Eötvös Loránd University

Faculty of Informatics

Erasmus Mundus Joint Master in

Intelligent Field Robotic Systems

Drones for autonomous descent in sewer manholes

Author:

Rahaf Haitham Omar Abu Hara

Intelligent Field Robotics MSc

Internal supervisor:

Zoltán Istenes

Associate Professor, ELTE

External supervisor:

Julián Cayero Becerra

Robotics and Automation Engineer

Budapest, 2023

Contents

Abstract

This Master's Thesis introduces an innovative two-phase solution for autonomous drone navigation in sewer manhole inspections, addressing the significant challenges presented by the complex and unknown environments of sewer systems. The approach combines autonomous drone flight, computer vision techniques, and path planning algorithms, with rigorous testing conducted in both simulated and real-world environments to ensure reliability. The research includes a novel application of the Arc Adjacency Matrix based Fast Ellipse Detection (AAMED) algorithm for sewer manhole detection and develops a unique algorithm for drone descent based on depth image data. By enhancing safety and efficiency in sewer inspections and potentially influencing similar applications requiring advanced navigational capabilities, this study significantly contributes to drone operations in constrained vertical environments and the management of essential urban infrastructures.

Keywords— Autonomous Drones, Sewer Inspections, Manhole Detection, Navigation, Controlled Descent, Localization, Simulation and Real-world Transferability, AAMED Algorithm, Back Projection Algorithm, Time-of-Flight (ToF) Camera, Visual Odometry, Sewer Systems, Infrastructure Management, Urban Health and Sanitation, Hazardous Environment Navigation, Drone Technology, Robust Systems, Infrastructure Maintenance, Image Detection, Sewer Manhole Localization, Sewer System Management, Utility Industry

Acknowledgements

I would like to express my deepest appreciation to everyone who provided me the possibility to complete this thesis.

A special gratitude I give to my internal supervisor, Associate Professor Zoltán Istenes of ELTE, whose contributions in stimulating suggestions and encouragement, helped me to coordinate my project and bring it to a successful completion.

I am heartily thankful to my external supervisor, Robotics and Automation Engineer Julián Cayero Becerra, whose patience, invaluable guidance, and immense knowledge guided me throughout this journey. I have been amazingly fortunate to have a supervisor who cared so much about my work and who responded to my queries so promptly.

I also wish to extend my thanks to my colleague, Andres Palomino Suescun, who provided insightful feedback and advice during the internship that formed a critical part of this research. Your help was much appreciated and will not be forgotten.

For my life partner, Hesham Fadl, words may fall short to express my gratitude. You have been my steadfast support in this journey, always encouraging me to reach for my goals, believing in me even when I doubted myself. Your constant love, motivation, and unwavering belief in my potential have been the foundation upon which this achievement was built. This work is as much yours as it is mine, for your support has been my strength and your faith my courage. I am truly grateful to have you by my side.

Finally, I would like to express my love and gratitude to my family; for their understanding, endless patience, and encouragement when it was most required. You have been my biggest cheerleaders, and this achievement would not have been possible without you.

Thank you, everyone, for your invaluable contribution to this work.

Chapter 1

Introduction

In the accelerating pace of our technological era, unmanned aerial vehicles (UAVs), commonly known as drones, have firmly established themselves as versatile instruments across a multitude of industries, evidencing a steady transition toward autonomous systems. The remarkable surge in their deployment, reflective of their vast potential in a variety of applications, has recently extended into the realm of sewer inspection. This thesis addresses an enduring challenge within urban infrastructure management – the inspection of sewer systems – proposing a groundbreaking solution. It explores how drones can revolutionize these inspections, enhancing safety and efficiency. Sewer systems present a distinctive environment teeming with unique navigational challenges, which the current state of drone technology is not yet fully equipped to surmount. Our work aspires to tackle these issues, presenting a two-phase approach to navigate drones through unknown environments, with a specific focus on sewer systems.

1.1 Background and context

Sewer systems, as vital infrastructural networks, transport and treat wastewater, crucial for urban health and sanitation. Due to aging, environmental stress, and chemical corrosion, they require regular scrutiny to prevent deterioration.

Traditionally, inspections were manual, posing safety risks to workers, while also being labor-intensive and time-consuming. Consequently, there's a pressing need for a safer, efficient alternative, which has come in the form of drones.

Drones can access hard-to-reach areas, reducing human risk and expediting inspections. With the capability to capture high-resolution data for further analysis, they offer significant advantages.

However, operating drones within sewer systems is fraught with challenges due to the confined, complex environments they need to navigate. Variable lighting, water presence, debris, and intricate sewer geometry further complicate the situation. Even with advancements in drone technology, reliable, autonomous sewer navigation remains a substantial challenge.

Given the widespread use of drones in various sectors and the urgent need to address sewer system management, the timeliness and relevance of this research are evident. By developing an efficient, safe solution for sewer inspections, this research aims to revolutionize the management and maintenance of essential urban infrastructures.

1.2 Problem statement

Sewer management, a crucial sector within the utility industry, continues to grapple with ongoing challenges due to the infrastructure's complexity and the hazardous nature of the environment. A promising solution lies in the use of drones for sewer inspections, but safe and effective navigation of these drones within sewer systems remains a formidable challenge.

This research focuses on tackling specific challenges intrinsic to this task. The central issues addressed include platform limitations, manhole detection, precise navigation above the manhole, controlled descent, sensor limitations, and ensuring sufficient battery life for mission completion.

It's important to note that these issues form part of a broader set of challenges faced by drones inspecting sewer manholes. The remaining challenges - notably localization, communication, environmental factors, and manoeuvring through the intricate sewer system - fall outside the scope of this thesis but are equally significant for achieving a comprehensive autonomous drone system for sewer inspections.

1.3 Research questions

This research's central question is to determine how we could develop and implement an autonomous solution capable of effectively executing a complete sewer inspection operation with a real drone. This question encompasses critical elements such as manhole detection, navigation, controlled descent, and localization, considering the transferability of simulated operations to real-world scenarios. To address this overarching question, we investigate the following specific questions:

1. Manhole Detection and Navigation: What is the most suitable algorithm for detecting manholes in the variable environments of sewers, and how can image detection be translated into real-world drone navigation commands?

2. Controlled Descent and Localization: Which algorithm can ensure a safe, controlled descent of the drone given the onboard sensors, and how can we mitigate typical errors such as localization noise and drift?

3. Simulation and Real-world Transferability: How can we accurately simulate the mission to guarantee that tested procedures are easily transferable to the real robot, especially when considering variations in tunnel geometry and camera position?

The following section outlines the objectives and scope required to address these research questions.

1.4 Objectives and scope

The purpose of this thesis is to tackle the challenges outlined in the Problem Statement, creating an innovative, real-time solution to sewer manhole inspection using drones. The crux of this study lies in the design of an autonomous navigation system for drones, engineered to efficiently navigate the complex task of detecting manholes and ensuring a safe descent in the demanding environments of sewers, considering practical constraints such as drone weight and maneuverability.

Our approach to the problem is divided into two distinct phases. The first phase concentrates on take-off, precise detection of the sewer manhole, and accurate positioning of the drone above the manhole. The second phase is focused on the controlled descent into the manhole, facilitated by a single downward-looking Time-of-Flight (ToF) camera. A significant part of this research is dedicated to the development of a mechanism that ensures a seamless transition between these two crucial phases.

The intent is to validate the proposed system through comprehensive simulations and real-world tests. The overarching goal of this research transcends the provision of an effective solution for sewer inspections. It aims to also potentially influence other applications that demand similar navigational capabilities. By succeeding in this endeavor, we aspire to make a significant contribution to drone operations in constrained vertical environments, thereby enhancing the safety and efficiency of such tasks.

Having outlined the objectives and scope of this thesis, the following section details the methodological approach designed to achieve these goals.

1.5 Methodology and approach

Our research strategy is structured into two distinct phases as outlined in the objectives and scope section, each with specific tasks to address the challenges articulated in the problem statement. The approach begins by setting up a simulation environment in the Gazebo 3D robotic simulator for preparations to tackle these tasks. An innovative integration of computer vision, reactive path planning, and robotic navigation is required for the tasks. Our methodology diverges from conventional practices that often resort to cutting-edge machine learning techniques.

These techniques often lead to extended operation periods, taxing the battery limitations of the drones [1], [2], [3]. Furthermore, they require extensive datasets, which are often not available for our specific tasks. In contrast, we opt to employ computer vision methods that leverage OpenCV functions to resolve the challenges at hand. This approach offers a balance of efficiency and practicality for our application.

Following the simulation phase, the developed solutions are transferred to a real-world drone and tested in a real-life environment. This step ensures that the proposed solutions are practically applicable, thereby underscoring the real-world transferability of our approach.

Our research, therefore, is not anchored on a mere quest for a more economically savvy solution. Instead, it is geared towards executing complex tasks, such as sewer manhole inspection, effectively and accurately given the drone's inherent constraints. By choosing resources best suited to our specific application, like ellipse detection algorithm in phase one and reactive path planing algorithm based on a single ToF camera system in the second phase, we target an optimal balance between cost, effectiveness, and accuracy.

Choosing this path does present its set of challenges — particularly doubts over applicability and reliability under varied real-world conditions. However, it also accentuates the potential of strategies that are precisely tuned to the task and resources at hand in drone navigation and inspection assignments. By aiming for a more achievable solution within these constraints, the success of this research could precipitate a reassessment of current standards and carve a path for similar methodologies in comparable applications.

This comprehensive approach seeks to balance effectiveness, efficiency, and real-world applicability, addressing the specific challenges of drone sewer inspection in a novel way. As the research progresses, we anticipate refining this methodology based on the emerging data and experiences.

1.6 Contribution of the thesis

The contributions of this thesis to the field of sewer manhole detection are substantial and multi-faceted.

For the first time, this research introduces the application of the Arc Adjacency Matrix based Fast Ellipse Detection (AAMED) algorithm in sewer manhole detection. This innovative use of the AAMED algorithm, which has been previously successful in fields such as satellite tracking and Unmanned Ground Vehicle (UGV) guidance, paves the way for a significant enhancement in the speed and accuracy of sewer detection methods. By successfully integrating this algorithm into the field of sewer inspection, this work not only enhances the precision and efficiency of sewer manhole detection but also contributes to improved management and maintenance of sewer systems.

Furthermore, this research goes beyond the conventional use of the AAMED algorithm by adding a filtration layer to optimize its performance specifically in the intricate context of sewer manholes. Utilizing area-based metrics, this additional filtration layer fine-tunes the detection process by selectively retaining only those ellipses that match the typical dimensions of a manhole. This modification substantially reduces the incidence of irrelevant detections and boosts the reliability and precision of the AAMED algorithm in this novel application.

Building upon the substantial advancements in sewer manhole detection, this research further contributes by creating and implementing a novel algorithm designed specifically for drone fully autonomous descent within manholes. A task that primarily relies on depth image data, this particular use of drones is groundbreaking, as indicated by a comprehensive literature review revealing minimal exploration in this specific area.

The development and successful implementation of such an algorithm signify a vital stride in leveraging drone technology for intricate navigation tasks. This unique contribution has practical implications not only for sewer inspection and management, but also for any other field that requires autonomous navigation in constrained and complex environments.

1.7 Overview of the thesis structure

This thesis is organized into seven significant chapters, each addressing a different aspect of the research:

1- The "Introduction" chapter sets the stage by outlining the context, problem statement, research questions, objectives, and scope of the research. It also highlights the methodology and the unique contributions of this study.

2- The "System Specifications" chapter delves into the technical and functional requirements of the drone system for sewer inspection, detailing aspects like hardware and software selection, and sensor configurations.

3- The "Literature Review" chapter provides an in-depth analysis of the current state of drone technologies, sewer inspection methodologies, and relevant algorithmic solutions, emphasizing the gaps this research intends to address.

4- The "Methodology" chapter provides a detailed step-by-step description of the ellipse detection algorithms, the process of developing a drone-controlled descent algorithm, and the setup of the simulation and real-world environments for testing and evaluation.

5- In the "Results" chapter, we present and dissect the outcomes of the simulated and real-world tests of the proposed system, focusing on its performance and robustness across varying sewer environments.

6- Lastly, the "Conclusion" chapter encapsulates the entirety of the thesis, providing a summary of the key findings, acknowledging the study's contributions, and suggesting potential directions for future research.

Chapter 2

System specifications

2.1 Overall framework

The primary objective of this research is the autonomous operation of a sewer-inspecting drone, facilitated by the integration of various tools and systems across two primary phases: manhole detection and positioning, and the subsequent descent into the sewer manhole.

Gazebo, a 3D robotic simulator, provides a controlled environment where we can rigorously test the algorithms designed for the autonomous operation of our drone. This process is streamlined with the Robot Operating System (ROS), serving as the backbone for operational software.

To ensure safe and controlled flight, both in simulation and real-world scenarios, the PX4 Autopilot is implemented. This open-source flight control software not only integrates smoothly with our simulation environment for effective testing and validation, but it also extends to the actual drone, delivering reliable, autonomous control.

Finally, to facilitate communication between the ROS-based control system and the drone, the MAVROS package is utilized. This system enhances MAVLink's capabilities within the ROS platform, allowing efficient transmission of commands from the ROS nodes to the drone and providing access to real-time telemetry data.

The following sections delve into each of these components, providing a comprehensive understanding of their roles within the framework.

2.1.1 Gazebo

Gazebo plays a pivotal role in the development and testing of the two main phases of our sewer inspection project, all within a controlled virtual setting. With its compatibility with ROS Noetic and support for PX4 Autopilot firmware, coupled with the ability to simulate realistic graphics, physics, and sensor data, Gazebo provides a highly accurate and realistic simulation environment tailored for intricate drone operations.

Furthermore, Gazebo offers a selection of pre-built models that can be customized to suit the specific needs of our project. This flexibility allows us to design a simulation environment that closely mimics the real-world conditions in which the drone will operate. Additionally, it enables us to adapt the drone model for our sewer inspection application by integrating the necessary sensors to replicate those onboard the actual drone.

2.1.2 Robot Operating System (ROS)

The Robot Operating System (ROS) is a flexible framework for writing robot software. In this project, ROS Noetic is specifically selected for its compatibility with Ubuntu 20.04 and Gazebo 11, and for the robust set of tools it provides for robot software development.

ROS plays a central role in our sewer inspection application, functioning as the primary platform for the development, and deployment of the two main algorithms — manhole detection and drone descent. These algorithms, encapsulated as ROS nodes, process sensor data to detect manholes and control the drone's movement, allowing it to descend accurately into the manhole. ROS's modular design and advanced message-passing interface enable seamless communication between nodes and other system components like the Gazebo simulator and PX4 Autopilot. This inter-node communication allows for real-time adjustments, critical for the accurate and autonomous operation of the drone during sewer inspections.

2.1.3 PX4 Autopilot

In our project, PX4 Autopilot, an open-source flight control software, controls the movements and operations of our sewer-inspecting drone. it is compatible with the Gazebo simulator that we use for testing our drone's manhole detection and descent algorithms. The communication between PX4 and Gazebo, as well as other elements of our system, occurs through the Simulator MAVLink API. This API facilitates the exchange of sensor data and motor/actuator values using MAVLink messages. It also manages communication with ground control stations such as QGroundControl and offboard APIs including MAVSDK and ROS, enabling us to seamlessly integrate our custom algorithms. Figure 2.1 provides a detailed illustration of the input and output flow of messages between the PX4 Autopilot system and the Gazebo simulator using this protocol.

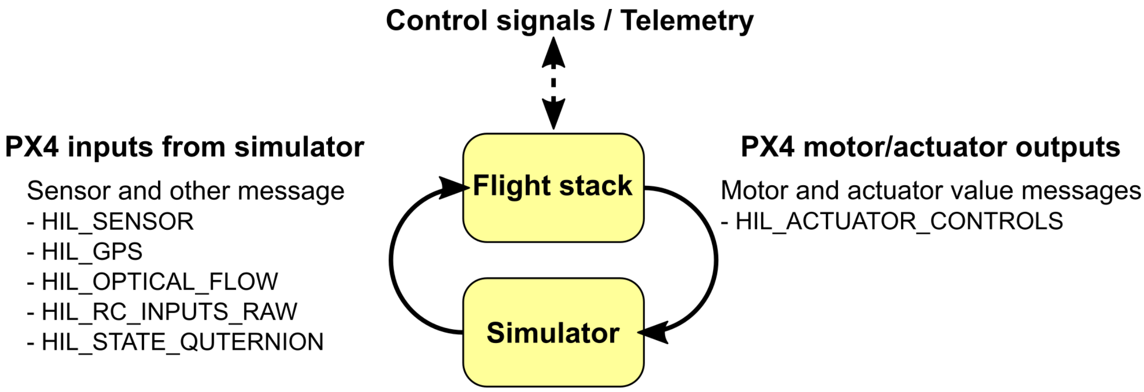


Figure 2.1: Diagram of message flow between PX4 and Gazebo [4]

During a SITL simulation, PX4, the Gazebo simulator, ground stations, and external developer APIs connect via UDP, either on the same computer or across different computers within a network. The SITL environment, depicted in Figure 2.2, features the main box on the left representing the PX4 autopilot system. The entire environment utilizes the UDP protocol to transfer information between tools, with PX4 employing three default port numbers: port 14540 for ROS, port 14550 for QGroundControl, and port 14560 for Gazebo. Gazebo automatically connects to port 14560, while QGroundControl listens on port 14550 for information from the autopilot system. The "API/Offboard" box in the top right corner indicates ROS communication and the "Simulator" box represents Gazebo communication.

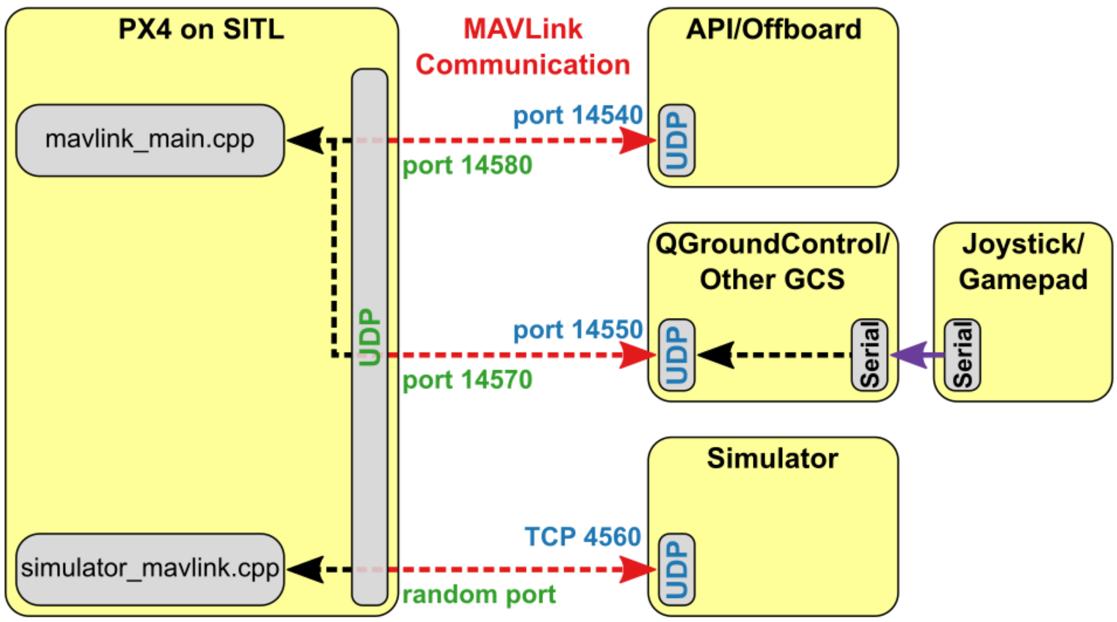


Figure 2.2: PX4 Software In the Loop environment overview diagram [4]

2.1.4 MAVROS

MAVROS plays a critical role in our sewer inspection project by serving as the communication bridge between the MAVLink-compatible PX4 Autopilot system and our ROS-based detection and descent algorithms. It translates MAVLink messages to ROS message types and vice versa, ensuring seamless information exchange within our system. This conversion enables the drone, equipped with MAVLink/MAVROS, to interact effectively with the ROS platform, facilitating successful execution of our sewer inspection application.

2.2 Drone specifications

The drone used in this project has a width of 257.80 mm, a height of 217.80 mm, and a limitted battery life of 20 minutes. It is specifically designed for sewer inspections, with a comprehensive set of cameras supporting the autonomous navigation into sewer manholes. To accomplish this task, the drone is equipped with four cameras that serve various purposes during the autonomous detection of sewer manholes and descent.

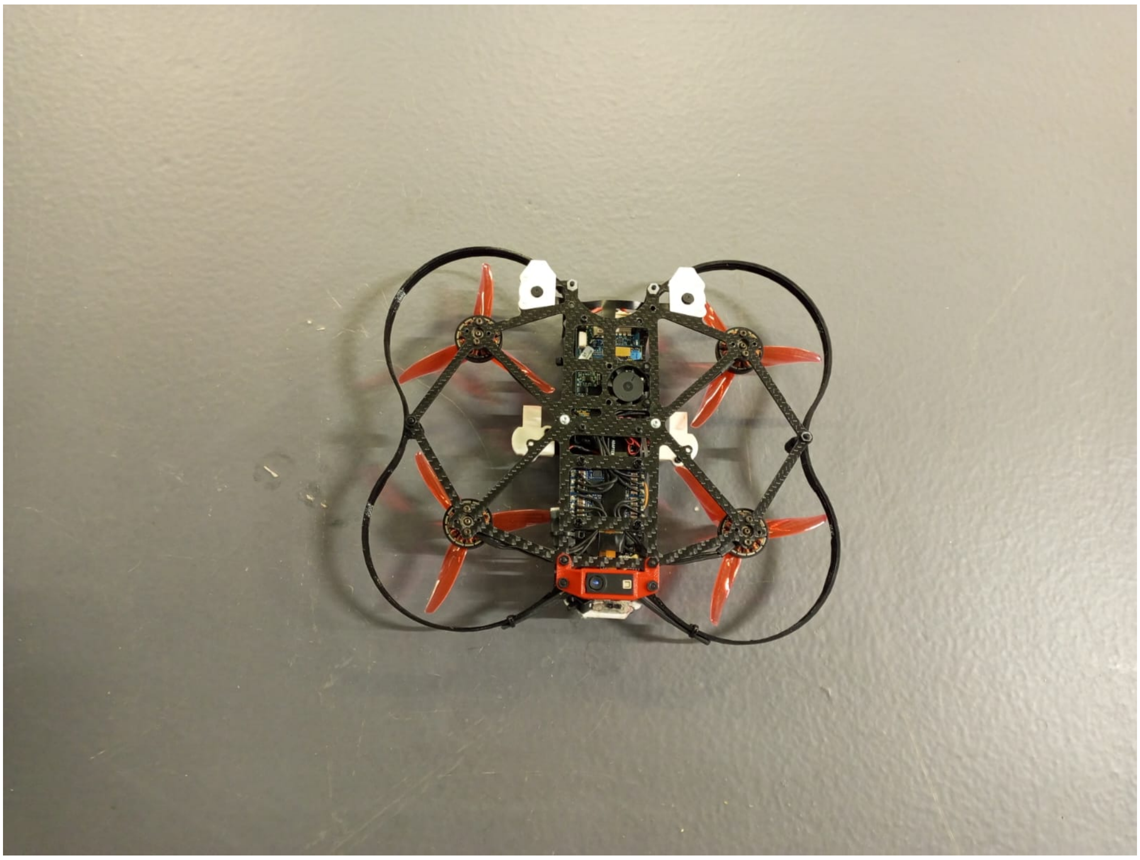


Figure 2.3: Image of the drone used in this project

2.2.1 Visualization camera

A front-facing camera (VOXL Hires Sensor M0024 IMX214) provides visual feedback and situational awareness for the operator or for processing by the drone's onboard computer.

2.2.2 Tracking camera

Another front-facing camera (VOXL Tracking Sensor M0014 OV7251) is used for dual purposes: to detect sewer manholes and perform visual odometry. It is placed beside the visualization camera, and both cameras are tilted at a 45-degree angle. This camera plays a crucial role in identifying the manhole and providing visual odometry data for localization.

2.2.3 Time-of-Flight (ToF) cameras

The Time-of-Flight (ToF) cameras used in the drone are based on PMD technology, which stands for Photonic Mixer Device. The PMD sensor works by emitting a short pulse of light and measuring the time it takes for the light to reflect back to the sensor. This time measurement is used to calculate the distance between the sensor and the object that reflected the light.

The PMD Time-of-Flight sensor used in the drone produces high-fidelity depth mapping indoors up to 6m. This allows the drone to accurately measure the distance to nearby surfaces and navigate autonomously within the sewer manhole. It is important to note that on the VOXL platform, the ToF sensor is mutually exclusive to the stereo cameras, meaning the stereo cameras need to be replaced with the ToF Add-on.

During the descent and ascent phases, the bottom and top ToF cameras play a crucial role in ensuring the drone maintains a safe distance from the walls and ladder while navigating the manhole. This is important to prevent collisions and ensure the safety of the drone and any potential human operators or bystanders.

Chapter 3

Literature review

The Literature Review chapter is structured to provide an in-depth look into the multidisciplinary areas that play a critical role in the development of an autonomous drone for sewer manhole inspection. This chapter begins with an overview of the relevant literature, presenting the context and motivation for this research, and identifying the unique value propositions of drones in this context. The theoretical framework delves into the key concepts, principles, and methodologies in the fields of Computer Vision, 3D Sensors, Deep Learning, Simultaneous Localization and Mapping (SLAM), and Ellipse Detection Algorithms. The section on related work examines the existing solutions and research findings related to sewer manhole detection, ellipse detection methods, and autonomous navigation. The final section focuses on identifying the gaps in the current literature, formulating the research questions, and defining the scope of this research.

3.1 Overview of relevant literature

The burgeoning application of drones in various fields has led to a surge of interest in their potential for sewer manhole inspections. Their inherent capabilities have been shown to enhance safety, efficiency, and cost-effectiveness significantly, outperforming traditional methods like CCTV and manhole cameras, as validated by multiple studies [5], [6], [7], [8], [9], and [10].

Building on this interest, the end of 2014 marked the initiation of a pilot project under the European Union's ECHORD++ innovation program. The project aimed to develop robots adept at performing inspection tasks within Barcelona's drainage and sewer network. The vision of this endeavor was to establish a satisfactory technological solution that could subsequently be applied to similar challenges in other cities across Europe and potentially worldwide.[11]

The unique value proposition of drones in this context, encompassing the following:

- Improved safety: Sewer manholes are often located in challenging and hazardous environments. The use of drones can minimize the risk of accidents and injuries to workers by reducing direct human exposure to toxic gases, contaminated water, and confined spaces.

- Enhanced efficiency: Manual inspection of sewer manholes is time-consuming and labor-intensive. Drones can cover large areas and inspect multiple manholes in less time, increasing the efficiency of the inspection process.

- High-quality data collection: Equipped with advanced sensors and cameras, drones can capture high-resolution images and accurate measurements of the sewer infrastructure. This data can be used to detect defects, assess structural integrity, and identify maintenance requirements more effectively.

- Cost-effectiveness: Drones can significantly reduce the costs associated with sewer manhole inspections by eliminating the need for expensive and specialized equipment such as confined space entry equipment, gas detectors, and safety gear.

- Accessibility: Drones can access hard-to-reach or hazardous areas that may be difficult or dangerous for human inspectors to access. This allows for more comprehensive inspections and a better overall assessment of the sewer infrastructure.

- Real-time monitoring: Drones can transmit live video feeds and data to inspection teams, enabling real-time monitoring and decision-making. This can help prioritize maintenance tasks and allocate resources more effectively.

- Reduced environmental impact: By minimizing the need for manual entry into the sewer system, drone inspections can help to reduce the environmental impact of sewer inspections. The use of drones can reduce the potential for contaminating water sources or releasing harmful gases into the environment during inspection processes.

However, the implementation of these innovative systems in practical sewer manhole inspections is fraught with significant technical challenges. The challenges faced by drone-based solution for this application are:

- Platform Limitations: Balancing the drone's stability, computational power, and size presents a significant challenge. Drones need stable controllers for maneuvering in confined spaces with potential obstacles and disturbances. They also require robust computational capabilities to handle real-time tasks like sensor data processing, image recognition, path planning, and potentially executing machine learning algorithms. Yet, these improvements must not compromise the drone's size, as navigating narrow sewer systems demands compact designs.

- Localization: Accurately determining the drone's position within the complex sewer network, as GPS signals are typically unavailable underground. This requires alternative localization methods, such as visual odometry or LiDAR-based solutions.
- Navigation: Navigate through the sewer system while avoiding collisions with pipes, debris, and other obstacles. Drones must be able to plan their path and adjust to changes in the environment autonomously. Particularly in the confined spaces of sewer systems, inherent sensor drift from Inertial Measurement Units (IMU) can exacerbate navigation challenges, causing accumulated errors over time and leading to positional inaccuracies.
- Manhole detection: Identifying manholes accurately and efficiently in diverse environments with varying lighting conditions, debris, and surface materials. This may involve the development of robust image processing algorithms or machine learning models to improve detection performance.
- Descent: Safely and autonomously descending into the sewer manhole while avoiding obstacles such as ladders, maintaining stability, and controlling the drone's speed. This task becomes particularly challenging due to confined spaces, airflow disturbances, and significantly reduced visibility in the dark environment of the sewer manhole [12], [13]. Conventional RGB cameras used for visual navigation struggle in low-light conditions, requiring alternative sensing solutions for effective operation.
- Communication: Maintaining reliable communication with the control station while operating in an underground environment with limited connectivity. This may involve the use of alternative communication methods or autonomous decision-making capabilities.
- Sensor limitations: Dealing with sensor limitations, such as reduced visibility for cameras and the potential for LiDAR sensors to become contaminated or obstructed by debris, water, or other materials present in the sewer system.

- Battery life: Ensuring sufficient battery life for the drone to complete its inspection tasks and return to the surface safely. The power constraints in such applications may require energy-efficient algorithms and hardware [1], [2], [3].

- Environmental factors: Operating in potentially hazardous environments with the presence of toxic gases, moisture, and corrosive substances. This requires drones to be designed with materials and components that can withstand these conditions.

This literature review supports our research's broader academic context, which is a part of the mission to develop a fully autonomous drone solution for sewer manhole inspection—a task that demands a period extending beyond three months. Given this time-bound constraint, the thesis narrows its focus to crucial aspects of the mission: autonomous take-off, sewer manhole detection, and safe descent through a vertical manhole while adeptly avoiding obstacles. Accordingly, our work primarily concentrates on addressing the challenges of platform limitations, manhole detection, navigation, descent, sensor limitations, and battery life. As we delve into these specific aspects, we aim to critically examine the existing work and prior research that tackle these challenges. The intention behind focusing on these aspects is twofold.

First, it aids in understanding the intricacies associated with the development of an autonomous drone system capable of conducting sewer inspections. By synthesizing the wealth of knowledge accumulated in these areas, we build a strong foundation for the rest of our research.

Second, this review takes the crucial step of identifying gaps in the existing body of literature, indicating which aspects have been extensively researched and which require further exploration. This investigation forms the basis for our research questions, providing a roadmap for our study and directing future research efforts in this field.

With this emphasis on the specified challenges, we are now equipped to critically analyze the relevant literature, identifying and synthesizing key findings to shape our study's specific research questions and objectives.

By the end of this comprehensive literature review, we will have painted a detailed picture of the opportunities and challenges in this field. This will lay the groundwork for detailing the gaps in the literature and formulating our research questions, thereby providing a clear trajectory for our research and guiding our contribution to the broader mission of creating an autonomous drone system for sewer manhole inspection.

3.2 Theoretical framework

3.2.1 Computer vision

In the realm of computer vision, the establishment of object recognition algorithm frameworks has reached a level of stability, allowing for the application of computer vision in a wide range of scenarios. Drones, or unmanned aerial vehicles (UAVs), are versatile robotic systems capable of flight, vertical take-off and landing, and impressive maneuverability. By integrating computer vision technology, drones have found extensive applications in navigation [14], precision landing, obstacle avoidance during flight, and visual simultaneous localization and mapping (Visual SLAM) [15].

In the context of sewer manhole inspection, drone-based object recognition and tracking technology leverage the capabilities of computer vision. During the flight, the drone captures images of the target objects (e.g., sewer manholes) while considering factors such as altitude and field of view. It then processes the extracted image features to recognize and track the objects of interest [16],[17]. Moreover, by combining this information with a relative position control algorithm, the drone can autonomously navigate and perform inspection tasks within the sewer system.

3.2.2 3D Sensors

LiDAR technology and Time-of-Flight (ToF) cameras significantly enhance the capabilities of drones for obstacle detection and avoidance, especially in challenging environments like sewer manhole inspections. Systems based on detection and avoidance employ these sensor technologies to identify and position obstacles, enabling the execution of maneuvers that assure drone safety. Both LiDAR and ToF cameras project infrared light beams in various directions, capturing the time of flight to create three-dimensional models of the navigational surroundings. As active technologies, they function independently of ambient lighting conditions, providing increased reliability in low-light or nighttime situations. Additionally, the time of flight measurements eliminates the need for extra processing to generate the 3D scene.

Mobile LiDAR systems utilize near-infrared radiations to measure the topology of an object and collect the reflectance returned by the measured objects. Due to the high pulse repetition rate (e.g., 550 kHz), the density of the acquired point clouds is very high (e.g., approximately 4000 points/m2 on the road surface), making mobile LiDAR systems a suitable solution for detecting manhole and sewer well covers on road surfaces.

In the context of sewer manhole inspections, incorporating LiDAR technology and ToF cameras into drones bolsters their capacity to detect and circumvent obstacles in the constrained and complex environment of sewer systems. While LiDAR provides a more accurate representation by emitting multiple light pulses, ToF cameras offer a simpler solution by sending out a single light pulse to obtain depth information. The 3D models produced by LiDAR and ToF cameras facilitate autonomous navigation for the drone, ensuring the aircraft's safety and the precision of the inspection procedure.[18]

3.2.3 Deep learning

Deep learning, a subset of machine learning, focuses on artificial neural networks with multiple layers that can learn and process complex patterns in data. It has become increasingly popular in various fields, including computer vision, natural language processing, and robotics. In the context of sewer manhole inspections, deep learning techniques can be applied to enhance object detection, recognition, and tracking of the manhole and other obstacles within the environment [19], [20],[21]. In addition, deep learning can assist in processing the data obtained from sensors like LiDAR and ToF cameras, contributing to better autonomous navigation and decision-making capabilities for the drone.

3.2.4 Simultaneous Localization and Mapping (SLAM)

SLAM is a computational approach that enables a robot or drone to build a map of its environment while simultaneously keeping track of its position within that map. This technique is particularly valuable for drones operating in GPS-denied environments, such as sewer manholes. By integrating SLAM with sensor data from LiDAR [22], ToF cameras [23], or computer vision systems [15] , drones can accurately estimate their position and orientation while navigating the complex and confined spaces of sewer systems. SLAM algorithms can also contribute to obstacle detection and avoidance, path planning, and efficient exploration of the sewer environment, enhancing the drone's ability to perform reliable and precise inspection tasks. [24]

3.2.5 Ellipse detection algorithms

Ellipse detection algorithms are fundamentally based on the distinct mathematical characteristics of an ellipse. To illustrate these properties, we can refer to Figure 3.1 as an example. The ellipse fitting or detection algorithms that have been developed so far can primarily be classified into three groups according to their architecture. The first group consists of algorithms that utilize algebraic or geometric least square methods. The second group employs clustering or voting-based Hough transform methods. The third and final group encompasses a range of uncategorized, statistical, or heuristic techniques that often involve combining different approaches [25].

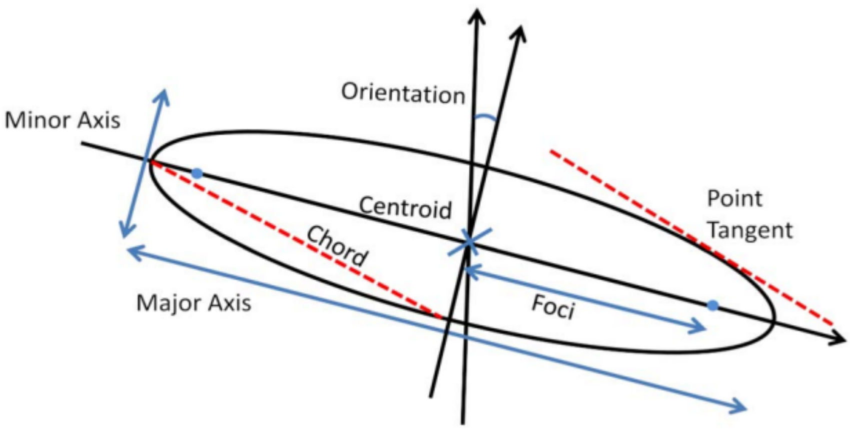


Figure 3.1: This figure highlights some useful parameters (blue) and geometric properties (red dotted line) of an ellipse [25].

3.3 Related work and previous research

3.3.1 Sewer manhole detection

A crucial aspect of the autonomous detection process is the drone's ability to recognize manholes while navigating the urban landscape. The literature offers various techniques and approaches to addressing the challenges associated with this unique perspective and the autonomous detection of manholes.

When it comes to detecting manholes and sewer wells in urban areas, LiDAR-based solutions are widely used due to the long-range, high-speed, and low-cost data acquisition. An automated stochastic object-oriented algorithm to detect sewer manholes from mobile LiDAR point clouds based on a marked point process of disks and rectangles is presented in [26]. The algorithm is based on the Bayesian paradigm in-order to efficiently process very large volumes of 3D point clouds, keeping in mind that the main scope here is to detect an unknown number and sizes of manholes and sewer wells in road surfaces. Mobile laser scanning (MLS) also known as Mobile LiDAR is used also for urban road manhole detection since laser scanning and navigation technologies have rapidly developed, and proven to be very efficient in acquiring very dense point clouds (over 800 points per square meter) along road corridors and been applied in many relevant applications such as surveying and mapping. In [27], they proposed an automated algorithm for detecting road manhole covers using MLS data. The mobile laser scanning system can carry the mapping mission day and night, which makes such a system a promising and cost-effective solution in similar tasks of detecting a big number of manholes in urban roads. They reduce the computational complexity and narrow down the searching regions by segmenting the road surface point from a raw point cloud and rasterizing into a georeferenced intensity image through inverse distance weighted (IDW) interpolation. Furthermore, a supervised deep learning model is developed and a random forest model is trained to detect road manhole covers. Another example regarding the MLS usage for road inventory and safety inspection, a feature recognition method presented in [28] is interpreting the whole MLS data according to the defined features. Initially, the raw MLS data are divided along road directions into sub-regions for segmentation and classification, and finally, further examination using the knowledge-based method to recognize the interested features for road inventory such as poles and crash barriers.

Another different approach to detect manhole covers from road surface images based on morphological techniques is presented by Tanaka and Mouri [16]. From the road surface image, they extract round shape components and use a black top-hat operation with disk shape structure elements to extract the round components. The resulting components go through a masking operation while applying a threshold input image, and by eliminating the regions with small areas the cover manholes were obtained.

A localization system presented in [29] includes a manhole detector module, that based on the depth images captured by a camera pointed upwards, checks whether the robot is under a manhole. Since the detected manholes offer a perfect opportunity to localize the robot according to the location of each manhole in the sewage system and the city drawings. The poor illumination conditions and high symmetry of the environment raised the need for machine learning approaches to train a classifier able to split the depth camera view between the regular ceiling and manhole [30]. The need for high-accuracy manhole detections and light-weight detection computations that are essential for the localization system, lead to the emergence of Convolutional Neural Networks (CNNs) as a convenient technique. Manhole detection is used to correct drift in the joint localization algorithm presented in [17], the location of the manholes can be either known a priori or mapped from above ground and then detected inside the pipes. They use a bag-of-features image recognition system to detect manholes through a linear support vector machine (SVM) classifier.

In the research paper by [31], the authors present an innovative approach to detecting manholes using depth images captured by drones navigating in dark, GPS-denied, and confined spaces. Rather than relying on traditional methods, their method employs a deep learning model specifically designed for this task. The study explores various network sizes to achieve the desired accuracy while maintaining low computational demands, making it suitable for implementation on a drone's parallel processing co-processor.

The core of their technique is the use of a temporal filter to enhance robustness and minimize false positives. This is achieved by requiring multiple detections within a specific timeframe before validating the final location of the manhole. The results of the study indicate that using deep learning on depth images is a viable solution for scene texture-agnostic manhole detection. The approach's effectiveness is demonstrated by a drone successfully navigating through a 1:1 size standard manhole found on a marine vessel.

3.3.2 Ellipse detection methods

A crucial aspect of the autonomous detection process is the drone's ability to recognize manholes while navigating the urban landscape. Due to the 45-degree tilted camera mounted on the drone, manholes are perceived as ellipses. This distinction is vital for developing robust computer vision and object recognition algorithms that can accurately identify and track manholes in real-time, enabling the drone to approach and descend into the manhole while avoiding obstacles. The literature offers various techniques and approaches to addressing the challenges associated with this unique perspective and the autonomous detection of manholes.

In the field of digital image processing, feature extraction is a key component that involves identifying and extracting simple features such as edges, corners, lines, and shapes, as well as more complex ones like rectangles and circles. Ellipse detection methods have matured over the past three decades but have not been extensively reviewed in the literature [25].

Ellipses are part of the family of conics in mathematics and can be viewed as a generalization of circles. Circles can appear as ellipses when viewed from a distant, oblique angle. The relationship between circles and ellipses contains meaningful information about size and orientation that can be used in various applications. Researchers have explored the properties of ellipses [32], such as the symmetric nature of their foci and the conjugate diameter relationship between parallel chords, to parameterize ellipses and predict their centers [33].

The importance of developing a robust ellipse detection system that can be applied in various real-world scenarios. The right balance between robustness and computational efficiency must be established for each specific application. The early research on ellipse detection focused on identifying a small number of computer-generated ellipses among primitive shapes, which can achieve high accuracy even with image noise or blur. However, this may not be suitable for more complex applications, such as tracking an elliptical face in a video sequence or detecting large numbers of elliptical cells in microscopy images [25]. Introducing the following set of real-world applications indicates the rule of ellipse detection.

- Surveillance and monitoring: Eye tracking [34], lips-reading [35], face detection [36], and human gait analysis [37].
- Agricultural applications: Segmentation of touching kernels for automatic sorting and picking [38].
- Industrial applications: Quality control in steel coil production using eccentricity measurement [39].
- Biomedical applications: Cell segmentation [40], [41], [42], 3D object shape reconstruction [43], and volume prediction of ellipsoid-shaped organs [44].
- Oceanography: Detection of elliptical water eddies in Surface Temperature maps [45].
- Spatial surveying: Identifying elliptical arcs in satellite images of natural contours and man-made structures [46].
- Space exploration: Analysis of solar corona with elliptical shapes [47].
- Electrical engineering: Impedance measurement system using ellipse fitting [48].
- Mechanical engineering: Strain and curvature analysis relying on ellipse fitting [49].
- Photovoltaic engineering: Tracking sunspot center orientation using ellipse fitting [50].
- Mechatronics applications: Localizing cameras on moving vehicles [51], [52] or unmanned helicopters using ellipse fitting on circular traffic signs [53].
- Character reconstruction: Chinese calligraphic stroke reconstruction using ellipse fitting [54].

Application-specific demands drive the need to increase computational efficiency and adapt to different types of images. Most projects involve introducing modifications to existing methods, but this practice hinders the development of a futuristic system capable of efficiently extracting ellipses from a variety of images automatically [25].

Ellipse detection algorithms can be broadly classified into three categories: least-square methods, clustering or voting-based methods, and uncategorized, statistical, or combined methods [25].

A. Least-square methods: These methods are popular for ellipse fitting due to their linear nature and simplicity, making them suitable for real-time applications [55], [56]. The direct Least-Square (DLS) method is a notable development [57], as it performs ellipse-specific fitting in a non-iterative manner. However, these methods can be sensitive to noise, biased, and limited in their ability to fit multiple primitives [58].

B. Clustering or voting-based methods: Hough Transform (HT) methods are widely used in computer vision for detecting geometric features. Despite their robustness towards occlusion, these methods are not ideal for multiple ellipse detection due to the increased computational demands. Some possible HT methods for multiple ellipse detection include Generalized HT [59], Straight Line HT [60], Fast Ellipse Hough Transform [61], and Randomized HT [62].

C. Uncategorized, statistical, or combined methods: Extended Kalman Filter (EKF) [63] and other statistical methods like RANSAC [64] and Genetic Algorithm [65] have been used for ellipse detection. However, they may perform poorly in high-noise environments or when multiple ellipse detection is required. Other approaches include ellipse growing, noise-inclusive statistical estimation methods such as maximum likelihood, and combined methods that use HT alongside least square methods or other techniques. While some of these methods can be computationally expensive, they may outperform other methods in certain situations.

In the paper [66] the authors propose a novel algorithm for detecting and tracking elliptical patterns in real-world robotics tasks. The algorithm fits ellipses to each contour in the image frame, discarding those that do not provide a good fit. This lightweight method is suitable for resource-limited onboard computers in robots and can handle lighting variations and partial views. The authors tested their approach on an autonomous UAV landing on a fast-moving vehicle, demonstrating its performance in indoor, outdoor, and simulated real-world scenarios. Compared to other well-known ellipse detection methods, the proposed algorithm achieved an F1 score of 0.981 on a dataset with over 1500 frames, indicating its potential for improved performance in certain robotics applications.

In the paper [67] the authors propose a novel ellipse detection method called arc adjacency matrix-based ellipse detection (AAMED). The method begins with segmenting edges into elliptic arcs and constructing a digraph-based arc adjacency matrix (AAM) to represent their triple sequential adjacency states. To make the AAM sparse, curvature and region constraints are applied. The algorithm then performs bidirectional searches on the

AAM to find potential ellipse candidates, utilizing cumulative-factor (CF) based cumulative matrices (CM) for efficiency. The ellipses are fitted from these candidates through the eigendecomposition of CM using the Jacobi method. Lastly, a comprehensive validation score is introduced to effectively eliminate false ellipses, taking into account adaptive shape, tangent similarity, and distribution compensation. The experiments demonstrate that the AAMED method outperforms 12 other state-of-the-art methods across 9 datasets in terms of recall, precision, F-measure, and time consumption.

In [68], the authors introduce a unique method for detecting elliptical features in digital images, which is not based on the widely used Hough transform or random sample consensus approaches. Instead, their technique is founded on the circle power theorem and its extension to conics. The primary concept involves transforming an image into multiple power ratio histograms, which record the distribution of power ratios calculated from specific reference point pairs. The authors demonstrate that peaks in power ratio histograms offer substantial evidence for the presence of ellipses. By employing peak verification based on the parallel chord property and multi-histogram cross-validation, the confidence of an identified peak is increased. The method's accuracy and robustness are validated through experiments on various synthetic and natural images.

3.3.3 Autonomous navigation

In the realm of autonomous navigation, existing solutions can be broadly categorized into global path planning (deliberative), local path planning (sensor-based), and hybrid methods [69].

Deliberative, or planning-based, approaches necessitate a map representation of the environment and localization information to determine safe local paths. These methods often rely on simultaneous localization and mapping (SLAM) techniques for autonomous operation in unknown environments [70]. For instance, the method in [71] combines an Unscented Kalman Filter (UKF) and a particle filter to process IMU and range measurements for UAV localization in dam penstocks with pre-existing maps. This semi-autonomous approach was later extended in [2] and [3] for more autonomous penstocks inspection.

In [2], UKF was employed to provide a 6-DOF estimation of the UAV pose by fusing data from the IMU, two range sensors, and four cameras against a known 3D occupancy grid map. However, a remote operator was still needed to provide waypoints to guide the UAV. In [3], a SLAM-based approach combining range and vision-based estimators was suggested. An algorithm was proposed for local mapping that applied a cylindrical model fitting to point clouds obtained from the heterogeneous sensors. Subsequently, the tunnel axis was estimated from the local map, and the UAV position was determined along the tunnel axis to guide the UAV.

When the robot can navigate freely in 3D space, some researchers [72],[73] employ adaptive path planners based on traditional techniques, such as A* [74], D* lite [75], PRM [76], or RRT [77]. However, these approaches often come with the drawback of high computational cost, and they consistently compute a collision-free path to the target location, which can be resource-intensive in unknown or dynamic environments. In contrast, refs [78] presented a different approach for operations in extremely narrow tunnels. This method finds safe paths using a modified A* algorithm in 2D occupancy maps generated by onboard SLAM, and it employs a low-level model predictive controller to track local trajectories based on the planned paths.

For inspections in tunnels, [12] and [13] proposed using a UAV equipped with a rotating camera to minimize field-of-view (FOV) obstructions when collecting images for inspection. The system also incorporated an array of laser range sensors, which leveraged prior knowledge of the tunnel's geometry to estimate the UAV's position and heading, allowing it to maintain its position at the tunnel's center. In the same studies [12] and [13], the team also conducted an experiment to test the UAV's autonomous performance in vertical shafts. Specifically, they simulated the access or exit from the tunnel via a DTSS entry shaft through a 5 m diameter manhole at ground level. The UAV was prepared with a safety tether for emergency retrieval, and the initial positioning at the shaft's centerline was manually controlled by a human pilot. Once in autonomous mode, the UAV was programmed to maintain its position in the shaft center, while the pilot manually controlled its vertical speed and altitude. The UAV conducted an autonomous flight of approximately 4.5 minutes, covering a vertical distance of roughly 8 m. The results of this experiment revealed an RMS position error of 0.53 m with a maximum deviation of 1.44 m from the shaft centerline, which was greater than the errors observed during the horizontal trial at the canal. The larger error may be attributed to the vertical nature of the flight, the manual control of vertical speed and altitude, and the inherent complexities of navigating vertical shafts. Despite the increased errors, this experiment demonstrated the potential applicability of UAVs for inspecting vertical tunnel shafts, although it also underscored the necessity for improvements in autonomous control during vertical navigation. Alternatively, deep learning approaches such as those presented in [19], [20], and [21] utilized Convolutional Neural Networks (CNNs) for navigation in underground mines with low-cost UAV systems, relying only on a single camera and LED light bar. In these approaches, the CNN classified images into three categories—left, center, and right—to adjust the UAV's heading and prevent collisions with tunnel walls, all without depending on localization information.

The UAV's motion was managed in the horizontal plane, while a remote operator provided a fixed altitude. The effectiveness of these methods hinges on the quality of the training dataset, posing challenges when implementing them in unfamiliar environments.

However, with an increase in the size of the search space, the computational cost of algorithms such as these becomes significantly more expensive [79]. Consequently, sensor-based methods, which are a subset of reactive approaches [80], can be more beneficial. These methods generate motion decisions based on the lightweight processing of current sensor observations, resulting in rapid and instinctive responses. Compared to planning-based methods, these approaches offer a more favorable computational cost and eliminate the need for precise localization.

In tunnel-like environments, reactive methods have been proposed, such as [81], [82], and [83], which focus on local sensory depth information of surrounding tunnel walls. These methods provide rigorous mathematical proofs of their performance and utilize control laws or estimations to maintain movement parallel to the tunnel axis while keeping a safe distance from the walls.

Merz and Kendoul [84] introduced a LiDAR-based collision avoidance solution for helicopters in structure inspection scenarios, which relies on a 2D LiDAR and the pirouette descent and waggle cruise flight modes for accurate mapping of the 3D environment. This method requires precise estimation of the vehicle's position for proper obstacle detection and allows for uninterrupted flights while avoiding obstacles, providing a real-time solution.

Hrabar [85] presented a reactive approach to collision avoidance that creates an elliptical search zone whenever an obstacle is detected in the planned path. The safety volume is composed of spheres that approximate a cylinder. The algorithm has been tested with both stereo vision and LiDAR-based systems, with the best results obtained using the LiDAR-only system. In [86], Hrabar's work was extended by incorporating a LiDAR sensor with 360-degree coverage and a range of up to 100 m, focusing on improving the search area for escape points, dynamic safety volume, local minima escape procedures, and limited search space.

Vanneste et al. [87] proposed a real-time approach that extrapolates the 2D VFH+ [88] method to 3D using a 2D polar histogram of the environment. The algorithm generates new robot motions with an average time of only 326 μs. However, this method requires tuning parameters depending on the application.

Some methods, such as those presented in [89], [90], [91], focus on collision avoidance for UAVs in shared spaces, considering the presence of other known UAVs and static obstacles. Alejo et al. [89] proposed an indoor method based on optimal reciprocal collision avoidance (ORCA) with centralized control.

Hybrid strategies aim to address the aforementioned drawbacks by combining both deliberative and reactive approaches for more efficient navigation behavior in unknown and dynamic environments. There exists a considerable body of literature on hybrid approaches, with examples found in [92], [93], [94], [95], [96], [97], [98] and the references cited therein.

Many existing hybrid navigation methods in three-dimensional (3D) environments employ search-based techniques for local planning components. While some utilize reactive-based approaches, they often only consider two-dimensional methods by constraining UAV movement to a fixed altitude, which fails to take full advantage of UAV capabilities. Thus, the primary contribution of this work is to propose a hybrid 3D navigation strategy for UAVs, enabling efficient navigation in partially unknown or dynamic environments. The proposed strategy combines a global path planning layer with a reactive obstacle avoidance control law, developed based on a general 3D kinematic model. The global path planning layer, which is founded on RRT-Connect, can generate efficient paths using available environmental knowledge. A sliding mode technique is employed to implement a boundary-following behavior in the reactive layer, offering quick obstacle responses at a low computational cost compared to search-based and optimization-based local planners. To create a well-rounded hybrid navigation strategy, a switching mechanism is introduced to manage the transition between the two control laws. Ultimately, the suggested method can overcome the limitations associated with relying solely on deliberative or reactive approaches.

A hybrid approach was proposed in [92] for navigation in dynamic environments, which integrated a potential field-based local planner with the A* algorithm serving as a global planner, relying on a topological map. The authors of [99] recommended an alternative hybrid approach for micro aerial vehicles, employing A* for both deliberative and local planning components. In this approach, the global path is refined locally around obstacles by undergoing re-planning processes.

3.4 Gaps in the literature and research questions

Upon critically reviewing the pertinent literature, two significant gaps in existing research are evident, guiding the focus of our study.

Firstly, there's a clear void regarding the detection of sewer manholes specifically aimed at precisely pinpointing their accurate center position. Existing literature mainly concentrates on recognizing manholes and sewer wells within urban contexts, primarily detecting manhole covers from the road surface. However, a lacuna becomes evident when one considers the accurate localization of a sewer manhole's center, a vital prerequisite for correct drone positioning for inspection missions. This dearth of research signifies a crucial area for exploration and development, imperative for furthering sewer inspection methodologies.

Secondly, the literature critically lacks discourse surrounding the drone's controlled descent during inspection missions. Much research has been carried out on navigation and detection for inspections of horizontal pipes. Still, the procedure of descending into vertical sewer manholes fully autonomous is glaringly neglected. The absence of substantial work in this area underscores the pressing need for research focused on devising robust and dependable methods for the drone's vertical descent into manholes. This gap presents a compelling opportunity for progress and innovation within the domain of autonomous drone technology for sewer inspections.

Beyond these identified gaps, the aspect of simulating the entire inspection operation and ensuring the real-world transferability of the developed solutions becomes a pertinent research question. While it doesn't directly correspond to the specific gaps identified, the issue of simulation and transferability is central to our study as it seeks to ensure the methods devised for manhole detection, navigation, and descent are not just theoretically sound but also practically applicable and transferable to real-world scenarios.

Given these discernible gaps and significant research questions, our study is shaped around the following key inquiries:

How can we devise an autonomous drone system capable of accurately detecting a sewer manhole's center to bridge the gap in current detection and localization techniques? How can we design an autonomous drone system that ensures a safe and dependable descent into vertical sewer manholes, effectively addressing the barely explored challenge of vertical descent in the context of sewer inspections? How can we accurately simulate the mission to guarantee that tested procedures are easily transferable to the real robot, especially when considering variations in tunnel geometry and camera position? In pursuit of answers to these questions, our research aims to make significant contributions to existing knowledge, providing innovative and pragmatic solutions to these identified gaps. We hope that our efforts will not only answer the aforementioned questions but also lay the foundation for future research, development, and innovation in the realm of autonomous drones for sewer inspections.

Chapter 4

Methodology

The system proposed for sewer manhole localization and descent using drone technology incorporates a blend of autonomous drone flight, image processing techniques, and path planning algorithms. These elements harmoniously collaborate to facilitate precise location and identification of sewer manholes in urban landscapes, ensure navigational accuracy towards them, and administer safe descent of the drone into these manholes.

The system comprises three major components: Drone Control, Manhole Center Localization and Navigation, and Drone Descending, as visually represented in Figure 4.1.

The procedure is divided into two phases, with the first phase encapsulating the Drone Control and Manhole Center Localization and Navigation components, and the second phase focusing on the Drone Descending component.

Phase One commences with the Drone Control component. The drone is readied for autonomous flight, a process that involves arming the drone, activating offboard control, and implementing takeoff. Upon achieving a pre-defined altitude, the procedure transitions to the Manhole Center Localization and Navigation component. This step utilizes images from the onboard tracking camera to identify and navigate to the sewer manholes. The captured images undergo processing, followed by the application of the detection algorithm, which serves to pinpoint the center of the manhole within the image frame. This identified center is then translated into real-world coordinates, directing the drone toward the recognized manhole center.

Phase Two is initiated once the drone has navigated to the identified manhole center. It consists solely of the Drone Descending component. This step relies heavily on a depth image sourced from a downward-facing ToF camera. The depth image is subdivided into smaller depth intervals, each of which is used to create an individual mask. These masks are then processed, facilitating the identification of the free area's center within each mask.

The identified centers are then translated into real-world coordinates, generating multiple free area centers. This process aids in establishing a safe and viable path for the drone to follow during its descent into the manhole, thereby ensuring the overall safety and efficacy of the operation.

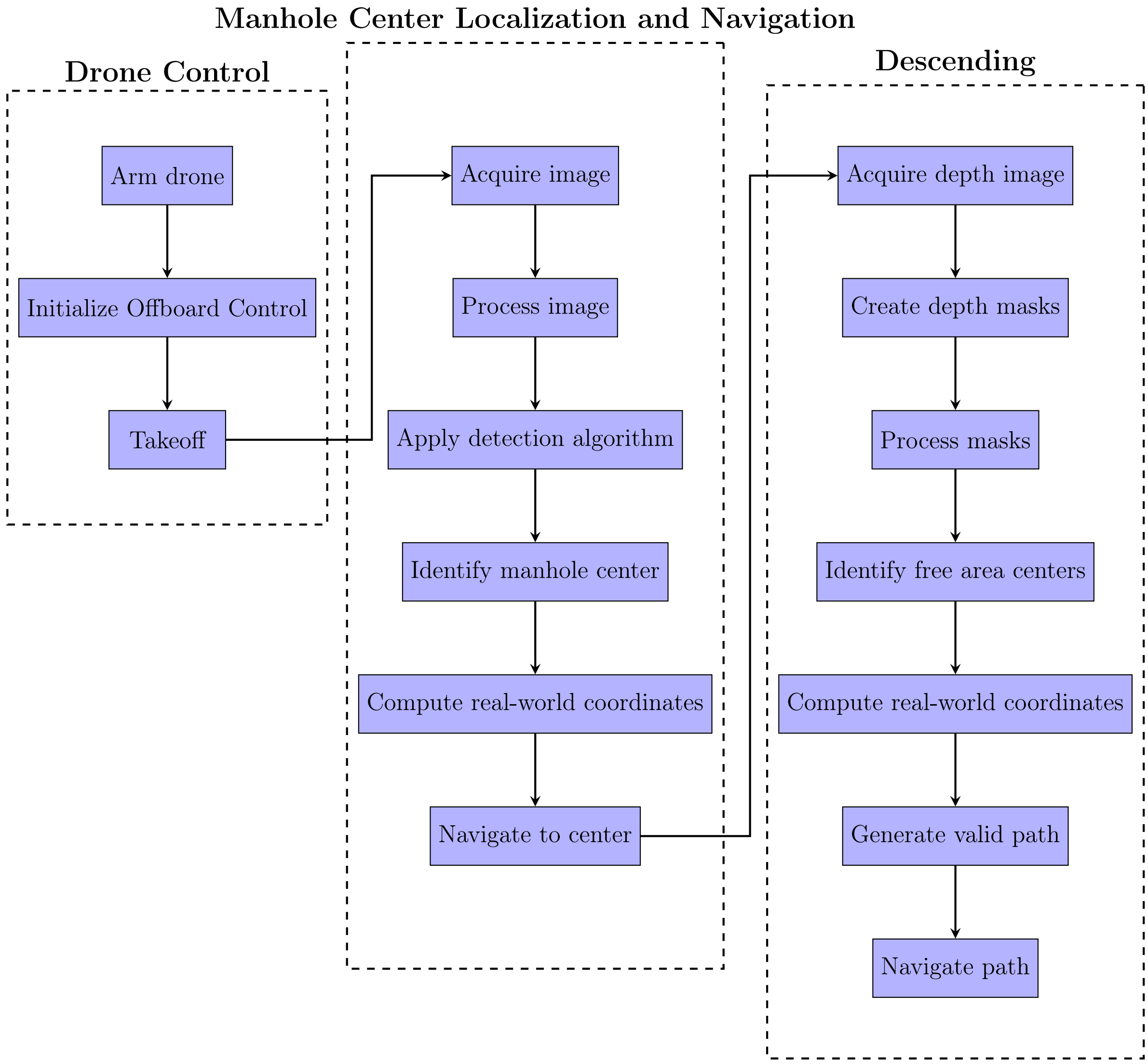


Figure 4.1: Flowchart of the drone-based sewer manhole localization and descending system.

4.1 Drone control

The drone control component prepares the drone for autonomous flight and ensures its safety and proper operation through the following actions:

- Arming the drone: The drone's motors and internal systems are activated. During arming, the drone performs self-checks to ensure essential components, such as the

IMU and sensors, are functioning correctly. Arming also unlocks the motors for lift generation.

- Off-board mode: Control is handed over to an external device, like a companion computer running ROS, instead of using the built-in remote controller. Off-board mode enables advanced control capabilities, autonomous missions, complex flight paths, and tasks that require precision and coordination based on instructions or real-time commands from ROS nodes.
- Initiating takeoff: The drone receives a command from a ROS node to lift off. Motors spin faster to generate lift, while onboard sensors and control algorithms maintain stability and control. Upon reaching a predefined altitude or a safe hovering height, the drone enters a stable hover mode, awaiting further ROS node commands to execute its mission, such as moving to the sewer manhole center and descending into it.

4.2 Manhole center localization and navigation

The Sewer Manhole Center Localization and Navigation component effectively locate and navigates to the manhole center using a sequence of stages:

4.2.1 Image acquisition and preprocessing

The drone captures images of the ground using its onboard tracking camera. The first stage involves undistorting the received images to account for the fisheye lens distortion. This step rectifies the images, ensuring accurate detection and localization of the sewer manholes.

4.2.2 Sewer manhole detection and center identification in images

Since the images are captured utilizing the front-tilted tracking camera positioned at a 45-degree angle, This tilted perspective causes the camera to capture the manhole's circular shape as an ellipse. Therefore in this stage, ellipse detection algorithms are applied to the undistorted images to identify sewer manholes. Two ellipse detection algorithms have been tested.

4.2.2.1 Algorithm 1

An OpenCV-based shape detection algorithm is employed. A step-by-step explanation of the algorithm is as follows:

1. **Noise filtering**: Convert the input image to grayscale, then apply a bilateral filter for noise reduction while preserving edges.

2. **Edge detection**: Use the Canny edge detector to extract edges from the filtered frame. Select thresholds thoughtfully to balance between capturing suitable edges in various conditions and avoiding excessive edge generation.

3. **Morphological operation**: Implement a morphological closing operation on the detected edges using a disk-shaped structuring element. This operation, which includes dilation followed by erosion, helps to fill minor gaps and connect disjointed components, especially beneficial when the manhole is slightly occluded.

4. **Contour extraction**: Extract contours from the processed image. Each contour is assessed individually to determine its elliptical nature. The algorithm verifies if there are sufficient points, typically 5 or more, to constitute an ellipse.

5. **Area range filtering**: Upon confirming the elliptical shape, the algorithm calculates the contour's area. The algorithm then checks if this area falls within the typical range for sewer manholes, which have diameters between 600 mm and 800 mm.

6. **Aspect ratio filtering**: If the calculated area lies within the specified range, the algorithm fits a rotated rectangle around the contour. This rectangle is instrumental in approximating the major and minor axes of the ellipse, independent of the ellipse's orientation. This enables the calculation of the aspect ratio, which differentiates between circular and elliptical shapes, a distinction that is critical in this context as the camera's 45-degree tilt causes circular manholes to appear elliptical in captured images.

The provided pseudo-code serves as a clear outline of the algorithm's steps and methodology.

4.2.2.2 Algorithm 2

This adapted ellipse detection algorithm is designed for the specific task of detecting sewer manholes in images. It is based on the method presented in [67], and includes the original five main steps (steps 1–5) from the paper. To better suit the sewer manhole

Algorithm 1 Ellipse Detection

```
procedure DETECT_ELLIPSE_CENTERS(image)
    gray ← CONVERT_TO_GRAYSCALE(image)
    blur ← APPLY_BILATERAL_FILTER(gray)
    edges ← DETECT_EDGES_USING_CANNY(blur)
    closed_edges ← APPLY_MORPHOLOGICAL_CLOSING(edges)
    contours ← DETECT_CONTOURS(closed_edges)
    filtered_contours ← INITIALIZE_LIST
    for each contour in contours do
        if contour has sufficient points then
            area ← CALCULATE_CONTOUR_AREA(contour)
            if area is within specified range then
                rotated_rect ← FIT_ROTATED_RECTANGLE(contour)
                aspect_ratio ← CALCULATE_ASPECT_RATIO(rotated_rect)
                if aspect ratio is within the specified range then
                    APPEND_TO_LIST(filtered_contours, contour)
                end if
            end if
        end if
    end for
    return filtered_contours
end procedure
```

detection application, two additional steps (steps 6 and 7) have been introduced, which consider the area range of sewer manholes and identify their centers.

1. **Arc extraction**: This step consists of three sub-steps: edge detection, contour extraction, and arc segmentation. First, the Gaussian blur method is used for smoothing the image, followed by the Canny edge detection method with adaptive thresholds to detect edges. Next, non-branched connected edge contours are extracted and approximated into multiple sets of line segments using the non-parametric Douglas-Peucker method. These sets of line segments are referred to as DP contours. Edge contours with lengths less than an adaptive threshold (T_{edge}) are considered noise and discarded. Arcs are extracted based on their convexity and curvature by calculating an extracting area using a vector and its rotated version. To determine if a point (A_{k+1}) belongs to the same arc (v_k) as the previous points (A_k, A_{k-1}), a formula with unknown parameters t and p is used: $v_{k+1} = tv_k + pR(s\theta_{\text{arc}}) \cdot v_k$. The formula represents a linear combination of the original vector (v_k) and its rotated version ($R(s\theta_{\text{arc}}) \cdot v_k$). The parameters t and p determine the weights of these two vectors in the combination. If the combination of the two vectors with the right weights (t and p) results in the vector v_{k+1}, then it means that the next point (A_{k+1}) belongs to the same arc as the previous points (A_k, A_{k-1}).

2. **Adjacency arc matrix construction**: Construct an Adjacency Arc Matrix (AAM) by defining an asymmetric matrix L to represent the connectivity between arc pairs. Use the KD-Tree to find adjacent arcs and verify if they can belong to the same ellipse by checking region and curvature constraints. The region constraint is satisfied if the starting point of one arc is in the search region of the other arc and vice versa. The curvature constraint is satisfied if the ending point of one arc and the starting point of the other arc meet certain conditions related to the curvature of the ellipse. The matrix L is then calculated for different cases based on the distance between the ending point of one arc and the starting point of the other arc, and the AAM is constructed after traversing all extracted arcs.

3. **AAM Based candidate generation**: Generate all possible arc combinations using Posterior Arc Search (PAS), Anterior Arc Search (AAS), and Bidirectional Combination Verification (BCV). PAS and AAS find all posterior and anterior arc combinations, respectively. BCV combines the results of PAS and AAS to produce all possible arc combinations and verifies them. The verified combinations are fitted to ellipses, and similar ellipses are removed using ellipse clustering.

4. **Ellipse fitting speedup**: The Ellipse Fitting Speedup method accelerates the ellipse fitting process by using cumulative factors and matrices to minimize redundant calculations when obtaining the fitting matrix, as two or more candidate combinations might contain the same arcs. This is achieved by pre-calculating cumulative factors for each point and summing them to form cumulative matrices. Fitted parameters are then calculated directly through twice real symmetrical matrix eigendecomposition using the Jacobi method.

5. **Ellipse validation**: Filter out false ellipses formed by candidate combinations using a set of sampling points and four indexes: shape index (SI), location index (LI), gradient index (GI), and weight index (WI). SI constrains the shape of the fitted ellipse, ensuring it represents a one-fourth ellipse. LI checks if a sampling point lies in the 8-neighborhood points of an edge point. GI measures the similarity between the estimated gradient and theoretical gradient at the sampling point. WI accounts for the non-uniform position distribution of points across the ellipse. The validated score, P_{score}, is computed using these indexes. If P_{score} is greater than the validation threshold T_{val}, the fitted ellipse is considered a true ellipse; otherwise, it is discarded.

6. **Area range filtering**: Filter the detected ellipses based on the typical area range of sewer manholes, which have diameters between 600 mm and 800 mm. This step

ensures that only ellipses corresponding to manholes are considered.

7. **Overlap removal and center identification**: Remove overlapping ellipses and identify the center of the remaining ellipse, which represents the manhole.

4.2.2.3 Operator selection

The detection algorithms identify and display all detected manholes along with their respective centers. Given that the drone operation is not authorized without an operator, this detection system offers an added benefit. It empowers the operator to select the intended manhole for inspection, especially in scenarios where multiple manholes are visible. Once selected, the center of the chosen manhole is back-projected to real-world coordinates.

4.2.3 Back projection

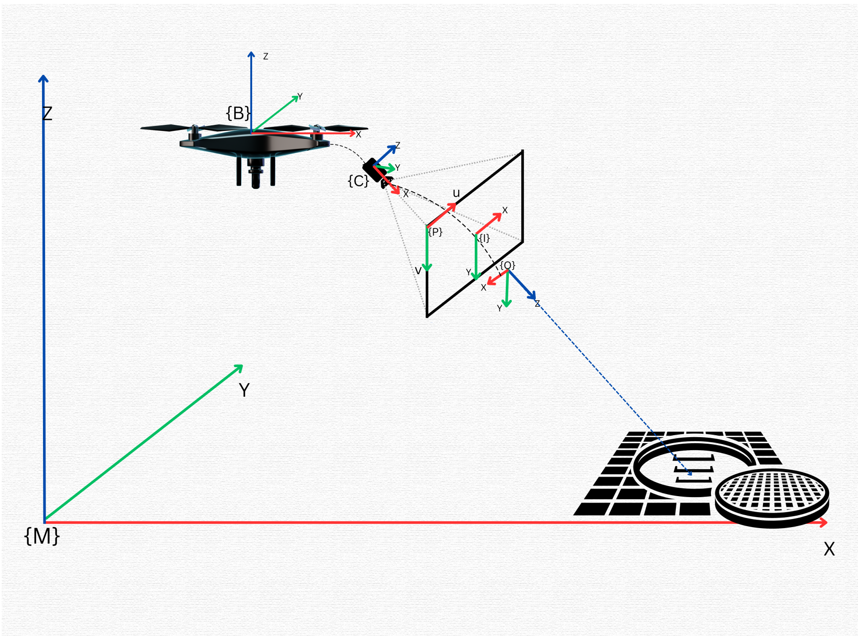


Figure 4.2: Visualization of coordinate frames and back projection transformation in the drone-manhole scenario

In this section, we explain the process of back-projection as shown in Figure 4.2, which involves computing the real-world coordinates (in our case, coordinates in the map frame) of a point detected in an image. This is particularly useful for our application, as we need to compute the real-world position of the detected manhole center to send it to the drone, allowing it to move towards the manhole center.

Fundamental to the process of back-projection are the intrinsic camera parameters, obtained through camera calibration. These parameters, often arranged in a matrix known as the camera matrix, form the relationship between the 3D real-world space and the 2D image space. Specifically, the intrinsic parameters are as follows:

$$K = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Where the principal point (u_0, v_0) is a point on the image plane through which the lens's line of sight passes. This point typically corresponds to the center of the image or lens. The focal lengths in the x and y directions (f_x, f_y) represent the distance between the camera's sensor and the lens, influencing the field of view. In our context, f_x and f_y refer to the focal lengths in pixels. These pixel-based focal lengths are calculated as the product of the physical focal length and the number of pixels per unit length (denoted by ρ) in the respective directions:

$$f_x = f_{\text{physical}} \times \rho_x, \quad f_y = f_{\text{physical}} \times \rho_y$$

The back-projection process can be divided into the following ordered steps:

1. **Converting pixel coordinates to camera frame coordinates**

   To convert the detected center of the manhole from the pixel coordinate frame (denoted as P in Figure 4.2) to the camera's optical frame coordinates (denoted as O in the same figure), we first subtract the principal point (u_0, v_0), effectively moving the origin of the pixel frame from the top-left corner to the center of the image, as shown in the following equation:

$$\begin{bmatrix} u' \\ v' \end{bmatrix}^P = \begin{bmatrix} u \\ v \end{bmatrix} - \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} \tag{4.1}$$

   Next, we divide these coordinates by (f_x, f_y), which represent the focal lengths in pixels in the x and y directions. This step serves two purposes: it converts the coordinates from pixels to meters, and it reverses the perspective projection effect.

$$\begin{bmatrix} x'' \\ y'' \end{bmatrix} = \begin{bmatrix} \frac{u'}{f_x} \\ \frac{v'}{f_y} \end{bmatrix} \tag{4.2}$$

   At this point, we know that the real-world point lies on a 3D vector expressed in the camera frame, starting from the (x'', y'') point and extending infinitely. The goal is to compute the length of this vector (scaling factor, denoted by λ) where it intersects the ground, as our manhole center lies on the ground.

2. **Computing the scaling factor λ**

To compute the scaling factor λ, we need the transformation matrix T_O^M, which represents the transformation from the camera's optical frame to the map frame. The optical frame, denoted as frame {O} in Figure 4.2, is a dummy frame created to represent the conventional camera frame, where X is right, Y is down, and Z is forward. In contrast, the ROS standard orientation, denoted as frame {C} in Figure 4.2, has X forward, Y left, and Z up.

To accommodate the difference between these orientations, we have created two different frames located at the same position: one named `camera_link` (following the ROS convention) and the other `camera_link_optical` (following the standard image convention). The Image and CameraInfo message headers should reference the `camera_link_optical` frame.

We have created a static transformation between the `camera_link` and `camera_link_optical` frames and placed it in the `transformation.launch` file. This allows us to take advantage of the tf library and directly access this transformation using the tf listener function.

With the transformation matrix T_O^M, we can compute the scaling factor λ using the following equation.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}^M = R_O^M \times \begin{bmatrix} x'' * \lambda \\ y'' * \lambda \\ 1 * \lambda \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}_O^M \tag{4.3}$$

By extracting the third row of the aforementioned equation, we can directly compute the scaling factor λ using the following expression:

$$\lambda = \frac{Z - t_z}{(R_O^M \times \begin{bmatrix} x'' \\ y'' \\ 1 \end{bmatrix})_3} \tag{4.4}$$

In this computation, we assume that the depth Z of the hole is zero, a reasonable assumption for flat ground surfaces. However, this assumption may not hold for uneven terrains. In such scenarios, alternative approaches might be required to estimate the position of the manhole

3. **Computing the 3D point in the optical frame**

   Once we have the scaling factor λ which represents the 3D vector that goes from the camera's optical center to the manhole's center in the optical frame, we can multiply it by the normalized image frame coordinates to obtain the 3D point in the optical frame:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}^O = \lambda \times \begin{bmatrix} x'' \\ y'' \\ 1 \end{bmatrix} \tag{4.5}$$

4. **Computing the real-world coordinates in the map frame**

   Finally, to compute the real-world coordinates of the manhole center in the map frame, we can apply the transformation matrix T_O^M to the 3D point in the optical frame:

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}^M = T_O^M \times \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}^O \tag{4.6}$$

   Now, we have the real-world 3D coordinates of the manhole center in the map frame, which will be sent to the drone to navigate to the correct location of the manhole center to start the descending phase.

4.3 Descending

The main concept of the descending algorithm, as illustrated in Figure 4.3, lies in the partitioning of each newly received depth image into multiple smaller depth masks. Each depth mask corresponds to a specific range of depth from the full depth range covered by the depth image. For every mask, the algorithm identifies the center of the free areas and converts these centers from pixel positions within the image to real-world coordinates. After processing all the masks, a path is established, composed of the centers of the free areas. This path is then relayed to the drone for execution. A detailed explanation of this process is as follows:

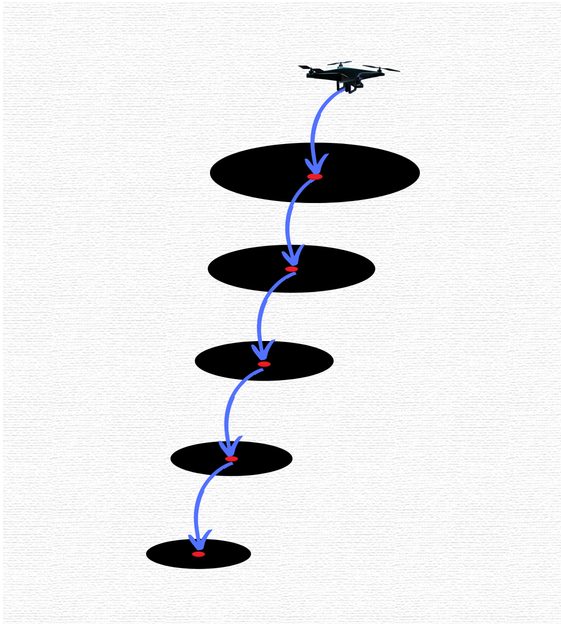


Figure 4.3: Illustration of the Descent Algorithm Concept: The depth image is segmented into multiple smaller depth masks (represented by black circles), each corresponding to a distinct depth range within the full depth range. The red dots signify the center of the free area in each mask, which forms the drone's descent path.

1. **Dividing the Depth Range into Smaller Intervals**

   Our ToF camera provides a depth range of 0.5m to 4m. When visualizing this data using a histogram as shown in Figure 5.14, we observed that closer depths yield more data points due to the occlusion of nearby obstacles. Directly segmenting this depth range into fixed intervals could result in a lack of data points in the farther intervals. To avoid this, we instead segment the full set of data points into multiple equal groups. As a result, each group represents a unique depth interval, as can be observed in Figure 5.15, where each interval is colorized differently. This approach not only ensures equal data distribution across all intervals but also generates multiple smaller intervals in the close range due to the higher density of nearby obstacles. Consequently, waypoints with smaller z-steps can be generated, improving our algorithm's ability to avoid close obstacles.

2. **Generating a mask for each depth interval**

   Upon creating depth intervals, we generate a binary mask for each one. Each mask maintains the dimensions of the original depth image but only includes depth values that fall within the corresponding interval. Within these resultant masks, a binary code is assigned to the pixels: a value of one denotes occupied areas (obstacles), while

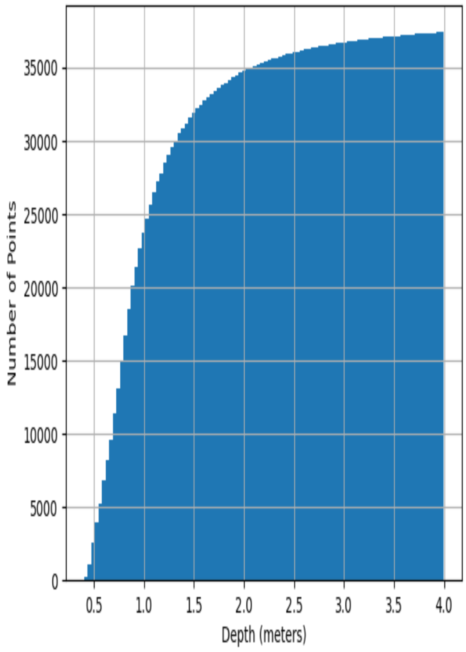


Figure 4.4: Histogram representing the distribution of data points across the depth range.

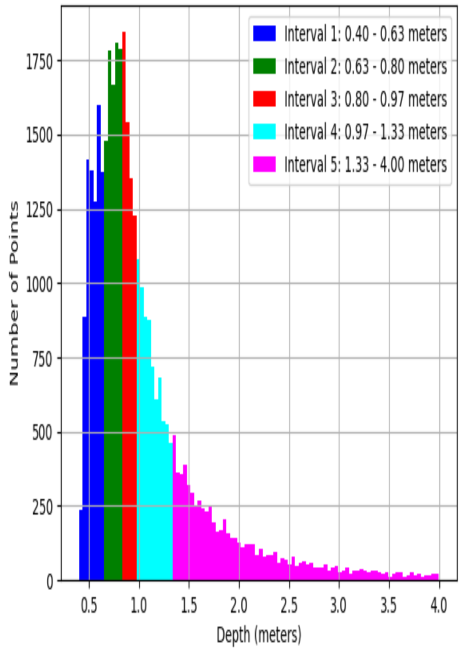


Figure 4.5: Visualization of the depth range divided into multiple equal groups, each represented by a unique color.

a value of zero signifies the free areas specific to that depth range interval. Figure 5.16 provides examples of these masks.

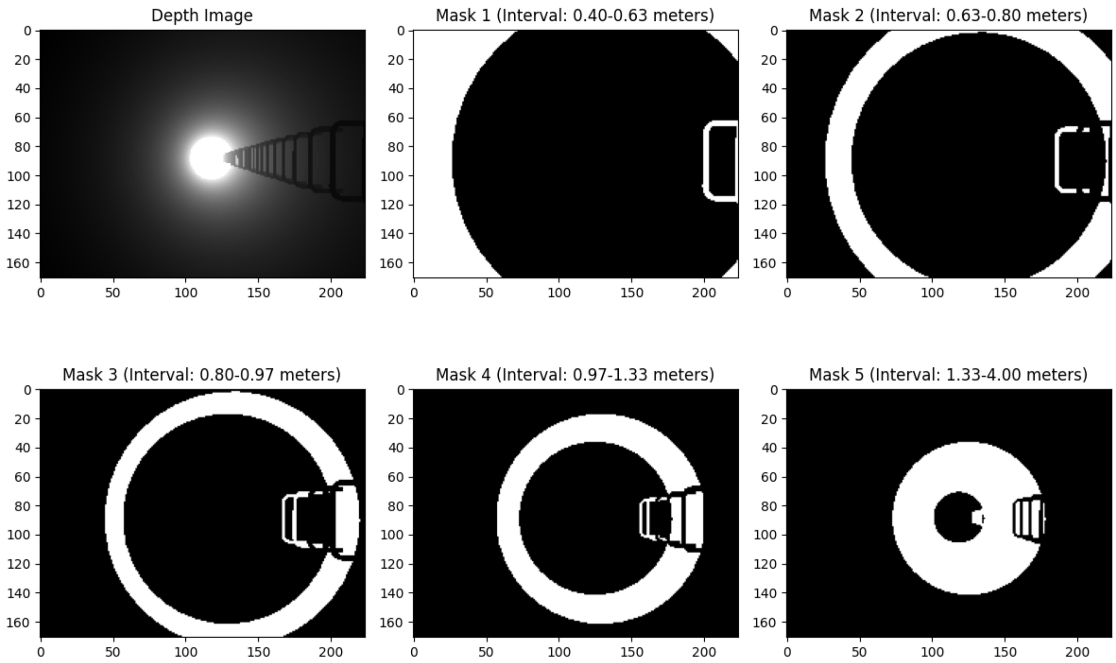


Figure 4.6: Visual representation of binary masks for each depth interval. The original depth image is shown in the first subplot, followed by binary masks for the corresponding depth intervals.

3. **Identifying the free area in each depth mask**

   The first step in this stage involves applying dilation in each depth mask. This operation enlarges the perceived size of the obstacles, which can cause the free area - originally circular in the absence of obstacles due to the manhole's shape - to take on an irregular form. This irregularity arises when an obstacle occupies part of the circular space within the manhole. This is illustrated in Figure 4.7, where the irregularly shaped free area is the black area enclosed by the white boundary.

   To enhance the geometric representation of these masks, we accumulate them. Consequently, each mask includes depth data corresponding to its depth range and all previous ranges. This approach yields more comprehensive depth data, generating a closed geometric representation with a single region signifying the free area. In Figure 4.8a, Mask 3 from Figure 5.16 is illustrated. Here, gaps representing occupied pixels from previous intervals are highlighted in red. However, through accumulation, these gaps are effectively filled, as demonstrated in Figure 4.8b.

   Given the irregularity of the free area, identifying its center requires a different approach. We do this by recognizing all the connected regions in the image and then calculating the distance transform for each region. The point with the maximum

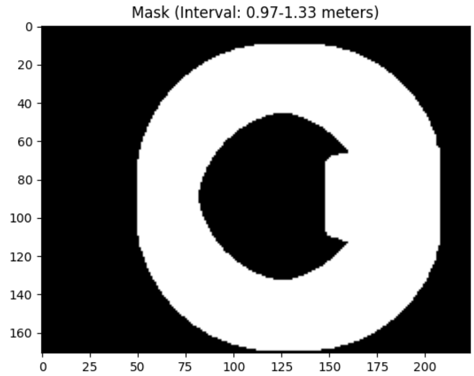
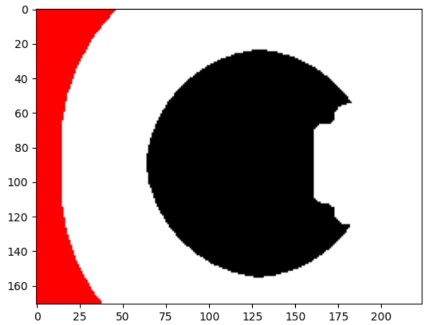
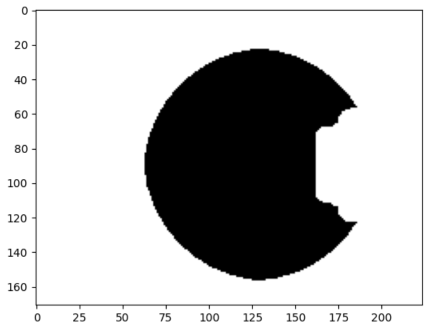


Figure 4.7: An illustration of the irregularly shaped free area after dilation is applied to Mask 4 from Figure 5.16. The free area is the black area enclosed by the white boundary.



(a) Mask 3: white region represents current obstacles; red region identifies previously detected obstacles.



(b) Accumulated mask: combination of depth data from Masks 1, 2, and 3, offering a more comprehensive depth view.

Figure 4.8: The process of generating and accumulating depth masks. Subfigure (a) illustrates an individual depth mask and Subfigure (b) demonstrates the accumulation of multiple depth masks.

distance to the boundary of its connected region is considered the center of the free area. However, as illustrated in 4.9, the resultant center (marked in red) may be misaligned with the actual center of the manhole (marked in blue). This misalignment can cause the ToF camera's reflected rays to only partially capture the manhole, resulting in a partially visualized manhole in the initial depth masks.

Attempting to identify the center of the free area under such circumstances using distance transform could lead to inaccuracies. This is particularly evident in instances like 4.10, where only a part of the manhole is visible, causing the center to be erroneously assigned to an incorrect location, indicated by the red marker in the figure.

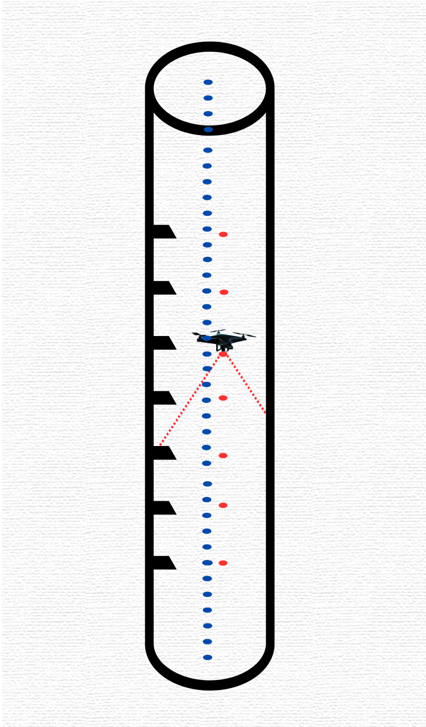


Figure 4.9: Depiction of the drone at the calculated center of the irregular free area (marked in red), which is shifted from the actual center of the manhole (marked in blue). The resulting proximity to the manhole's right side causes the drone's ToF camera to capture a shorter depth on the right, leading to partial visualization of the manhole in the initial depth masks.

To address this issue, centers that may result in errors are excluded. We utilize the center of the entire depth image as a reference and only consider those centers whose distances from the main center fall within a defined threshold, which was empirically determined. These selected centers, originally in pixel coordinates, are then back-projected into real-world coordinates. An adjustment is made for the Z value used in Equation (4.4). Rather than employing zero, we add the mid-point of the depth interval of the mask currently being processed to the drone's current z-depth value, as shown in Equation (4.7).

$$Z = [Z]_B^M + \frac{\text{Interval}_{\text{start}} + \text{Interval}_{\text{end}}}{2} \tag{4.7}$$

It is important to note that simply disregarding depth mask intervals with such centers and directly sending a valid center is not optimal. The suggested center may be significantly distanced from the drone, increasing the risk of collision with a near surface. To mitigate this risk, we transmit the x, and y coordinates of the center along with smaller z steps from the drone's current position to the center depth. This approach is feasible as the center is identified not based on individual masks,

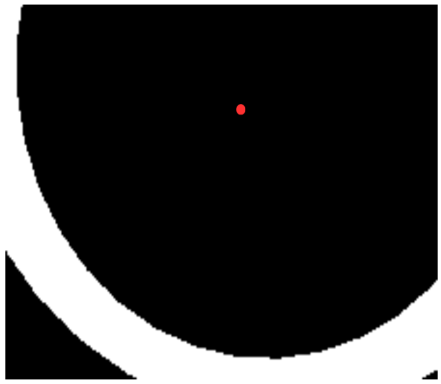


Figure 4.10: An illustration demonstrating the issue with using distance transform for center identification under incomplete manhole visibility. The erroneous center assigned by the distance transform method is marked in red.

but on accumulated masks. As a result, all obstacles up to that interval are taken into account.

Upon determining the center of the free area for all depth interval masks of the depth image, we consider them valid paths for the drone to follow. To facilitate reactive descent, the drone regenerates this path each time it receives a new depth image. The descending process continues until the drone reaches a position directly above the manhole ground, within a certain predefined threshold. Upon reaching this position, the drone is well-positioned to collect data from the sensor located at the bottom of the manhole.

4.4 Evaluation methodology

The algorithms discussed in the previous section form the foundation for the drone's ability to autonomously locate, identify sewer manholes in urban areas, navigate towards them, and safely descend into them. However, their practical utility and effectiveness can only be truly validated through rigorous testing. The subsequent sections present the evaluation methodology employed, outlining the procedures undertaken to assess the drone's performance within both simulated and real-world environments.

4.4.1 Simulation environment

Before actual deployment in real-world conditions, it is vital to validate the performance and reliability of the drone and its associated algorithms within a simulated environment. This step not only helps in mitigating potential risks and failures but also provides a better understanding of the system's capabilities in controlled and consistent settings. The simulation environment created for this purpose comprises three significant components: Creating an urban environment, integrating the sewer manhole models within it, and the refined Iris Quadrotor model to enhance realism and performance testing.

4.4.1.1 Creating urban environment

The first component of our simulation environment is the cityscape, designed with Gazebo software to create a representative urban scenario. As depicted in Figure 4.11, a Gazebo city world was meticulously designed to replicate a typical urban setting, replete with houses, shops, and streets. This design process utilized pre-built models from the Gazebo Library[100], which provided a diverse array of realistic models to construct the simulation setting.

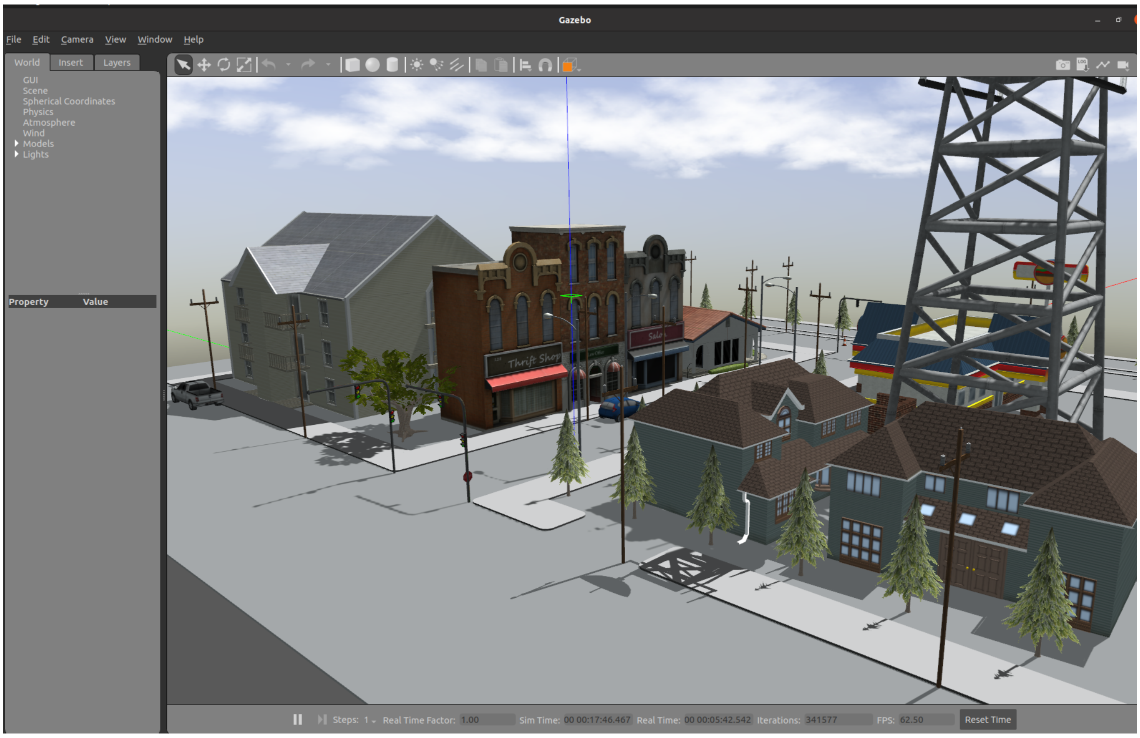
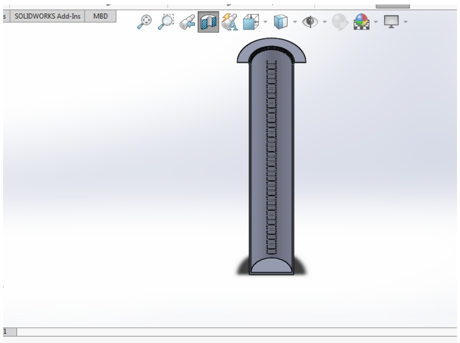
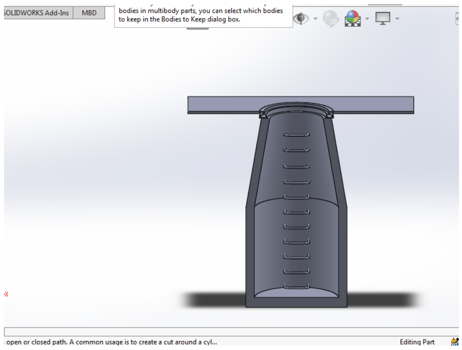


Figure 4.11: Customized Gazebo city world



(a) Manhole with 600 mm diameter



(b) Manhole with 600-900 mm diameter

Figure 4.12: Solids Drawings

4.4.1.2 Designing and integrating the sewer manhole models

In order to test the drone's performance under a variety of conditions and challenges, we incorporated custom sewer manhole models into the simulation. These models, designed in SolidWorks, adhered to standard sewer manhole dimensions, ranging from 600 mm to 800 mm in diameter. We created two types of models to replicate the diversity encountered in real-world scenarios: one type maintained a constant diameter of 600 mm and included a ladder of 140 mm, while the other type showcased varying diameters, with the smallest being 600 mm and also incorporating a ladder. Figure 4.12 provides an illustration of these models.

To successfully incorporate the custom sewer manhole model into the Gazebo city world, the following steps were taken:

1. The SolidWorks model was converted to a Gazebo-compatible format, such as STL (.stl), ensuring smooth integration with the simulation environment and compatibility with the pre-built Gazebo library models.

2. The Blender program was used to accurately define the model's center before exporting it in the (.stl) format, guaranteeing proper positioning and interaction within the city world.

3. The sewer manhole model was integrated into the city world file using appropriate XML tags that specify essential attributes, such as the model's position, orientation, and scale within the urban simulation. This provided a realistic representation and seamless interaction with the Gazebo library models.

4.4.1.3 Refining the Iris Quadrotor model and simulated sewer manhole for improved realsim

The PX4-SITL and Gazebo package includes a 3DR Iris quadrotor model for simulation purposes. The dimensions of the simulated Iris, with a diameter of 665 mm, propellers with a diameter of 256 mm and a height of 110 mm, are larger than the actual drone utilized in this project. To maintain the real-world proportions between the drone and the sewer manhole, the diameter of the simulated sewer tunnel and the ladder width were adjusted accordingly.

Furthering this pursuit of realism, four cameras were integrated into the Simulation Description Format (SDF) file of the customized Iris model. The position, orientation, and distinct properties of each camera such as image format, resolution, lens intrinsic, distortion parameters, and noise characteristics, were meticulously defined in the SDF file.

Incorporating noise characteristics into the simulation was essential to provide a more realistic representation of the camera's performance, accounting for potential real-world imperfections and uncertainties like varying illumination conditions. This inclusion of simulated noise enabled a more accurate evaluation and improvement of the drone system's robustness.

To facilitate the usage of the simulated camera's data for further analysis, processing, visualization, or control within the ROS Noetic environment, a Gazebo plugin was configured to establish a connection between the simulated camera and the ROS environment.

These adaptations significantly enhanced the simulation environment, allowing for a more representative depiction of the drone's performance. Consequently, it facilitated more effective testing and development, assuring that the drone's behavior in the simulation closely mirrored its real-life counterpart. This methodology successfully addresses the third research question, proving that with careful design and customization, mission simulations can ensure seamless real-world transferability of tested procedures.

4.4.2 Real-World environment

In order to ensure the robustness of the algorithm and to validate its performance under real-world conditions, extensive tests were conducted in various environments.

4.4.2.1 Real-World test site description

An initial site visit was conducted to a sewer manhole in Girona, Spain. During this visit, data from the drone's cameras were collected and stored in a ROS bag for subsequent analysis. The sewer manhole observed during this visit can be seen in Figure 4.13.

Figure 4.13: The sewer manhole in Girona, Spain

To facilitate a comprehensive testing regime, a dedicated test site was established within the premises of the Eurecat Robotics Company, located in Cerdanyola, Spain. This setup involved the installation of two cylindrical tubes each with a diameter of 600mm and height of 1500mm, thereby emulating the physical characteristics of an actual sewer system. This arrangement is depicted in Figure 4.14.

Figure 4.14: The cylindrical tubes installed in the lab at Eurecat Robotics Company

Additionally, to aid the visual detection tasks, a printed image (of 600mm diameter) representing the top view of a manhole was stationed within the lab, as shown in Figure 4.15.

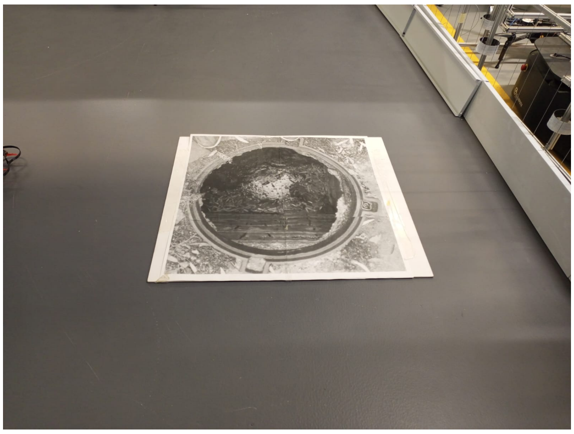


Figure 4.15: The printed image of a manhole used for testing

Chapter 5

Results

This chapter presents the findings from the various tests conducted in both simulation and real-world environments. The results are analyzed to ascertain the strengths and limitations of the system and to propose potential improvements.

5.1 Simulation test results

This section presents the results of the simulation tests and discusses their implications. The simulations were conducted in the customized Gazebo city world with the addition of the sewer manhole as shown in Figure 5.1.

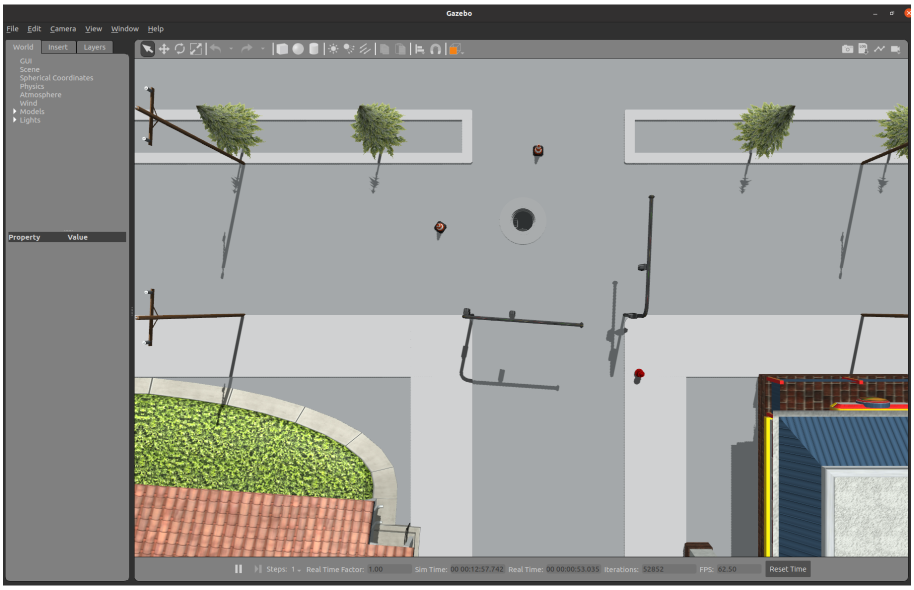


Figure 5.1: Customized Gazebo city world with the addition of the sewer manhole model at street intersections

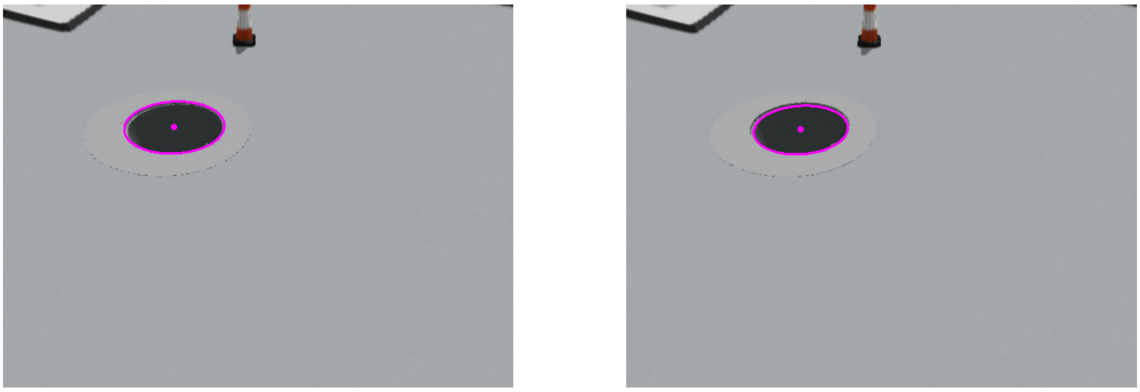
5.1.1 Phase 1 - Autonomous sewer manhole localization and navigation

To assess the efficacy of the implemented algorithms in a simulated environment, a series of tests were executed as follows:

- **Test 1.1: Ability of detection algorithms to identify sewer manhole**

  In this test, the drone's ability to autonomously detect a sewer manhole was evaluated using two distinct detection algorithms. These algorithms were applied to images captured by the drone's tracking camera in a simulated environment, with each image processed using dedicated ROS nodes. The accuracy and reliability of the algorithms in identifying the manhole and its center were assessed, with results visualized via an OpenCV window.

  The results of Test 1.1 are presented in Figure 5.2. Both algorithms performed well in the simulated environment, as the manhole's elliptical shape was well-defined and unobstructed, and there were no other objects of a similar shape that could cause confusion.



(a) Output of the first detection algorithm on a sample image from the simulated environment

(b) Output of the second detection algorithm on a sample image from the simulated environment

Figure 5.2: Comparison of the output from two detection algorithms in the simulated environment

The key performance metrics for the two algorithms are accurately summarized in Table 5.1. It can be observed that the second algorithm demonstrated a notable increase in processing speed over the first one by 66%. This boost in performance makes it a more attractive option for real-time applications, especially when paired with powerful processing hardware. Nevertheless, the trade-off comes with its higher memory utilization, which may pose challenges in constrained hardware setups or longer-duration missions where power efficiency is critical.

Algorithm	Processing Speed (fps)	Memory Consumption (GB)
Algorithm 1	22	0.00487
Algorithm 2	36	0.109

Table 5.1: Comparison of key performance metrics for the two detection algorithms

- **Test 1.2 Navigation accuracy following coordinate conversion**

  This test evaluates the ability of the back-projection algorithm to convert the detected center of the manhole from pixel coordinates to real-world coordinates for navigation purposes. Subsequently, the real-world coordinates generated are used as waypoints for the drone's autonomous navigation. The drone is launched from various starting positions, and its trajectory toward the manhole center is logged using a ROS bag. This procedure aids in verifying whether the drone's endpoint of the path aligns with the known position of the sewer manhole within the simulated world, thereby evaluating the algorithm's performance under different starting conditions.

  This was validated through trajectory data from a ROS bag, revealing the drone's successful approach towards the actual center of the manhole with minimal deviation - just 9 mm in the x-direction and 20 mm in the y-direction (see Figure 5.3).

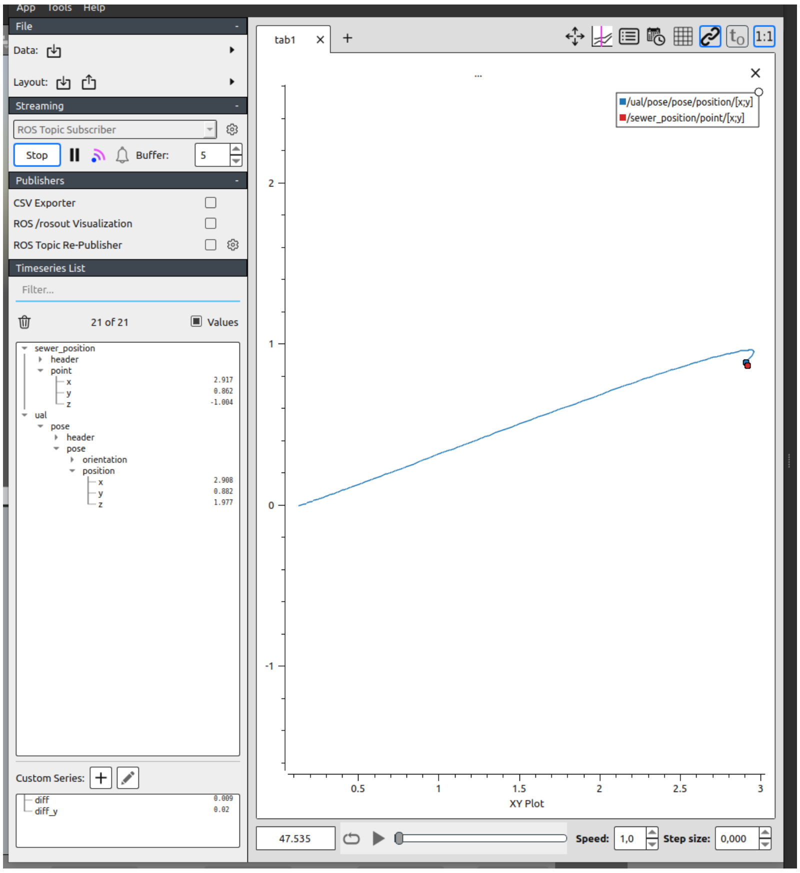


Figure 5.3: Drone's trajectory illustrating its approach to the actual center of the manhole

We then analyzed the performance of the back-projection algorithm under varying initial drone positions within the world frame. The sewer manhole's real-world coordinates were noted at position (46.0264 m, 1.44992 m), and associated errors were meticulously scrutinized. Both detection algorithms contributed to the error plots shown in Figures 5.4 and 5.5.

A significant source of error, common to both scenarios, is the undistortion process. Despite utilizing the camera's intrinsic parameters to correct the inherent distortion caused by the fisheye lens, it doesn't always achieve flawless results, resulting in residual distortion.

This residual distortion becomes more pronounced as the drone's distance from the manhole increases, leading to the manhole's image appearing as an elongated ellipse towards the image edge. This distortion is especially noticeable when the drone's initial position diverges considerably from the actual manhole center.

In the first scenario, a notable error was observed when the y-coordinate of the drone's initial position deviated significantly from the actual manhole center, while the x-coordinate remained constant. However, the second detection algorithm improved the fitting of ellipses to distorted manhole images, resulting in more accurate real-world coordinate representations and smaller errors. Significantly, when the drone's y-coordinate stayed within a 1.5 m range on either side of the actual manhole center, the error stayed within the acceptable limit of 100 mm (see Figure 5.4).

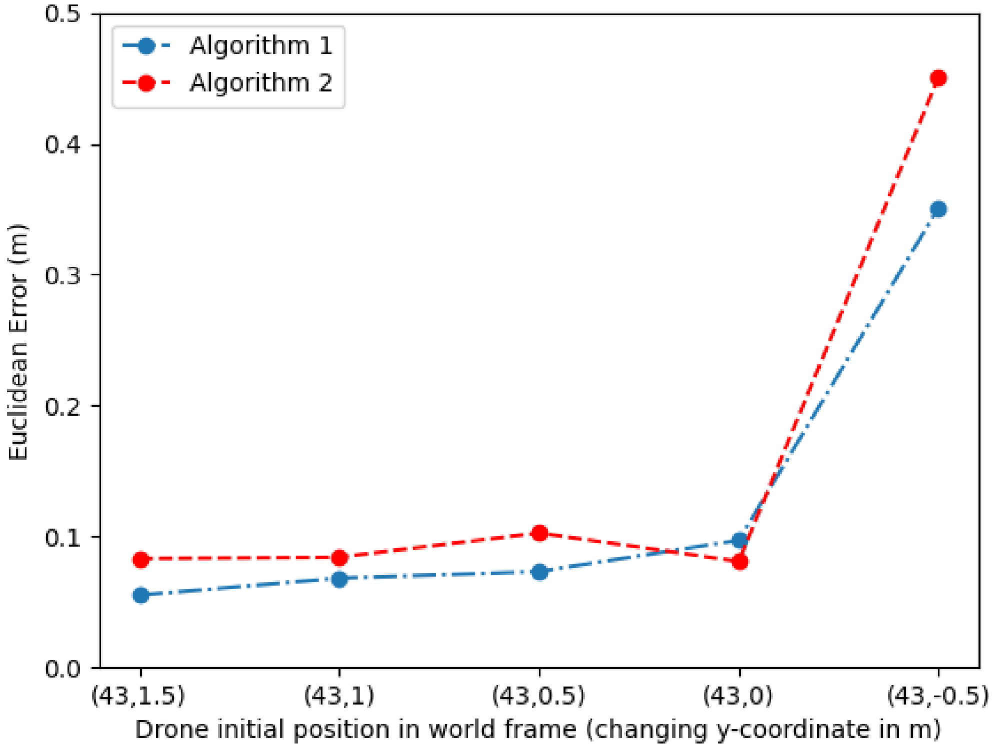


Figure 5.4: Error in the drone's position with varying y-coordinate

In the second scenario, while keeping the y-coordinate constant, the x-coordinate of the drone's initial position was varied. The error increased when the drone was either too close or too far from the manhole center. However, the second detection algorithm's refined ellipse fitting led to more accurate manhole center detection and smaller errors, surpassing the performance of the first algorithm. When the drone's x-coordinate was within a 1-4 m range from the actual manhole center, the error was less than 100 mm, indicating satisfactory performance within this range (see Figure 5.5).

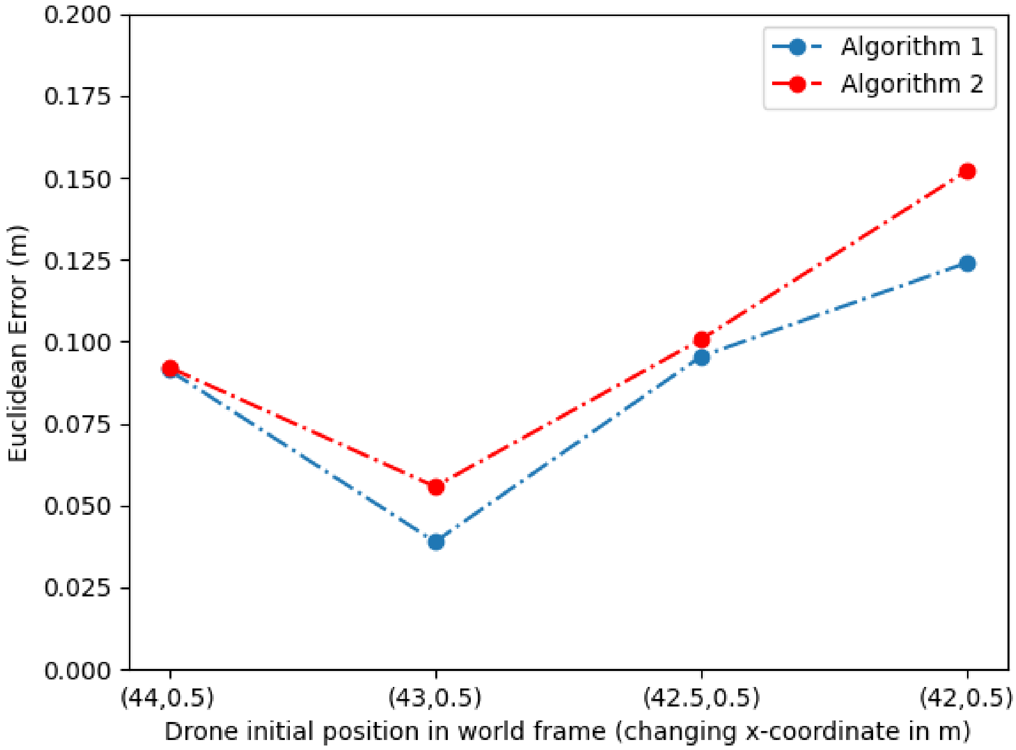


Figure 5.5: Error in the drone's position with varying x-coordinate

An error within 100 mm is considered acceptable, as this would still allow the drone to remain over the manhole, even if its center is slightly shifted. Any discrepancy in alignment can be corrected during the descending phase of the drone's operation.

All measurements were conducted at the drone's default takeoff altitude for this mission, set at 2m. It's important to note that the altitude, can significantly influence the algorithm's performance. Changes in altitude directly affect the perspective from which the drone's camera views the scene, as well as the resolution of the captured image.

For instance, when the drone is flying at a low altitude and is significantly far in the x-coordinate from the manhole, the manhole might not be visible in the captured image due to the limited field of view. On the other hand, if the drone is at a low altitude and is very close to the manhole, only a part of the manhole might be visible in the image. These scenarios, among others, would naturally affect the ability of the algorithm to accurately detect the manhole and determine its center.

This underscores the necessity for careful consideration of the drone's initial altitude and position to ensure the entire manhole is within the camera's field of view. This would facilitate accurate manhole detection and subsequent center determination.

The complexity of this problem extends beyond mere coordinate conversion and requires a thorough understanding of diverse positional and environmental factors.

5.1.2 Phase 2 - Descending through the manhole

- **Test 2.1 Autonomous descent through the manhole:**

  In this test, the drone was tasked to descend autonomously through the sewer manhole. This phase was initiated after the completion of Phase 1, with the drone positioned above the manhole. The drone was tested in simulated sewer manholes of different lengths, allowing an assessment of the algorithm's performance and stability under realistic and challenging conditions.

  Initially, a ROS bag was recorded while the drone descended through a 4.5 m sewer manhole. The drone was positioned near the center of the manhole, with coordinates (2.85, 1.05) m in the map frame and an altitude of 0.5 m. The path generated by the drone is depicted in Figure 5.6, which comprises three subfigures.

  The left half of Figure 5.6 (a) showcases the 3D trajectory of the drone, with the trajectory itself representing the drone's center. The drone, in our simulation environment, has a diameter of 921 mm. Given a manhole diameter of 1260 mm and a ladder width of 215 mm, the available movement space for the drone's center narrows to only 120 mm. This region defines the safe zone within which the drone must navigate to avoid a collision. Despite these constraints, the drone successfully adheres to the safe zone, avoiding any contact with the manhole boundaries and the ladder. Notably, prior to the drone entering the manhole, the algorithm guides it towards a waypoint positioned relatively close to the center of the manhole's free area. This movement is noticeable as the drone transitions from its starting position to the center of the safe zone.

  For a detailed evaluation of the drone's navigational accuracy, we examined the drone's positional variation along the x and y axes during the descent.

  The top right quarter of Figure 5.6 (b) illustrates the drone's position along the x-axis throughout the descent. At one point, the drone came close to the boundary but remained within the safe zone. This was possible because only the walls of the manhole served as obstacles on this axis, which results in an elongated safe zone in the x direction due to the exclusion of the ladder width.

  The bottom right quarter of Figure 5.6 (c) demonstrates the fluctuations in the drone's position along the y-axis during the descent. This axis aligns with the ladder.

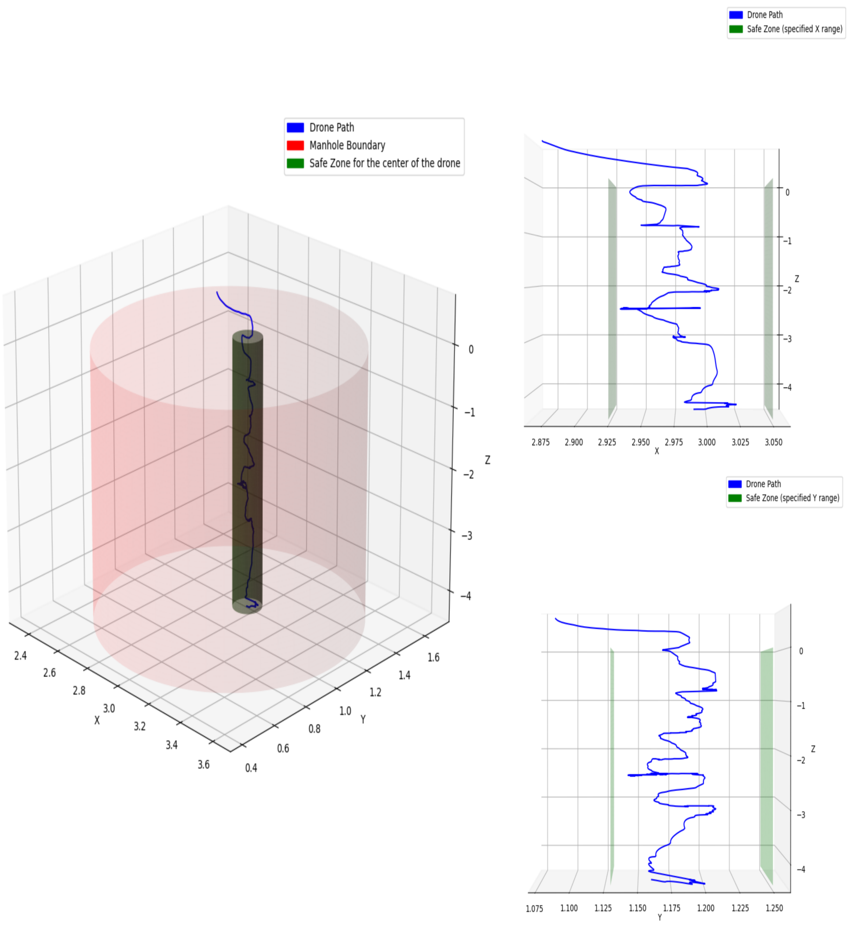


Figure 5.6: Composite figure showing the 3D path of the drone and the variation along the X and Y axes. The left half of the figure shows the 3D Path (a), the top right quarter shows the X-Axis Variation (b), and the bottom right quarter shows the Y-Axis Variation (c).

The variations in the drone's y-coordinate stemmed from our algorithm's process of determining the free center of each mask derived from the depth image. During instances where the ladder was not included in the mask, minor adjustments were made to the drone's y-coordinate. Despite these adjustments, the drone consistently remained within the safe boundaries throughout the descent.

As previously discussed, in Phase One there can be instances where the generated center might deviate from the actual center of the manhole. However, this doesn't pose an issue due to the efficiency of our descending algorithm. This algorithm is designed to generate a valid path as soon as the drone's downward-facing TOF camera comes within the boundaries of the manhole.

we placed the drone at the starting position (43, -0.3)m in the world frame. This position was identified in **Test 1.2 Results and Discussion** as causing a deviation between the generated center and the actual center of the manhole.

The center generated during the first phase was located at (3.273, 2.064)m with an altitude of 1m in the map frame. On the other hand, the actual center was found to be at (2.985, 1.750)m in the same map frame. As shown in Figure 5.7, after reaching the generated center, part of the drone was found to be outside the manhole in the Gazebo simulation.

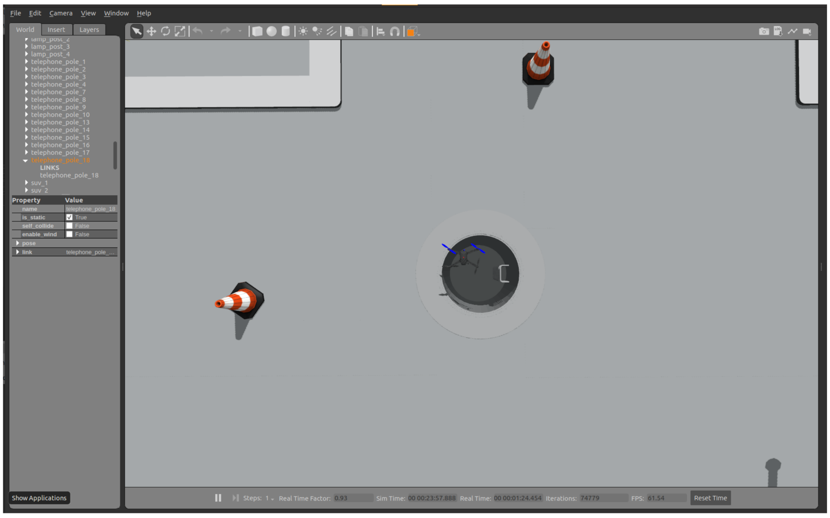


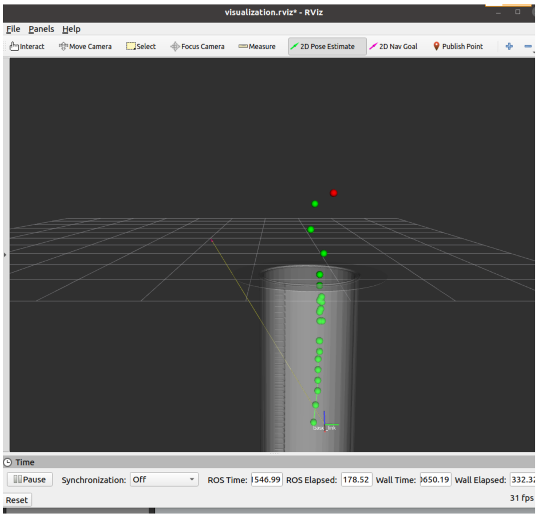
Figure 5.7: The drone's position shows a deviation from the manhole center.

Upon reaching the generated center, the descending algorithm was initiated. We tracked the generated path using RViz and PlotJuggler (Figures 5.8a and 5.8b).

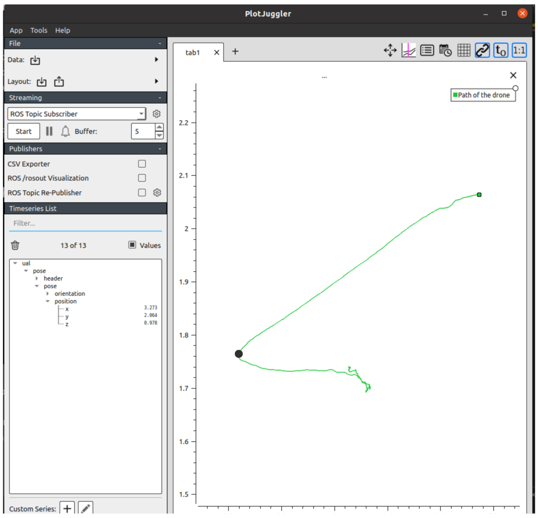
Despite the initial deviation, where the initial masks might only represent a portion of the manhole, the algorithm succeeds in first generating a waypoint at (2.82, 1.754)m—very close to the center of the manhole—marked in black in subplot 1. This is possible because our approach cumulatively considers these depth masks until a mask representative of the manhole's geometry is found. The center of this area, while accounting for all previously identified obstacles, is then sent as a waypoint to the drone.

The drone reacts to this waypoint by incrementally descending towards it, starting from its current altitude and ending at the altitude corresponding to the waypoint. Crucially, with our approach of generating a new path each time a depth image is received, the algorithm has generated an additional waypoint close to the center of the free area. This occurs before the drone enters the manhole. Such a strategy is essential to circumvent any possible collisions with obstructions like the ladder, which may not be visible in the initial depth image.

Subplot 2 presents an additional visualization using RViz to underline the success of our algorithm. The red ellipse marks the initial position, while the green ellipses illustrate the waypoints generated by our algorithm.



(a) Visualization of the drone's path using RViz.



(b) Visualization of the drone's path using PlotJuggler.

Figure 5.8: Visualization of the drone's navigational path using two different tools.

In another scenario, we evaluated our drone's descending capabilities within a longer, 6-meter sewer manhole. We leveraged RViz for visualizing the path generated by the drone throughout its descent. As illustrated in Figure 5.9, the drone's path is represented by a sequence of green ellipses. Each ellipse corresponds to a waypoint produced by our algorithm. In addition, the green line indicates the path generated by our algorithm for each incoming depth image. Consequently, this line depicts the most recent depth image path.

Before initiating the descending algorithm, it is notable that the drone was positioned almost centrally within the manhole's free area with 0.5 altitude above the manhole. The illustration in Figure 5.9 clearly demonstrates the adherence of the drone's path to this central area during the course of the descent, showcasing the effectiveness of our algorithm in guiding the drone safely down the manhole.

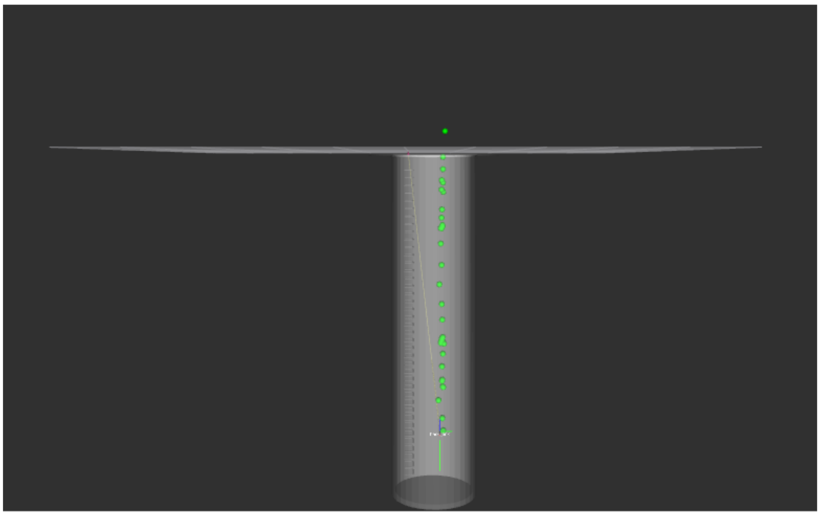


Figure 5.9: Visualization of the drone's path in a 6-meter sewer manhole using RViz. The green ellipses represent waypoints generated by our algorithm, while the green line represents the path published by our algorithm for each new depth image.

5.2 Real-World test results

5.2.1 Phase 1 - Autonomous sewer manhole localization and navigation

A series of tests were performed to evaluate different aspects of the drone's performance in real-world:

- **Test 2.1 Drone control evaluation**

  The drone's autonomous control capabilities of autonomous arming, takeoff, and landing are tested in the lab to confirm operational readiness.

  The results of Test 2.1 revealed inconsistent outcomes during the autonomous takeoff phase. The drone managed a controlled ascent in some instances, while in others it abruptly rose and rapidly descended, ultimately leading to a crash.

  The primary cause for these erratic behaviors can be traced back to the drone's visual odometry system. This system, designed to track visual features in the environment to estimate the drone's position and orientation, struggled during the ascent phase.

  Specifically, when the drone was positioned over a uniformly patterned area or featureless ground, the 45-degree tilted tracking camera faced regions with insufficient visual features. This deficiency hindered the visual odometry system from accurately estimating the drone's position and orientation. Consequently, the drone was led to take corrective maneuvers based on inaccurate data.

For instance, if the visual odometry system failed to register the drone's upward movement due to the lack of distinct features, it could mistakenly perceive that it was not gaining altitude. This misinterpretation could trigger an overcorrection, resulting in an abrupt, uncontrolled ascent. The erroneous positional data could then generate a feedback loop, causing a rapid descent and subsequent crash.

In Figure 5.10 the altitude data for one of these scenarios is presented. It vividly shows the sudden spike in altitude followed by a sharp descent, with subsequent fluctuations due to the crash impact.

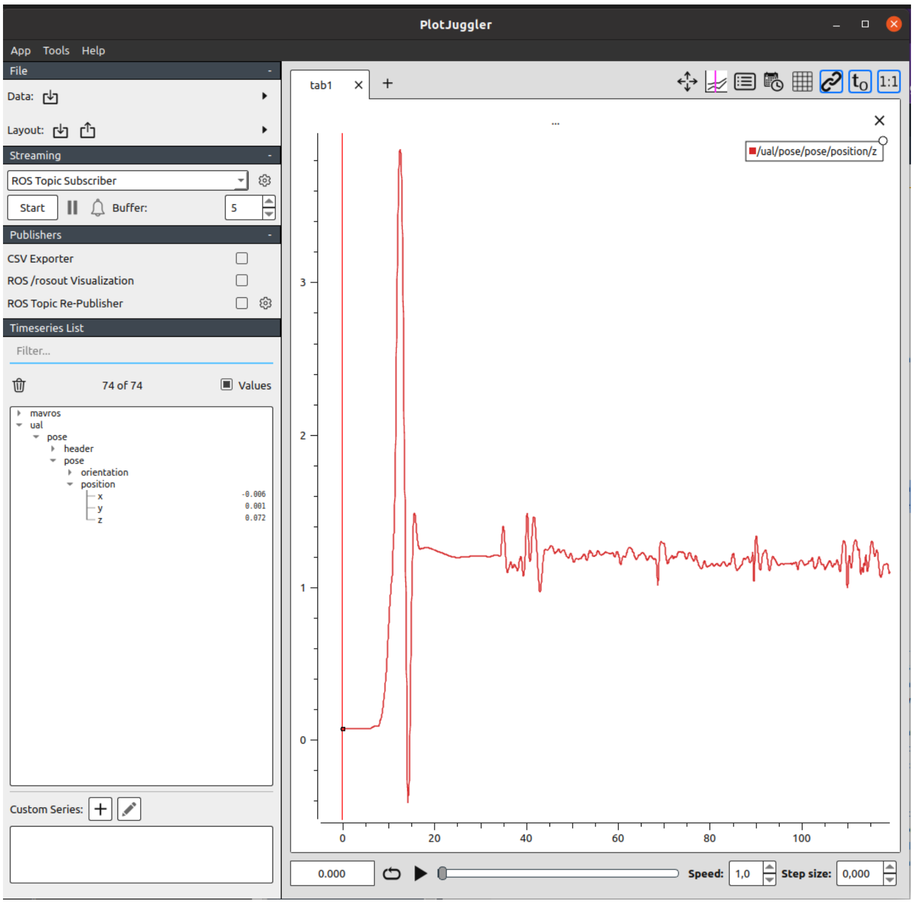


Figure 5.10: Drone altitude data during a failed takeoff attempt

These findings underscore the limitations of relying solely on visual odometry, particularly in environments featuring uniform or repetitive patterns. The outcomes also suggest that performing further autonomous tests solely relying on this method could pose considerable risks, as it increases the likelihood of crashes. These insights are valuable in addressing the second part of our research question - "how can we mitigate typical errors such as localization noise and drift?" By emphasizing these risks, we highlight the need for a more comprehensive approach for a more robust solution to localization challenges.

- **Test 2.2 Assessment of detection algorithms with pre-recorded ROS bag data**

  In this test, the performance of the detection algorithms is evaluated using pre-recorded ROS bag data from the tracking camera aimed at the Girona manhole. The detection results are visualized using an OpenCV window.

  The first algorithm had difficulty detecting the manhole due to partial occlusion by grass. This manhole, located in a grassy area, had a portion of it obscured, thus offering fewer clear edges for the algorithm to latch onto. Although our algorithm is designed to detect well-defined elliptical shapes and can fill small gaps in instances of minor occlusions, the degree of occlusion presented in this case was beyond its capacity. Its reliance on clear, defined edges for accurate detection became a limiting factor in this scenario, causing it to fail in identifying the sewer manhole.

  Figure 5.11 below illustrates this difficulty, with the manhole going undetected by our algorithm due to the grass occlusion.

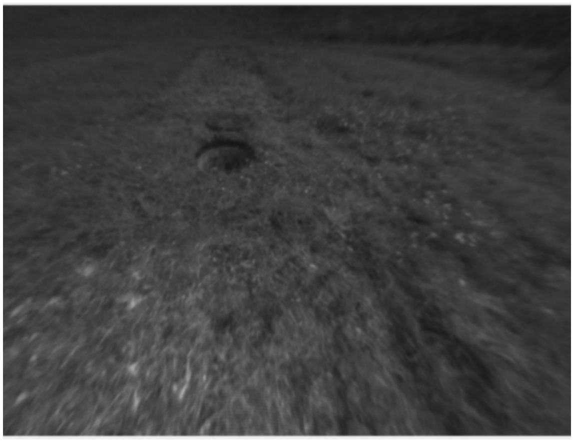


Figure 5.11: Output of the first detection algorithm showing failure to detect the occluded manhole

Contrastingly, the second algorithm successfully detected the manhole despite the partial occlusion. This algorithm employs a more advanced technique, which includes arc extraction and ellipse validation steps that enable it to handle cases where the target object, in this instance the manhole, is somewhat obscured. Specifically, in the arc extraction step, the algorithm extracts arcs based on their convexity and curvature, allowing it to continue to detect parts of the manhole even when parts of it are obscured. In the ellipse validation step, the algorithm uses a set of sampling points and four indexes to validate the shape of the fitted ellipse. This helps to confirm the detection of the manhole even when parts of it are obscured. These techniques together enable the second algorithm to successfully identify the occluded manhole.

Figure 5.12 below demonstrates the successful detection by the second algorithm.

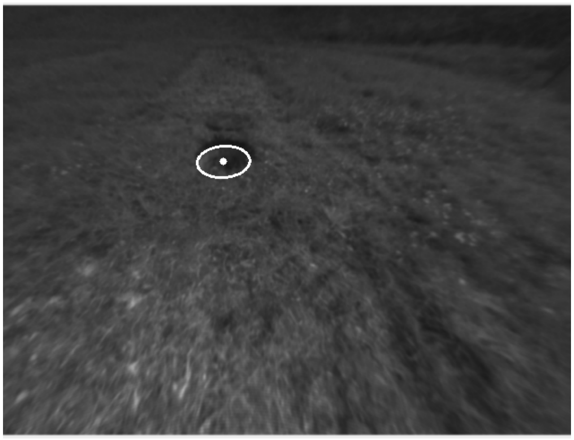


Figure 5.12: Output of the second detection algorithm successfully identifying the occluded manhole

The results from Test 2.2 provide significant insights into the robustness and adaptability of the two algorithms under varying conditions. The first algorithm, while efficient in scenarios where the manhole is mostly unobstructed, demonstrates limitations when the manhole is partially occluded. On the other hand, the second algorithm shows greater resilience in handling complex detection scenarios, such as those with partial occlusion. This resilience suggests its potential suitability for sewer manhole detection applications, effectively addressing the first research question on how sewer manholes can be accurately detected.

- **Test 2.3 Comprehensive assessment of phase 1**

  This test involves a comprehensive evaluation of Phase 1. The drone is tasked with autonomously taking off, detecting the printed manhole image, identifying the real-world center coordinates of the detected manhole, navigating towards it, and finally landing on top of it.

  Figure 5.13 visualizes the outcomes of Test 2.4, showing the test setup on the left and two result plots on the right. The testing environment was augmented with ArUco markers and other objects to enrich the visual odometry, improving the features available for the drone's navigation system and contributing to the success of the tests, as shown in Figure 5.13 (left).

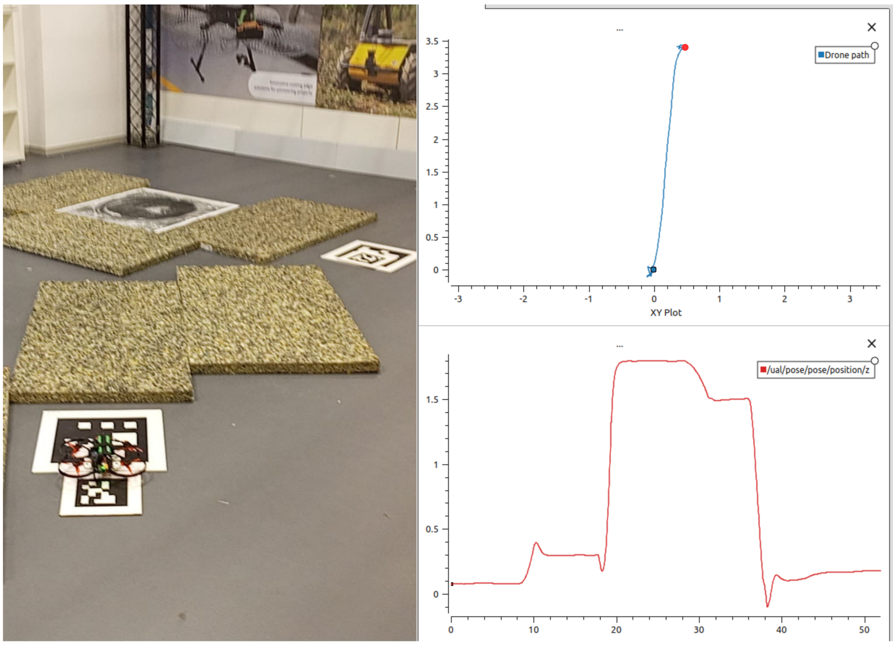


Figure 5.13: (Left) Test 2.4 setup showing the drone, printed manhole, and ArUco markers. (Top Right) Drone's navigation path in the x,y-plane. (Bottom Right) Altitude plot of the drone's flight.

As seen in the x,y-path plot in Figure 5.13 (top right), the drone navigates directly towards the manhole after detecting it and identifying its real-world center coordinates. The effectiveness of the algorithm is evident as the drone accurately reaches the real center, marked in red. This highlights the success of converting pixel coordinates to real-world coordinates.

The altitude plot in Figure 5.13 (bottom right) depicts the phases of the drone's flight: a sharp increase in altitude corresponds to the take-off, a nearly constant altitude is maintained during navigation towards the manhole center, and a controlled decrease in altitude occurs during the landing phase.

The outcomes of this operation affirm the effectiveness of Phase 1. The drone capably executed autonomous take-off, manhole detection and center identification, navigation, and precision landing. This achievement successfully addresses our first research question - "Manhole Detection and Navigation: What is the most suitable algorithm for detecting manholes in the variable environments of sewers, and how can image detection be translated into real-world drone navigation commands?" These findings substantiate the aptness of detecting manholes as ellipses using the AAMED algorithm in diverse sewer environments. Additionally, they illustrate how the detection

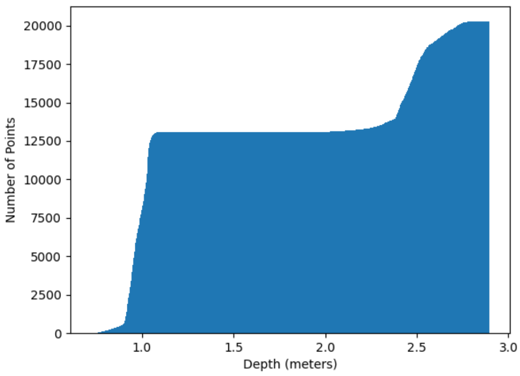


Figure 5.14: Histogram representing the distribution of the real ToF data points across the depth range.

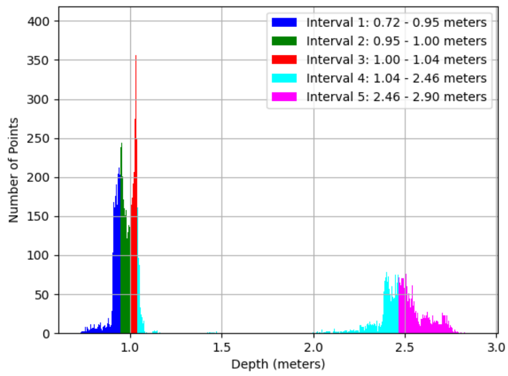


Figure 5.15: Visualization of the depth range divided into five equal groups, each represented by a unique color.

of a center in an image can be translated into practical drone navigation commands using the back projection algorithm.

5.2.2 Phase 2 - Descending through the manhole

In this phase, we tested the drone under real-world conditions to demonstrate the effectiveness of our algorithm. Visual odometry, often used for localization, was found to be unreliable for our needs. Consequently, we decided to operate the descent algorithm concurrently with a manual drone descent, extending one meter into the tube. This methodology enabled us to visualize and compare the actual path of descent with the path generated by our algorithm. Nevertheless, it's crucial to note that the outcomes could be affected by potential inaccuracies due to the manual nature of the descent and possible initial misalignment with the tube's center.

We recorded both the real path taken during the manual descent and the path generated by the algorithm for further comparison. Additionally, we collected Time-of-Flight (ToF) data throughout this process to gain more insights.

We started by illustrating several steps of our algorithm applied to the actual depth data. Initially, we generated a histogram from a depth image taken at the beginning of the descent (see Figure 5.14). This histogram provides a comprehensive view of the depth values in the image. Furthermore, we segmented the entire set of data points into five equal groups, each group denoted by a distinct color in a separate graph (refer to Figure 5.15). This segmentation demonstrated that the range of depth is more confined at closer intervals and more extensive at farther intervals, a consequence of the occlusion of nearby obstacles.

We then partitioned the depth image into multiple masks based on the generated depth intervals. As demonstrated in Figure 5.16, even though the attached surface is flat, Masks 1 and 3 show different depth levels due to the drone's tilt during the descent.

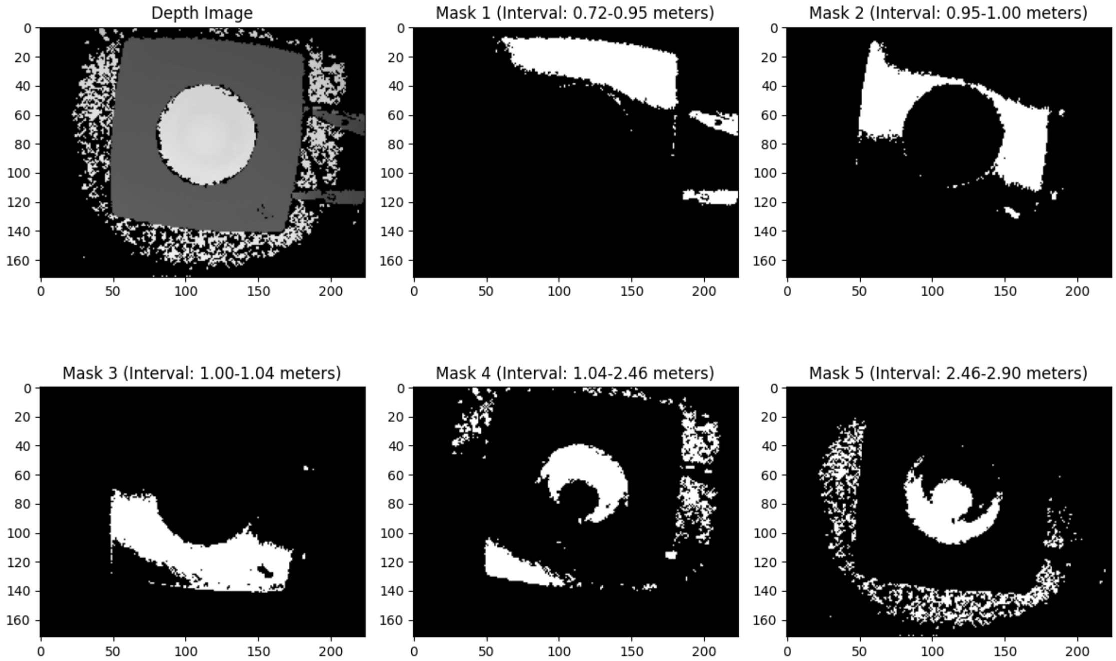


Figure 5.16: Visual representation of binary masks for each depth interval. The real depth image is shown in the first subplot, followed by binary masks for the corresponding depth intervals.

Our analysis progressed by visualizing the centers generated by our algorithm for multiple masks obtained at different depths during the descent (Figure 5.17). The red dot in these images represents the center calculated by our algorithm, and the green dot denotes the actual center of the tube. In most cases, the calculated center was close to the actual center, with identical centers in some cases. For instances where the mask does not fully represent the shape geometry, such as in Mask 2, the algorithm successfully computes the center by relying on the accumulated mask from previous intervals.

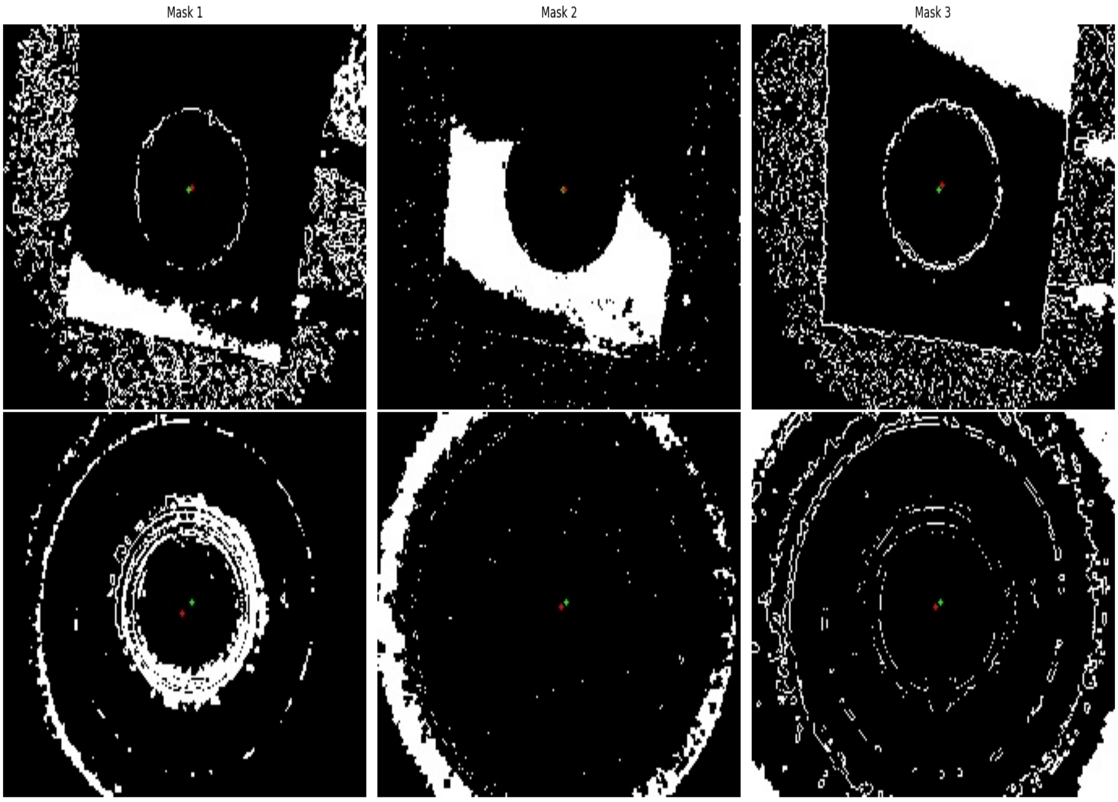


Figure 5.17: Visual representation of algorithm-generated centers (red dots) and actual tube centers (green dots) at varying depth intervals.

Lastly, we rendered a 3D plot to visualize both the paths followed during the manual descent and those generated by the algorithm. These paths were recorded when the drone's initial position was approximately 300 mm above the tube. It is important to note that the drone's initial position exhibited a significant shift in the y-direction. However, in the x-direction, it was nearly aligned with the center of the manhole.

As depicted in Figure 5.18, the y-positions of the generated centers were closely aligned with those of the actual path. However, a noticeable discrepancy was observed between the x-positions of the generated centers and the actual path, especially outside the manhole, with the discrepancy being less than 100 mm. Given the absence of a ladder in the tube, the drone remained within the safe zone despite this discrepancy. This discrepancy could be attributed to inaccuracies during manual descent and potential misalignment of the initial position with the actual center of the tube.

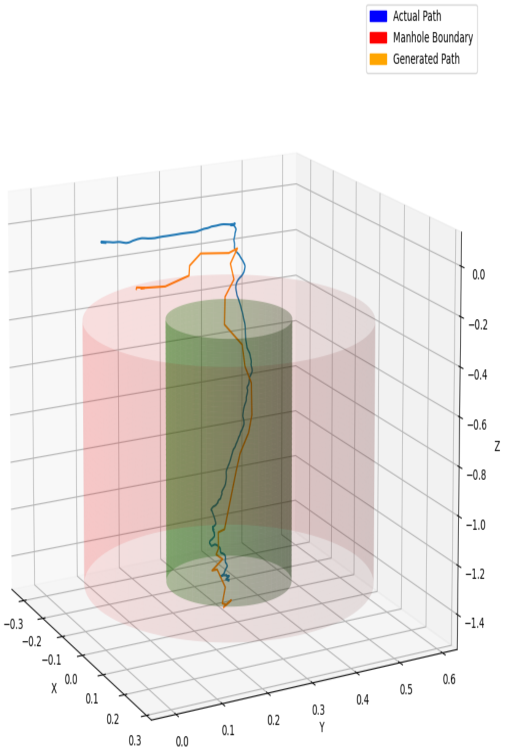


Figure 5.18: 3D visualization of the path followed during the manual descent (blue) and the one generated by the descending algorithm (orange). The green cylinder represents the safe zone of the drone's center, while the red cylinder represents the tube boundary.

The visualization of the generated centers in the images and the results from this test validate the concept of our algorithm. By demonstrating its performance in real-world conditions, we have provided evidence of its resilience and effectiveness for the intended application.

This phase's outcomes provide evidence of the algorithm's reliability and its potential applicability in similar environmental conditions, thus answering our second research question affirmatively.

The outcomes of this phase offer substantial evidence regarding the reliability of our descending algorithm and its potential applicability in similar environmental conditions. These findings positively address the first part of our second research question - "Which algorithm can ensure a safe, controlled descent of the drone given the onboard sensors?" Indeed, our uniquely developed descending algorithm, which relies solely on a single downward Time-of-Flight (ToF) camera, exhibits considerable promise for ensuring a safe, controlled descent of the drone, demonstrating robust performance even under real-world conditions.

Chapter 6

Conclusion

In addressing the daunting task of achieving effective autonomous drone navigation for sewer inspections, this research examined three fundamental questions: optimal manhole detection and navigation, the development of a reliable descent and localization mechanism, and the transferability of simulated operations to real-world scenarios.

Firstly, the AAMED algorithm was identified as a suitable method for detecting manholes in diverse sewer environments. Successful autonomous take-off, manhole detection, center identification, navigation, and precision landing demonstrated how the detection of a center in an image can be translated into practical drone navigation commands using the back projection algorithm. This finding effectively addressed our first research question, validating the application of the AAMED algorithm in real-world scenarios.

Secondly, the developed descending algorithm, which relies on a single downward ToF camera, demonstrated its potential in ensuring a safe, controlled descent of the drone. The reliability of this algorithm, tested under real-world conditions, satisfied the first part of the second research question. However, our research also underlined the limitations of sole reliance on visual odometry, particularly in environments with uniform or repetitive patterns. This insight underscores the need for a more comprehensive approach, possibly involving additional sensors or the use of ToF data for localization, to enhance system robustness and minimize the likelihood of unexpected flight behaviors.

Lastly, careful design and customization of the simulation environment facilitated more effective testing and development, ensuring the drone's behavior in the simulation closely mirrored its real-life counterpart. This approach demonstrated that mission simulations can ensure seamless real-world transferability of tested procedures, successfully addressing the third research question.

While this study provides promising solutions, it is important to note that the challenges of sewer inspections using autonomous drones are vast and interconnected. The remaining challenges - including localization, communication, environmental factors, and maneuvering through intricate sewer systems - require further research. The insights and methodologies derived from this research, nevertheless, offer a significant step towards a comprehensive autonomous drone system for sewer inspections. We hope our contributions will spur further innovations in the realm of autonomous drones for sewer inspections, revolutionizing the management and maintenance of essential urban infrastructures.

Bibliography

[1] T. Özaslan et al. "Inspection of penstocks and featureless tunnel-like environments using micro UAVs". In: *Field and Service Robotics: Results of the 9th International Conference*. Vol. 105. STAR. 2015, 123–136.

[2] T. Özaslan et al. "Towards Fully Autonomous Visual Inspection of Dark Featureless Dam Penstocks Using MAVs". In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 4998–5005.

[3] T. Özaslan et al. "Autonomous navigation and mapping for inspection of penstocks and tunnels with MAVs". In: *IEEE Rob. Autom. Lett.* 2.3 (2017), pp. 1740–1747.

[4] PX4 Development Team. *Gazebo Simulation*. Accessed: 2023-03-10. URL: `https://docs.px4.io/main/en/sim_gazebo_gz/`.

[5] Inagaki Y et al. "Application of UAV for Sewer Pipe Inspection". In: *International Journal of Water and Wastewater Treatment* 6 (Jan. 2020). DOI: `10.16966/2381-5299.169`.

[6] Kenzo Nonami. *All About Industrial Application of Drones*. Japan: Ohmsha, 2018.

[7] Ippei Yamashita. *Drone Textbook Advanced Text Unmanned Aerial Worker Test (Drone Certification) Level 2 Correspondence*. Japan, 2018.

[8] Japan Sewage Works Association. *Guideline of Sewer Pipe Facility Stock Management*. Japan, 2016.

[9] Japan Sewage Works Association. *Manual of Sewage Facility Management Works Cost Estimate*. Japan, 2020.

[10] Yusuke Inagaki et al. "An effective measure for evaluating sewer condition: UAV screening in comparison with CCTVS and manhole cameras". In: *Water Practice and Technology* 15 (May 2020). DOI: 10.2166/wpt.2020.038.

[11] Echord++. *CHALLENGE BRIEF – RELATED TO THE ECHORD++ CALL FOR RD PROPOSALS*. Accessed: 2023-03-15. URL: https://echord.eu/pdti/pdti-urban-robotics-sewer-inspection/.

[12] C. H. Tan et al. "A smart unmanned aerial vehicle (UAV) based imaging system for inspection of deep hazardous tunnels". In: *Water Practice Technol.* 13.4 (2018), pp. 991–1000.

[13] C. H. Tan et al. "Design optimization of sparse sensing array for extended aerial robot navigation in deep hazardous tunnels". In: *IEEE Rob. and Autom. Lett.* 4.2 (2019), pp. 862–869.

[14] Yago da Silva et al. "Computer Vision Based Path Following for Autonomous Unmanned Aerial Systems in Unburied Pipeline Onshore Inspection". In: *Drones* 6.12 (2022), p. 410. ISSN: 2504-446X. DOI: 10.3390/drones6120410. URL: http://dx.doi.org/10.3390/drones6120410.

[15] Jorge Fuentes-Pacheco, Jose Ascencio, and J. Rendon-Mancha. "Visual Simultaneous Localization and Mapping: A Survey". In: *Artificial Intelligence Review* 43 (Nov. 2015). DOI: 10.1007/s10462-012-9365-8.

[16] Naoki Tanaka and Masayo Mouri. "A Detection Method of Cracks and Structural Objects of the Road Surface Image". In: Jan. 2000, pp. 387–390.

[17] S. Edwards et al. "A Robust Method for Approximate Visual Robot Localization in Feature-Sparse Sewer Pipes". In: *Frontiers in Robotics and AI* 10 (2023). DOI: 10.3389/frobt.2023.1150508.

[18] Chris H. Bahnsen et al. "3D Sensors for Sewer Inspection: A Quantitative Review and Analysis". In: *Sensors* 21.7 (2021), p. 2553. ISSN: 1424-8220. DOI: 10.3390/s21072553. URL: http://dx.doi.org/10.3390/s21072553.

[19] S. S. Mansouri et al. "Towards MAV Navigation in Underground Mine Using Deep Learning". In: *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2018, pp. 880–885.

[20] S. S. Mansouri et al. "Deploying MAVs for autonomous navigation in dark underground mine environments". In: *Rob. Auton. Syst.* 126 (2020), p. 103472. URL: https://www.sciencedirect.com/science/article/pii/S0921889019306256.

[21] S. S. Mansouri et al. "MAV Navigation in Unknown Dark Underground Mines Using Deep Learning". In: *European Control Conference*. 2020.

[22] Misha Urooj Khan et al. "A Comparative Survey of LiDAR-SLAM and LiDAR based Sensor Technologies". In: Aug. 2021. DOI: 10.1109/MAJICC53071.2021.9526266.

[23] Victor Castañeda, Diana Mateus, and Nassir Navab. "SLAM combining ToF and High-Resolution cameras". In: Apr. 2011, pp. 672–678. DOI: 10.1109/WACV.2011.5711569.

[24] Tzu-Yi Chuang and Cheng-Che Sung. "Learning and SLAM Based Decision Support Platform for Sewer Inspection". In: *Remote Sensing* 12.6 (2020), p. 968. ISSN: 2072-4292. DOI: 10.3390/rs12060968. URL: http://dx.doi.org/10.3390/rs12060968.

[25] C. Wong et al. "A Survey on Ellipse Detection Methods". In: *IEEE International Symposium on Industrial Electronics* (May 2012). DOI: 10.1109/ISIE.2012.6237243.

[26] Yongtao Yu et al. "Automated Detection of Road Manhole and Sewer Well Covers From Mobile LiDAR Point Clouds". In: *Geoscience and Remote Sensing Letters, IEEE* 11 (Sept. 2014), pp. 1549–1553. DOI: 10.1109/LGRS.2014.2301195.

[27] Yongtao Yu, Haiyan Guan, and Wenli Mi. "Automated Detection of Urban Road Manhole Covers Using Mobile Laser Scanning Data". In: *IEEE Transactions on Intelligent Transportation Systems* 16 (Apr. 2015), pp. 1–12. DOI: 10.1109/TITS.2015.2413812.

[28] S. Pu et al. "Recognising basic structures from mobile laser scanning data for road inventory studies". In: *Photogrammetry Remote Sensing* 66 (Jan. 2011), pp. 528–539.

[29] D. Alejo, Fernando Caballero, and Luis Merino. "RGBD-based robot localization in sewer networks". In: Sept. 2017, pp. 4070–4076. DOI: 10.1109/IROS.2017.8206263.

[30] David Alejo, Fernando Caballero, and Luis Merino. "A Robust Localization System for Inspection Robots in Sewer Networks". In: *Sensors* 19 (Nov. 2019), p. 4946. DOI: 10.3390/s19224946.

[31] Rasmus Andersen et al. "Depth-based Deep Learning for Manhole Detection in UAV Navigation". In: June 2022, pp. 1–6. DOI: 10.1109/IST55454.2022.9827720.

[32] G. Song and H. Wang. "A fast and robust ellipse detection algorithm based on pseudo-random sample consensus". In: *Proceedings of the 12th International Conference on Computer Analysis of Images and Patterns*. Springer-Verlag, 2007, pp. 669–676.

[33] D. Shane Barwick. "Very Fast Best-Fit Circular and Elliptical Boundaries by Chord Data". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.6 (2009), pp. 1147–1152. DOI: 10.1109/TPAMI.2008.279.

[34] T. Takegami, T. Gotoh, and G. Ohyama. "An Algorithm for Model-Based Stable Pupil Detection for Eye Tracking System". In: *Systems and Computers in Japan* 35.13 (2004), pp. 21–31.

[35] M. Park et al. "Automatic Cell Segmentation in Microscopic Color Images Using Ellipse Fitting and Watershed". In: *2010 IEEE/ICME International Conference on Complex Medical Engineering*. IEEE. 2010, pp. 69–74.

[36] D. Maio and D. Maltoni. "Real-Time Face Location on Gray-Scale Static Images". In: *Pattern Recognition* 33.9 (2000), pp. 1525–1540.

[37] R.D. Xu and M. Kemp. "Fitting Multiple Connected Ellipses to an Image Silhouette Hierarchically". In: *IEEE Transactions on Image Processing* 19.7 (2010), pp. 1673–1682.

[38] Gong Zhang, Digvir Jayas, and Noel White. "Separation of Touching Grain Kernels in an Image by Ellipse Fitting Algorithm". In: *Biosystems Engineering* 92 (Oct. 2005), pp. 135–142. DOI: 10.1016/j.biosystemseng.2005.06.010.

[39] Daniel Schleicher and Bernhard Zagar. "Image processing to estimate the ellipticity of steel coils using a concentric ellipse fitting algorithm". In: Nov. 2008, pp. 884 –890. DOI: 10.1109/ICOSP.2008.4697270.

[40] F. Dufrenois. "Ellipse Fitting with Uncertainty and Fuzzy Decision Stage for Detection. Application in Videomicroscopy". In: *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*. 2001, pp. 432–443.

[41] S. Crotty and K.R. Choudhury. "Morphometric Analysis for Early Detection of Changes in Cellular Structure in a Toxicological Experiment". In: *Statistics in Medicine* 26.29 (2007), pp. 5253–5266.

[42] N. Kharma et al. "Automatic Segmentation of Cells from Microscopic Imagery Using Ellipse Detection". In: *IET Image Processing* 1.1 (2007), pp. 39–47.

[43] J. Pu et al. "An Ellipse-Fitting Based Method for Efficient Registration of Breast Masses on Two Mammographic Views". In: *Medical Physics* 35 (2008), p. 487.

[44] S. Mahdavi et al. "3D Prostate Segmentation in Ultrasound Images Based on Tapered and Deformed Ellipsoids". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2009*. 2009, pp. 960–967.

[45] A. Fernandes and S. Nascimento. "Automatic Water Eddy Detection in SST Maps Using Random Ellipse Fitting and Vectorial Fields for Image Segmentation". In: *Discovery Science*. Springer, 2006, pp. 77–88.

[46] Y. Soh et al. "A New Method for Ellipse Fitting in Satellite Images". In: *Second International Conference on Intelligent Computation Technology and Automation*. IEEE. 2009, pp. 502–506.

[47] S. Sellah and O. Nasraoui. "An Incremental Hough Transform for Detecting Ellipses in Image Data Streams". In: *20th IEEE International Conference on Tools with Artificial Intelligence*. IEEE. 2008, pp. 45–48.

[48] P. Ramos et al. "Recent Developments on Impedance Measurements with DSP-Based Ellipse-Fitting Algorithms". In: *IEEE Transactions on Instrumentation and Measurement* 58.5 (2009), pp. 1680–1689.

[49] T. Wynn and S. Stewart. "Comparative Testing of Ellipse-Fitting Algorithms: Implications for Analysis of Strain and Curvature". In: *Journal of Structural Geology* 27.11 (2005), pp. 1973–1985.

[50] S. Xiaofang and G. Wencheng. "A Sun Spot Center Orientation Method Based on Ellipse Fitting in the Application of CPV Solar Tracker". In: *International Conference on Intelligent System Design and Engineering Application*. IEEE. 2010, pp. 175–178.

[51] A. Chia et al. "A Split and Merge Based Ellipse Detector with Self-Correcting Capability". In: *IEEE Transactions on Image Processing* (2010), pp. 1–1.

[52] K. Kanatani. "Ellipse Fitting with Hyperaccuracy". In: *Computer Vision–ECCV 2006*. 2006, pp. 484–495.

[53] L. Mak and T. Furukawa. "A 6 DOF Visual Tracking System for a Miniature Helicopter". In: *2nd International Conference on Sensing Technology*. 2007, pp. 32–37.

[54] S. Wong, H. Leung, and H. Ip. "Fitting Ellipses to a Region with Application in Calligraphic Stroke Reconstruction". In: *2006 IEEE International Conference on Image Processing*. IEEE. 2006, pp. 397–400.

[55] P. Rosin. "Analysing Error of Fit Functions for Ellipses". In: *Pattern Recognition Letters* 17.14 (1996), pp. 1461–1470.

[56] A. Fitzgibbon, M. Pilu, and R. Fisher. "Direct Least Square Fitting of Ellipses". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 21.5 (1999), pp. 476–480.

[57] W. Gander, G. Golub, and R. Strebel. "Least-Squares Fitting of Circles and Ellipses". In: *BIT Numerical Mathematics* 34.4 (1994), pp. 558–578.

[58] R. Halir and J. Flusser. "Numerically Stable Direct Least Squares Fitting of Ellipses". In: *The Sixth International Conference in Central Europe on Computer Graphics and Visualization*. 1998, pp. 59–108.

[59] D. Ballard. "Generalizing the Hough Transform to Detect Arbitrary Shapes". In: *Pattern Recognition* 13.2 (1981), pp. 111–122.

[60] P. Nair and A. Saunders. "Hough Transform Based Ellipse Detection Algorithm". In: *Pattern Recognition Letters* 17.7 (1996), pp. 777–784.

[61] N. Guil and E. Zapata. "Lower Order Circle and Ellipse Hough Transform". In: *Pattern Recognition* 30.10 (1997), pp. 1729–1744.

[62] R. McLaughlin. "Randomized Hough Transform: Improved Ellipse Detection with Comparison". In: *Pattern Recognition Letters* 19.3-4 (1998), pp. 299–305.

[63] J. Porrill. "Fitting Ellipses and Predicting Confidence Envelopes Using a Bias Corrected Kalman Filter". In: *Image and Vision Computing* 8.1 (1990), pp. 37–41.

[64] R. Bolles and M. Fischler. "A Ransac-Based Approach to Model Fitting and its Application to Finding Cylinders in Range Data". In: *International Joint Conference on Artificial Intelligence*. 1981, pp. 637–643.

[65] P. Yin. "A New Circle/Ellipse Detector Using Genetic Algorithms". In: *Pattern Recognition Letters* 20.7 (1999), pp. 731–740.

[66] Azarakhsh Keipour, Guilherme Pereira, and Sebastian Scherer. *Real-Time Ellipse Detection for Robotics Applications*. Feb. 2021.

[67] Cai Meng et al. "Arc Adjacency Matrix-Based Fast Ellipse Detection". In: *IEEE Transactions on Image Processing* PP (Jan. 2020), pp. 1–1. DOI: 10.1109/TIP.2020.2967601.

[68] Min Liu, Bodi Yuan, and Jing Bai. "Ellipse Detection on Images Using Conic Power of Two Points". In: *British Machine Vision Conference*. 2018. URL: https://www.semanticscholar.org/paper/Ellipse-Detection-on-Images-Using-Conic-Power-of-Liu-Yuan/e05fd6c7237cee0127d5424d532958b7f8b70171.

[69] Taha Elmokadem and Andrey Savkin. "A Hybrid Approach for Autonomous Collision-Free UAV Navigation in 3D Partially Unknown Dynamic Environments". In: *Drones* 5 (July 2021), p. 57. DOI: 10.3390/drones5030057.

[70] Frank Mascarich et al. "A multi-modal mapping unit for autonomous exploration and mapping of underground tunnels". In: Mar. 2018, pp. 1–7. DOI: 10.1109/AERO.2018.8396595.

[71] Tolga Ozaslan et al. "Inspection of penstocks and featureless tunnel-like environments using micro UAVs T Özaslan, S Shen, Y Mulgaonkar, N Michael, V Kumar - Field and Service Robotics, 2015". In: Dec. 2013.

[72] M. Nieuwenhuisen and S. Behnke. "Hierarchical Planning with 3D Local Multiresolution Obstacle Avoidance for Micro Aerial Vehicles". In: *Proceedings of the ISR/Robotik 2014, 41st International Symposium on Robotics*. Munich, Germany, 2014, pp. 1–7.

[73] S. Grzonka, G. Grisetti, and W. Burgard. "A Fully Autonomous Indoor Quadrotor". In: *IEEE Trans. Robot.* 28 (2012), pp. 90–100. DOI: 10.1109/TRO.2011.2160634.

[74] P.E. Hart, N.J. Nilsson, and B. Raphael. "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". In: *IEEE Trans. Syst. Sci. Cybern.* 4 (1968), pp. 100–107. DOI: 10.1109/TSSC.1968.300136.

[75] S. Koenig and M. Likhachev. "Fast replanning for navigation in unknown terrain". In: *IEEE Trans. Robot.* 21 (2005), pp. 354–363. DOI: 10.1109/TRO.2004.838026.

[76] L.E. Kavraki et al. "Probabilistic roadmaps for path planning in high-dimensional configuration spaces". In: *IEEE Trans. Robot. Autom.* 12 (1996), pp. 566–580. DOI: 10.1109/70.508439.

[77] S.M. Lavalle. *Rapidly-Exploring Random Trees: A New Tool for Path Planning*. Tech. rep. Ames, IA, USA: Computer Science Department, Iowa State University, 1998.

[78] M. Petrlík et al. "A robust UAV system for operations in a constrained environment". In: *IEEE Rob. Autom. Lett.* 5.2 (2020), pp. 2169–2176.

[79] Y. Li and C. Liu. "Efficient and safe motion planning for quadrotors based on unconstrained quadratic programming". In: *Robotica* 39.2 (2021), pp. 317–333.

[80] Michael Hoy, Alexey Matveev, and Andrey Savkin. "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: A survey". In: *Robotica* 33 (Mar. 2014), pp. 1–35. DOI: 10.1017/S0263574714000289.

[81] A. V. Savkin and C. Wang. "A Method for Collision Free Navigation of Non-Holonomic 3D Robots in Unknown Tunnel like Environments". In: *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2017, pp. 936–940.

[82] A. S. Matveev and A. V. Savkin. "Proofs of Technical Results Justifying an Algorithm of Reactive 3D Navigation of a Mobile Robot through an Unknown Tunnel". In: (2018). arXiv: 1803.00803 [cs.RO].

[83] A. S. Matveev, V. Magerkin, and A. V. Savkin. "A method of reactive control for 3D navigation of a nonholonomic robot in tunnel-like environments". In: *Automatica* 114 (2020), p. 108831. URL: https://www.sciencedirect.com/science/article/pii/S0005109820300297.

[84] T. Merz and F. Kendoul. "Beyond visual range obstacle avoidance and infrastructure inspection by an autonomous helicopter". In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. San Francisco, CA, USA, 2011, pp. 4953–4960. DOI: 10.1109/IROS.2011.6048763.

[85] S. Hrabar. "Reactive obstacle avoidance for Rotorcraft UAVs". In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. San Francisco, CA, USA, 2011, pp. 4967–4974. DOI: 10.1109/IROS.2011.6094762.

[86] F. Azevedo et al. "Collision avoidance for safe structure inspection with multirotor UAV". In: *2017 European Conference on Mobile Robots (ECMR)*. Paris, France, 2017, pp. 1–7. DOI: 10.1109/ECMR.2017.8098671.

[87] S. Vanneste, B. Bellekens, and M. Weyn. "3DVFH+: Real-Time Three-Dimensional Obstacle Avoidance Using an Octomap". In: *CEUR Workshop Proceedings*. Vol. 1319. York, UK, 2014.

[88] I. Ulrich and J. Borenstein. "VFH+: Reliable obstacle avoidance for fast mobile robots". In: *1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*. Vol. 2. Leuven, Belgium, 1998, pp. 1572–1577. DOI: 10.1109/ROBOT.1998.677288.

[89] D. Alejo et al. "Optimal Reciprocal Collision Avoidance with mobile and static obstacles for multi-UAV systems". In: *2014 International Conference*

on Unmanned Aircraft Systems (ICUAS). Orlando, FL, USA, 2014, pp. 1259–1266. DOI: 10.1109/ICUAS.2014.6842294.

[90] L. Blasi et al. "Path Planning and Real-Time Collision Avoidance Based on the Essential Visibility Graph". In: *Appl. Sci.* 10.5613 (2020). DOI: 10.3390/app10165613.

[91] Y. Du, X. Zhang, and Z. Nie. "A Real-Time Collision Avoidance Strategy in Dynamic Airspace Based on Dynamic Artificial Potential Field Algorithm". In: *IEEE Access* 7 (2019), pp. 169469–169479. DOI: 10.1109/ACCESS.2019.2956897.

[92] C. Urdiales et al. "A hybrid architecture for autonomous navigation in dynamic environments". In: *Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology*. Halifax, NS, Canada, 2003.

[93] Y. Zhu et al. "A new hybrid navigation algorithm for mobile robots in environments with incomplete knowledge". In: *Knowl. Based Syst.* 27 (2012), pp. 302–313. DOI: 10.1016/j.knosys.2011.11.012.

[94] M. Nieuwenhuisen and S. Behnke. "Layered mission and path planning for MAV navigation with partial environment knowledge". In: *Intelligent Autonomous Systems 13*. Berlin/Heidelberg, Germany: Springer, 2016, pp. 307–319.

[95] M. Hank and M. Haddad. "A hybrid approach for autonomous navigation of mobile robots in partially-known environments". In: *Robot. Auton. Syst.* 86 (2016), pp. 113–127. DOI: 10.1016/j.robot.2016.09.009.

[96] M. Wzorek, C. Berger, and P. Doherty. "A Framework for Safe Navigation of Unmanned Aerial Vehicles in Unknown Environments". In: *2017 25th International Conference on Systems Engineering (ICSEng)*. Las Vegas, NV, USA, 2017, pp. 11–20.

[97] M. D'Arcy, P. Fazli, and D. Simon. "Safe navigation in dynamic, unknown, continuous, and cluttered environments". In: *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. Shanghai, China, 2017, pp. 238–244.

[98] L. Adouane. "Reactive versus cognitive vehicle navigation based on optimal local and global PELC*". In: *Robot. Auton. Syst.* 88 (2017), pp. 51–70. DOI: 10.1016/j.robot.2016.11.014.

[99] Matthias Nieuwenhuisen and Sven Behnke. "Layered Mission and Path Planning for MAV Navigation with Partial Environment Knowledge". In: vol. 302. Jan. 2014. ISBN: 978-3-319-08337-7. DOI: 10.1007/978-3-319-08338-4_23.

[100] Gazebo Model Library. *Gazebo Models*. Accessed: 2023-03-05. URL: http://models.gazebosim.org/.

List of Figures

List of Tables

List of Algorithms