in monitoring and tracking. Thus, we are offering autonomy for monitoring with drones that are capable of flying autonomously, e.g., covering a given area, and are able to perform certain monitoring tasks, e.g., identifying dead trees.

In this thesis, we implement a decision process for an autonomous drone for environmental monitoring. We implement an improved vertical decomposition algorithm in order to create complete coverage path planning in environments with complex obstacles. Further, we investigate, implement and compare two different cellular decomposition algorithms (trapezoidal decomposition and boustrophedon decomposition) and present the performance of these algorithms in various scenarios. Finally, we test our system in a monitoring scenario where we collect information about leftover bottles on the campus. We execute simulations on the given scenario and present the results we obtained. Our simulation results prove that our suggested system is able to execute a variety of environmental monitoring tasks autonomously.

Keywords: Target detection, environmental monitoring, UAV, decision making, autonomous navigation

ÖZET

Otonom Çoklu Yer Hedefi Teşhis ve Takibi

Ömür YILDIRIM

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

Danışman: Doç. Dr. Revna ACAR VURAL

Çevresel izleme, doğa ve vahşi yaşam üzerindeki olumsuz insan etkisini azaltmak için önemli bir role sahiptir. Yoğun takip ve gelişmiş gözlem yeteneği gerektirdiği için çevresel izleme pahalı bir yöntemdir. Bu nedenle, otonom gözlem maliyeti düşürdüğü ve verimliliği artırdığı için otonomluğun çevresel izleme için uygulanması günümüzde önemli bir tartışma konusudur. İnsansız hava araçları izleme ve takip konularında ispatlanmış başarıları ile otonom çevresel izleme için umut verici adaylardır. Bu çalışmada, otonom hareket edebilen insansız bir hava aracı ile çevresel izleme için otonom çalışan bir sistem öneriyoruz.

Bu tez çalışmasında, çevresel izleme yapabilen bir otonom insansız hava aracı için bir karar verme mekanizması geliştirdik. İçersinde kompleks engeller bulunan bir alanın komple taranmasını sağlayacak bir rota oluşturmak için geliştirilmiş bir dikey ayrışma algoritması geliştirdik. Bu algoritmaya ek olarak, iki değişik hücresel ayrışma algoritmasını (trapezoidal ayrışma ve boustrophedon ayrışma) tasarladığımız ayrışma algoritması ile birlikte kullanarak, söz konusu algoritmaların değişik uygulamalardaki performanslarını karsılaştırdık. Son olarak, tasarladığımız otonom çevresel izleme sistemini kampüste bırakılmış boş şişelerin takip edildiği bir çevresel izleme senaryosunda simule ettik. Simülasyon sonuçlarımız, otonom insansız hava araçlarının çeşitli çevresel izleme görevlerinde kullanabileceklerini ispatlamaktadır.

Anahtar Kelimeler: Hedef tespiti, çevresel izleme, İHA, karar verme, otonom navigasyon

1
INTRODUCTION

1.1 Literature Review

Environmental monitoring as described in the European Commission's Environmental monitoring (2018) article, persistent systematic studies in which the environmental conditions are unveiled with the purpose of appraising the progress made to accomplish given environmental targets, and also to facilitate the detection of new environmental circumstances. Results of environmental monitoring effects outlines of environmental policies. Environmental monitoring is divided into subcategories; air monitoring, water monitoring, ecosystem services monitoring, biodiversity monitoring, fire, drought and flood monitoring, by European Union Science Hub.

Environmental monitoring is being used to determine possible harm caused by humans to the environment. Responsible foundations and governments try to monitor and manage natural resources, track species at risk, and save protected areas. Environmental monitoring tasks require state of art observation ability. An observer should be able to observe a specified environment periodically or continuously. Monitoring tasks often require access to uncivilized areas. Depending on the target, the observer needs to track several data sources. Further, due to the big size of fields, monitoring tasks take a long time. Apart from these challenging conditions, the quality of gathered data and the certainty of complete coverage is key to characterize the environment. Executing such highly demanding tasks with human intervention is expensive and unstable. Further, in uncivilized areas, a manned investigation may not be possible. Therefore, organizations seek a way to limit human intervention, increase the accessibility of monitoring, and reduce the costs of environmental monitoring. In this manner, autonomous systems are good candidates for environmental monitoring to reduce cost and expedite the monitoring process and to be used in conditions where a manned investigation is not possible.

Monitoring tasks demand intensive workload and continuity. To satisfy these

demands, space-borne (e.g. satellites) and airborne (e.g. drones) remote sensing can be used. A space-borne remote sensor has a higher distance to the surface compared to an air-borne remote sensor. While a higher distance to the surface boosts the coverage capability, it struggles with data extraction difficulties such as the resolution of the image and cloud effect. Further, deployment and maintenance of a space-borne sensor cost drastically more compared to an air-borne sensor. Hence, air vehicles like UAVs can achieve better results with less cost compared to satellites.

To give a brief summary of the foundation of this study, we will describe the concepts of autonomy, autonomous navigation and autonomous vehicles. Following, we will explain environmental monitoring and its applications. Finally, we will give insight into monitoring using autonomous vehicles.

1.1.1 Definitions

Autonomy, the ability or a state of being self-governing; acting separately from others. Application of this concept in aerial monitoring represents air vehicles that are able to operate monitoring tasks without any human interaction.

Autonomous navigation, the ability to reach a target location autonomously through ascertaining a collision-free path from starting point to target point, often called path planning. In order to calculate the path, the system requires a map of the environment. For applications in which an initial map of the environment is not provided, the system has to calculate the map by processing the sensor measurements while traveling in the environment. Simultaneous localization and mapping (SLAM) is a common approach for calculating the map of an environment without prior knowledge of the environment. SLAM is widely used for indoor autonomous scenarios. In cases an initial map of the environment is provided, path planning can be done before the system starts to travel [1]. Yet, even if the system has prior knowledge of the environment, the system needs to recognize its position while traveling, which is known as localization, to follow the path with adequate efficiency. For outdoor scenarios, the position of the vehicle can be estimated from the integration of GPS and inertial measurements. By introducing the control module [2], the autonomous flying system can navigate throughout the path given the estimated pose by the localization in consideration of the measurements of the onboard sensors.

Autonomous vehicles, can move from point A to point B without any driver interaction. Autonomous vehicles aim to increase safety by leveraging their capability of sensing the surroundings, avoiding collisions, and detecting and identifying objects and the environment. Autonomous vehicles reduce the error rate caused by drivers

with help of numerous safety policies and they eliminate the negative effects of stress or tiredness from traffic.

1.1.2 Environmental Monitoring

The incessant change in the state of ecosystems, biological communities, and species is a result of natural events and the activities of humans owing to the increase in human population and its increasing dependency on the Earth's resources. There is no doubt that we are experiencing one of that times of environmental change. This reveals the importance of putting effective environmental monitoring measures in place to be able to observe and catch these changes, also the contributing causes that affect them as [3] states.

Environmental monitoring has been determined as the observation and study of the environment which serves to assess the state of the environment. The objective observations that develop data, which in return support useful information such as promoting the public water supply protection, vegetation, radioactive waste management, or global climate change as described by [4]. From the perspective of the environment, monitoring its meaning does not only stands for measuring or observing, rather implies watching closely for control purposes; surveillance; tracking; checking continually; detecting change. The change hints at time, thus monitoring measures that change in an environment over time and space.

This systematic measurement of variables and processes related to a particular issue, is designed to supply information on the features of the issue and their changes over time, thus monitoring is necessary for a specific purpose and specifying targets against it, those targets may measure compliance with environmental legislation or policy, or ensuring whether given standards that designed to protect the environment are being met, as well as their effectiveness. For instance, it might take the form of an early warning for a vegetation change or for the purpose of taking preventive actions in monitoring the levels of effluent from a given factory draining into a river. The purpose in mind and narrowing targets of resources grants environmental monitoring plans cost-effective.

As described in **Fondriest Environmental Inc.**'s *Environmental Monitoring Applications* article, as is known what makes statistical data reliable and robust, the quality of the practices in the progress of monitoring might be influenced through (i) information to have accessibility in a format that people can easily access at a reasonable cost, and being transparent, (ii) information to be accurate at the current state of the art, (iii) coherent information to able to combined with other related information

previously collected, (iv) the methods and process of data collection should be given as supplementary information for ease of interpretation, (v) relevance information that is important to user, (vi) information to be the time that is ready as soon and as close to the measurement time. Besides the quality control monitoring plan should be well-coordinated with other applications that observed the same area, such as through the BACI (Before-After-Control-Impact) model that had designed to provide an assessment of the effects of a given activity upon the environment at a particular location, e.g., monitoring the extent of salinity of agricultural land due to fluctuation in vegetation allows comparisons to be made between the current data and the previous data before there is a noticeable or risky impact.

The task of environmental monitoring continues to become more important as residual development and technological progress are made. The data collection gathered through environmental monitoring is essential to data-driven decision-making, therefore, this task involves policymakers as well as organizations and individuals, even, municipal engineers, public health officials, health care providers, or farmers. Besides its sphere of influence, there are many benefits of technological advancement in terms of the methods, devices, and techniques used to monitor the environment in regard to the existence of some challenges.

1.1.3 Monitoring Using Autonomous Vehicles

Autonomous vehicles, such as flying, naval and territorial robots, have a great effect on exploration and tracking tasks such as leak detection, pollution monitoring, research and rescue, resources exploration, wildlife tracking and many other fields. With respect to natural protection and efficient usage of natural resources, monitoring tasks are important as well as challenging problems.

Monitoring tasks usually take a long time due to the size of fields, besides, monitoring tasks often take place in uncivilized areas. Therefore, the compelling conditions of monitoring tasks becloud maintaining high-quality observations. To increase the precision and robustness of a manual monitoring system, manual monitoring systems need to be invested more workforce which increases the costs. Though organizations' intentions to reduce human workforce and costs contradict manual monitoring systems, hence, there appeared a variety of more qualified methods and devices that are vital in implementing management tasks and measuring environmental programs such as satellite techniques, or unmanned vehicles and personal assistants. As in [5] satellite techniques have been used for coastal management plans to locate areas prone to erosion due to variable conditions on shores to control the potential risk. However, this intention to reduce human workforce and costs advanced in the way to reach the

least human workforce, thus, demand for the capability of a higher level of autonomy increased as the technology improved. In this regard, autonomous systems are used to lower the cost of the monitoring process and to execute monitoring tasks going through where a manned observation is not possible.

Approaches for environmental monitoring problems can be categorized based on the observation approach. As [6] describes, the first approach is creating a network of sensors, where sensors collect measurements and exchange measurements with each other in order to track the environment and estimate the location of the event. This localization approach requires the usage of reference points which causes expensive and time-consuming deployments. Furthermore, this strategy limits the coverage area. The second approach contains the usage of the vehicle(s) equipped with the appropriate sensor(s) that moves towards the defined area. This approach is more dynamic and more flexible compared to a fixed network of sensors. Yet, it requires solving many challenging technical issues such as planning, navigation and endurance. Collaborative research of [7] shows that resolution plays an essential role in identifying the more accurate variety of structures through a UAV equipped with autonomous search which is able to capture higher-resolution images to identify features in the water along the River Dee in Wales. This experience, in turn, holds importance for the ecological qualifications of that specific river. As an instance to further implication of [8], a specific method called ultrasonic beacons, is used to supply positional data that enables UAVs to conduct an autonomous task through collecting video data to ensure the structural health of a building or bridge, in turn, harms or cracks in the structure successfully detected.

To be able to conduct a monitoring task autonomously, the proposed system needs to have certain capabilities. First, the proposed system needs to be able to sweep a defined field autonomously. To achieve this, the proposed system needs to produce a route that ensures complete coverage of the field. Following, the proposed system needs to identify the targets during the complete coverage of the field and once the target is identified the proposed system needs to investigate the target. Finally, the proposed system needs to execute previously stated tasks while ensuring the safety of the environment and itself. Therefore, we identify the following three capabilities that are required for a system that executes monitoring tasks using autonomous vehicles; complete coverage path planning, autonomous inspection and flight safety.

1.1.3.1 Complete Coverage Path Planning

Complete coverage path planning (CCPP) constructs a path that leads a vehicle to discover every part of a defined area. CCPP resolves the essential problem of coverage

in environmental monitoring missions. There are robotic applications such as cleaning robots [9], autonomous underwater covering vehicles [10], painter robots [11], lawn mowers [12], agricultural crop harvesting equipment [13], have implemented a special type of path planning in a two-dimensional (2D) environment.

Complete coverage tasks can be done by ground or aerial vehicles [14, 15]. Yet, complete coverage of wild areas, e.g., monitoring a forest, can be more challenging to execute for a ground vehicle compared to an aerial vehicle. Further, an aerial view of fields provides a better angle to execute cost-efficient complete coverage.

Depending on the monitoring target, the proposed approaches can be characterized by different requirements. Coverage algorithms differ as algorithms that require the basics of the environment, which describes the blocked and free space, and the algorithms that can be executed in an unknown environment.

There are several methods that tackles path planning problem in unknown environments, e.g., template-based model, fuzzy logic and neural networks. In study [1] template-based approach is used for CCPP where algorithms rely on the prior map knowledge of the environment and cope with unknown obstacles. The template-based approach is applied to direct the path planning, thus applied in the incorporation of the kinematic and the geometric model of the mobile robot during path planning. Neural networks and fuzzy logic based approaches to CCPP, develops a model which generates a path through self-learning. Neural network-based approaches are being used for unknown environments. Such as study of [9] develops a self-learning CCPP approach, which can avoid obstacles in an unknown environment. Neural network approaches require high computational power which causes complications in hardware design and power consumption. In the study of [16] a behavior model based on fuzzy logic is suggested. The model provides a solution to guide a robot to clean an unknown room environment from any start point. Yet, difficulty in defining suitable fuzzy rules has affected the smoothness of paths and resulted in paths that are not suitable enough for maneuvering and traversing.

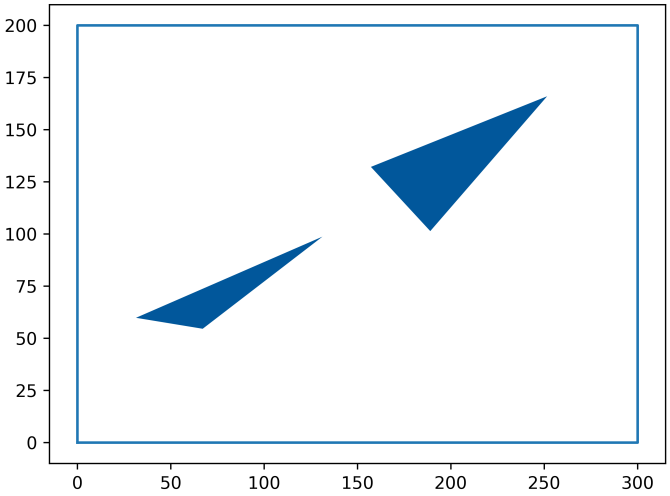
In this study, we focus on static path planning. Our missions take place in environments where the obstacles are fixed and known in advance. Our system is expected to achieve complete coverage in 2D space. There are many studies in the literature that tries to find an optimal path in a 2D static environment, e.g., artificial potential field, roadmap method, rapidly-exploring random trees and cellular decomposition.

Artificial potential field approaches such as [17] use a method to discretize the field and the robot motion. Based on obstacles and goal points they are creating a repulsive

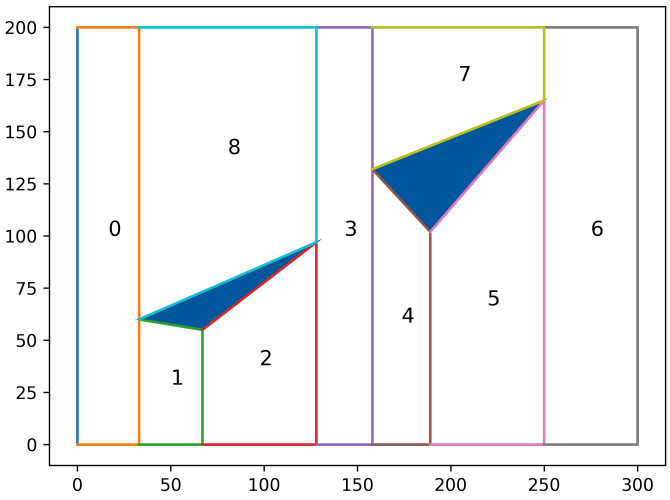
potential field and an attractive potential field in the workspace. Results show that the algorithm can construct a path to achieve complete coverage. Though the algorithm suffers with the local minimum, hence they become inefficient for the exploration of complex maps such as the TUM campus map we use in this study.

The roadmap methods produce a path by connecting collision-free areas in the field by developing a network of collision-free curves. The curves form a collection of standardized paths which is called a roadmap. Once the roadmap has been built, path planning continues by calculating the start and the target points of roadmap curves and establishing a connection between these plotted points. There are several methods available for developing the roadmap. In the study [18], a visibility graph method is used for the path planning of an autonomous land-based robot. In another study [19], a Voronoi diagram method was used to find the optimal path between the start and destination in an environment with simple disjoint polygonal obstacles. Although the results of the studies show that the roadmap method is able to compute the optimal path, roadmap calculation requires high computational effort.

The rapidly-exploring random tree approach introduces a randomized data structure to solve the path planning problem [20]. An application of rapidly-exploring random tree [21] demonstrates a successive extraction of frontiers and coverage of the entire field in a reasonable amount of time. The disadvantage of the rapidly-exploring random tree approach is that it is hard to optimize the randomness for the exploration range of the UAV we used in this study. Further, due to the limited resources of the UAV, continuous computational power requirement during the flight makes the algorithm less preferable.



(a) Polygonal environment consists of 2D trapezoid-like shapes. Blue colored polygons are obstacles. White areas represents collision-free space of environment.



(b) Determined trapezoidal cells

Figure 1.1 An application of trapezoidal cell decomposition

Cellular decomposition approaches are based on a concept that decomposes the field into a stack of non-overlapping cells. Then, the system creates a graph from connections of cells that expresses the adjacency between cells. In each cell, the system makes back-and-forth motions to achieve complete coverage. The most common cellular decomposition method is trapezoidal decomposition [22], where applied to a special 2D polygonal environment consisting of non-intersecting polygonal obstacles and a polygonal boundary. Each cell lies between two successive polygon vertices. In a further study of the boustrophedon decomposition [14], the system minimizes

the number of cells used in cell decomposition by combining cell decomposition and template-based approaches. In the study [23], a new method is presented as Morse decomposition which generalizes boustrophedon decomposition to include non-polygonal obstacles. However, it cannot handle surfaces parallel to the sweep line.

Many studies in the literature suggest a novel method to do the path planning of an autonomous vehicle. When we compared their trade-offs, we decided to use the cellular decomposition method in favor of efficiency and simplicity. Other methods we evaluated above don't offer a significant improvement to the cellular decomposition, instead, they increase the complexity and the cost. Further, we decided to improve the decomposition algorithm to support path planning in maps with complex obstacles.

1.1.3.2 Autonomous Inspection

Inspection tasks such as pipeline monitoring or forestry tracking, demand extensive decision processes and intensive labor. Approaches can use machines, tools, human labor, and/or other suitable items to execute the tasks. Various inspection techniques can be needed to monitor the objects or the environment, e.g., tracking suspected damage events or determining whether scheduled or preventative maintenance is required.

One of the issues that inspection systems need to solve is continuous or periodic data analysis which is needed to detect and monitor the condition of the subjects. Furthermore, inspection tasks often need to be executed at the fields has limited access. Data acquisition for the inspection task may vary depending on the subject type, such as methane gas detection which needs a methane sensor or deer tracks detection which needs appropriate image processing. Considering these challenges and demands of an inspection task, autonomous approaches are advantageous compared to manual approaches. Common approaches for autonomous inspection systems consist UAVs with image acquisition capabilities. UAVs provide an aerial perspective of the subject which often provides better conditions to discover large areas. Yet, hardware limitations of the UAVs create issues for the onboard computational capability which is highly demanded most of the image processing applications.

In study [15], an autonomous UAV is used for search and rescue tasks. They used a stereo camera to localize the UAV depending on the surrounding objects and to detect and inspect targets. The UAV flies autonomously until detects a target or gathers an emergency exception. In another study [24], an autonomous inspection system

suggested to detect emergencies in wild areas. They provide a multi-UAV application to provide real-time data from ongoing fires. The suggested system kept up-to-date overview images of the affected area to the officers.

1.1.3.3 Flight Safety

UAV consists of a highly integrated multitask technology that involves orchestrating software and hardware to ensure flight and communication. Therefore, it is not surprising that the UAVs will take the place of manned air crafts as technology advances. However, despite its great potential for the future, there are still limits to ever-developing UAVs.

First of all, as every aircraft, UAV must be safely integrated into civilian airspace. Critical safety issues related to UAVs are the risk of collision with other UAVs and the risk of loss of control as stated in **Flight Safety Foundation**'s *Unmanned Aircraft Systems Key Safety Issues* article. Furthermore, if UAV flights autonomously, safety risks should be tracked and avoided by the UAV. In this way, Critical safety issues for autonomous UAVs are the loss of control during flight, the loss of localization, the mechanical issues, the collisions, the wildlife issues and the fatigue. The loss of control can be caused by a control mistake or by an opposing weather condition such as wind or rain. During the implementation of a task, the flight safety from the takeoff until the return to the starting point is highly dependent on the sensor technology of the UAV. However, even if it is required the sensors to have hyper-spectral, pressure, or any higher technology to accomplish oversensitive control, small-sized or micro UAVs are yet not able to carry such technology.

During an outdoor autonomous flight, localization is commonly provided through a GPS module. The UAV leads to the target location, controls its speed by leveraging supplied position from the GPS module, and this is not different compared to what a pilot in a manned aircraft actually does. In case of a shortcoming of a weak GPS signal, or any cut in connection UAV may face decision instability, or worse, crashes and accidents. Therefore, UAV is expected to improve its capability as the data link technology advances towards higher anti-jamming capability, high speed and better reliability.

Mechanical issues, e.g., damaged propeller, can cause flight instability. These issues may lead the UAV to lose control and crash. Further, a possible intersection between an UAV and an object or a structure or another UAV can cause a collision. Even more, a collision can be caused by a potential animal strike, e.g., conflicts with birds. After a certain flight time, fatigue of the UAV can affect flight stability. To overcome these

challenges, the compliance of artificial intelligence technology and the UAVs is also expected in the future, such as object recognition systems, an avoid technology, etc.

One another challenge for the UAVs is the need for more lasting power or viability. The viability depends on the size of the batteries and the power consumption of the system. Therefore, flight time varies depending on the features and the size of the UAV, e.g. while a high altitude large UAV with turbofan engines can fly for hours despite being weighted over tons, a multi-rotor UAV with electric driven motors is not durable over an hour flight time though they are weighted around 100g. In this regard, new energy plants like hydrogen power, solar power or so, can be used as the energy source for the small UAV models.

1.2 Objective of the Thesis

To solve the autonomous environmental monitoring problem, the designed system should be able to execute certain tasks such as coverage and object detection. Between tasks, there should be a robust decision-making mechanism to keep the balance between protecting the UAV and surroundings, and accomplishing a successful mission. We suggest dividing monitoring tasks into cognition and vision layers, to distinguish the autonomy of UAV and object detection. The vision layer is responsible to detect and identify targets. It detects objects and identifies the detected target. In this work, we calculate the output of the vision layer by estimating the probability of target detection concerning certain conditions of the environment and UAV.

The cognition layer is responsible to create an efficient path for total coverage of the monitored field, to make the decision to switch between tasks, and protecting the UAV and surroundings from potential harm.

This work investigates the decision process of an autonomous UAV for environmental monitoring. In particular, we present a scenario in which UAVs could play a key role. We execute several simulations on the given scenario and present the results we obtained.

We aim to introduce a system that can plan and execute an environmental monitoring task autonomously. We aspire to design a decision-making framework that can be used with a variety of monitoring tasks with minor adjustments and we expect our system to increase the safety and the maintainability of the monitoring missions compared to earlier implementations of monitoring systems. We introduce an improved decomposition algorithm to add the capability to plan routes in environments with complex obstacles.

This thesis project has several objectives:

1. Present theories related to the field and previous approaches of environmental monitoring.
2. Present autonomy and autonomous vehicles with their application to monitoring tasks.
3. Design a framework for an autonomous UAV for environmental monitoring.
4. Present logical approach and implemented algorithms of autonomy.
5. Design a scenario where an autonomous UAV executes environmental monitoring tasks.
6. Build a simulation environment and simulate designed scenario of environmental monitoring with presented approach.

1.3 Hypothesis

Achieving complete autonomy for environmental monitoring requires solving many complex problems. Complete autonomy demands intense data analysis as well as structured decision-making processes. With respect to these issues hypothesis of this thesis project:

> **We purpose a cognitive system, which is aware of its surroundings and capable to make decisions depending on environmental conditions and its well-being, can be used for the autonomous decision-making process of an unmanned aerial vehicle for environmental monitoring.**

2
APPROACH

In this chapter, the study of this thesis is presented. First, we describe the conceptual framework of the thesis. The following methodology and algorithms of the autonomous approach are being presented.

With our proposed approach, we aim to introduce a system that can plan and execute an environmental monitoring task autonomously. We expect our system to increase the safety and the maintainability of the monitoring missions compared to earlier implementations of monitoring systems. We introduce an improved decomposition algorithm to add the capability to plan routes in environments with complex obstacles. Hence, the proposed system increases efficiency and reduces costs while increasing the safety of monitoring tasks.

2.1 Conceptual Framework

Autonomous information gathering in an uncontrolled field requires a powerful and robust decision-making architecture. The autonomous vehicle must have an awareness of its surroundings, position, target, events of the environment and well-being. It should observe and evaluate the state by using certain awareness metrics.

In this thesis project, an autonomous system for environmental monitoring is constructed upon two main layers, cognition and vision. The vision layer receives raw sensor data as input and outputs the estimation of the possible target and identified target data. The cognition layer executes tasks that are responsible for decision-making and taking real-time actions. To be able to fulfill cognition layer tasks, management of UAV's control unit, chain of actions depending on the output of the vision layer and tracking events in the environment, e.g., collision or mechanical issues, are needed. This study focuses on the cognitive decision process, in other words, the cognition layer of an autonomous environmental monitoring system.

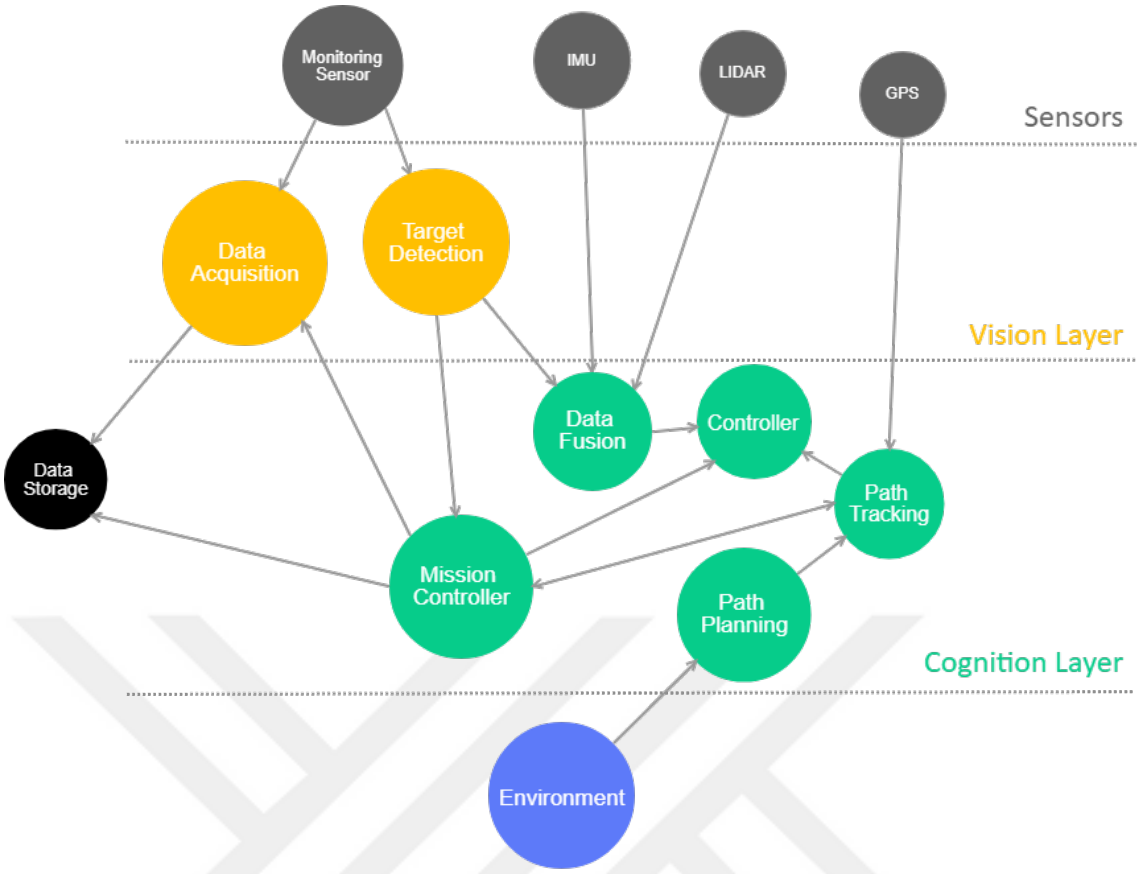


Figure 2.1 Autonomous environmental monitoring system architecture

2.2 Cognition Layer

Monitoring tasks can be divided into levels in regard to tracking. In the first level of monitoring, the definition of the environment and recognition of targets should be done. Recognition is subjective in the manner of quality of inspection, e.g., image observation may be affected by the angle of the camera or daylight. In the next level of monitoring, the observer should be certain to not overlook any part or target of the field. With respect to levels of monitoring, a cognition layer for an autonomous monitoring system is designed to act in three decision states; path tracking, target acquisition and emergency. To handle each decision state, the cognition layer is separated into three modules; coverage module, investigation module and safety module. While each module has separated action and decision logic, each module is in the effect by other modules for their actions and decisions. The design of modules gives flexibility in usage for different monitoring applications.

The main goal of the cognition layer is to serve optimal observation conditions to the vision layer while ensuring the safety of the UAV. To achieve this goal, a robust decision-making structure is constructed. We first clustered actions and decisions to the states of the monitoring task. Following this, we defined several policies to

successfully perform tasks of monitoring. Finally, we implemented algorithms for the policies with respect to efficient resource usage.

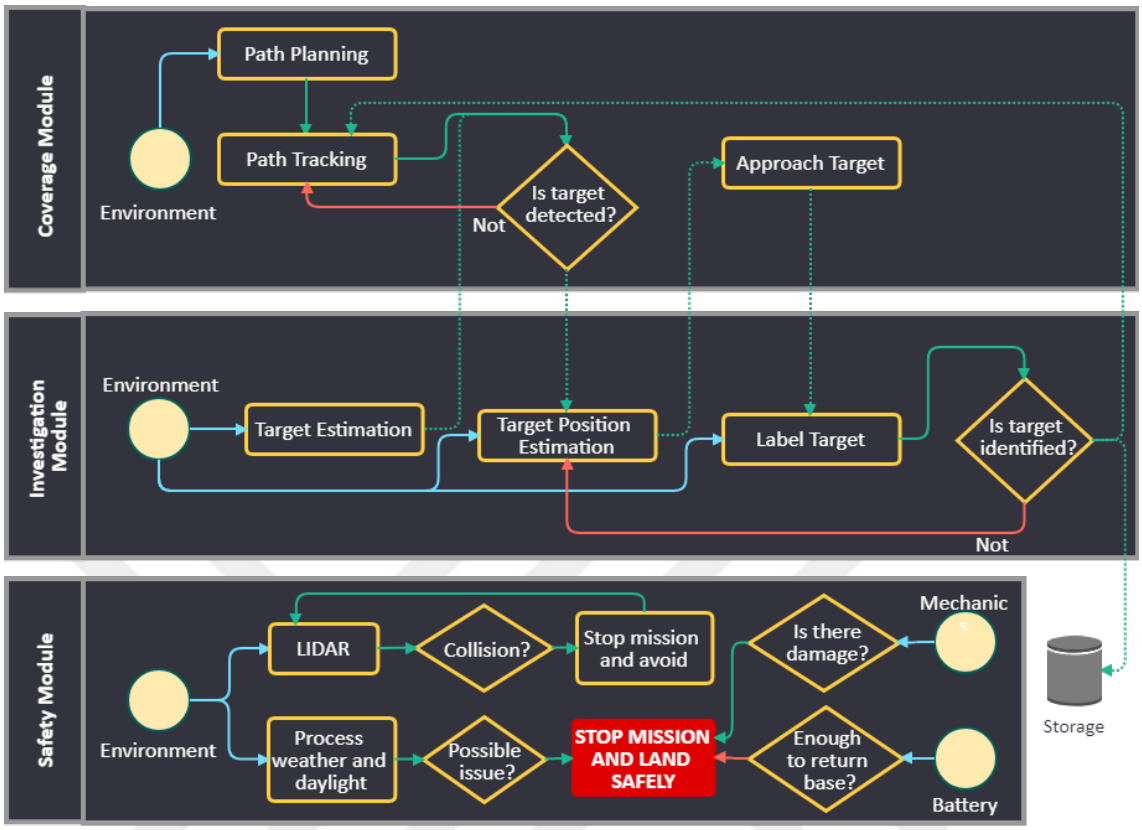


Figure 2.2 Cognition module framework and algorithm

2.2.1 Decision Making

The ability to make a decision depends on deep knowledge about the specifics of a state, such as understanding the conditions for a state X in order to prevent Y from happening. The most advanced level of decision-making ability belongs to intelligent living beings. To take an action, they need to observe or experience. The biology of living beings supports decision-making in a way to protect themselves. They may use their instincts or reflexes to balance the gain/loss ratio of decisions. Yet, biological capabilities like instincts can not be more useful.

The structure of a decision depends on the logical comparison of input(s). During the process of the decision, the decider needs measurement(s) and measure(s). The success rate of a decision is depended on the quality of the measurement and the measure. To successfully imitate a human-like decision logic, machines need a model of the decision.

Modern approaches such as neural networks (NN), integrate human-like learning through observation without using a predefined set of rules. Yet, to create a successful

decision logic, training data of NN must be adequate to model the decision. The increasing complexity of decision logic leads to demanding data preparation processes. Furthermore, advanced NN approaches demand expensive computation. For the design of an autonomous UAV, one of the important metrics is the usage of resources. In an application such as environmental monitoring, the computational power of the identification and tracking tasks consumes a greater part of the power resources. Because of the limited resources of UAVs, using an on-board real-time NN based monitoring approach is not efficient. One way to solve the power resource problem of UAVs is using a ground station for making the computations. Yet, a ground station brings restrictions to fully autonomous systems such as delays in real-time reactions and limit in flight range. Thus, we decided to use a policy-based approach to aid the decision-making process of our autonomous UAV. The basis of decisions is represented by a set of rules, e.g. if the detected object is the target then get near to it. In our study, decisions are categorized according to the three states of the monitoring mission; path tracking, target acquisition and emergency.

2.2.2 Decision States

Decision states are the general expression of different measurements and rules. States represent the status of the UAV with respect to the environment and the present task. To clarify the structure of the states, we first explain the rules and the measurements for each state.

2.2.2.1 Path Tracking State

The aim of the path tracking is the complete coverage of the field by using the calculated path. Tracking of the path is based on localization information and motion estimation of the UAV. GPS data provides position information to the UAV. Localization is calculated by combining prior knowledge of the environment location and GPS data. With tracking information from IMU, UAV is capable to know the direction and the dynamics of itself.

Before the path tracking state is becoming active, the map of the environment is processed to a group of target points. Then calculated target points are sequenced to achieve the minimum movement. The path of the UAV is based on these sequenced target points. UAV travels from one target point to another in the given sequence. The path tracking policy keeps UAV's direction in the way of the next target. Furthermore, the percentage of total coverage is calculated during the path-tracking state. The UAV starts the mission with a path tracking state and tries to keep the path tracking state until it reaches complete coverage.

2.2.2.2 Target Acquisition State

The cognition layer tracks the target estimation output of the vision layer. If the vision layer outputs a possible target, then the UAV gets into the target acquisition state. During the target acquisition state, the main task of the UAV is to investigate the possible target and collect necessary data. To be able to investigate the target from a better perspective, UAV needs to position itself to the relative position of the target according to the UAV.

Estimation of the target position is one of the outputs of the vision layer, as represented in Figure 2.1. After processing the target position and the GPS data, UAV takes a chain of actions. During the investigation, the UAV takes decisions based on the image quality feedback of the vision layer. Depending on the image quality, UAV tries to get to an optimal distance for observation. The optimal distance of the UAV may change depending on the type of monitoring task. In image processing based monitoring, the optimal distance is usually closer to the target, so that UAV is expected to be at a certain distance to serve a better perspective for the vision layer. If the UAV gets a bad image quality feedback at the optimal distance, then it decides to move between optimal positions, e.g. drawing circles around the target. To increase the efficiency of the resource usage, UAV may decide to break the target acquisition. Decision merit to cut target acquisition state is based on the average time spent during target identification. Once the target acquisition is complete the UAV returns to the path tracking state.

2.2.2.3 Emergency State

During the path tracking and the target acquisition states, the UAV watches for potential emergencies. In case of the sensor measurements indicate potential harm, the UAV gets into the emergency state. In the emergency state, the main task of the UAV is to ensure to avoid potential harm from itself and the surroundings. During the emergency state, all other tasks become inactive. Once the emergency state is recovered, the UAV returns to the last state before the emergency.

2.2.3 Coverage Module

The coverage module is responsible for sweeping the field completely. During the autonomous flight, the module is active continuously because the monitoring task begins with the start of coverage and ends once complete coverage is achieved.

The main tasks for the coverage module are; calculating the path, moving along the path and tracking the coverage percentage. Before the take-off for the monitoring mission, the path for the complete coverage is calculated. The calculation requires

prior knowledge of the coordinates of the field and the restricted areas inside the field, e.g., fixed obstacles or not interesting parts of the field. The module uses an extended vertical cell decomposition algorithm to divide the field into cells. Following it calculates a graph of the adjacency of the cells. It searches for the Eulerian path [25] inside the graph. Finally, it calculates back-and-forth motion paths (boustrophedon path) in each cell and by connecting each motion path from the first cell to the last cell, it generates the complete coverage path.

During the flight, UAV uses GPS for localization. It tracks the coverage path and estimates covered areas. It has a certain altitude (2 m) and speed (10 m/s) during the cruise. However, tracking of the path can get interrupted by the investigation module due to target detection. In the case of target detection, the state changes from path tracking to target acquisition. The coverage module handles the transition between states. Furthermore, the safety module can output an emergency during coverage. Before the safety module takes an emergency action, the coverage module is responsible to store the latest coverage progress. After getting through the emergency, the coverage module ensures returning to the latest state and to resume to the mission.

2.2.4 Investigation Module

In a possible target detection input, the investigation module takes charge of the decision-making. The main goal of the investigation module is to serve better conditions to satisfy input quality needs for the vision module. Depending on the output of the vision module, the investigation module takes a chain of actions. To achieve the investigation task, the module takes complete control of the UAV. The module changes the position of the UAV or opens or closes a sensor to increase the quality of the observation.

The UAV is always in the target acquisition state during the lead of the investigation module. Change of the state can occur once the vision module gathers complete information about the target or on an emergency during the investigation. In an emergency, the investigation module is responsible for backup all gathered information from the target.

2.2.5 Safety Module

The goal of the safety module is to avoid or to reduce potential harm to the UAV and the environment. The safety module uses a set of rules to identify emergencies. Rules are created from a collection of previous experiences, the standards of autonomous flights and the health merits of the mechanical parts of the UAV. It depends on certain

sensor measurements to check safety rules.

Decisions of the safety module can change the state of the UAV to an emergency state. In most cases, if the module decides on a possible emergency, the safety module actions interrupt the monitoring mission and the system will need a restart. Only the detection of collision does not interrupt the mission. In a potential collision detection, the safety module takes actions to avoid the collision and if the module gets success, the UAV continues to the monitoring task.

The safety module uses several sensors. Laser Imaging, Detection And Ranging (LIDAR) is used for collision calculations. It is expected to be a collision-free space during cruising in flight altitude. Therefore, LIDAR can be used to detect any blockage in the cruising direction. In addition to the LIDAR, a more advanced image processing approach can be used to detect the objects. Collision avoidance can be done more efficiently With the detected object information compared to LIDAR. Though, the required computational power for a LIDAR based approach is significantly lower compared to an image processing based approach.

Weather conditions of the environment can be used to avoid potential instabilities for flight such as strong winds or rainstorms. Weather forecasts can be examined before the flight to predict potential disruptions. A more advanced weather condition tracking can be done by using onboard sensors. Although onboard sensors can provide real-time data about the environment, their size and weight is making them inefficient.

Daylight condition is another important merit for flight safety. Bad light conditions cause lower image quality for the cameras. Therefore, flight safety can be affected by reduced image processing capability. Hence, the daylight can be tracked by an onboard photo-diode and daylight information can be used to foresee potential vision disruption.

2.3 Technical Requirements

An autonomous system for environmental monitoring should be able to cover a certain field with minimal human control/interaction. The system should avoid obstacles and situations that might disrupt the tasks. An autonomous environmental monitoring system aims to achieve minimal interruption, hence starting and finishing a monitoring task in one attempt. To measure the success of the monitoring task, we specified the following technical requirements.

In an autonomous environmental monitoring task, the UAV is expected to cover the

field with maximum battery usage efficiency. One of the main challenges of UAVs is their limited capability of short flight times due to their limited payload capacity. Further, during the monitoring task they need to investigate any detected target and gather detailed information about the target which will reduce the battery capacity for the coverage task. One of the solutions to increase the flight time of UAVs is increasing battery size. But increasing battery size affects the total load of the UAV, which increases the required lifting force. Hence, causing the need to use more powerful motors or to increase the number of motors.

UAVs can be categorized according to motor usage. They can be separated as multirotor, fixed wings and single rotor (helicopter). Multirotor UAVs are easy to manufacture, and the most economical compared to other types. Yet, multirotor UAVs come with reduced flying time, reduced endurance, and reduced speed compared to other types of UAVs. Fixed-wing UAVs use the principle of airplanes. They generate the lift by using a wing instead of vertical thrust generating rotors. Using the airplane flight principle produces higher speeds and travel distance but causes difficulty to loiter over one point of interest. Single-rotor UAVs use vertical thrust generating rotors to fly. They have increased capability for flight time. Yet, single-rotor UAVs have the drawback of the increased maneuvering complexity, cost, vibration and danger of large spinning blades which is causing a safety issue for autonomous flight. In comparison to these types of UAVs, we decided to use multirotor UAVs because of the reduced cost and their maneuvering capability in limited spaces.

Multirotor UAVs are classified according to size; small, medium and professional (industrial). Small and medium sized multirotor UAVs have small batteries due to the payload capacity of motors. This causes a short flight time. Currently, they are capable of maximum flight times between 20 to 30 min with a minimal payload. Industrial UAVs have increased payload capacity. Yet, their flight time is not much longer than small and medium sized UAVs, and they cost much more compared to other types.

In this thesis, we use a medium sized autonomous UAV with 4500 mAh battery capacity during simulation. The average current of the UAV is calculated as 8000 mA with 10 m/s velocity. In maximum usage of the battery, the UAV can fly for around 34 min. Yet, the battery should not be reduced under 30% to protect the battery from the harm of reduced voltage. With consideration of battery safety percentage, the average flight time of the UAV should be around 24 min.

The UAV is expected to cruise at 10 m altitude where the UAV will face fewer dynamic obstacles compared to the surface of the field. We decided to take an offline approach for path planning of the complete coverage of the field. Once the coverage path is

calculated, the UAV is expected to lift and start the monitoring mission.

The UAV is equipped with an adjustable camera according to the angle of the slope on the surface. The camera is used to monitor the surface and detect the targets. We decided to use a single camera with an adjustable connection instead of using multiple cameras. One of the drawbacks of this decision is not having the stereo effect to calculate the depth. Though, we compensate stereo effect by relying on the elevation calculation of the onboard sensor. Another idea was to increase the quality of the lens to increase the detention range. We selected a 20mm lens because of the weight-angle of view balance. Using a fish-eye lens was another option. But to avoid the normalization computations, hence to save power, we avoided the fish-eye lenses. Using a mechanical focus lens was another option to avoid vertical movement when investigating a potential target. Though the weight increase with a mechanical focus lens would reduce the flight time noticeably, thus we decided to use a fixed-focus lens. Additionally, a LIDAR sensor is placed in the direction of the horizontal movement axis. LIDAR sensor is used to detect the obstacles. During the monitoring, the UAV may be interrupted due to detected targets or safety issues. To handle the navigation of the UAV in these occurrences, we used an online path-tracking approach.

At the 10 m altitude, the detection range is calculated as 2.5 m from the center of the camera. When the UAV moves straight from one point to another as in Figure 1.1, it covers a rectangular area with 2.5 m width and travel distance length. Because of the circular detection range, it also covers one circle with 2.5 m radius. To formulate coverage calculation, we can use the equation

$$x \star 2.5 + 2.5^2 \star \pi/2 = y \tag{2.1}$$

where x is total travel distance in meters and y is total coverage in square meters.

To summarize our technical requirements:

1. Planned path should cover whole surface of field. Path planning is done with an offline approach.
2. During the coverage, the UAV should be able to handle interruptions.
3. The UAV is expected to use maximum 70% of its 4500 mAh battery.
4. Expected maximum flight time of the UAV is 23 min and 38 s.
5. Maximum cruise speed of the UAV is 10 m/s.

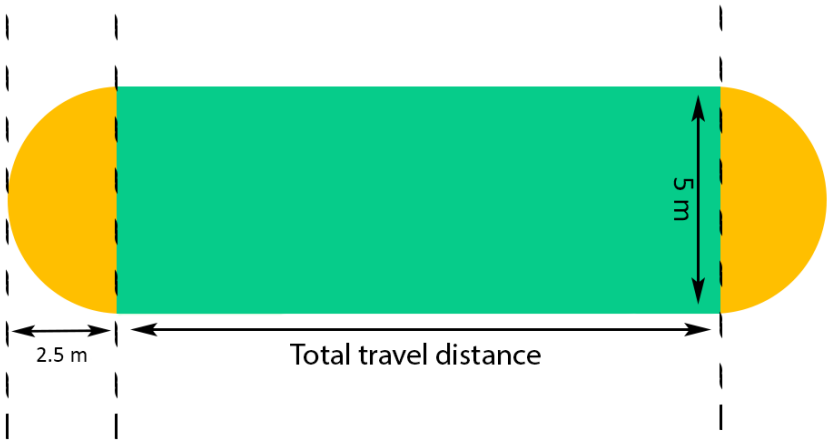


Figure 2.3 Total coverage for travelled distance
Green part is covered are due to movement while yellow part is covered area due to circular target detection range

6. The UAV is expected to cover 50 m^2 per 1 m movement. After completion of movement 9.81 m^2 additional coverage is expected due to 2.5 m target detection radius

2.4 Algorithms

In this section, the studied approach of each cognition module and its algorithms are explained. First, we introduce the decision making procedure. Then we explain the complete coverage approach and the path planning algorithm. Finally, we introduce the emergencies, the safety rules and the emergency actions.

2.4.1 Complete Coverage Path Planning and Path Tracking

In this thesis, we introduce an extended Vertical Cell Decomposition (VCD) algorithm in order to achieve complete coverage. Common trapezoidal approaches such as VCD assumes that all obstacles in the field are in general position. An obstacle in general position can be described as an object without an edge on vertical lines and without two vertices located on the same vertical line. Assuming all obstacles are in general position reduces the complexity of the algorithm, hence reducing the power consumption. But it makes the algorithm less useful in scenarios with complicated surface obstacles. Since we want to provide a monitoring solution that can work in any field definition, we improved the VCD and suggested a new extended VCD algorithm to decompose fields with obstacles without being in general position.

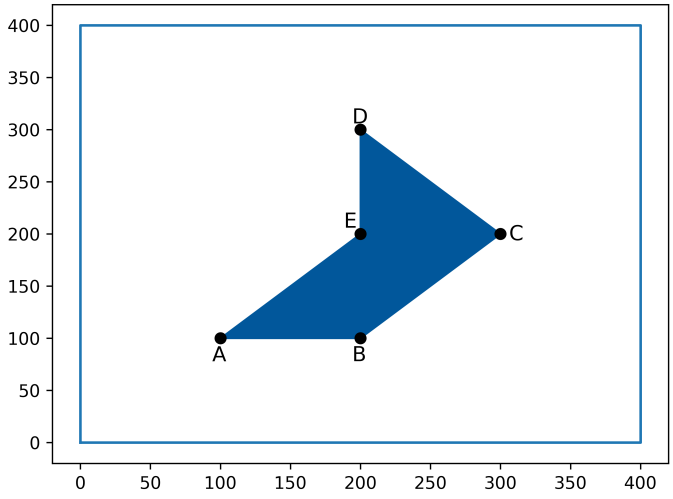
As other cellular decomposition algorithms, [26] explains that VCD works as follows:

1. Divide free space of environment to connected regions often called cells.
2. Determine adjacent cells and construct an adjacency graph. Cells are the vertices of the graph that have a common boundary.
3. Search for a path in the adjacency graph between start and end cells.
4. Compute a path connecting certain points of cells common their centroids via the center points of the boundaries.

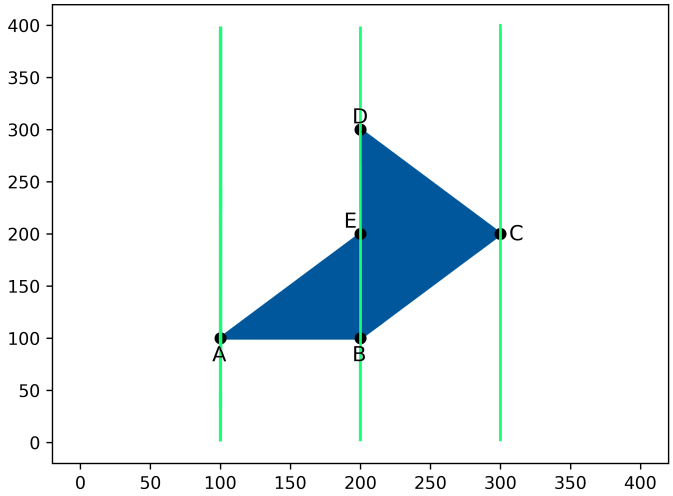
A vertex in 2D space can be defined with its x and y coordinates. If two vertices are in the same vertical line, the extended VCD calculates virtual vertices which distinctively hold neighbor vertex coordinates. The algorithm decides the virtuality of a vertex depending on the connection between its neighbor vertex. If an imaginary line between two vertices intersects with the obstacle then the algorithm keeps the vertices as normal. If the imaginary line does not intersect with the obstacle then the algorithm labels vertices as virtual.

The term virtual vertex comes from its behavior during cell calculation. When a virtual vertex is used to calculate cross product with another vertex to determine the possible intersection between vertices and obstacles, a virtual vertex acts like it's making vertical movement between its position and its neighbour's position.

As it's displayed in Figure 2.4, vertex B, D and E is placed on the same vertical line at x=200. Vertex B has the above neighbor vertex (E). Yet, an imaginary line between vertex B and vertex E has an intersection with the polygon. With this reasoning, the algorithm calculates vertex B as a real vertex, therefore the variable for vertex B holds only its coordinates. Vertex E has one below (B) and one above (D) neighbor. An imaginary line between vertex E and its below neighbor vertex B has an intersection with the polygon. Yet, vertex E and vertex D are vertices of an edge, therefore the line between vertex E and vertex D does not intersect with the polygon. In this case, the algorithm marks vertex E and vertex D as virtual vertices and the variables for vertex E and vertex D holds virtual coordinates.



(a) Blue colored polygon with named vertices



(b) Detected vertical lines in green color

Figure 2.4 A sample polygon and vertical lines from vertices

Algorithm is as follows:

In the common VCD approach, during the cell calculation phase for a field as in Figure 2.4, vertex A will be matched with vertex B and the algorithm misses cell between vertex A, D, E and upper boundary of the field. If we don't use the virtual vertex concept, our algorithm will math vertex A with vertex B and vertex A. It will miss vertex D because it will select the first possible vertex match for the upper cell which is vertex E. With virtual vertex D and E, the algorithm matches vertex A and vertex E as well. But because vertex E is a virtual vertex, it checks other points stored in a variable of vertex E and it decides that vertex D is a better match for the cell calculation. The

Algorithm 1 Extended vertical cellular decomposition

```
Get vertices from obstacles
verticalLines ← x coordinate of vertices
cells ← []
for verticalLine in verticalLines do
    for verticesOnLine in verticalLine do
        vertticesLen ← length of verticesOnLine
        if vertticesLen > 1 then
            verticesOnLine ← createVirtualVertices(verticesOnLine)
        end if
    end for
end for
for verticalLine in verticalLines do
    for verticesOnLine in verticalLine do
        for vertice in verticesOnLine do
            cell ← Null
            matchedVertice ← findNearestVertice(vertice)
            while cell is Null and matchedVertice is not Null do
                lineBetweenVertices ← createLineBetweenVertices(vertice, matchedVertice)
                if checkIntersectionWithObstacles(lineBetweenVertices) then
                    cell ← calculateCell(vertice, matchedVertice)
                    cells.append(cell)
                end if
                matchedVertice ← findNearestVertice(vertice)
            end while
        end for
    end for
end for
graph ← calculate graph from cell centroids
tour ← find a path in graph
Decompose cells in given order of tour
```

decision for vertex D selection comes from a y-coordinate comparison. If two vertices are matched as a successive vertex for a cell, a vertex with a higher y-coordinate is selected by the algorithm. For our example in Figure 2.4, vertex A's successive vertices are vertex B and vertex D. This matches create two cells, cell 1 and 2 as we can see in Figure 2.5.

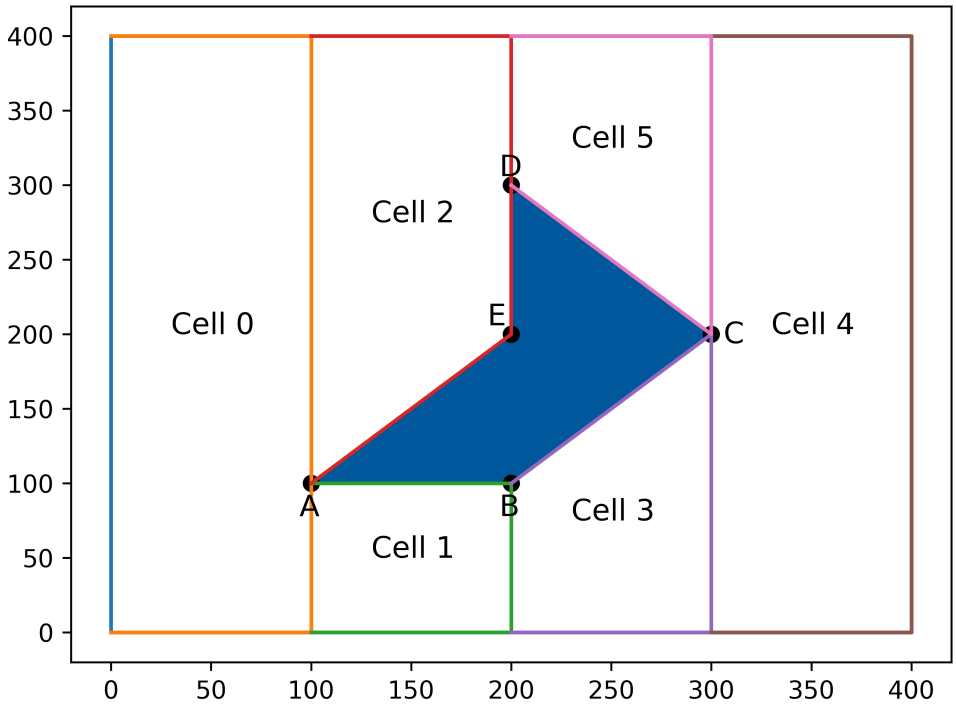


Figure 2.5 Trapezoidal cells of sample map

We used two different algorithms to determine the cells of the given map: trapezoidal decomposition and boustrophedon decomposition.

Trapezoidal decomposition: Trapezoidal cells are the space between two successive polygon vertices. Successive vertices are generally the closest neighbor vertices and an imaginary line from one vertex to another should have no intersection with given obstacles of the map. As in Figure 2.5 where the blue part is the obstacle, vertex A and vertex B are neighbors. An imaginary line between vertices A and B has no intersection with the obstacle. Thus, the space between vertex A and B is the cell 1. But vertex A and vertex C are not the closest neighbors. Further, an imaginary line from vertex A to vertex C has an intersection with the obstacle.

Boustrophedon decomposition: Boustrophedon cells [14] are merged trapezoidal cells. They reduce total cell count and increase simplicity and efficiency due to the fact that reduced cell count minimizes lengthwise transition movements between cells. In both decomposition methods, cells are created between two vertices. The difference between these two approaches is the middle events [27]. The trapezoidal decomposition closes or opens a cell when a middle event occurs. But, the boustrophedon decomposition simply updates the current cell with new border points during the middle events.

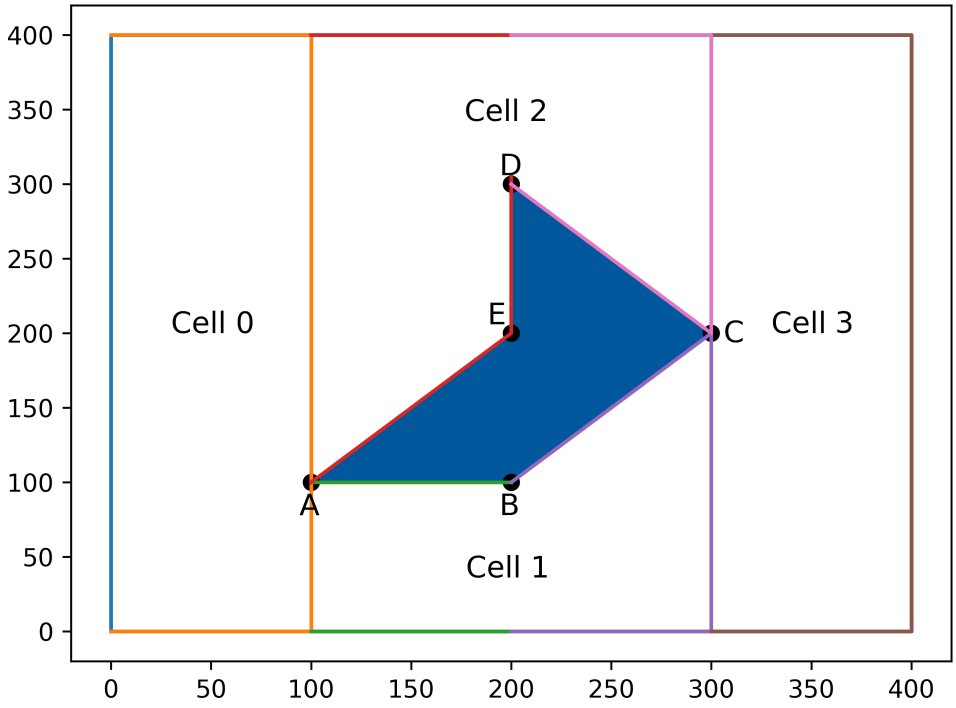


Figure 2.6 Boustrophedon cells of sample map

As in Figure 2.5 where the blue part is the obstacle, cell 1 is defined between vertices A and B and cell 2 is defined between vertices B and C by the trapezoidal decomposition algorithm. But for the boustrophedon decomposition, the intersection point for vertex B is a middle event and causes the extension of cell 1 until the vertex C. Simply, the boustrophedon decomposition merges trapezoidal cells 1 and 3 and produces the boustrophedon cell 1 as in Figure 2.6.

The difference between the boustrophedon decomposition and the trapezoidal decomposition can be seen better with a comparison of figures 2.7 and 2.8. For the TUM map, the trapezoidal decomposition resolves the map to fourteen cells whereas the boustrophedon decomposition resolves it to nine cells.

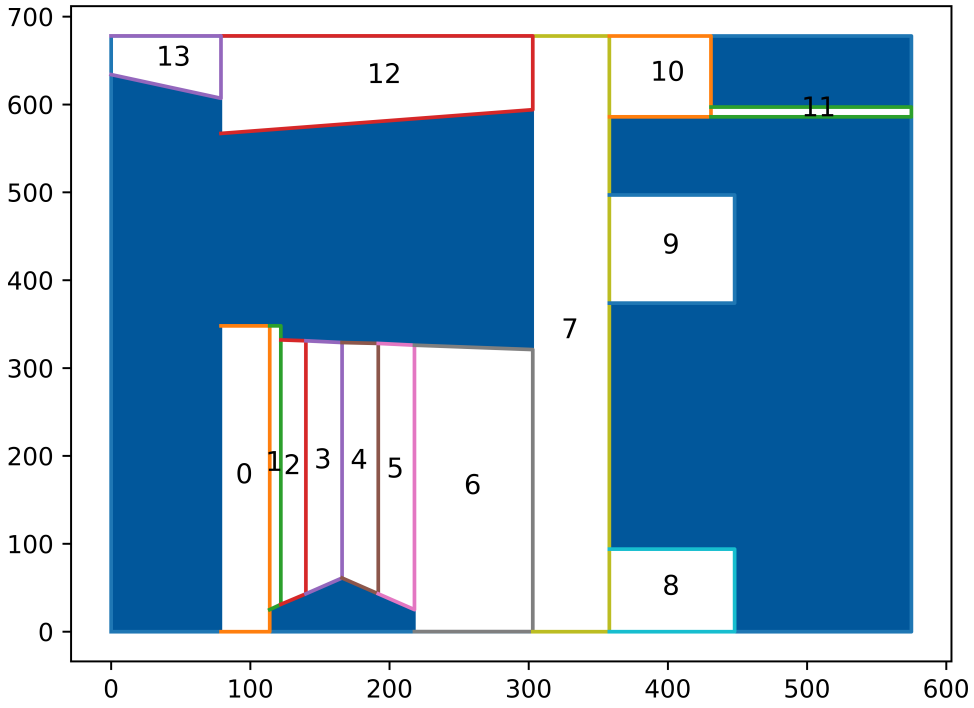


Figure 2.7 Calculated trapezoidal cells in TUM map

Following, from the calculated cells, a graph and a tour from the graph are calculated. The tour starts at point (0,0) and finishes at point (0,0). To satisfy start and end points in Figure 2.6, the order of cells may be arranged as 0, 2, 3, 1 or 0, 1, 3, 2 depending on the last visited point of cell 0. If the last visited point of cell 0 is the top right corner of cell 0, then it makes more sense to continue the tour with cell 2. If the last visited point of cell 0 is the bottom right corner of cell 0, then it makes more sense to continue the tour with cell 1. After tour calculation, the algorithm decomposes each cell and creates the path of coverage as shown in Figure 2.9.

The path of coverage starts from one of the corners of starting cell. Instead of starting coverage from the same corner of each cell, starting from exactly where previous cell coverage ended provides efficiency. If a cell coverage finishes at a point that doesn't have adjacency then it will calculate the nearest corner of the next cell as the starting point. For example in 2.9 (b), coverage starts at cell 0's top-left corner. Coverage of cell 0 finishes at the bottom-right corner which makes cell 1 suitable for the next cell. After coverage of cell 1 coverage continues following cells 3, 4, 5 and 2.

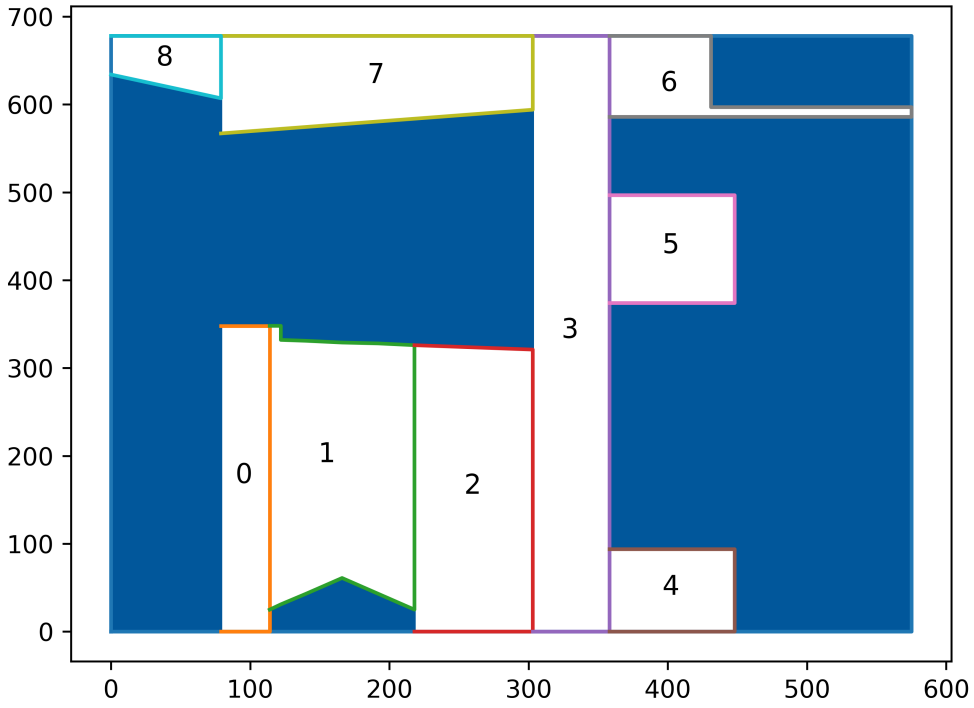


Figure 2.8 Calculated boustrophedon cells in TUM map

2.4.1.1 Path Tracking

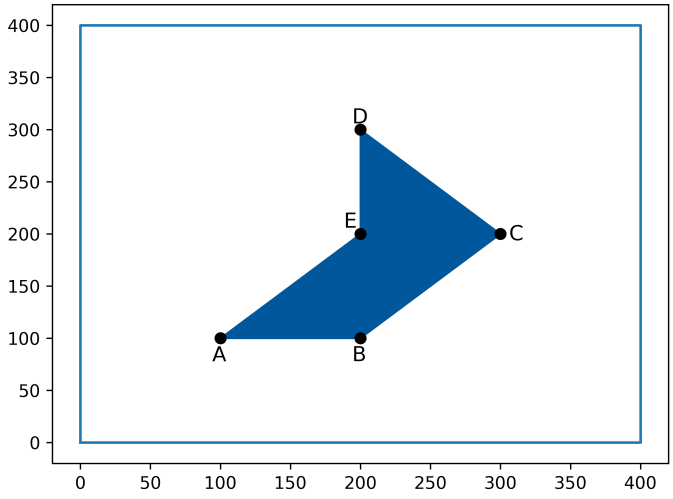
The path tracking algorithm ensures the completion of planned coverage by following the constructed path with help of GPS data. The coverage path stores a sequence of points from the field. Path tracking algorithm defines the position of the UAV according to GPS coordinates. Then, it controls motions from the start point towards the end point. If the coverage module receives a possible target message from the investigation module, path tracking stops until the investigation of the target finishes.

Path tracking algorithms is as follows:

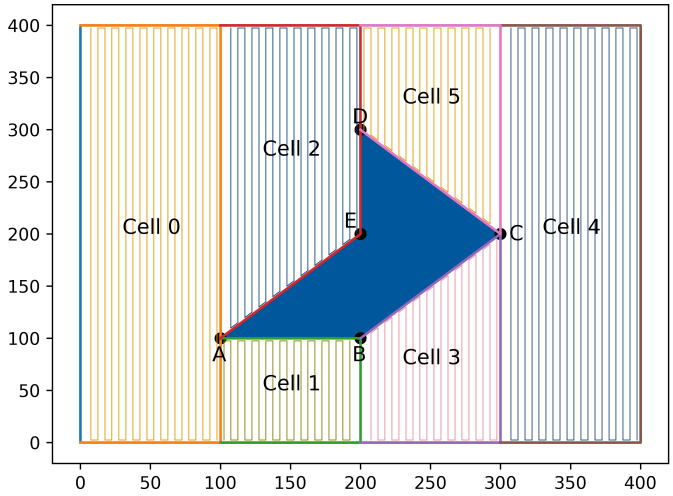
To be able to track total coverage as in Figure 2.10, the algorithm first calculates the total target detection area $A_{detection}$. It is a complete circle as in equation (2.2):

$$A_{detection} = \pi tan^2(\alpha) r^2_{detection} \varepsilon_{altitude} \tag{2.2}$$

where $r_{detection}$ is target detection range, α is the angle of the camera, $\varepsilon_{altitude}$ is error rate of altitude sensor. After, UAV calculates complete coverage $C_{coverage}$ as in equation (2.3):



(a) Reference map, obstacle is displayed with blue



(b) Decomposed map

Figure 2.9 Vertical cell decomposition applied to map

$$C_{coverage} = \pi tan^2(\alpha) r^2_{detection} \varepsilon_{altitude} + \Delta t V 2 r \varepsilon_{acclrm} \tag{2.3}$$

where Δt is the travel time, V is the velocity of vehicle and ε_{acclrm} is the error rate of the accelerometer. Yet, movement during the target inspection is not included complete coverage calculation due to fact that UAV can pass through a position that is already covered.

Algorithm 2 Path Tracking

```
path ← [(x, y) of path]
numberOfPoints ← len(path)
pointNumber ← Checkstoreforrecentpointorstartfrom0
targetPoint ← path[pointNumber]
                                          ▷ While path is not finished
while pointNumber != numberOfPoints do
    gpsLocation ← readGPSInput()
                                          ▷ Check if UAV reached to target point
    if gpsLocation == targetPoint then
        pointNumber++
        targetPoint ← path[pointNumber]
    else
        moveTowards(targetPoint)
    end if
                                          ▷ Break path tracking if target detected
    possibleTarget ← Target information from vision layer
    if possibleTarget then
        store(pointNumber)
        Wait for investigation module output
    end if
end while
```

2.4.2 Avoiding Potential Harm

Detection of potential harm can occur in a series of scenarios where the UAV and/or surroundings may get damaged in case of continuity of the present action. To create cognitive understanding, conditions of a scenario that may lead to potential harm are modeled as safety rules. The system tracks and processes several sensor measurements and determines potential harm by comparing the measurements with safety rules. The safety module applies a set of policies to avoid detected potential harm if any is detected.

Conditions for potential harm scenarios are defined as following:

Figure 2.10 Calculation of complete coverage in simulation, area highlighted as green represents the covered area

1. Obstacle in the direction of the motion.
2. Power level reduces under the minimum level that can protect battery from being permanently damaged.
3. Power level reduces under the minimum level that is required to return to the base station.
4. Damage on the hardware.
5. Loss of the localization data.
6. Optimal daylight reduces under the minimum level that is required to ensure minimum viable picture quality captured by the camera.
7. Adverse weather condition.

2.4.2.1 Collision

Collisions may be caused by objects in the environment, e.g., walls, dynamic objects or another UAV. In the proposed approach, the collisions are predicted by the processing of LIDAR data. Due to the fact the system requires a map of the field before starting the monitoring mission, static obstacles such as walls, fences and unreachable areas are expected to be avoided during path calculation. Further, in the detection of an unpredictable obstacle such as interference with another UAV, the safety module declares an emergency and tries to avoid the potential collision. In the way of identification of the unknown obstacles, our approach may be improved by using an advanced object detection algorithm such as image processing based object detection. Identifying details of an unknown object can lead to better logical decisions, e.g. in the case of identifying an unknown object as a UAV, the system can try to stop instead of making a maneuver to avoid it. Therefore, it can reduce the speed and hence the effect of the collision. Though, due to limitations of the computational power of the UAV, we didn't implement an advanced object detection algorithm.

2.4.2.2 Power Management

Lithium Polymer Battery (LiPo) is a common choice for UAVs due to being a lightweight battery type and the high power performance associated with LiPo. But besides these advantages, the battery needs to be protected against potential voltage swings. Therefore, the remaining level of the LiPo should be tracked. The battery should exceed a certain level at any time, to support a minimum voltage level often defined as 3.0V for each cell of the battery.

The safety module tracks the level of the battery and may declare an emergency depending on the low level of the battery. During the flight, if the battery drops under 30 percent, the safety module stops the monitoring task and lands the UAV safely. Following, it publishes an emergency message with attached location information.

One other metric for the battery level is the minimum power level that is required to return to the base station. The module periodically tracks the balance between the battery level and future power consumption. In case of the power demand exceeds the remaining power of the battery, the safety module tries to return to the predefined safety location, e.g., the base station. Following, if the UAV can't reach the safety location within the batter limits, the algorithm applies the battery protection policy stated in previous paragraphs.

2.4.2.3 Hardware Damage Control

In order to detect the damage to hardware components, the safety module tracks UAV's reactions to the IMU control commands. In the event of an unexpected wobble, a sudden descent of altitude or a lack of reaction to a motion command, the safety module declares an emergency. Yet, the detection precision of hardware harm may be improved with additional sensors and trackers, e.g. checking propeller speeds with cadence censors.

2.4.2.4 Position Controller

Loss of the GPS data is a common issue of monitoring tasks in wild fields. Due to natural effects such as clouds, GPS signal may be blocked or interrupted. In this proposed approach, GPS aids localization of the UAV. In the event of GPS data loss, the UAV continues its task for a certain amount of time and waits for getting the GPS signal again. In case of continuous interruption of the GPS data, the safety module declares an emergency and applies the policy to land.

To reduce the dependency on the GPS data, the position of the UAV may be estimated with an advanced approach such as SLAM. The UAV can have a local position estimation related to the start point or the prior placed objects/markers. As mentioned before, in our proposed approach, advanced localization approaches can not be implemented due to the computational power limit of the UAV.

2.4.2.5 Daylight Controller

The efficiency of the target detention and identification is highly dependent on the image quality. The level of daylight can affect image quality drastically. Therefore, we implemented an algorithm to detect if the level of the light reduces under optimal level. In case of low light, the UAV stops monitoring the mission and returns to the base location. Daylight levels can be tracked by an on-board sensor such as a photo-diode or scheduled daylight database for the monitored field.

2.4.2.6 Weather Controller

Due to the vulnerable dynamics of UAVs, changes in weather conditions can easily affect the stability of the flight, e.g., disturbing wind. Furthermore, the conditions of rain and snow may reduce the efficiency of monitoring. In such cases, the safety module declares an emergency and tries to return to the safety location. Weather data may be gathered before starting the monitoring task from a forecaster.

Algorithm for avoiding harm is as follows:

Algorithm 3 Avoiding Possible Harm

while UAV is flying **do**

▷ Read sensor data

distanceToObstacle ← Processed LIDAR data
batteryLevel ← Measurement of battery level
powerDemand ← Calculate required power to return base
dataOfIMU ← $IMUoutput$
currentAction ← $Decisionstateandnextmotion$
gpsLocation ← readGPSInput()
totalGPSIntteruptionTime ← 0
daylightLevel ← Read daylight sensor or database
weather ← Data from forecaster

▷ Check for collision

if *distanceToObstacle* lower then distance limit **then**
Change direction of motion
end if

▷ Check battery level

if *batteryLevel* lower then safety level **then**
Land safely
else if *batteryLevel*is lower then*powerDemand* **then**
Return to base
end if

▷ Check mechanics

if *compareActionAndReaction*($currentAction, dataOfIMU$) **then**
Land safely
end if

▷ Check GPS interruption

if *gpsLocation*is not defined **then**
totalGPSIntteruptionTime ← unit time
if *totalGPSIntteruptionTime*is higher then time limit **then**
Land safely
end if
else
totalGPSIntteruptionTime ← 0
end if

▷ Check Daylight

if *daylightLevel*is under optimal **then**
Return to base
end if

▷ Check weather conditions

if *weather*is windy or rainy or snowy **then**
Return to base
end if
end while

3
IMPLEMENTATION

In this section, first, we explain the dependencies of our implementation. Following, we discuss the Director tool. Finally, we present the test scenario for environmental monitoring and we explain how we implemented the simulation with the presented scenario.

3.1 Dependencies

The implementation tools play an important role to increase the simplicity and effectiveness of the implementation. After the definition of our mission and requirements, we decided to implement our algorithms in Python. Although there are various programming languages that we could have chosen, the agility and stability of Python programming was the main factor for our decision. Another key factor was the fact that raising popularity of Python language in the developer community. Because of the increasing usage of Python, there is a vast resource of examples with related works. Furthermore, Python is accepted and broadly used for many popular scientific applications. Hence, the Python community creates a lot of useful tools such as Artificial Intelligence (AI) tools Shogun, Panda, Scikit-Learn and Tensorflow.

In the implementation, we wanted to test our algorithms in a virtual testing environment to simulate our proposed approach and prove the capabilities of our system. After we had made our decision to use Python, we researched to find a robotic simulation environment. There are several cutting-edge simulation tools, such as LPZRobots, MARS and Synthesis. Yet, we have used the MIT Robot Locomotion Groups simulation tool Director [28] in order to run and visualize our simulation. We selected Director because of its easy implementation compared to the other tools. Furthermore, Director has detailed documentation with great example usages and Director had the best visuals for the simulation preview.

There are several Python packages that are used for implementing our algorithms.

First, we discuss them. Then we discuss our simulation tool.

3.1.1 Python

For the implementation of the algorithms of this project, we used Python version 2.7. In order to install Python 2.7, one of the POSIX systems or a Windows system is required. Further, to install scientific Python packages, the Python package management tool *pip* or *virtualenv* is needed, also a Python distribution like Anaconda may be used.

We used several Python packages to increase the consistency and maintainability of the work.

3.1.1.1 NumPy

NumPy is a Python package for scientific computing. It's useful for a variety of computations such as linear algebra, Fourier transform, and random number capabilities. We used NumPy to make complex matrix computations and to process our error probability equations for creating a realistic testing environment.

3.1.1.2 Matplotlib

Matplotlib is a library for Python to plot 2D figures in a variety of hard copy formats and interactive environments across platforms. We used Matplotlib to produce outputs of the path planning algorithm.

3.1.1.3 Shapely

Shapely is a Python package for the manipulation and analysis of planar geometric objects. It brings a great set of instances of geometric objects such as Point, LineString, or Polygons. Shapely finds predicates and relationships within object instances which we commonly used in our simulation environment to calculate intersections between objects or to find container objects.

3.1.2 Director

Director is a robotics interface and 3D visualization framework developed with C++ and Python. The tool supports an easy integration with outsourced algorithms and serves a rich viable environment for visualization. Yet, it's challenging to find example

implementations of the Director. We used Director to create and run our simulation environment.

Dependencies of director is as follows:

- Python 2.7 and NumPy
- Qt4 or Qt5. Qt is a cross-platform application development framework.
- VTK 6.2+, VTK or Visualization Toolkit is an open-source software system for 3D computer graphics, image processing, and visualization.

Additionally, we used PythonQt package to bind Qt framework to our Python app.

3.2 Monitoring Scenario

As we discussed through the thesis, the main goal of this project is developing a cognition system to control an UAV for autonomous environmental monitoring. To test the success rate of the proposed approach, we designed a monitoring scenario. While we researched for a potential scenario, we realized that students tend to leave empty bottles of their drinks at various places inside the campus. We decided to monitor the campus to find and identify the leftover bottles.

We created the map of the main building of Technische Universität München (TUM) campus at Arcisstraße 21 as in Figure 3.1. The main task of the designed monitoring scenario is the detection of bottles and the identification of the bottle labels around the campus. To be recognized as successful, the system is required to identify all bottles placed on the campus while achieving complete coverage of the field without causing any damage to itself and its surroundings. Further measurements to calculate the success rate of the mission are the location precision of the detected bottles, the mission completion time and the total travel distance.

3.3 Simulation

The monitoring scenario as described in section 3.2 is simulated with Python implementation and visualized by Director. In the simulation, an UAV is executing the monitoring task in the field as shown in Figure 3.1 a. The UAV executes the task by relying on the algorithms explained as in 2.

The simulation environment is created uniquely before each execution of the simulation. The blueprints of the campus in Figure 3.2 b are being used to create a 3D

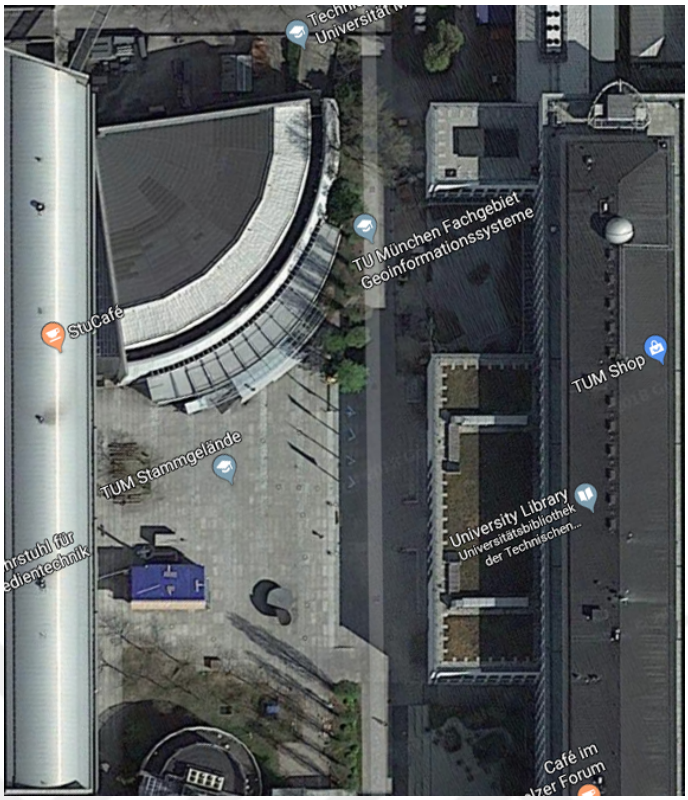


Figure 3.1 TUM main campus from aerial angle
Captured from Google Maps at 48°08'56.4"N 11°34'01.1"E

representation of the environment in Director. After the creation of the environment, 10 empty bottles were randomly placed. Further, bottles are labeled randomly between different brands.

The 3D model of the UAV which is based on an open-source model project **Air Drone 3d model (2018)** in *Free3D.com* is being rendered as in Figure 3.3. Additionally, the LIDAR sensor with its interaction with the environment and target detection range of the drone are visualized as in Figure 3.4.

In the simulation, between the parts of the main building, the UAV searches for randomly placed. The UAV is not aware of the total bottle count at the beginning. Therefore, the UAV travels the whole map to achieve complete coverage of the collision-free space of the field to find as much as the bottle it can. Further, UAV detects and identifies the bottles, and stores all detected bottles with their positions in memory. Finally, it tries to avoid collisions and tries to use power sources efficiently.

In the simulation, the UAV is equipped with a LIDAR sensor with 10 m detection range, a high precision GPS sensor, the target detection and the identification module and a 4500 mAh LiPo battery. The UAV is controlled in 3D space with adjustable velocity and acceleration. The UAV is capable rotating in 3-axis.

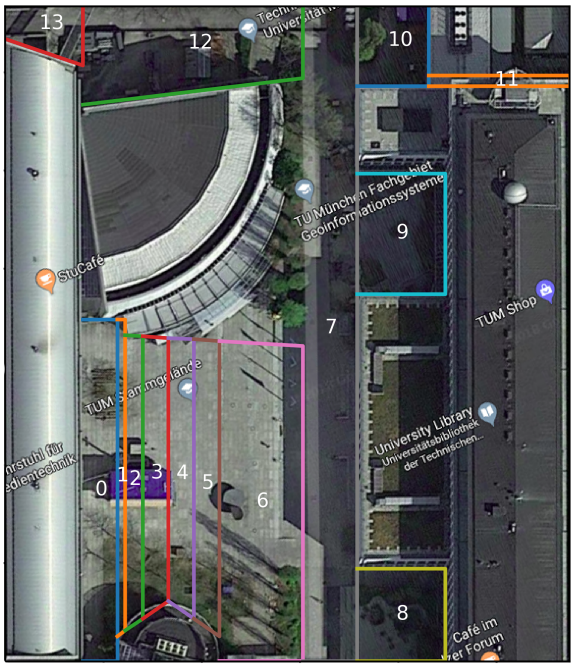


Figure 3.2 Calculated cells displayed on TUM map

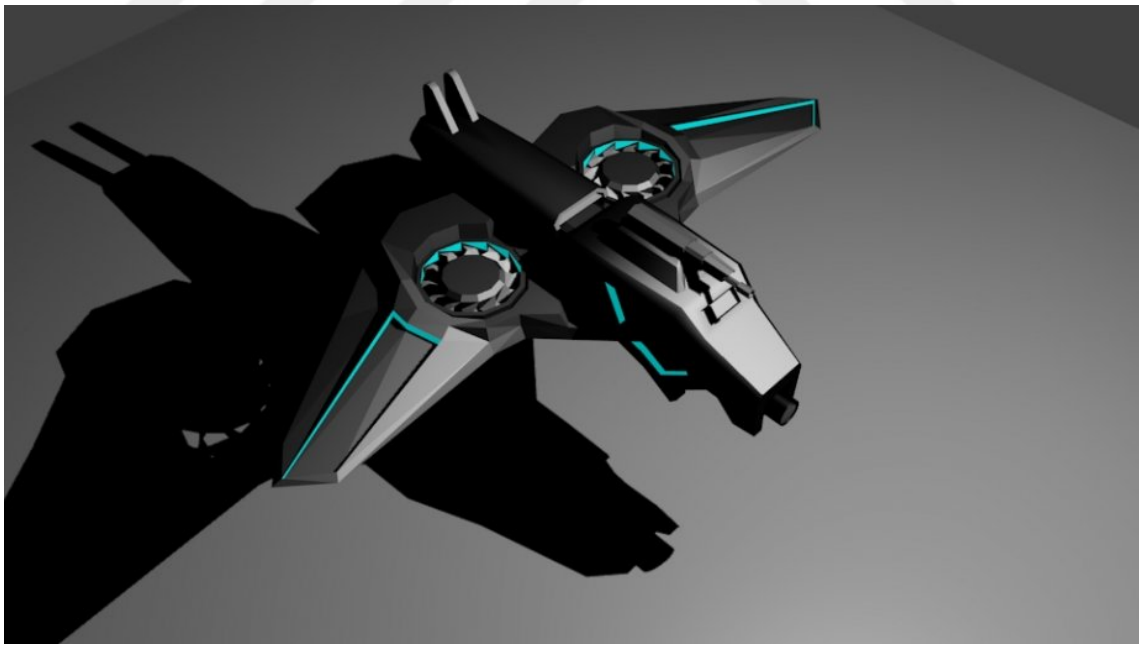


Figure 3.3 3D model of drone, *image from Free3D.com*

The UAV has vision and cognition layers as described in Figure 2.1. We simulated the probability of the target estimation and the label identification outputs of the vision layer by using the Gaussian distribution function (3.1);

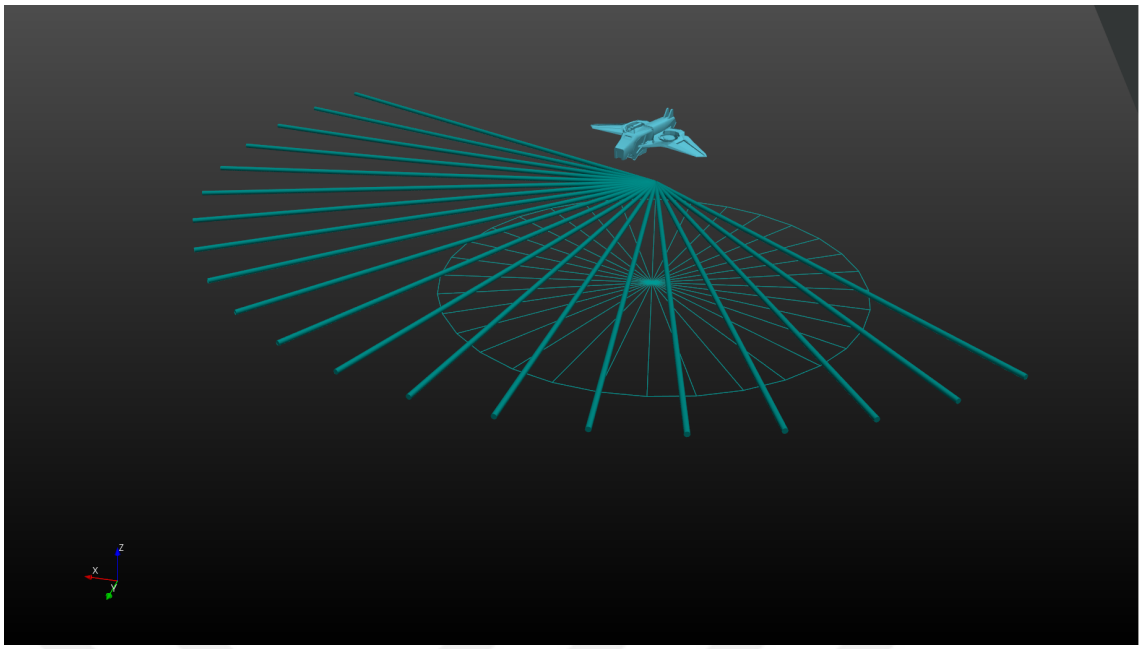


Figure 3.4 3D model of drone and on-board sensors rendered in simulation

$$P(x) = \frac{100}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \tag{3.1}$$

where μ is 0.7 m, x is the parametric distance between the UAV and the target, σ is 1 m. Further, to simulate real-world side effects such as a light density or blurry camera output, we applied a randomized success ratio by using Pythons random uniform function which returns a value between 0 and 100. Randomly generated side effect estimation is used to reduce the target detection precision. For example, assume that we calculate the Gaussian value of a position as 95% and we calculate the side effects chance for 5%. Further, assume that we have a 90% precision threshold to decide whether the target is detected or not. We calculate 20% side effect loss of 95% precision. Therefore, we end up with 90.25% target detection precision. Hence, we decide the target is detected and start investigating the target.

The simulation starts with the processing of the given map. The cognition layer gets TUM map as input and processes it with path planning algorithm as in Figure 3.5 a. Following, the UAV placed to a random location inside the available free space of the map. After placement, the UAV takes off and rises to 10 m altitude. The UAV starts to move to the first calculated cells start point and starts coverage of the map. During the cruise, it has a fixed 10 m/s velocity. The UAV moves in x-y coordinate in path tracking state at 10 m fixed altitude. The UAV changes its altitude if the flight state changes to target acquisition state or an emergency state.

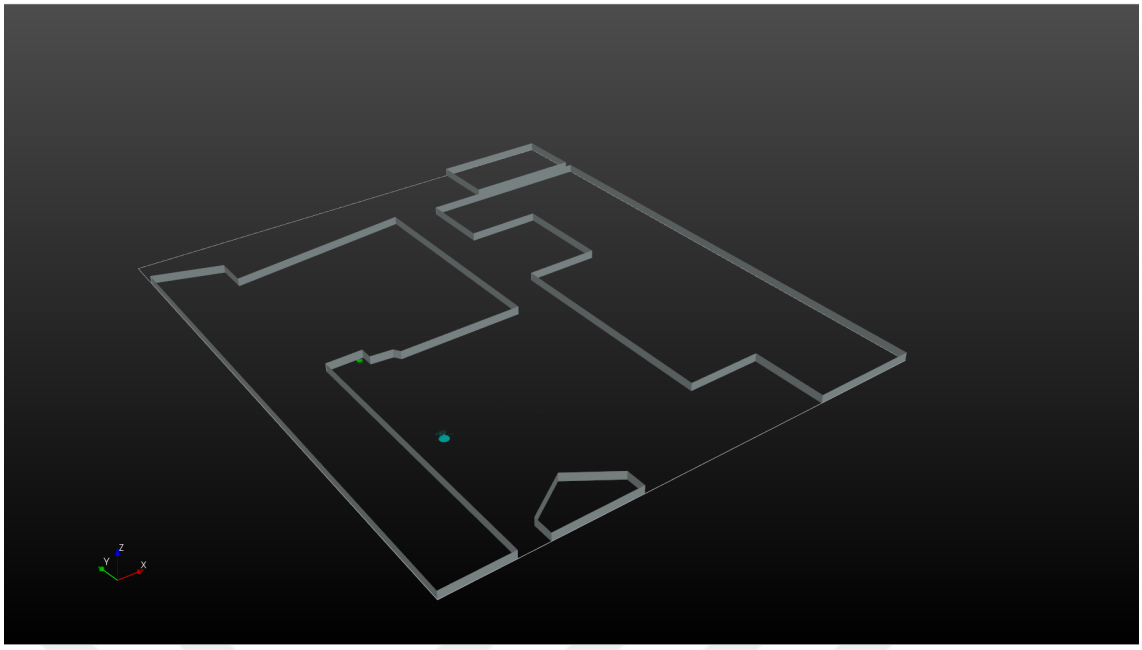


Figure 3.5 3D rendered map

In the target acquisition state, the cognition layer gets position estimation from the vision layer. Target position estimation has an exponential error rate depending on the distance to the target. According to the position of the bottle, UAV set its position to 0.7 m (optimal distance for bottle labeling) distance from the target bottle. To achieve this, the investigation module of the cognition layer controls the UAV in 3D space.

Following the UAV makes a circular movement until the vision layer outputs the label of the bottle. If the vision layer can't detect the label for one complete circle around the target bottle, the cognition layer stops the investigation of the bottle, outputs the message that the target label cannot be determined and returns to path tracking state. During the process, the list of detected target positions and labeled targets are stored in the memory. A success rate is calculated by comparing information of collected bottles and placed bottles. Although the success rate of the simulation has a direct connection with the estimations we make for target detection and labeling which might lead to unrealistic results, with the usage of parameters like the distance between UAV and the target and the speed of the UAV in probability calculation, we achieved a reliable and a robust output from the simulation.

While the UAV cruises and investigates the detected targets, the safety module checks battery percentage to calculate the flight time if the UAV can complete the mission successfully. Further, the safety module checks for potential collisions. As described in section 2.2.5, the safety module tracks potential collisions, which may be caused by obstacles. In our scenario, the obstacles are buildings, hence they are fixed. The drone knows potential collision zones and acts accordingly. To detect a collision, the

safety module uses collected data from LIDAR. LIDAR spread rays in forward direction of the UAV. Although the UAV is free to move in 360 degrees, the UAV uses LIDAR's output for 180 degrees to save computational power. The resolution of LIDAR rays is 8.18 degrees which means that 22 rays flow through LIDAR to the environment at 180 degrees angle facing forward.

One other task of the safety module is detecting a possible crash caused by a battery outage. The safety module calculates unit power consumption by average current usage which is 8000 mA. As stated earlier, the UAV has equipped with a 4500 mAh battery which 33 min 45 s flight time in complete usage. Yet, to protect LiPo battery, the safety module tracks the remaining percentage of the battery and keeps it higher than 30% at any given time. After exclusion of the battery protection percentage, the UAV is left with 23 min 37.5 s flight time. If the flight time is about to exceed the limit, then the safety module interrupts the mission and lands the UAV to the surface. Additionally, during the cruise, the safety module tracks the distance to the base point, and by using the maximum cruise velocity (10 m/s) it calculates the required power to return to the base. If the safety module detects the remaining power is about the same as the required power to return the base, the safety module declares an emergency and moves the UAV towards the base station.

4
RESULTS AND DISCUSSION

4.1 Results

In this section, we demonstrate our findings from the simulation runs. First, we share the results of the simple obstacle map simulation. Following, we share the results of the TUM map simulation.

As explained before, the proposed system is tested on two different maps. We ran the simulations of both maps for two different decomposition algorithms; trapezoidal decomposition and boustrophedon decomposition. We aimed to show the trade-offs between these algorithms. For each simulation run, we randomly placed ten randomly labeled bottles in the environment.

In simulations, the following data were recorded to measure the monitoring capability of the proposed system; detected target count, labeled target count, location precision, total travel distance, total mission execution time and total interruption due to emergencies.

As mentioned in 3.2, the evaluation of the successful completion of the monitoring mission is defined as collecting all bottle positions and their labels in the field with the highest power consumption efficiency. However, finding an optimal solution for a monitoring mission is not realistic due to the fact that the targets are distributed randomly and the system is expected to have real-time interactions with the environment. Thus, it's difficult to extrapolate an optimal path from the input map. Yet, assuring complete coverage of the field is a must in the monitoring scenarios we have. After considering these facts, we decided to take the following approach to judge the coverage efficiency of the field. We calculated the collision-free space of the maps. Then, we calculated the minimum travel distance required to achieve complete coverage of the field by dividing the total coverage area by the space that the UAV can cover in a second with a maximum cruise speed of 10 m/s. Despite the fact that UAV is not expected to cruise constantly during the monitoring but to

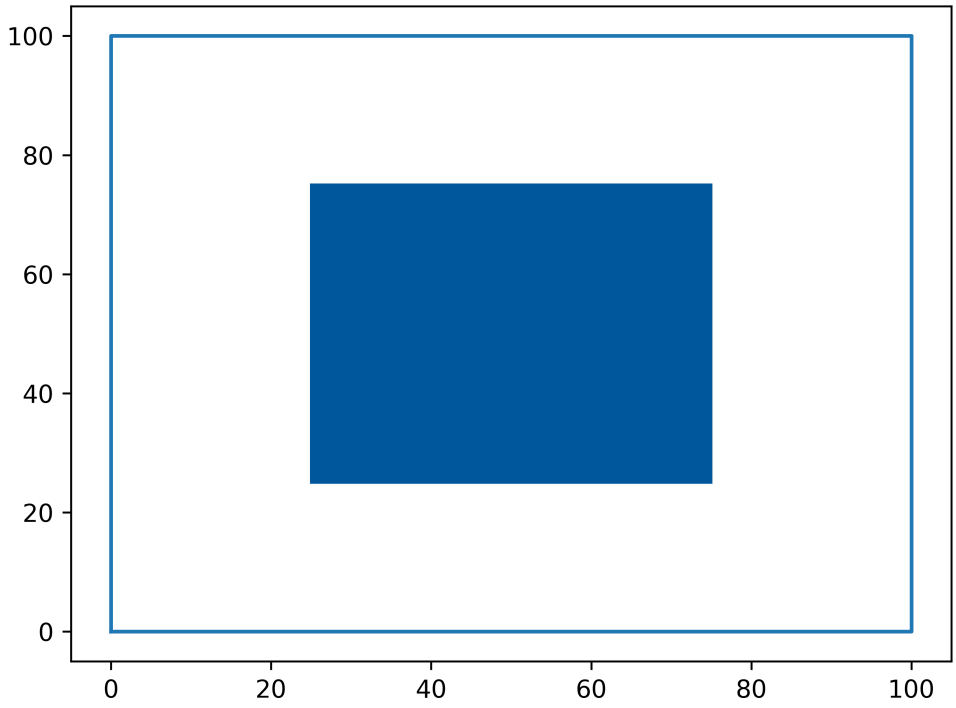


Figure 4.1 The simple obstacle map
Blue colored area is obstacle

investigate the detected targets, we accepted the minimum travel distance required to achieve complete coverage as the optimal solution. At the end of each simulation, we compared the minimum travel distance of the map with the total travel distance of the system during the mission. This comparison derived the efficiency of our system.

4.1.1 The Simulation of the Simple Map

We created an artificial map with a simple rectangular obstacle placed in the middle as presented in Figure 4.1. The reason we created the simple map to test different decomposition algorithms in a less complicated environment and provide a variety of sample data for comparison.

The simple map has 100 m width to 100 m length. The obstacle in the middle of the field is 50 m wide and 50 m long. We randomly placed 10 bottles in the collision-free space around the obstacle. The total coverage space of the map is 7500 m^2. The approximated minimum travel distance for complete coverage of the simple map was calculated as 1500 m from equation 2.3. The cruise speed of the UAV is 10 m/s. To travel 1500 m, the UAV needs to fly at least 150 s.

Simulations of the simple map were performed for two decomposition methods the boustrophedon decomposition and trapezoidal decomposition as described in section 2.4.1. For the simple map, the boustrophedon decomposition method calculated the same cells as trapezoidal decomposition. Hence, they are expected to produce similar results. For each method, ten simulations were performed.

Results of the simple obstacle map simulation for the trapezoidal decomposition method were gathered in table 4.1. Following averages derived from ten simulation executions: 84% target detection rate, 80% target labeling rate, 92.96% location precision, 1831 total travel distance and 7.34 min of mission execution time. These findings suggest that the UAV traveled 331 m more than the minimum travel distance for complete coverage of 1500 m. Thus, the UAV needs 22% more power for traveling compared to the minimum required power for the mission. The UAV spent 4.86 min more for the entire mission execution where the minimum time was 2.5 min. Hence, the UAV needs 193% more time compared to the minimum required time for the mission.

Table 4.1 Results of the Simple Obstacle Map Simulation for Trapezoidal Decomposition

Simulation Run	Detected Targets	Labeled Targets	Location Precision	Travel Distance	Mission Time	Total Interruption
1	8	8	92.4%	1808	7.1	0
2	8	8	93.4%	1821	7.37	0
3	9	7	92.6%	1903	7.9	0
4	6	5	93.8%	1783	7.11	0
5	9	9	92.3%	1813	7.27	0
6	7	7	92.9%	1788	6.99	0
7	9	8	93.7%	1823	7.35	0
8	8	8	93.4%	1853	7.88	0
9	10	10	92.7%	1855	7.18	0
10	10	10	92.4%	1863	7.31	0

Results of the simple obstacle map simulation for the boustrophedon decomposition method were gathered in table 4.2. Following averages derived from ten simulation executions: 85% target detection rate, 84% target labeling rate, 93.12% location precision, 1827.7 m total travel distance and 7.29 min of mission execution time. These findings suggest that the UAV traveled 327.7 m more than the minimum travel distance for complete coverage of 1500 m. Thus, the UAV needs 21.8% more power for traveling compared to the minimum required power for the mission. the UAV spent 4.79 min more for the entire mission execution where the minimum time was 2.5 min. Hence, the UAV needs 191% more time compared to the minimum required time for the mission.

Table 4.2 Results of the Simple Obstacle Map Simulation for Boustrophedon Decomposition

Simulation Run	Detected Targets	Labeled Targets	Location Precision	Travel Distance	Mission Time	Total Interruption
1	9	9	92.9%	1849	7.20	0
2	9	9	93.6%	1822	7.31	0
3	9	8	92.4%	1887	7.4	0
4	7	7	93.5%	1771	7.21	0
5	10	10	92.7%	1863	7.32	0
6	8	8	92.4%	1794	7.03	0
7	8	8	93.1%	1812	7.16	0
8	10	10	92.9%	1893	7.82	0
9	9	9	93.9%	1837	7.61	0
10	6	6	93.8%	1749	6.91	0

The results of both methods don't suggest conspicuous differences. Hence, the results prove that usage of the boustrophedon decomposition method for simple maps is not efficient, since it demands more computational power compared to the trapezoidal decomposition. Further, the results prove that the proposed system has a high success rate for the identification and labeling of targets and their positions.

4.1.2 The Simulation of the TUM Map

TUM main campus as in Figure 3.1 were used as the second environment for simulations. The TUM campus map has 575 m width and 678 m length. The map contains four obstacles that represent buildings of the TUM campus. The first obstacle is on the western part of the campus, and it has 104995 m^2 surface. The second obstacle is on the south part of the campus, and it has 3536 m^2 surface. The third obstacle is on the east part of the campus, and it has 107632 m^2 surface. The fourth obstacle is on the north part of campus and it has 11664 m^2 surface. The obstacles blocks access to 227827 m^2 space of the map. Thus, the available coverage space of the TUM map is 162023 m^2. In simulations, ten beer bottles were randomly placed to free space of the TUM campus.

The approximated minimum travel distance for complete coverage of the TUM campus map was calculated as 32404.6 m from equation 2.3. The cruise speed of the UAV is 10 m/s. To travel 32404.6 m, the UAV needs to fly at least 3241 s (54.01 min).

As simulations of the simple map, simulations of the TUM campus map were performed for two decomposition methods, the boustrophedon decomposition and the trapezoidal decomposition as described in section 2.4.1. For the TUM campus

map, the boustrophedon decomposition method calculated nine cells whereas the trapezoidal decomposition calculated fourteen cells. Hence, the simulation results of the two methods are expected to show conspicuous differences. For each method, fifteen simulations were performed.

Results of the TUM campus map simulation for the trapezoidal decomposition method were gathered in table 4.3. Following averages derived from the fifteen simulation executions: 92.66% target detection rate, 88.66% target labeling rate, 93.17% location precision rate, 36266.6 m total travel distance, 80.02 min of mission execution time and three emergency interruptions. These findings suggest that the UAV traveled 3862 m more than the minimum travel distance for complete coverage of 32404.6 m. Thus, the UAV needs 11.9% more power for traveling compared to the minimum required power for the mission. The UAV spent 26.01 min more for the entire mission execution where the minimum time was 54.01 min. Hence, the UAV needs 48.15% more time compared to the minimum required time for the mission. Further, the UAV faced three emergency interruptions due to a battery outage. For each emergency interruption, the UAV returned to the base station and recharged its battery to continue the mission. The complete recharge of the battery takes approximately four hours. Hence, three times the complete recharge of the battery caused twelve hours delay for the completion of the mission.

Table 4.3 Results of TUM Campus Map Simulation for Trapezoidal Decomposition

Simulation Run	Detected Targets	Labeled Targets	Location Precision	Travel Distance	Mission Time	Total Interruption
1	10	9	93.5%	36608	82.79	3
2	8	8	92.4%	36253	78.84	3
3	10	9	92.7%	36547	80.33	3
4	8	7	92.5%	36276	80.82	3
5	10	10	94.0%	36353	79.67	3
6	10	10	93.5%	36304	82.76	3
7	9	9	93.7%	36126	79.61	3
8	10	10	93.8%	36226	79.71	3
9	10	10	92.7%	36280	78.78	3
10	9	8	93.7%	36380	80.20	3
11	10	10	92.7%	36289	78.75	3
12	9	9	92.1%	35932	79.29	3
13	8	8	92.8%	35697	78.89	3
14	9	9	93.5%	36493	79.69	3
15	9	7	93.9%	36235	80.27	3

Results of the TUM campus map simulation for the boustrophedon decomposition method were gathered in table 4.4. Following averages derived from fifteen simulation

executions: 86.66% target detection rate, 82% target labeling rate, 93.08% location precision rate, 35901.7 m total travel distance, 79.69 min of mission execution time and three emergency interruptions. These findings suggest that the UAV traveled 3497.1 m more compared to the minimum travel distance for complete coverage of 32404.6 m. Thus, the UAV needs 10.79% more power for traveling compared to the minimum required power for the mission. The UAV spent 25.68 min more for the entire mission execution where the minimum time was 54.01 min. Hence, the UAV needs 47.54% more time compared to the minimum required time for the mission. Further, the UAV faced three emergency interruptions due to a battery outage. As explained before, three times the complete recharge of the battery caused twelve hours delay for the completion of the mission.

Table 4.4 Results of TUM Campus Map Simulation for Boustrophedon Decomposition

Simulation Run	Detected Targets	Labeled Targets	Location Precision	Travel Distance	Mission Time	Total Interruption
1	10	10	92.8%	35974	80.03	3
2	10	9	94.0%	35571	82.58	3
3	9	9	93.3%	36128	79.36	3
4	7	7	92.5%	35910	78.58	3
5	9	8	92.3%	35483	79.46	3
6	9	9	92.6%	36081	79.51	3
7	10	10	93.0%	35896	79.81	3
8	8	8	92.4%	36143	79.73	3
9	9	9	93.8%	35724	79.00	3
10	6	6	92.5%	35502	78.67	3
11	9	9	93.9%	35859	80.08	3
12	10	8	94.0%	36150	80.61	3
13	9	9	93.6%	36127	79.14	3
14	6	5	92.2%	35743	78.66	3
15	9	7	93.3%	36235	80.27	3

Results in tables 4.3 and 4.4 prove that the boustrophedon decomposition method improves the efficiency of the system by decreasing total energy consumption and total mission time compared to the trapezoidal decomposition method in the TUM campus map. The minimum travel distance of the UAV was reduced by 364.9 m and the mission execution time dropped by 0.33 min. However, the target detection rate dropped 6% and the target labeling rate dropped 6.6% for the boustrophedon method compared to the trapezoidal method.

The target detection and the target labeling are modeled with the probabilistic functions explained in 3.3. The target labeling is directly affected by target detection because the labeling procedure starts only after the detection of a target. These facts

corroborate that an increase in the mission time and the total travel distance improves the rate of target detection and target labeling. Hence, the performance drawback of the target detection and the target labeling for the boustrophedon decomposition method was mainly caused due to fact that the UAV traveled less compared to the trapezoidal decomposition method.

Simulation results of the TUM campus map prove that the proposed system is capable of executing the monitoring task with a high precision and success rate in large fields. Yet, they show that the proposed system faces delays due to its battery size.

4.2 Discussion

Environmental monitoring is a thriving process and activity that gives insights into the environmental impact. Environmental monitoring tasks are characterized by continuous intensive observation challenges in wild environments. They require a highly educated observer with sharp focusing skills. Hence, environmental monitoring tasks are expensive to execute and maintain. Solving such a problem with autonomy can decrease costs and boost efficiency and safety.

In this study, we introduced an autonomous monitoring system that can plan and execute environmental monitoring tasks. We designed a decision-making framework that can conduct a UAV in a variety of monitoring missions without being affected by the target type. Our system standardizes flight safety at the core of the decision-making framework. Thus, the proposed system has a flexible framework that increases the safety and the maintainability of the monitoring missions compared to earlier implementations of monitoring systems. Further, we introduced an improved decomposition algorithm to add the capability to plan routes in environments with complex obstacles. Hence, the proposed system increases efficiency and reduces costs while regularizing security in monitoring tasks.

In this study, we used a multirotor UAV with a 4500 mAh battery to execute the given monitoring tasks in section 3.2. The UAV was expected to have a 10 m/s maximum velocity. The system was expected to use a maximum of 70% of UAVs battery to protect the battery from potential harm. The optimal target detection range was given as 2.5 m. the UAV was expected to cover the field 14.81 m^2 for every 1 m movement. Further, the UAV was expected to cover 50 m^2 for each second of flight.

The simulations were performed to test the performance of two decomposition methods (boustrophedon decomposition and trapezoidal decomposition) in two different environments (the simple obstacle map and the TUM campus map). The

results of the simulations are presented in tables 4.1, 4.2, 4.3 and 4.4.

The results of the simple obstacle map simulations showed that the system collected truthful information for 8.2 bottles of every 10 bottles. Further, the results provided that the system traveled 21.9% more distance compared to the minimum travel distance required for the complete coverage of the simple map.

The results of the TUM campus map simulations showed that the system collected truthful information for 8.5 bottles of every 10 bottles. Further, results provided that system traveled 11.3% more distance compared to the minimum travel distance required for the complete coverage of the simple map.

The simulation results of both environments proved that the proposed system has a robust monitoring capability while ensuring safety standards. However, the proposed system performed 78.1% efficiency in the simple obstacle map simulations while it performed 88.7% efficiency in the TUM campus map simulations. We suspect that the main reason behind the efficiency loss is the time and power loss during the investigation of targets. In each environment, ten targets were placed on the field. After the detection of a target, the UAV reduces its speed and investigates the detected target for detailed information which is causing time loss and additional power usage. Therefore, in a smaller map with the same number of targets, it's expected that the system will perform with reduced efficiency. Another reason that affects the efficiency is the transition movements between cells. The transitions between cells sometimes cause unwanted resource usage because it forces the UAV to move over a part of the map that is already covered.

In the TUM campus map simulation, due to the large size of the field, the UAV returned to the base station to recharge itself three times. The approximate four hours of recharge time of the battery causes twelve hours delay for the completion of the mission. These findings show that the proposed system needs a better capacity for flight time to minimize interruptions. Therefore, the multirotor UAVs are inefficient for environmental monitoring tasks in large fields, due to the battery capacity of the multirotor UAVs.

This study used two offline path planning algorithms to calculate the coverage path. The results showed that both of the calculated routes are suitable for complete coverage. However, for complex maps, the boustrophedon decomposition method improves the efficiency of the system due to the reduced cell number. The efficiency gain increases with the introduction of more complicated obstacles in the environment.

We suggested an online decision-making system that can orchestrate the tasks of a monitoring mission. Our system is capable of detecting targets, investigating targets and predicting and resolving emergencies. We can achieve better results with our system by improving the robustness of the decision-making and by improving the autonomy of flight.

4.2.1 Through Robust Decision-making

To conduct environmental monitoring tasks that include a variety of scenarios where a rich set of decisions with challenging complexity is expected to be made, the system needs to reduce uncertainty to improve the robustness of the decision-making. To reduce the uncertainty, data collection and characterization of collected data must be appropriate.

As we presented in this thesis, the proposed autonomous system can conduct environmental monitoring tasks. Our approach was focused on identifying the decision requirements of autonomy and designing robust decision-making. We believe there are three ways to improve our decision-making architecture: using improved data collection methods, improving data analysis and updating the decision process with experiences.

Improving data collection In our approach, observation of the environment happens in two processes. The vision layer collects the image sequences of the environment and then it extracts reliable information related to the monitored target. The success of the target detection is greatly depended on the quality of the captured image. There may be a condition that leads to distortion of the imagery such as low quality of the ambient light. Due to these conditions, it may be impossible to gather data. Therefore, there may be additional supportive observations to identify the surroundings. LIDAR data can be used to identify certain shapes which can help to improve target prediction precision. Another observation process is done by the cognition layer to identify obstacles by using LIDAR. But our approach simplifies the obstacle detection with the distance calculation. Our approach doesn't extract any further meaningful information to identify the obstacles. Our obstacle-avoidance approach can be improved by detecting the shapes of obstacles and taking particular obstacle-avoiding actions depending on the nature of the obstacle.

Improving data analysis Data analysis is a critical factor that supports robust decision-making by providing the reliable meaning of the data for state comparison. Our approach to switching between the target tracking and the target investigation tasks depends on the target estimation precision of the vision layer. Yet, to raise

the usage efficiency of power resources, the system may determine to ignore certain targets. Investigating each detected potential target reduces the efficiency of the system and may lead to interruptions of the mission. In consideration of the environmental monitoring tasks mostly taking place in the wild areas, it's hard to reach monitoring fields. Furthermore, if an autonomous vehicle aborts its mission in the middle of the field because of a drained battery, human intervention is required to collect the vehicle from the location it stands. The system can decide to investigate certain detected targets and manage the trade-off between resources and mission completion.

Consideration of previous experiences The basis of our decision-making comes from a complete analysis of a monitoring task. By considering the results of recent events, the system can adapt decision logic to the present state and conditions. The system can identify certain events and decide on a different policy. Furthermore, it can connect events and results, e.g., it can determine to do further investigation on a certain part of the field in particular conditions that may lead to emergencies.

4.2.2 Autonomy of Flight

Autonomous flying depends on real-time localization. During monitoring task executions, the quality of the position estimation critically influences the ability to make required motions for path tracking. In our monitoring scenarios, the UAV is expected to conduct flights autonomously during the complete coverage of the field, during the approach to the target and the during the safety reactions.

In the proposed approach, we presented a static motion model for autonomous flight. We believe there are four ways to improve the autonomy of flight: improving path planning, using independent localization, improving investigation motions, and improving emergency reactions.

Improving path planning To be able to achieve complete coverage of the field, the system should calculate a path that ensures the certainty of the investigation of every bit of the field. In the proposed approach, we used a static path planning method. Our approach forbids calculating dynamic obstacles in the field and requires prior knowledge of the environment. Path planning can be adaptive to dynamic obstacles by using AI approaches such as reinforcement learning.

Independent localization Localization may be the most crucial factor for an autonomous vehicle. Data for the localization should be precise enough to avoid losing the track of the mission and the potential disappearance of the vehicle. In the

proposed approach, we used GPS data as the main data source of the positioning. Yet, GPS is not a precise source when there is an interference between the receiver and the satellites. Thus, GPS brings interference dependency for the system. We believe that localization can be improved by processing real-time data. Approaches like SLAM can support localization based on the surroundings. Yet, approaches like SLAM require high computational power due to intense image processing. These approaches may be used as a backup for GPS data where GPS data is lost.

Improving investigation motions After the detection of the target, the system tries to gather more data about the detected target to be able to identify it. In the proposed approach, we used a policy-based motion depending on the state to handle the investigation of the target. In the proposed approach, during the investigation, the UAV gets to the optimal identification distance, then changes the angle of view related to the target by moving in a circle around the target while staying at the same altitude. These policy based motions can be improved in a way to adapt motions to the target position and condition, e.g., making 3-axis rotations.

Improving emergency reactions Due to the regulations of autonomous flying and the cost of autonomous vehicles, the safety of the vehicle and the environment is a necessity of autonomous flying. As described in earlier chapters 1, 2 and 3, emergencies can be caused by outside factors as well as mechanical issues of the UAV. Predicting emergencies is an optimal way to avoid potential harm. Yet, it's hard to define the merits to detect potential emergencies. In the proposed approach, we predict a potential collision depending on the static obstacle data of the map and the LIDAR data analysis. For other emergencies, the system tracks several sensors and uses a set of safety rules to detect the emergencies. One way to improve this approach is increasing the number of defined emergencies. Furthermore, the UAV can employ a dynamic prediction of potential emergencies by comparing the real-time conditions with the recent emergency conditions.

REFERENCES

[1] R. N. De Carvalho, H. Vidal, P. Vieira, M. Ribeiro, "Complete coverage path planning and guidance for cleaning robots," in *Industrial Electronics, 1997. ISIE'97., Proceedings of the IEEE International Symposium on*, IEEE, vol. 2, 1997, pp. 677–682.

[2] S. Grzonka, G. Grisetti, W. Burgard, "Towards a navigation system for autonomous indoor flying," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, IEEE, 2009, pp. 2878–2883.

[3] I. F. Spellerberg, *Monitoring ecological change*. Cambridge University Press, 2005.

[4] J. Awange, *GNSS Environmental Sensing*. Springer, 2017.

[5] M. J. McCarthy, K. E. Colna, M. M. El-Mezayen, A. E. Laureano-Rosario, P. Méndez-Lázaro, D. B. Otis, G. Toro-Farmer, M. Vega-Rodriguez, F. E. Muller-Karger, "Satellite remote sensing for coastal management: A review of successful applications," *Environmental management*, vol. 60, no. 2, pp. 323–339, 2017.

[6] B. Bayat, N. Crasta, A. Crespi, A. M. Pascoal, A. Ijspeert, "Environmental monitoring using autonomous vehicles: A survey of recent searching techniques," *Current opinion in biotechnology*, vol. 45, pp. 76–84, 2017.

[7] M. Rivas Casado, R. Ballesteros Gonzalez, R. Wright, P. Bellamy, "Quantifying the effect of aerial imagery resolution in automated hydromorphological river characterisation," *Remote Sensing*, vol. 8, no. 8, p. 650, 2016.

[8] S. Rathinam, P. Almeida, Z. Kim, S. Jackson, A. Tinka, W. Grossman, R. Sengupta, "Autonomous searching and tracking of a river using an uav," in *American Control Conference, 2007. ACC'07*, IEEE, 2007, pp. 359–364.

[9] S. X. Yang, C. Luo, "A neural network approach to complete coverage path planning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 1, pp. 718–724, 2004.

[10] S. Hert, S. Tiwari, V. Lumelsky, "A terrain-covering algorithm for an auv," in *Underwater Robots*, Springer, 1996, pp. 17–45.

[11] P. N. Atkar, H. Choset, A. A. Rizzi, E. U. Acar, "Exact cellular decomposition of closed orientable surfaces embedded in/spl rfr//sup 3," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, IEEE, vol. 1, 2001, pp. 699–704.

[12] Z. L. Cao, Y. Huang, E. L. Hall, "Region filling operations with random obstacle avoidance for mobile robots," *Journal of Robotic systems*, vol. 5, no. 2, pp. 87–102, 1988.

[13] M. Ollis, A. Stentz, "First results in vision-based crop line tracking," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, IEEE, vol. 1, 1996, pp. 951–956.

[14] H. Choset, "Coverage of known spaces: The boustrophedon cellular decomposition," *Autonomous Robots*, vol. 9, no. 3, pp. 247–253, 2000.

[15] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. L. Grixa, F. Ruess, M. Suppa, D. Burschka, "Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue," *IEEE robotics & automation magazine*, vol. 19, no. 3, pp. 46–56, 2012.

[16] S. Y. Lang, B.-Y. Chee, "Coordination of behaviours for mobile robot floor cleaning," in *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, IEEE, vol. 2, 1998, pp. 1236–1241.

[17] P. Vadakkepat, K. C. Tan, W. Ming-Liang, "Evolutionary artificial potential fields and their application in real time robot path planning," in *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, IEEE, vol. 1, 2000, pp. 256–263.

[18] N. Tran, D.-T. Nguyen, D.-L. Vu, N.-V. Truong, "Global path planning for autonomous robots using modified visibility-graph," in *2013 International Conference on Control, Automation and Information Sciences (ICCAIS)*, IEEE, 2013, pp. 317–321.

[19] P. Bhattacharya, M. L. Gavrilova, "Voronoi diagram in optimal path planning," in *4th International Symposium on Voronoi Diagrams in Science and Engineering (ISVD 2007)*, IEEE, 2007, pp. 38–47.

[20] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," 1998.

[21] Y. Shi, Q. Li, S. Bu, J. Yang, L. Zhu, "Research on intelligent vehicle path planning based on rapidly-exploring random tree," *Mathematical Problems in Engineering*, vol. 2020, 2020.

[22] J. Barraquand, J.-C. Latombe, "Robot motion planning: A distributed representation approach," *The International Journal of Robotics Research*, vol. 10, no. 6, pp. 628–649, 1991.

[23] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, D. Hull, "Morse decompositions for coverage tasks," *The international journal of robotics research*, vol. 21, no. 4, pp. 331–344, 2002.

[24] M. Quaritsch, R. Kuschnig, H. Hellwagner, B. Rinner, A. Adria, U. Klagenfurt, "Fast aerial image acquisition and mosaicking for emergency response operations by collaborative uavs," in *Proceedings for the International ISCRAM Conference*, 2011, pp. 1–5.

[25] L. Euler, "Solutio problematis ad geometriam situs pertinensis," *Comm. Acad. Sci. Imper. Petropol.*, vol. 8, pp. 128–140, 1736.

[26] S. J. Russell, P. Norvig, *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited, 2016.

[27] H. Choset, P. Pignon, "Coverage path planning: The boustrophedon cellular decomposition," in *Field and service robotics*, Springer, 1998, pp. 203–209.

[28] P. Marion, *Director: A robotics interface and visualization framework*, 2015. [Online]. Available: `http://github.com/RobotLocomotion/director`.

PUBLICATIONS FROM THE THESIS

Papers

1. Ö. Yildirim, R.A. Vural, K. Diepold, “Improving coverage method of autonomous drones for environmental monitoring,” Turkish Journal of Electrical Engineering and Computer Sciences, vol. 28, no. 6, pp. 3352-3367, 2020.

Conference Papers

1. Ö. Yildirim, K. Diepold, R.A. Vural, "Decision Process of Autonomous Drones for Environmental Monitoring," in 2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA), 2019, IEEE, p. 1-6.

Awards

1. Best Paper Award for the paper entitled Decision Process of Autonomous Drones for Environmental Monitoring. IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA) held in Sofia, Bulgaria from July 3-5, 2019.