Czech Technical University in Prague

Faculty of Electrical Engineering
Department of Computer Graphics and Interaction
Multi-robot Systems

# Fusion of radiation and depth measurements for an unmanned aerial vehicle

**Bachelor's Thesis**

## Tadeáš Zribko

Prague, May 2023

Study programme: Open informatics
Branch of study: Computer games and graphics

**Supervisor: Ing. Petr Štibinger**

## Declaration

I declare that presented work was developed independently, and that I have listed all sources of information used within, in accordance with the Methodical instructions for observing ethical principles in preparation of university theses.

Prague, date ............... ......................

# BACHELOR‘S THESIS ASSIGNMENT

## I. Personal and study details

| | | | |
|---|---|---|---|
| Student's name: | **Zribko Tadeáš** | Personal ID number: | **498830** |
| Faculty / Institute: | **Faculty of Electrical Engineering** | | |
| Department / Institute: | **Department of Computer Graphics and Interaction** | | |
| Study program: | **Open Informatics** | | |
| Specialisation: | **Computer Games and Graphics** | | |

## II. Bachelor’s thesis details

Bachelor’s thesis title in English:

**Fusion of radiation and depth mesurements for an unmanned aerial vehicle**

Bachelor’s thesis title in Czech:

**Fúze radiačních a hloubkových měření pro bezpilotní helikoptéru**

Guidelines:

1) Familiarize yourself with the measurement principle of a Compton camera and the output data format (https://github.com/rospix/gazebo_compton_camera_python). Examine the simulation model of the sensor for the MRS UAV system [1]. 2) Study the available methods for localizing multiple signal sources simultaneously. 3) Design a method that accepts the Comtpon camera measurements and estimates the positions of potential radiation sources. 4) Familiarize yourself with the OctoMap data structure and the environment mapping framework used in the MRS UAV system (https://github.com/ctu-mrs/octomap_mapping_planning). Study the use of a depth camera as well as a 3D LIDAR for object detection and environment mapping [3]. 5) Assuming the radiation source is a solid object, design a method to eliminate false-positive detections by incorporating environment perception by the available distance sensors. 6) Evaluate the proposed method for localization of radioactive sources in a simulated environment and on real-world datasets. Discuss the advantage of incorporating distance mesurements.

Bibliography / sources:

1. T. Baca, M. Petrlik, M. Vrba, V. Spurny, R. Penicka, D. Hert and M. Saska. The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles. Journal of Intelligent & Robotic Systems 102(26):1–28, May 2021. 2. T. Baca, P. Stibinger, D. Doubravova, D. Turecek, J. Solc, J. Rusnak, M. Saska and J. Jakubek. Gamma Radiation Source Localization for Micro Aerial Vehicles with a Miniature Single-Detector Compton Event Camera. In 2021 International Conference on Unmanned Aircraft Systems (ICUAS). June 2021, 338–346. 3. Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., & Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. Autonomous robots, 34, 189-206. 4. Gao, W., Wang, W., Zhu, H., Huang, G., Wu, D., & Du, Z. (2018). Robust radiation sources localization based on the peak suppressed particle filter for mixed multi-modal environments. Sensors, 18(11), 3784. 5. Djuric, P. M., Kotecha, J. H., Zhang, J., Huang, Y., Ghirmai, T., Bugallo, M. F., & Miguez, J. (2003). Particle filtering. IEEE signal processing magazine, 20(5), 19-38. 6. Sato, Y., Ozawa, S., Terasaka, Y., Minemoto, K., Tamura, S., Shingu, K., & Torii, T. (2020). Remote detection of radioactive hotspot using a Compton camera mounted on a moving multi-copter drone above a contaminated area in Fukushima. Journal of nuclear science and technology, 57(6), 734-744.

Name and workplace of bachelor’s thesis supervisor:

**Ing. Petr Štibinger Multi-robot Systems FEE**

Name and workplace of second bachelor’s thesis supervisor or consultant:

Date of bachelor’s thesis assignment: **09.02.2023** Deadline for bachelor thesis submission: **26.05.2023**

Assignment valid until: **22.09.2024**

| | | |
|---|---|---|
| Ing. Petr Štibinger<br>Supervisor’s signature | Head of department’s signature | prof. Mgr. Petr Páta, Ph.D.<br>Dean’s signature |

## III. Assignment receipt

The student acknowledges that the bachelor’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor’s thesis, the author must state the names of consultants and include a list of references.

.

| | |
|---|---|
| Date of assignment receipt | Student’s signature |

## Acknowledgments

First of all, I would like to thank my supervisor, Ing. Petr Štibinger, for his two years of patience, guidence and the opportunity to learn from a person who is knowledgeable in the field of study. I express my gratitude to the Multi-Robot System group for the opportunity to test the work on real equipment. Finally, I would like to thank my whole family for their support during my studies and love they have shown me throughout my life, my girlfriend for her encouragement and inspiration and my friends.

## Abstract

Radiation localization is an important topic for the protection of humans and the environment at present. It is the main task that is tackled in this thesis. The task uses a combination of the flexibility of an Unmanned Aerial Vehicle (UAV) and sensors to acquire data about the environment in which they are operating. The most essential sensor is a single layer Compton camera based on the advanced TimePix3 technology for radiation detection. The camera has a small size and low weight which is a clear advantage over other detectors for use onboard small UAVs. Together with the depth sensor for obstacle perception, a Light Detection and Ranging (LIDAR) or a depth camera are placed on the UAV structure and serve as data sources for radiation localization. The measurements from the depth sensors are stored in the form of an OctoMap for a convenient 3D representation of the environment. The radiation localization task is performed by processing the sensor data using various sampling methods and fusing the radiation and depth measurements. The advantage is the detection functionality even in more complex environments. The estimation of the positions of the radiation sources is solved using a particle filter, which is implemented in different variants to achieve the best possible results. The last sensor that is used onboard the UAV is the RGB camera. The image obtained from the camera is used as a visual output for the human operators. In the image, the color-coded objects identified as the radiation sources are clearly visualized. The radiation source detection procedure is applicable in many fields such as nuclear energy, medical and industrial applications. The methods developed in this work considerably improve the radiation localization approach developed by the Multi-robot Systems Group (MRS) group at Faculty of Electrical Engineering (FEE) Czech Technical University (CTU). A key advantage of the proposed solution is the ability to localize multiple radiation sources simultaneously which has a great potential for safety and efficiency improvements in all related fields.

**Keywords** Unmanned Aerial Vehicles, Compton camera, TimePix, Radiation source detection, Particle filter, Image processing, Object detection

# Abstrakt

Lokalizácia radiácie je dôležitá téma pre ochranu človeka a súčasne aj životného prostredia. Je to hlavná úloha, ktorá sa rieši prostredníctvom tejto práce. Pri plnení úlohy sa využíva kombinácia flexibility bezpilotného lietadla a senzorov pre získavanie dát o prostredí v ktorom sú umiestnené. Najdôležitejším senzorom je jednovrstvová Comptonova kamera založená na pokročilej technológii TimePix3 na detekciu žiarenia. Disponuje nízkymi rozmermi a nízkou váhou čo je jednoznačná výhoda oproti ostatným detektorom. Spolu s hĺbkovým senzorom na vnímanie prekážok, LiDAR alebo hĺbková kamera sú umiestnené na konštrukcii bezpilotného lietadla a slúžia ako zdroje dát na lokalizáciu žiarenia radiácie. Merania hĺbkových senzorov sa ukladajú vo forme mapy OctoMap na pohodlné 3D zobrazenie prostredia. Úloha lokalizácie radiácie začína spracovaním dát zo senzorov pomocou rôznych vzorkovacích metód a fúziou radiačných a hĺbkových meraní. Výhodou je funkčnosť detekcie aj v komplexnejšom prostredí. Odhad polôh zdrojov radiácie rieši časticový filter, ktorý je implementovaný v rôznych variantách pre dosiahnutie čo najlepších výsledkov. Posledným senzorom ktorý je súčasťou UAV je RGB kamera. Obraz získaný z kamery je využitý ako vizuálny výstup pre pozorovateľa v ktorom môže jasne vidieť farebne vyznačene zdroje radiácie. Postup detekcie zdrojov radiácie je využiteľný v mnohých oblastiach, ako napríklad v jadrovej energetike, v medicíne a v priemysle. Metódy vyvinuté v tejto práci výrazne zlepšujú schopnosť lokalizácie žiarenia ktorý bol vyvinutý skupinou MRS na ČVUT FEL v Prahe. Hlavnou výhodou navrhovaného riešenia je možnosť lokalizovať viacero zdrojov žiarenia súčasne, čo má veľký potenciál pre zvýšenie bezpečnosti a efektívnosti vo všetkých súvisiacich oblastiach.

**Kľúčové slová** Bezpilotné lietadlo, Comptonová kamera, TimePix, Detekcia zdrojov radiácie, Časticový filter, Spracovanie obrazu, Detekcia objektov

## Abbreviations

**CTU** Czech Technical University

**FEE** Faculty of Electrical Engineering

**GPS** Global Positioning System

**RTK** Real Time Kinematics

**LIDAR** Light Detection and Ranging

**MRS** Multi-robot Systems Group

**ROS** Robot Operating System

**RTK** Real-time Kinematics

**UAV** Unmanned Aerial Vehicle

**DNA** Deoxyribonucleic acid

**RPA** Remotely piloted aircraft

**ToA** Time-of-Arrival

**ToT** Time-over-Threshold

**GUI** Graphic User Interface

**CdTe** Cadmium Telluride

# Contents

# Chapter 1

# Introduction

Now technology is part of everyday life. We see and use technology all the time, making our lives more enjoyable and taking it to the next level. Day by day, new inventions and means of improving their features and techniques of execution are coming up. Unmanned Aerial Vehicle or UAV is one of the new technologies that the world offers us. The main motivation for the creation of UAVs was for military purposes. Motivations were primarily formed by the ability to explore, infiltrate enemy territory, and observe. Remote control is one of the many advantages that UAVs possess. It provides the option to operate in the sky without risking human lives. Over time, new types of Remotely piloted aircraft (RPA) such as rotor aircraft have evolved from the classic RPA. Their use has gradually spread from military purposes to every possible field of humanity[31]. It would be a shame not to use the UAV technology majorly for the protection of humans, living creatures and the environment we live. UAV allows us unmanned and, most importantly, safe access for humans. The UAVs can be divided into several groups. They are classified according to their weight, speed, flying height, or the weight they can carry. Many of them are also designed for commercial use. Shops today offer many different types of UAVs. The most common types of UAVs are currently four or six-propeller vehicles. People started using them for a number of new disciplines. The most common use is image recording or just as a gadget. UAVs are also used as a common inspection tool for a number of industries.

Nuclear power plants are among the most powerful sources of electricity. The electricity produced makes up a significant portion of overall electricity generation. Nuclear fission plays a major role in the process of electricity generation.

X-ray computed tomography is already common in almost every large hospital. A discovery which has fatally improved medical diagnosis and saved countless lives. The principle of the X-ray topographer is also based on radiation with very short wavelengths. These electromagnetic waves are generated by the X-ray tube, which converts electricity into electromagnetic waves.

All this, besides many advantages, contains disadvantages as well. The main disadvantage is ionizing radiation, which can permeates through the air and has adverse effects on organic materials. Prolonged exposure to radiation can mean death for a person. Radiation causes changes to cells in the tissue. In most cases, it is a change of genetic information encoded by the Deoxyribonucleic acid (DNA). The adverse effect has led to the development of products that are even unbeneficial to mankind. Nuclear weapons are an example of such a product. Nuclear weapons are mainly known for their power and the untamed energy that the weapon possesses.

In 2011, Japan was hit by a strong earthquake in the east. The earthquake caused a tsunami, which subsequently damaged the Fukushima nuclear reactor. This damage resulted in the release of radioactive material from the power station. The event affected the lives of many people living in the area. The accident is still one of the worst in the world. The

Fukushima or Chernobyl nuclear power plant disasters shook the world and changed the perception of nuclear power. Power plant accidents or the radioactive waste produced by nuclear power plants are a problem that affects us all, because we do not know at first sight what can and cannot be radioactive. The human body has no organ to detect low doses of ionizing radiation, it ignores them. Reducing the consequences of radioactivity is an ongoing effort in the field. The mitigation has resulted in the development of new ways of detecting leaked substances and reducing the concentration of radioactivity in the air or in the ground. Detecting and removing radioactive material is the most effective weapon the humanity has against radiation in terms of safety.

Several types of radiation detectors are available to detect different types of ionizing radiation. The most well-known types of radiation detectors include semiconductor, gas-filled, scintillation, and solid-state devices. Radiation detectors use different properties of radiation. The most common property of radiation is the disruption of electrons. By measuring the disruption of electrons, we can determine not only the existence of radiation, but also its intensity and even its direction. Measuring radiation in most cases means that the detector must also be under the influence of radiation.

Precisely in order to avoid radiation coming into contact with people, the best solution is to exclude people from the problem altogether. Combining a UAV and a sensor to detect radiation is one of the best options for human safety. The UAV will provide easy access for the sensor to an otherwise toxic or difficult-to-reach environment. As the result, we can perform measurements even in close proximity to the radiation source, which would have been impossible otherwise.

The goal of this work is to use a combination of a UAV and a radiation detector to locate potential possible sources of radiation. Radiation sources can be at different distances from each other and can also be placed between or within obstacles. The essential aspect is that the sensor receives gamma radiation from the source. Locating potential radiation sources is only possible by processing sensor output data. The data relates to both the radiation detector and the mapped environment. Data from the sensors are fused to create an estimate of the radiation source. All data is collected by the UAV and evaluated in real-time. With the least amount of human involvement, this method enables fast and precise localization of multiple radiation sources. It can be extremely useful when exploring dangerous environments in a short amount of time. Even the best UAV has a very short battery life. Additionally, some battery power has to be conserved for the return from a dangerous area. Thanks to the measurement speed, the position of the source can be quickly estimated after a single charge of the UAV.

## 1.1 State of the art

UAVs, as we know them today, were created as a result of the motivation of safety. Because of their availability, UAVs have become widely used in a number of projects. They can assist with various activities, such as detection of different objects. An example of detection is an odour source[14] or radiation source. Also UAVs can help in firefighting[4], [9], airspace protection[3], [16], wildfire detection in forested areas[10], [24] and scanning crops for pests[7]. Due to their small size and ability to move in complex environments, they are also well suited for mapping and documentation of historical buildings[8].

Radiation search and localization are particularly important today. It is essential to ensure the safety of the environment from the effects of radiation. Several methods already exist for detecting radiation. One method is based on the gradient ascent algorithm[26]. The algorithm tries to find a local maximum and thus determine the position of the radiation source. Another method similar to the previous one, is iterative[25]. This method attempts to estimate the maximum likelihood for a source using the Gradient method along with the Newton's method. The use of reinforcement learning algorithms has applications in this area as well. Liu Z. and Abbaszadeh S. [15] work on a learning algorithm consisting of double Q-learning. This algorithm, based on states from measured data and possible actions, evaluates the position of a radiation source. Compared to the gradient method, Q-learning even achieves better and faster results.

The Fukushima accident caused radiation contamination in the environment. A number of works have been written on the concentrations of radioactivity in the air[21], [22]. The primary task was to map the environment and assign the radiation to a given location. In [23], radiation was measured by a small research team and a helicopter. To protect the helicopter crew, an unmanned helicopter was created that can move along a pre-programmed trajectory. New possibilities for unmanned aerial measurement were created. Nor can we forget the water contamination that this disaster has caused. This topic is discussed by Buesseler, Ken in[28].

Radiation detection can also help protect against terrorist attacks. The attacks could be nuclear weapons in the form of “dirty bombs”. The term “dirty bomb” is used to describe a conventional explosive that, after detonation, contaminates the air with radioactive material. Sean M. Brennan[34] used a network of detectors to find the threat. The network of detectors cooperates with each other in order to obtain the best possible detection result.

The MRS[1] at the Department of Cybernetics, FEE, CTU in Prague, participates in project RaDron[2], which tackles the problem of locating unknown sources of gamma radiation. The project employs one or more UAVs to detect the radiation source quickly[11]. A Compton camera sensor based on Timepix3 technology is used for radiation detection[12]. The Compton camera based on the Timepix3 technology is a novel device which has only one chip. The chip does not need cooling and the main advantage is the small size of the sensor. The small size and low weight allow us to use the sensor directly on the UAVs. Compton sensor with TimePix3 chip is today's accurate and fast technology build on semiconductor detectors[17][13][6].

Other approaches similar to this work are also under development. They are based on a particle filter, which processes sensor measurements to determine the position of the radiation source. This involves localizing the sources in an already pre-partitioned environment on a grid of points. Subsequently, these points are assigned a weight according to the measurement [1] or localizing sources by processing batches of data gathered in advance.[18].

## 1.2 Problem definition

This work tackles the problem of detecting possible radioactive substances by an autonomously operating UAV. The first step of the work is to process the output data from the Compton camera. All sensors are placed onboard a flying UAV. The main sensors used in the work are for radiation detection, environment mapping, and image capture. The sensors collaborate to achieve a single goal: to locate and detect potential sources of radiation. The drone moves around the possible occurrence of radiation, and the sensor continuously collects radiation data. The UAV is using Rospix[17], a Robot Operating System (ROS) driver for

---

[1] `http://mrs.felk.cvut.cz`

[2] `http://mrs.felk.cvut.cz/projects/tacr-radron-project`

Timepix-based detectors, which was developed as a part of the RaDron project. The sensor readings are provided in the form of Compton cones. The Compton cone is a 3D representation of all possible directions to a hypothetical radiation source. The idea is to convert the Compton cone into a suitable mathematical equation. The equation is converted to a set of 3D points using an appropriate method and data access. The sampled points serve as an assumption of the radiation source's position. To improve hypothesis placement, the Compton camera measurements are fused with data from onboard 3D sensors. It is assumed that the source of the radiation is not in the air. Therefore, it is possible to reduce the number of suitable locations by focusing on the surfaces and objects in the 3D map. The fusing dramatically simplifies our work with the data in the next steps of the work.

Designing an appropriate iterative algorithm based on the particle filter concept is the next phase in this project. The particle filter is able to respond to new measured point sets. The response of the particle filter is based on discarding unnecessary measurement points. The output of the particle filter is a set of points of approximate source localization generated from the measured values. The output is then processed by a rounding method capable of finding the most appropriate locations of radiation occurrence.

The final step is the visualization of the results. The main output is represented by a colored BGR[3] or RGB camera located on the UAVs. The 3D points obtained by the previous methods are converted into the 2D coordinate system of the camera. Then, the points are used as a prerequisite to mark the radioactive object. This is further augmented by detecting features directly in the image around the suspected radioactive objects. The result is a real-time video of the tagged objects. An additional output is a 3D visualization created with the Rviz tool. The 3D visualization displays the scanned area along with the points from the sensor and filter at the same time.

## 1.3 Contributions

This thesis presents a complete processing pipeline for quick detection of multiple radiation sources by a compact UAV. The procedure starts by processing the radiation measurements in the form of Compton cones obtained from the Compton camera, and depth measurements obtained from a LIDAR or a depth camera. The processing starts by formulating a parametric equation describing the Compton cone envelope. The cone envelope is then sampled using one of the two definition methods as a set of points. The set of points is reduced using fusion with depth measurements. The procedure continues with the detection of radiation sources, which is achieved by a filtering model. The thesis presents and compares three possibilities of these filtering models. The models vary by the means of weight assignment to the particles used by the filter. The last step of the procedure is the visualization of the radiation sources to the user in the color camera image. The step involves detecting objects using an edge detector and assigning the estimated radiation source to these objects. The result is a color bounded region of the object in the image that can be considered as radioactive.

---

[3] Blue, Green and Red color scheme

# Chapter 2

# Preliminaries

This chapter introduces the basic definitions and presents the reader with the information on which the thesis is based or contributed to its creation. It consists of an introduction to the ROS framework, which was used in the implementation part of this thesis. The chapter describes the test environment Gazebo and its role in the development and validation. Last but not least, it discusses the fundamental problem to which this thesis is devoted, namely radiation, the impact of radiation on the environment, and the impact on humans. It provides a brief overview of radiation detection methods, and the principle of the radiation measurement technique using the single-layer Compton camera.

## 2.1 Robot Operating System

Robot Operating System[1] is a popular open-source framework used to develop robotic software. Willow Garage launched an Open Source ROS project in 2007 to produce a framework that can be used in various robotic applications. ROS is popular with researchers and enthusiasts because it offers a variety of tools and libraries to make a complex robot system easier. Thanks to the modular framework design of ROS, multiple software components can be connected to each other via a common interface. For this reason, it is easier to create complex robotic systems consisting of many software programs running on different computers. The framework offers several communication protocols that enable the flow of data between different components, including messages and services.

The foundation of ROS is the idea of a node, which is a standalone piece of software that carries out a single function. Through the ROS protocols[2], it is possible to communicate between the different nodes. As a result, it is easy to build modular robotic systems, that allow the combination of various nodes to produce complex behaviors. The extensive development community, that supports ROS is one of its main advantages. For ROS, there are thousands of packages[3], that offer a wide variety of capabilities, from high-level planning and decision-making to low-level hardware control. This implies, that developers can start creating their own robotic systems immediately by finding the packages that easily provide the characteristics they require. The platform independence of ROS allows it to function on a variety of hardware and operating systems. This makes it simple to utilize ROS with a variety of robotic gear and to adapt ROS-based apps to various platforms. In addition to being widely used in academia and research, ROS is also employed in business to create robotic applications.

Many robotic systems, including autonomous cars, manipulators, and drones, are developed using ROS. Moreover, ROS is applied in a number of fields, including industry, health-

---

[1] `https://wiki.ros.org/ROS/Introduction`

[2] `https://wiki.ros.org/ROS/Technical`

[3] `https://index.ros.org/packages`

care, and agriculture. ROS's adaptability is one of its main benefits. Since ROS is an open-source framework, programmers are free to alter and expand it as necessary. As a result, programmers can extend ROS with new capabilities or change its existing ones to better suit a given application. In order to make the creation and debugging of robotic systems easier, ROS also offers a variety of debugging and visualization tools. Developers can view the behavior of their robotic systems in real time using the framework's integrated visualization tool, RViz. Several debugging tools are also offered by ROS, such as the ROSbag tool, which enables programmers to capture and replay data from their robotic systems.

In conclusion, ROS offers a strong and adaptable platform for the creation of robotics software. Its modularity, networked architecture, and community support make it easier to build sophisticated robotic systems, that can function on a variety of hardware and operating systems. With its and a thriving developer community, ROS is probably going to play an important role in the advancement of robotics technologies in the oncoming years. [19]

### 2.1.1 Robot Operating System communication

In ROS, network communication can take place between individual parts of the program, which are called nodes. Several possible ways of communication and their parts are described below.

The ROS*message*[4] is usually used to transfer information between nodes. A message is a simple data structure, that consists of two parts. The first part is the message type and the second part is the message data. Each message has its own specification and own type. A type is a form of predefined structure, that can consist of a simple format of one type, such as a string. It can also be represented by a more complex structure consisting of names and their data types. Messages are mainly used to control the robot, to determine its current state or to obtain information from sensor output. They are a useful tool to simplify communication between components and in particular, provide modular usability and reusability.

A key component of the ROS is communication, which allows many nodes to share information and cooperate to accomplish a shared objective. Communication is accomplished using a variety of protocols that let nodes call and deliver services, publish and subscribe to topics, and send messages using the parameter server. In ROS, publish/subscribe (Pub/Sub) is the most used form of communication. Nodes in a Pub/Sub network can exchange messages by publishing them to a topic or subscribing to it to receive messages. There is a designated buffer over which messages are sent back and forth. To send a message to a topic, a node creates a *publisher* and to receive a message from a topic, it creates an object known as a *subscriber*. Many nodes can simultaneously post to and subscribe to the same topic using the flexible and scalable Pub/Sub architecture.

ROS *services* are another typical communication method used in ROS. A node makes a service call to another node, which subsequently responds to the request. A message pair consisting of a request message and a response message defines the service. The objects in a node that make requests are called service clients, while the objects that provide services are called service servers. Service calls are useful when nodes need to perform complex calculations. The main difference between service and message is that the service call returns a reply, it is a confirmed communication. The main advantage of using a server is if you need to get a response. Typically an essential control command where it is necessary to know if the

---

[4] `https://wiki.ros.org/msg`

command was executed or not. Alternatively, the server is used to obtain information for some trajectory planning where it is important to know if the path exists or not.

Each node in the ROS system has access to a global dictionary called *a parameter server*, which is another feature provided by ROS. Parameter servers track key-value pairs, that can be used to store configuration data such as system parameters or robot configuration. Nodes can access parameter servers to read or write parameters and because parameter values can be changed dynamically, the system can be dynamically reconfigured.

### 2.1.2 Rosbag

Rosbag, a component of ROS, was created to collect data from topics produced by ROS nodes, while the system was in operation. Data from robot sensors like cameras, LiDARs or microphones as well as details about the robot's state like its position, velocity, and orientation are frequently captured using rosbag. This information is packed in a file with the ".bag" suffix, which may then be played back for review or testing. Replaying the recorded data is one of the most important benefits of using rosbag. This enables programmers to test algorithms and debug software without the use of physical robots or sensors. It can also create several arbitrary scenarios and possible robot states, which can then be processed in several ways to achieve the best results.

## 2.2 Gazebo simulator

Gazebo is an open-source program that simulates real-world environments. It is primarily focused on robotics and allows developers to model and test robotic systems in a virtual environment. In addition to the graphical interface, it also includes many tools for generating and managing robots or environments. It includes a built-in physics engine. It is used especially in the research and development of new robotic devices. The developers can test their systems and robot management in a near-realistic environment. The simulator allows to test many scenarios and states in which the robot can be in. This is aided by the aforementioned Gazebo physics engine, which simulates the physical properties of the world and interaction with the environment. It is suitable for simulating natural variables and interactions such as wind, gravity, collisions, and everything that can affect the movement of the robot even in the real world. Thanks to this, the programmer can improve, and test his system. This improves the probability of the robot's functionality in the real world. Also, the simulation is used to protect both the robot itself and prevent injury to people around it. The Graphic User Interface (GUI) in the Gazebo system which is shown in Fig. 2.1, can observe the robot according to the user's requirements from different angles or perspectives. Additionally, Gazebo offers tools for creating and managing models. Thanks to the environment generation integrated in the Gazebo simulator is possible to create an environment for testing. It is possible to set up different lighting conditions or topography of the space. The Gazebo simulator also has the ability to import your own models and then position them in the simulation. The Gazebo simulator is also fully compatible with ROS. Thanks to this, it is possible to design any kind of system for robotic devices. Gazebo also includes some modules that extend its functionality and can be connected to robots. These include various sensors, cameras, and other peripherals. The developer can also use his own modules and plugins that are suitable for his project. These plugins can then be freely adapted to the user's requirements and used in many custom products[35].

Real Time Factor: 1.00 Sim Time: 00 00:03:43.352 Real Time: 00 00:03:48.315 Iterations: 55838 FPS: 63.45

Figure 2.1: Screenshot from Gazebo simulator. Includes user interface for inserting objects.

## 2.3 Radioactivity

Radioactivity is the process of releasing charged particles or electromagnetic waves from the nucleus of an atom. The emitted particles or waves are called radiation. The phenomenon of radiation can be observed in some types of atoms that have unstable nuclei. Atoms with unstable nuclei contain excess energy which is usually released in the form of ionizing radiation. Radioactivity is a natural process that persists in various elements or can be induced artificially by human activity.

### 2.3.1 Types of radiation

Radiation is divided into three types: alpha, beta and gamma radiation. Each type of radiation has different properties and characteristics.

#### Alpha radiation

Alpha radiation is the release of alpha particles from the nucleus of an atom. It consists of a helium nucleus composed of two protons and two neutrons. Alpha particles are relatively large and heavy. They have a low penetrating power, which means they can be stopped by a sheet of paper or even the outer layer of human skin. The problem occurs if swallowed or inhaled, in which case alpha particles can cause significant damage to body tissues and organs. Alpha radiation is commonly emitted by heavy elements such as uranium and plutonium.

#### Beta radiation

Beta radiation is the release of beta particles from the nucleus of an atom, which are high-energy electrons or positrons. Beta particles are much smaller and lighter than alpha particles and have greater penetrating power, which means they can penetrate several layers of human tissue. It is possible to stop them with a wooden barrier, a layer of aluminium or any material that has sufficient density to allow the beta particles to interact with matter. Despite having lower mass than alpha particles, their better transmissivity and mobility make them more dangerous. Beta radiation is commonly emitted by elements such as carbon-14 and strontium-90, which is a by-product of nuclear reactors.

### Gamma radiation

Gamma radiation is the release of gamma particles from the nucleus of an atom. They take the form of electromagnetic waves and are high-energy photons without any mass or electric charge. This makes it one of the most deadly and invasive forms of radiation. Without losing energy, gamma radiation rays can go through different materials or air. Gamma photons may travel several meters through concrete, several centimeters through lead, and, of course, the human body. Gamma rays are produced by the decay of various radioactive elements, including uranium, radium or thorium. Apart from the obvious disadvantages of gamma radiation, it is also possible to take advantage of its positive properties. Due to its high energy and ability to penetrate materials, gamma radiation is an important aspect in various fields. Its application in medicine for the radiation treatment of specific tumors is a typical example[32].

### Summary

Every radiation type has unique characteristics, that influence how it can be transferred and interact with various materials. This indicates that there are various levels of risk associated with each form of radioactivity for both individuals and the environment. For instance, gamma radiation can pass through the body and harm internal organs, whereas alpha radiation can be deadly if it penetrates the body. The key is to understand the many types of radiation that may be present in a particular environment, then identify them correctly and respond appropriately.

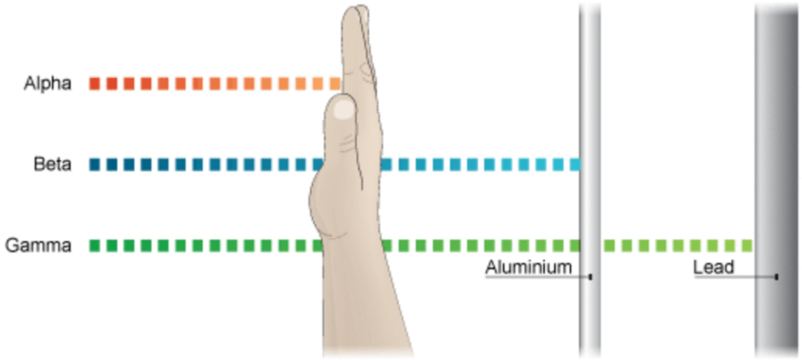


Figure 2.2: Demonstration of various forms of ionizing radiation and its penetration through materials. Source: https://owlcation.com/stem/The-Three-Types-of-Radiation

### 2.3.2 Impact of radiation

Radioactivity has a variety of helpful applications around the world. Almost every technology that helps humans today is powered by electricity. A large amount of this energy is also produced by radioactive materials. When radioactive material is present in sufficient amounts, it is possible to effectively integrate the energy into our electrical system. Another indisputably advantageous example is the beneficial use of radiation in medicine. The X-ray imaging and the treatment of some cancers[32] are the two most common medical applications. Today, radiation has become an almost indistinguishable part of medicine and electricity generation.

Although radiation has unquestionable advantages, it can also be hazardous and damage people, the environment, and the animals that live there. Environment contamination can occur in a variety of ways. The human factor and inappropriate control and storage of the radiation material are two factors that contribute to environmental pollution. The use of nuclear weapons, nuclear power plant accidents, and human-caused radioactive material leakage into the environment are all examples of this. In addition, radioactivity can also be released into the environment naturally, such as when radioactive material in rocks, minerals decays. The most dangerous characteristic of radioactive substances is their ability to rapidly contaminate the environment and not knowing about the radiation when people are exposed to it. Radiation can spread most frequently through the water or air. Pollution of the sea, rivers, and lakes has major effects not only for the local fauna and flora but also for the whole environment. When contaminated water evaporates, radiation can be dispersed through rain over distances of several tens of kilometers from its source, and contamination can even reach subsurface water supplies. Another, way of spreading contamination is through the air. Radioactive dust or fallout can spread rapidly and unstoppably through the wind over a distance of hundreds of kilometers. Radiation enters organisms through simple breathing or skin absorption. High radiation dosages are extremely dangerous to human beings. There are numerous health issues that it may cause. Radiation exposure can alter a person's DNA and lead to cancer or other disorders. It can also make a person nauseous or make them vomit. Death is the final manifestation of a high radiation dosage in a person. Radiation also poses a hazard to animal life, thus they are not an exception. Animals that are exposed to radiation develop mutations, live shorter lifetimes, or have trouble reproducing. This leads to the extinction of some species and the subsequent collapse of the food chain, which has an impact on the entire ecosystem.

The impact of radiation does not only affect living organisms. Rain and air are two ways that radiation contaminates the ground. As a result, the soil is no longer suitable for continued usage or agriculture. Fruits and vegetables produced on contaminated soil become contaminated and hazardous to the consumer. Contamination in the soil can occur over a very long period of time, on the scale of hundreds to thousands of years. This results in long-term adverse effects for its use. Cleaning such a contaminated environment is a very time-consuming and costly task. Therefore, it is important to take precautions to entirely eradicate radioactive material contamination. The regulation of ionizing radiation is handled by a number of standards and laws. These regulations establish safeguards for the environment, humans, animals, and the plants that grow there. There are treaties that govern the use of nuclear weapons and treaties for nuclear power stations to cooperate with one another in the case of an accident. Despite the fact that radiation is typically exceedingly dangerous, it can actually be extremely advantageous and useful to humans when handled properly.

## 2.4 Radiation detector

There are many radiation detection devices. Ionization chamber detectors were among the first devices capable of detecting ionizing radiation. This detector works by determining the electric field potential in a gas-filled chamber that is exposed to radiation. The Geiger-Müller detector works on a similar principle to the ionization chamber, but has a higher detection voltage. Because of the higher voltage, the sensitivity of this detector is much higher and it can detect even small amounts of radiation. The sensor has as output a value representing the number of measured particles. It cannot measure the energy or determine the type of particles measured by the sensor. The scintillation detector, which transforms ionizing radiation into photons of light, is another crucial kind of radiation detector. A high energy particle hitting a scintillator causes the release of lower energy photon. The energy of the emitted photon is proportional to the energy of incoming radiation. This process of conversion is known as luminescence. These detectors offer excellent sensitivity, but tend to be large and heavy due to the large volume of scintillating material and accompanying electronics. This makes them less suitable for use onboard small-scale UAVs.

In this work, a detector based on a completely different technology is used to find and localize gamma radiation. The detector uses a single-layer Compton camera based on the hybrid semiconductor chip Timepix3. Whole semiconductor detectors operate on the electromagnetic characteristics of the materials they are made of. Their great advantage is the direct conversion of high energy particles into electric signal. This allows the detector to be very compact while maintaining very high sensitivity, being able to detect individual particles.

### 2.4.1 TimePix3

TimePix3 is the third generation of TimePix chips. It was developed in 2014 to take the place of the MidiPix chips and be an enhanced version of the TimePix base chip. These chips were created by a collaboration based at CERN and CTU in Prague also participated in the development. This chip works on pixel semiconductor detector technology, where each photon of radiation hitting the sensor is instantaneously detected and transmitted for processing. The TimePix3 sensor consists of a 256 x 256 pixel matrix with a pixel size of 55 µm. Each pixel serves as a separate detector capable of responding to ionizing radiation. Si, GaAs, or Cadmium Telluride (CdTe) are the materials that are used to make the sensor. The sensor pixels can operate in three modes. The first mode is called **Time-of-Arrival (ToA)** mod. The **ToA** mode is used to measure the time of the radiation detection event for each pixel. The next mode is **Time-over-Threshold (ToT)**. After the threshold of the detected signal is crossed, **ToT** captures the precise time. The third mode is a particle counter, often called MediPix mode. The main difference from the previous generation TimePix is the ability to measure simultaneously in two modes of the device, **ToA** and **ToT**.

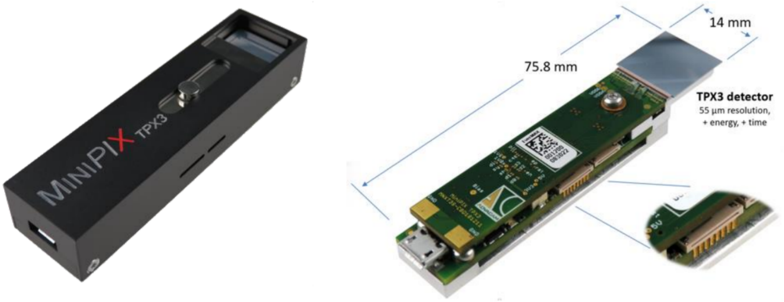


Figure 2.3: Picture of the MiniPIX TimePix3 device and a picture of the circuit board of MiniPix together with the footprint details. Source: [12]

### 2.4.2 Compton camera

The Compton camera is an ionizing radiation detector that uses the TimePix3 chip technology. The Compton camera captures the radiation particles, its detection time and energy value. Contrary to conventional radiation detectors, the camera also provides the ability to estimate the direction from which the ionizing particles came. The compact size of the camera also means that the sensor volume/area exposed to the ionizing particles is limited. The Compton camera relies on detecting the products of an event known as Compton scattering. In this event, a high energy photon changes direction and transfers a portion of its energy to an electron bound in the detector. This allows us to reconstruct the set of directions of a possible radiation source in the form of a Compton cone.

The Compton camera construction usually consists of at least two detector layers. The first layer is known as the scatterer, while the second layer below is known as the absorber. The scatterer layer is a semiconductor (e.g. Si or CdTe) and is usually thinner than the absorber layer. This layer triggers the Compton scattering effect. The contact of a photon of radiation with the first layer results in a recoiled electron. The electron's position, time of arrival, and energy are recorded by the sensor. After the scattering effect, the second layer absorbs the scattered photon and records its position, time, and energy as well. The absorber layer is usually thicker and composed of a material with a larger proton number, such as CdTe. Afterwards, the scattering angle Θ can be determined using Compton's Eq. 2.1 as illustrated in Fig. 2.4.

$$\begin{aligned} \cos\Theta &= 1 - m_e c^2 \frac{E_1}{E_0\left(E_0 - E_1\right)} \\ \bar{E}_0 &= E_1 + E_2 \end{aligned} \tag{2.1}$$

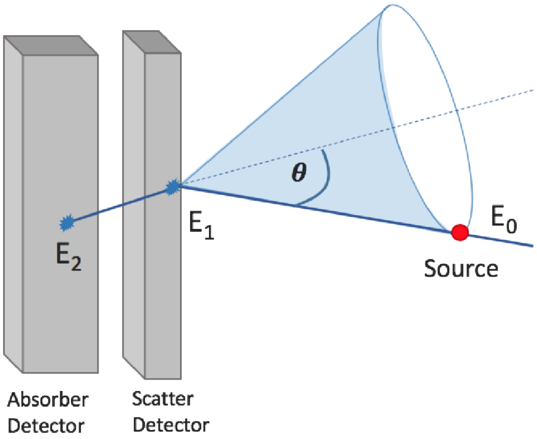


Figure 2.4: A gamma photon with energy $E_0$ emitted from the source is then scattered in the first layer. There, the reflected electron leaves the energy $E_1$ and the scattered photon is absorbed in the second layer, leaving an energy of $E_2$. Source: [20]

In this study, a modified version of the Compton camera with only one layer is employed instead of a multi layer Compton camera. The combined functionality of the scatterer and absorber is provided by a single 2 mm thick layer of CdTe. The advantages of Compton's single-layer design lie in its small size, lightweight and simple construction. The single layer Compton camera can accurately determine the position of the coordinates, where the radiation particles were detected on the sensor, but the challenge is to determine the depth. Depth is the vertical distance between the recoiled electron and the scattered photon. For depth calculation, the TimePix 3 chip is used to record the time of the detected particles by the sensor with **ToA** and **ToT** modes. Depth is then computed from the time difference between the two events and a known constant velocity of signal propagation through the sensor material.

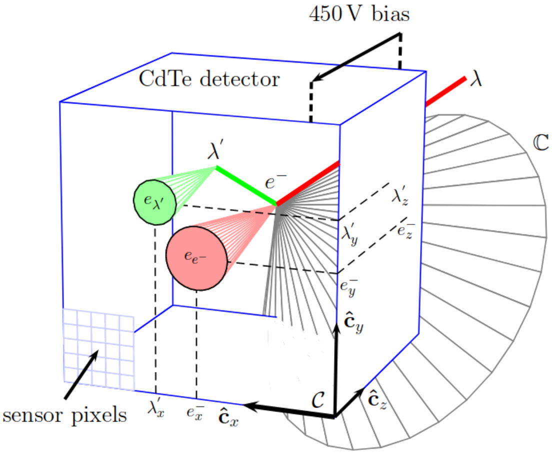


Figure 2.5: Visualisation of a single-layer Compton camera in the $C$ coordinate system. Where $e^-$ shows the position of the incident electron, $\lambda^{`}$ is the position of the incident photon and $\lambda$ is the direction towards the radiation source. $\mathbb{C}$ is the Compton cone reconstructed from the two events. $e_\lambda$ and $e_e$ show the measured footprint of the respective scattering products. Source: [6]

Fig. 2.5 represents a visualization of the processing of radiation particles. The CdTe detector has a bias voltage of 450 V with the event propagation velocity of $2325.6ms^{-1}$. If $e_t^-$ is defined as the arrival time for the scattered electron and $\lambda_t^{'}$ as the arrival time for the photon, then the depth is calculated as Eq. 2.2.

$$\Delta z = 2325.6ms^{-1}\left(e_t^- - \lambda_t^{'}\right) \tag{2.2}$$

After calculating the depth, the positions of the scattered electron and the photon are defined in the Compton camera coordinate system $C$ as $\mathbf{e_c^-} = \left[e_x^-, e_y^-, \Delta z\right]^T$ for the electron and for the photon as $\lambda_\mathbf{c}^{'} = \left[\lambda_x^{'}, \lambda_y^{'}, 0\right]^T$. The scattering angle $\Theta$ is calculated from the energies recorded by the sensor for the electron $E_{e^-}$ and photon $E_\lambda$ in units of joules from the Compton equation expressed as the Eq. 2.3:

$$\Theta = \cos^{-1}\left(1 + m_c c^2\left(\frac{1}{E_{e^-} + E_\lambda} - \frac{1}{E_\lambda}\right)\right), \tag{2.3}$$

where $m_c = 9.10938356 \cdot 10^{-31}$ kg is the rest mass of an electron, $c = 299792458ms^{-1}$ is the speed of light in vacuum, and the argument in $\cos^{-1}$ is a number belonging to the interval between $(-1, 1)$. Resulting from the computations are identified potential directions for the particle's origin or the potential location of the ionizing radiation source. The possible directions of radiation are described using the Compton cone's surface. Sec. 3.1 provides more detailed information. The definition of the parameters from the Table 3.1 required to represent Compton's cone is for the origin at point $\mathbf{e_c^-}$, for the direction of Compton's cone is determined as $\mathbf{e_c^-} - \lambda_\mathbf{c}^{'}$, and the internal angle is $\Theta$.

## 2.5 Depth measurement

Accurate depth measuring is essential in a variety of industries, ranging from navigation, cartography, and environmental monitoring to construction, resource exploration, mining, and construction inspection In this work, depth measurements are primarily used to map the environment. LIDAR and depth cameras are two relatively new techniques for measuring depth that have grown in popularity recently.

LIDAR is a type of remote depth monitoring that measures the distances of objects using lasers. During this process, a laser beam is emitted from the LIDAR sensor, which bounces back to the sensor after hitting the object. The time it takes for the laser beam to travel to and from the object is used to calculate the distance. The distances calculated by the lasers are captured according to the frequency that is set for the device. By quickly sweeping the laser beam in different directions, the surroundings may be mapped out in three dimensions. The LIDAR can measure depths accurately, with precision typically ranging from one millimeter to one centimeter.

The process of the camera consists of the camera lens emitting infrared light, which is reflected from surfaces and returned back to the lens. It is done using multiple camera lenses at the same time. The lenses have a specified distance between each other and each lens acquires different data about the same space that it captures. Then the built-in calculation unit calculates the distance of the object from the camera by combining all outputs from the lenses. The captured object more distant from the camera has a smaller deviation of data between camera lenses than the object closer to the camera. This process can also be performed without infrared light, but it gives us an advantage in poorly illuminated environments.

Two types of sensors, LIDAR and depth cameras, are employed in numerous fields, including robotics, autonomous driving, and 3D scanning. LIDAR provides a number of benefits, including great precision and accuracy, a long range, high resolution, and the capacity to identify obstacles. However, it can be expensive and its weight can also be a barrier to its deployment in robotics. Therefore, it is not very suitable for the low-performance small UAV this thesis is aimed towards. Depth cameras, on the other hand, are less expensive than LIDAR sensors, easier to use, and weigh significantly less, making them appropriate for a wide range of applications. The depth camera is shown for example in Fig. 2.6 The depth camera has a shorter range of only a few meters, lower resolution, and is not able to produce 3D point clouds with the same degree of information as LIDAR. Ambient lighting has an impact on depth cameras, which can decrease their accuracy in direct sunshine or dim lighting.

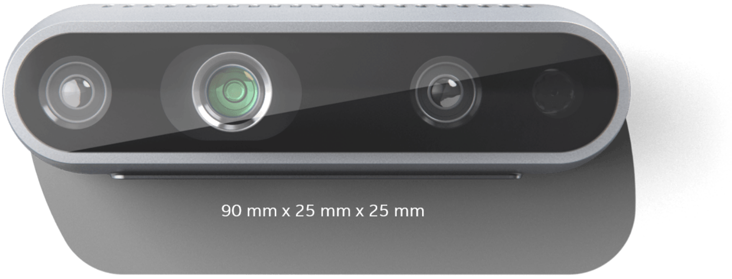


Figure 2.6: Picture of the Intel RealSense™ D435 with its dimensions. Source: `https://www.intelrealsense.com/depth-camera-d435/`

Range, resolution, field of view and price are the main differences between depth cameras and LIDAR. LIDAR is more suitable for applications in complex environments, because it can create 3D point clouds with a high level of detail at distances of up to several hundred meters in each direction, and we have sufficient financial resources to buy it. Depth cameras are more appropriate for applications where it is important to reduce each weight load and in simpler environments due to their lower accuracy, they are less expensive and have a smaller field of view.

### 2.5.1 OctoMap

The OctoMap is an open source library for C++ programming language. The OctoMap provides a highly efficient and flexible 3D mapping framework, that enables creating environment maps using depth sensors. It can accurately represent space with a probabilistic voxel grid. The OctoMap is used frequently in the robotics industry. A robot often needs to know about the space it is in and perform certain actions accordingly. When a robot moves in a complex or dynamic environment, recording the environment is very important and necessary. Using techniques like the A* algorithm or other route planning techniques, allows the robot to plan routes efficiently while avoiding obstacles.

An essential feature of the Octomap is the efficient representation of space. The OctoMap is based on the octree structure. The octree is a tree data structure that recursively partitions space into cubic cells. Each cell represents a volume in space, and the depth level of iteration in the recursion determines its size. The occupancy of the cells is determined according to a probabilistic model. The octree structure is capable of real-time updates along with newly arriving measurements from sensors. This performance is achieved by updating only the altered elements of the octree structure and avoiding needless changes to the structure. As a result of the short time requirements, the environment in which the robot is placed can be recorded very precisely and recently. Another benefit of the OctoMap is its adaptable interface, which enables quick integration into unique project designs. Because the library is open source, it is constantly evolving and new enhancements are being added. Overall, it is a very flexible and powerful library for capturing 3D environments and its constant improvement will ensure that it remains a part of robotics for many years to come[27].

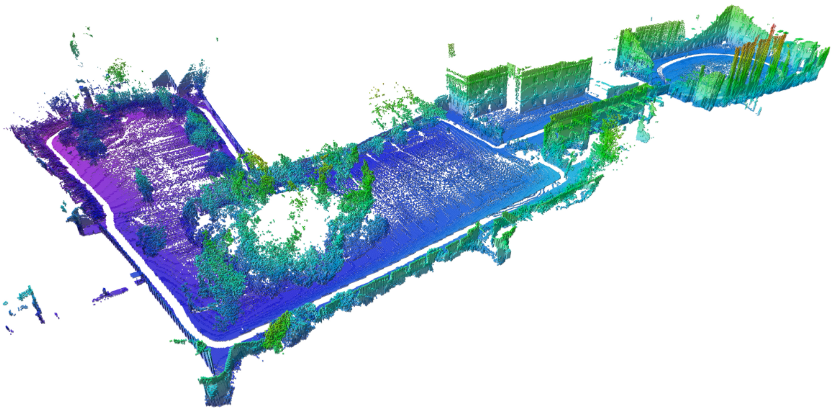


Figure 2.7: 3D visualization of the environment using the OctoMap, in which the world is divided into cubes. Source: `https://octomap.github.io/`

# Chapter 3

# Sampling

This chapter focuses on the processing of the input data obtained from the main sensors. The goal is to convert the measured values into points located in $\mathbb{R}^3$ space, which can be easily and quickly processed in the next steps needed for radiation localization. The UAV is equipped with several sensors. Each sensor has a unique purpose and is needed for the whole work to function. The dosimetric sensor is represented by a Compton camera and the depth sensor can be represented by a LIDAR or a depth camera. By combining the resulting measurements from the sensors, a small set of points can be obtained. This work presents several possible ways to obtain these points.

## 3.1 Dosimetry evaluation

This focuses on processing of the output data obtained from the Compton camera. The Compton camera is used to determine the direction in which a radioactive source could be located. The processed sensor output has a conical shape called a Compton cone. The cone is described by two vectors and two scalars that define its position, orientation and shape. All vectors are assumed to be located in $\mathbb{R}^3$. The first vector is the origin, which defines the apex of the cone. The second vector is the direction of the cone, which defines the orientation in which the cone points. The first scalar represents the angle between the direction vector and the surface of the cone. The second scalar represents the distance between the origin of the cone and its base. This parameter is not used in our calculations. Instead, it is chosen dynamically as needed. The parameter notation is shown in Table 3.1 and an example of a Compton cone in Fig. 3.1.

| | |
|---|---|
| $\mathbf{o}$ | The vector represents the origin of the cone |
| $\mathbf{d}$ | The vector of the direction vector of the cone |
| $\alpha$ | The scalar represents the angle, where $\alpha \in \left(0, \frac{\pi}{2}\right)$ |

Table 3.1: Mathematical notation of the Compton cone.

The goal is to transform the cone into the most appropriate set of points representing its surface other than the origin. Therefore, it is necessary to create a generator that has a cone description as the input and a set of points as the output. The basic principle of the generator is to find a suitable equation of a cone that can be used to sample the cone at a given resolution. The basic form used to solve the problem is the parametric equation, which is suitable for an iterative approach and computationally unconstrained. The parametric expression provides a suitable solution of the problem with a minimum number of parameter variables. The idea is to describe cones as circles.

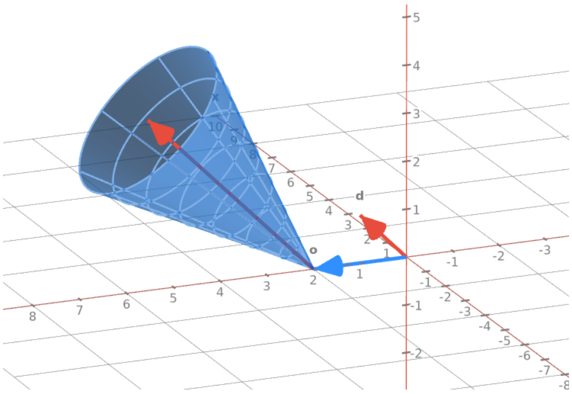


Figure 3.1: Example of a Compton cone in $\mathbb{R}^3$ space with the origin and direction vectors from the Compton cone notation Table 3.1. The blue vector represents $\mathbf{o}$ origin of cone, the red vectors represent the $\mathbf{d}$ direction.

### 3.1.1 Circle equation

A circle in $\mathbb{R}^2$ can be defined as the locus of all points that satisfy the equations

$$\begin{aligned} x &= r \cdot \cos(\theta) \\ y &= r \cdot \sin(\theta), \end{aligned} \tag{3.1}$$

where $x, y$ are the coordinates of any point on the circle, $r \in \mathbb{R}_{\geq 0}$ is the radius of the circle and $\theta \in \langle 0, 2\pi \rangle$ is the parameter. Equation 3.1 also represents the conversion from polar coordinates $(r, \theta)$ to Cartesian coordinates $(x, y)$. In this section, the cone is to be partitioned into a set of circles. For this to work, the parametric equation has to be expanded into $\mathbb{R}^3$. Let $(\mathbf{u}, \mathbf{v}) \in \mathbb{R}^3$ be unit vectors which define the plane in which the circle is located. The task of $\mathbf{u}$ and $\mathbf{v}$ is to rotate the circle according to the direction of the cone. In other words, they are the basis vectors of the circle. After adding the new vectors, it is possible to introduce the basic form of the circle equation in space of $\mathbb{R}^3$ as Eq. 3.2, Eq. 3.3.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = r \cdot \cos(\theta) \cdot \mathbf{u} + r \cdot \sin(\theta) \cdot \mathbf{v} \tag{3.2}$$

$$\begin{aligned} x &= r \cdot \cos(\theta) \cdot u_x + r \cdot \sin(\theta) \cdot v_x \\ y &= r \cdot \cos(\theta) \cdot u_y + r \cdot \sin(\theta) \cdot v_y \\ z &= r \cdot \cos(\theta) \cdot u_z + r \cdot \sin(\theta) \cdot v_z \end{aligned} \tag{3.3}$$

The equation Eq. 3.2 contains two unknown vectors $(\mathbf{u}, \mathbf{v})$ and a scalar $r$ which represents the radius of the circle. Vectors $\mathbf{u}$ and $\mathbf{v}$ are two orthogonal vectors that reside in the plane of the circle. To calculate the vectors $\mathbf{u}$ and $\mathbf{v}$ it is necessary to use the vector describing the direction of the cone $\mathbf{d}$ from Table 3.1. The vector $\mathbf{d}$ represents the normal vector of the plane defined by $\mathbf{u}$ and $\mathbf{v}$. The first step is calculate $\mathbf{u}$ from the equation $\mathbf{u} \cdot \mathbf{d} = 0$ or $\mathbf{u} = (\mathbf{d}_y, -\mathbf{d}_x, 0)^T$ and to normalize the vector $\mathbf{u}$. The calculation of the third orthogonal vector $\mathbf{v}$ is using the vector product method and normalizing the vector, $\mathbf{v} = \frac{\mathbf{u} \times \mathbf{d}}{\|\mathbf{u} \times \mathbf{d}\|}$. Now the basis vectors $\mathbf{u}$ and $\mathbf{v}$, together with the orientation of all circles according to the direction of the cone, are calculated.

The next step is to calculate the radius of the circle. The circle represents the base of the cone. For simplicity, assume that the cone has origin at $(0, 0, 0)$ and a height of one unit.

The assumption allows us to turn the problem into a triangle problem. The triangle is right-angled, where the radius of the circle represents the opposite side of the triangle, the adjecent side is the height of the cone and the hypotenuse is the length of the side of the cone. The triangle is illustrated in Fig. 3.2. The problem can already be solved by goniometric functions. In the case of calculating the radius where $adjecent = 1$ and $\alpha$ is from Table 3.1, the result is $r = \tan\alpha$, because $\tan\alpha = \frac{opposite}{adjecent}$. If the radius is not calculated according to the given assumption, but according to the assumption that the magnitude of the $hypotenuse = 1$, the result is $r = \sin\alpha = \frac{opposite}{hypotenuse}$. This approach allows us to proceed uniformly along the surface of the cone rather than by its height.

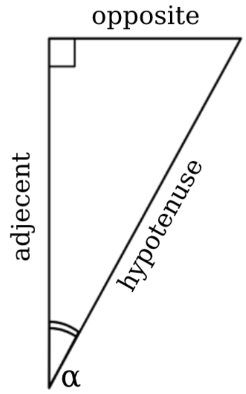


Figure 3.2: A right-angled triangle with marked sides and the $\alpha$, which represents the angle of the cone.

### 3.1.2 Cone equation

The equation Eq. 3.2 is further modified to express any point on the surface of the cone surface. It changes from a parametric curve equation to a parametric surface equation. Parametric surface equations contain two parameters. One is the interpreted $\theta$ and the other is the height parameter, which scales and shifts the circle along the cone. The height parameter is a scalar which we will recognize as $h$, where $h \in \mathbb{R}_{\geq 0}$. The cone may not be centered at the origin of the coordinate system, so the whole equation needs to be shifted by the vector $o$ from Table 3.2. As a final step to develop the equation, it remains to define the vector responsible for the direction of motion and velocity of the circle, called $\mathbf{p}$. If we define $r$ by $r = \tan\alpha$ the vector $\mathbf{p}$ is defined by $\mathbf{d}$, since it moves at unit speed. Using a uniform approach to the cone surface, vector $\mathbf{p}$ must be multiplied by $\cos\alpha$ to reduce the velocity with respect to the angle of the cone. The main equation that can uniformly describe the cone cover using parametrization, and notation in Table 3.2 is as follows

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{o} + h \cdot (\mathbf{p} + r \cdot (\cos(\theta) \cdot \mathbf{u} + \sin(\theta) \cdot \mathbf{v})). \tag{3.4}$$

| | |
|---|---|
| $\mathbf{o}$ | The vector $\mathbf{o} \in \mathbb{R}^3$ represents an offset from the origin. |
| $\mathbf{p}$ | The vector $\mathbf{p} \in \mathbb{R}^3$ represents the direction and magnitude of the step of the circle on the cone. |
| $\mathbf{u}, \mathbf{v}$ | The vectors $(\mathbf{u}, \mathbf{v}) \in \mathbb{R}^3$ describes the orientation of a circle in $\mathbb{R}^3$ space . |
| $\theta$ | The parameter $\theta \in \langle 0, 2\pi \rangle$ represents a central angle of a circle. |
| $h$ | The parameter $0 \leq h \leq H$ denotes the circle's location on the cone, and $H$ is the cone's height. |
| $r$ | Radius of the circle cut from the cone at the defined height, $r \in \mathbb{R}$. |

Table 3.2: Notation of the parametric Eq. 3.4, describing a cone.

## 3.2 Cone sampling

The section presents several methods to sample the surface of a cone by points using a parametric surface equation. Thanks to the Eq. 3.4, we can define any point on the Compton cone. The objective is to approximate the cone as precisely as possible with the lowest number of points. Low computational complexity is also a quality that the method must meet. The method must be able to generate a set of points from the cone almost instantaneously for a real-time application. The need for a fast point generation motivated the development of several cone sampling methods.

### 3.2.1 Line sampling method

The simplest of the methods is the approach using straight lines. When applying the method, the part of the cone to be processed is determined. This is determined by the start and end of the part which are defined as the distance from the origin of the Compton cone. The cone processing starts by defining points on the circle at unit distance of the circumference of the circle from the origin of the cone. The number of points that are generated depends on the input parameter. In parallel, the number of points also represents the number of lines that will be generated. Eq. 3.4 is then used to create a predetermined number of points on the line in the chosen cone portion. The line is created as a ray from the origin of the cone towards the generated point. If a point from the circle is defined as a vector $\mathbf{c}$, vector $\mathbf{o}$ is origin from Table 3.1 and $h$ is a parameter, then the resulting point $\mathbf{q} \in \mathbb{R}^3$ is calculated as

$$\mathbf{q} = \frac{(\mathbf{c} - \mathbf{o})}{\|(\mathbf{c} - \mathbf{o})\|} \cdot h + \mathbf{o}. \tag{3.5}$$

Then, a certain number of points on the line are sampled using the parametric equation. The sampling process is described in Alg. 1.

**Algorithm 1** Compton cone sampling with lines

```
function SampleConeLine($startHeight, endHeight, lineCount, pointsPerLine$)
    $step \leftarrow \frac{2\pi}{lineCount}$    ▷ Dividing the circle into steps.
    $heightShift \leftarrow \frac{endHeight - startHeight}{pointsPerLine}$    ▷ Dividing the line into steps.
    $\theta \leftarrow 0$
    while $\theta < 2\pi$ do
        $pointOnCircle \leftarrow generatePoint(1, \theta)$    ▷ Generating a point using Eq. 3.4, $h = 1$.
        $lineHeight \leftarrow startHeight$
        while $lineHeight < endHeight$ do
            $pointOnLine \leftarrow getPointOnline(pointOnCircle, lineHeight)$    ▷ Using Eq. 3.5.
            $points$.Add(point)
            $lineHeight \leftarrow lineHeight + heightShift$
        end while
        $\theta \leftarrow \theta + step$
    end while
    return $points$
end function
```

Method SampleConeLine needs the least computational time of all the methods listed in this work, because it describes only a few points using the equation of a circle Eq. 3.4. However, it has one significant disadvantage. The problem is that the distance of the points between the lines is proportional to the distance from the origin of the cone. Inaccuracy can have a serious impact on the accuracy of the next measurement. The other listed method tries to solve the problem, but at the expense of performance. An example of this sampling method is shown in Fig. 3.3. To reduce the error, which line sampling has with the proportion, ray tracing is used to calculate the point set instead of the parametric Eq. 3.5 , as described in more detail in Sec. 3.3. The solution to the problem can also be to set a higher line density during sampling.

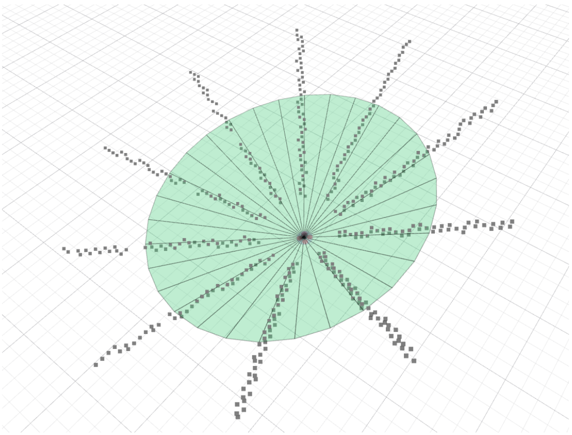


Figure 3.3: Example of line sampling with ten defined points on a circle. Lines are created by casting ten rays using ray tracing.

### 3.2.2 Uniform sampling method

The most accurate and controllable way to approximate cones is by a uniform sampling. The technique is distinguished by the fact that all neighboring points are equally far from one another. The algorithm, like the earlier ones, is capable of describing a particular region of the Compton cone. The part of the cone to be processed is defined by the distance of the start and end from the origin of the cone. The parameter that controls the distance between the generated points is called range. The algorithm starts by processing the cone from the defined start distance of the segment to the end, proceeding with the step of range. At each step, a slice of the cone has to be taken using the circle Eq. 3.4. The circle is then sampled as described earlier. The points on the circle are then determined from the equation. The sampling step on the circle is derived as the range divided by the height of the cone at this distance. The results of the algorithm are points in $\mathbb{R}^3$ representing the cone. The biggest disadvantage is the added computational load caused by defining of the points using just the parametric equation of the circle in space. The solution is therefore to not generate more points than needed, i.e. to reduce the sampling rate. An example of this sampling method is shown in Fig. 3.4. The process is procedurally described in Alg. 2

**Algorithm 2** Uniform Compton cone sampling by range

```
function SampleConeUniform($startHeight, endHeight, range$)
    $height \leftarrow startHeight$
    while $height < endHeight$ do
        $step \leftarrow \frac{range}{height \cdot \sin\alpha}$        ▷ Dividing the circle into steps according to range.
        $\theta \leftarrow 0$
        while $\theta < 2\pi$ do
            $point \leftarrow generatePoint(height, \theta)$        ▷ Generating a point using Eq. 3.4.
            $points$.Add(point)
            $\theta \leftarrow \theta + step$
        end while
        $height \leftarrow height + range$        ▷ Height shift according to range.
    end while
    return $points$
end function
```

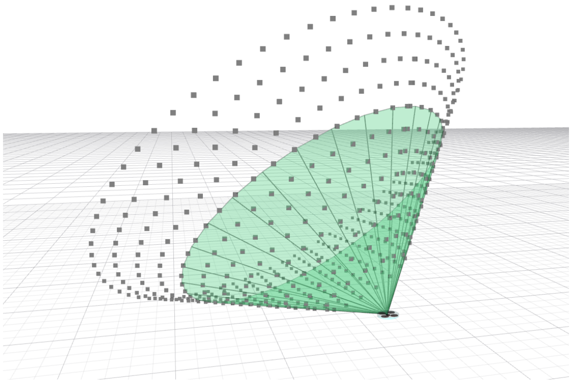


Figure 3.4: Example of the Uniform sampling, with range equal to 1

## 3.3 Measurement fusion

After applying one of the algorithms from Sec. 3.2 to process the Compton cone, the set of points that need to be processed is obtained. Hundreds of points are used to accurately represent the entire surface of the cone. The high sampling density leads to a longer time required to process the points. To reduce the time required, it is necessary to limit the number of active points. Some of the generated points may not be useful for locating the radiation source. A typical example are the points that do not pass through any object, because the radiation source is assumed a physical object, not a gas. The above assumption allows us to exclude points that do not pass through objects. This speeds up the data processing and allows prior exclusion of unlikely radiation locations.

To achieve the restriction of generated points, it is necessary to obtain information about the space in which the Compton camera is located. Depth measurements taken by the sensors are used to map the environment. The combination of dosimetric and depth measurements is therefore an effective option to access processed data faster. Depth measurements are taken with a LIDAR or a depth camera and then processed as a PointCloud. Thanks to the OctoMap library[1] and the OctoMap server[2], the PointCloud is then converted to the octree structure. The octree structure enables the efficient representation of $\mathbb{R}^3$ space in which the UAV operates. This is necessary, as the overall goal is to localize potential radiation sources in $\mathbb{R}^3$. To implement the enhancement, the point sampling algorithms need to be modified. The idea is to associate the generated points with voxels in the OctoMap.

### 3.3.1 Ray tracing

Additionally, ray tracing from the OctoMap library is utilized for the octree. Ray tracing in the thesis serves as a replacement and refinement of point generation in three-dimensional space instead of the sampling method described in the Sec. 3.2.1. Ray tracing searches for voxels[3] in the space from which the OctoMap is constructed. The found blocks are then transformed into a set of points that describe the ray and eventually the Compton cone's envelope. A ray is defined by its start and end points in space. When using Eq. 3.5 the generated points are similar to the points obtained by ray tracing, but not exactly identical.

The ray-tracing method is based on the principle [40] by Amanatides and Woo. The method first determines if the points are a part of the OctoMap and calculate the ray direction and voxel sizes beforehand. The ray tracing algorithm begins with the voxel at the start point and gradually moves in the direction of the defined ray. When selecting the next voxel, the algorithm selects the neighboring voxel in the direction of the dimensions and finally selects the one closest to the beam towards the front. By its approach, the algorithm achieves the optimal selection of voxels whose coordinates are converted into a set of points. The algorithm terminates when it reaches a voxel that represents the endpoint of the ray. Ray tracing in fusion with depth measurements improves the Compton cone sampling so that each voxel passing through the ray is a newly generated point. The improvement is especially noticeable with objects close to the UAV, where the beams are placed close together and ray tracing generates a point for each voxel in the OctoMap. The ray tracing and also generates a point for each voxel in the OctoMap passing through the ray.

---

[1] `https://github.com/ctu-mrs/OctoMap_mapping_planning`

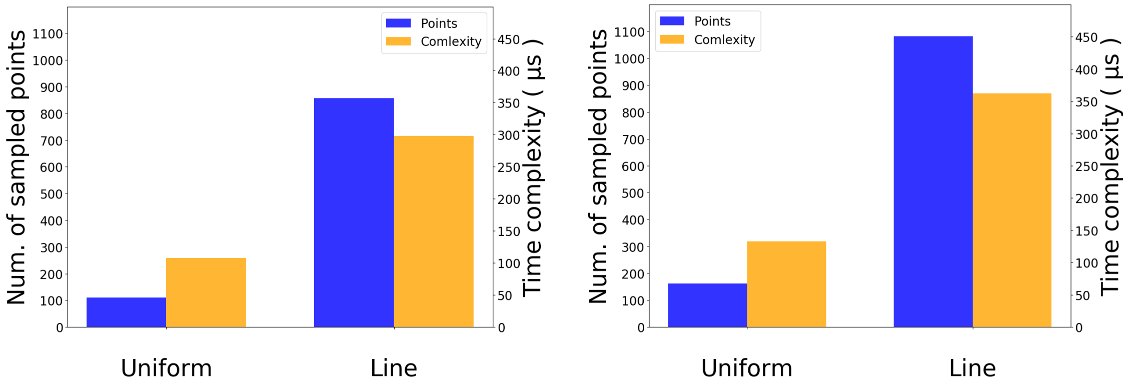
[2] `https://github.com/ctu-mrs/mrs_OctoMap_server`

[3] a cube cell into which the OctoMap is divided

## 3.4 Comparison

The section discusses the advantages and disadvantages of the developed sampling algorithms. Cone sampling is performed by two algorithms. Sample line method from the Sec. 3.2.1 and Uniform sampling method from the Sec. 3.2.2. Both algorithms use a different way of sampling the surface of the Compton cone by using the same parametric equation. The essential part is the fusion of the sampled radiation data with the depth measurements. The fusion is also performed differently for each algorithm. The SampleConeLine algorithm uses ray tracing for the highest accuracy. Ray tracing takes advantage of all available data from depth sensors represented by the OctoMap. The uniform sampling on the other side first generates points describing the cone independent of depth measurements. Then, the algorithm checks the overlap between the points and the OctoMap data defined as occupied. Various approaches to data fusion, achieve different results in time complexity or number of points in the set obtained by the sampling algorithm.

Different environments created with the OctoMap are used to compare the sampling algorithms. The test is performed on a simple environment defined as a flat surface without obstacles and for comparison an environment with obstacles is also created, representing a real deployment. The algorithms are tested in a way where thousands of randomly generated Compton cones are sampled in each of the environments. The data are recorded and then displayed in the following plots Fig. 3.5a and Fig. 3.5b. It is important to note that the time complexity of the algorithms for each test is on the order of microseconds. This demonstrates the possibility of real-time deployment.



(a) The graph shows data obtained by comparing sampling algorithms in an obstacle-free environment.

(b) The graph shows data obtained by comparing sampling algorithms in an environment with obstacles.

Figure 3.5: The figures show a comparison of the two sampling algorithms developed in this thesis. The data is obtained by testing in an environment without (a) and with obstacles (b). The graph is a bar graph where each algorithm is represented as two columns. The first columns for each algorithm represent the average number of points generated during sampling and the second column is the average time complexity of the algorithm. It can be seen that when using a uniform algorithm, the computational cost is significantly lower at the expense of generating fewer points.

The first of the graphs represents the comparison of algorithms in the environment without obstacles and the other with obstacles. The data displayed on the different graphs have elements of similarity. In Fig. 3.5, it can be clearly seen that sampling in a complex environment has an effect on the number of points that are generated during the fusion and this increases the time complexity over time. With the Uniform Sampling, it can be seen that all the values are lower than those measured with Line Sampling. This is mainly caused by the default value at the input of the sampling algorithm. In this thesis, the value of the range parameter is set to 1. This value is empirically set according to the resolution of the OctoMap at which the fusion is performed. When the value is reduced, the number of points generated increases due to the increased density of the sampling algorithm. More information about each algorithm is offered by pairwise comparisons, such as the number of sampled points and the time required for their generation. It can be seen that the uniform algorithm needs proportionally more time to process the Compton cone than the Line algorithm with respect to the number of points generated. Thus, the time complexity is a disadvantage of the uniform algorithm. However, the overall time complexity of Uniform processing is lower than using Line Sampling because the total number of points generated is lower. The Line algorithm, because of the number of points, has an increased time complexity in the next stages of the work. The performance of the Line Sampling algorithm could be improved by adjusting the number of samples or the OctoMap resolution, but the performance of the Uniform sampling was deemed satisfactory. The Uniform sampling is the preferred method and is used in the following sections.

# Chapter 4

# Radiation source estimation

The aim of this chapter is to describe how to find possible locations of a radiation source. There are many methods for estimating the position of radiation, but this article presents a method focusing on the processing of data obtained from discretization of measurements. The data to be processed are presented in the Sampling chapter. The data are acquired sequentially at a certain frequency, and are also represented as a set of independent points. Therefore, the best choice is to use an iterative algorithm capable of continuous updates with new incoming measurements. The algorithm should also sort these points into essential and non-essential. An important part of the work is to detect not just one, but multiple sources of radiation. The sources must therefore be found using an algorithm that does not converge to a single location in the environment. The algorithm must also be fully modular and easily re-configurable, as the mission of parameters and objectives may change during the development or even on the fly. One of the algorithms that can analyze incoming data and filter out the unnecessary ones is the particle filter.

## 4.1 Particle Filter

Particle filter is a sequential method based on the Monte Carlo principle designed to solve nonlinear and non-Gaussian problems. The method's purpose is to estimate the present but unknown probability density in the state space in order to derive assertions about a dynamical system's most likely state. More information can be found in [36]. The method uses particle sets to represent probability distribution and can be used with various forms of a state space model. The state model may be nonlinear, and the particle set, including the noise distribution, may take any form. The particle set may also be called a sample.

Nowadays, the particle filters are applied in many industries such as data science, robotics, statistics, medicine, and reinforcement learning. The particle filter solves many problems in applied robotics. A typical example is tracking the position of dynamic objects in space, such as drones or underground robots. If the dynamic object is the robot itself, a particle filter may be used for localization within the environment using its onboard sensors[33][30]. Another problem common in robotics is human tracking. There are many ways to tackle this problem[39][29], and the particle filter method is one of them [38]. Multimodality is also a characteristic of the particle filter, which allows tracking of not only one, but many objects at the same time.

The necessity for high volumes of data is one of the issues that these algorithm face. A vast number of samples must be input in order to acquire the desired results. Furthermore, there is no clear demonstration of particle filter algorithm's convergence[37].

## 4.2 KD-Tree

An essential component for speeding up data processing in this work is the KD-Tree. KD-Tree is a data structure that is used for more efficient particle searching. The structure stores particles in a binary tree, whose levels are distributed among three dimensions. There, each dimension represents a different coordinate axis of the particle. In comparison to a linear search, the point search algorithm in a KD-Tree has an average time complexity of $\mathcal{O}(\log n)$, which unquestionably speeds up the algorithm.

The KD-tree is built by recursively splitting the space by finding a midpoint along one of the coordinate axes for each level of the tree. Points to the left of the midpoint are inserted into the left subtree, and points to the right of the midpoint are inserted into the right subtree. For each of the remaining axes, this process is carried out repeatedly until all points have been inserted to the tree. The KD-tree can be used for various spatial queries. For this work, a custom KD-tree has been created and it is possible to find the nearest neighbor of a given point, find all points within a certain distance from a given point, or to find the N closest points to a given point. These queries are effective because a significant percentage of the search space can be pruned using the tree's hierarchical structure.

## 4.3 Estimation algorithm

The estimation of the radiation source location is using an iterative algorithm. The algorithm is based on the particle filter principle. The particle filter takes the sampled cones as the input for each iteration of new particles generation. The particles represent the possible locations of the radiation source as described in Chapter 3. The particles are compared and ranked with the particles from the previous measurements. The principle of one iteration of the algorithm is described in the following steps.

1. Accept new particles from the latest measurement generated by the sampling method.
2. Assign a weight to each new particle using the stored particles.
3. Update the weights of the stored particles according to the new measurement.
4. Merge new and stored particles.
5. Exclude the particles which are unlikely to represent a radiation source.
6. Store particles that have not been excluded.
7. Find potential radiation sources' locations using stored particles.

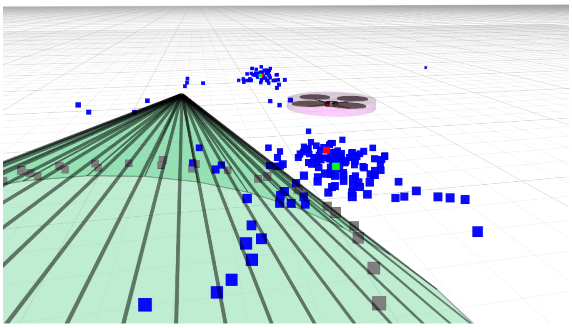


Figure 4.1: 3D visualization of particles processed by the particle filter algorithm. The green object represents one of the measured Compton cones. The grey points are created using the sampling method from Sec. 3.2. The blue points represent the stored particles from the particle filter algorithm. The red points represent the actual position of the radiation source, the green point is the radiation source estimate from the filter.

The algorithm is run on each new input which updates the measurement result. The results of the algorithm after each update should be more accurate and better approximate to the actual radiation source's position. A key part of the algorithm is the model method that assigns weights to all new particles. The weights of a particle are computed both for new particles and for particles already stored. It is the basis of the algorithm according to which the particles in the next step are excluded. Point elimination enables data processing with a low computational load and the use of a real-time particle filter. An example of the functionality of the particle filter algorithm can be seen in Fig. 4.1. In the next sections, several models are introduced and tested for calculating particle weights.

### 4.3.1 Average model

The first model for calculating the weights is called the Average model. This model determines the weights for particles based on the distance of the nearest particles. The model aims to keep the particles, which are close to other particles in the incoming measurement and to discard the most distant particles. The model contains two parameters that determine the number of particles in the calculations. The first parameter determines how many particles from the dataset are used to calculate the weight of a new incoming particle. The second parameter determines how many particles are used when updating the weights of the stored particle by the incoming particle. The model starts by accepting a new measurement. Once accepted, for each particle in the new measurement, the N nearest distances between the stored particles in the dataset and the selected particle are found. The number N is given as an input parameter. At the same time, the nearest distances for a particle in the dataset are similarly found. The calculation of the particle distance is replaced by the squared distance instead of the Euclidean distance to improve computational performance. The square distance is defined as the Eq. 4.1, for two particles in $\mathbb{R}^3$ space.

$$d = (x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2 \tag{4.1}$$

After finding the nearest distance to the measurement, its weight is calculated as

$$w = \frac{\sum_{t=1}^{n} d_t}{n}. \tag{4.2}$$

In Eq. 4.2 $w$ is the particle's weight, $n$ is the number of the nearest distances, and $d_t$ is the distance from another particle. For a particle in the dataset, the weight is calculated as

$$w = \frac{w_p \cdot c_{w_p} + \sum_{t=1}^{m} d_t}{c_{w_p} + m}. \tag{4.3}$$

The equation handles updating the weight for the particle, where it uses the weight from the previous iteration $w_p$ and the new average distance. The variable $c_{w_p}$ is the total number of particles used in the calculation of the previous weight, $m$ is the number of the nearest distances obtained from the input parameter, and $d_t$ is the distance. After updating the weights, the particles are merged and sorted according to the newly found weights. The particles with the worst weights, which have the largest average distances do not get inserted into the capacity-limited dataset. The functionality of this model is repeated periodically for each new measurement.

The Average model can have different settings that establish the input parameters. The weight of a particle can be calculated using the all-to-one method, where here the weight is calculated as the average of all other particles. Another method is the one-to-one, where the weight is calculated based on the nearest particle only. In developing the algorithm, several different combinations of parameters were tested using percentages of the total number of particles. The most appropriate combination of parameters is to calculate weights from the 10 percent of the nearest newly arriving particles and from all the stored particles in the dataset.

### 4.3.2 Worst model

The Worst model uses a similar technique to the Average model to calculate the weights. The main difference is that instead of the best distances, it calculates the average of the worst ones. This allows us to approach the particles differently and to focus on removing mainly the most outlying particles, i.e. the particles which are the least probable for the presence of a radiation source. It requires two parameters. One specifies the number of particles used from the dataset to calculate the weight of a new incoming particle. The second parameter specifies the number of particles used in updating the stored particle by the incoming particle. The calculations are performed according to Eq. 4.2 for the new particles and Eq. 4.3 for the stored particles. The difference from the Average model in the equations is that the model takes M and N as the number of the farthest measured distances. At the end, the particles with the largest weights, where the weight represents the average distance to the farthest particles from the other measurements, are removed. The number of particle removals is based on the capacity of the model dataset.

### 4.3.3 Surrounding model

The most adaptable filter model is the Surrounding model. This model calculates the weight of a particle according to whether it hits or exists in its surroundings. In the surrounding model, each particle has a counter which records the number of hits. A hit for a particle means that there is a distance between it and another particle that is less than a specified threshold distance. The working principle of the model is described in the following steps.

The initial step is to go through all the particles already stored in the model dataset and for each particle, the number of hits in the new measurement is counted. At the same time, the hits for the particles from the new measurement are also counted using the particles already in the dataset.

The next step is to evaluate the weights of the particles. The evaluation of the weights is divided into two parts. The first part solves the hit count for particles already in the dataset and the second for newly acquired particles. The weight of a particle in the dataset is determined as follows. If the number of hits exceeds a threshold value, typically set to zero, then the weight of the particle decreases by a specified amount. If the number of hits exceeds the hit threshold then the weight increases by a specified amount. For the newly received particles, the weight is determined differently. The possibility of adaptation for not yet explored particle positions was also considered in the model design, as well as the possibility of moving the UAV to a different location. This is why the weights for the new particles are calculated from the weights of the dataset particles. The weight value is obtained as the weight of a particle at a certain position in the dataset. If the number of hits for a new particle is above a threshold, the particle has a weight gained from a better position in the dataset than particles that do not reach the threshold.

The last step is to merge the particles of the new measurement and the dataset. Subsequently, the particles with the largest weights that no longer fit into the limited capacity of the dataset are discarded. After this step, the iteration of the model is done, and it is ready for the next input of new particles.

A keyfeature of this model is that the particle weights increase during the source search instead of decreasing. This leads to a significantly higher number of particles that, although they may represent the position of the radiation, do not interfere with other input data measurements.

## 4.4 Estimation clustering algorithm

The filter output is too inaccurate to be used directly. The output contains a lot of information in the form of positions where radiation sources could be located. Ideally, each radiation source should be represented as a single point in $\mathbb{R}^3$ space. To implement such a solution, an algorithm is developed that processes the output data of the particle filter and returns only the estimated positions of the radiation sources. The algorithm is based on BFS[1] and divides the particle into clusters, where one cluster represents one position of the radiation source. The clusters are generated according to two parameters, the first describes the minimum positional distance between two particles belonging to the same cluster. The second parameter expresses the minimum particle number at which the cluster is considered a radiation source. In addition to the clustering parameters, the number of highest-rated particles from the dataset can also be determined, which is taken into account when applying the algorithm.

The algorithm starts by selecting an initial particle, which is the first particle without an assigned cluster in the filtered dataset. The algorithm then finds all neighboring particles that are within a certain threshold distance from the starting particle. These neighboring particles are added to the cluster and their positions are used to update the estimated position of the source and find other particles belonging to the same cluster. This process is repeated for each unclustered particle in the particle dataset until all clusters have been found, and all particles have been processed. The input parameters are designed to handle noisy measurements and

---

[1] Breadth First Search – an algorithm for traversing graph vertices

tolerate missing or incomplete data. It works well in situations where the sources are widely separated and the noise level is low. However, it may not work well in situations where the sources are close together below the input parameter threshold or when the noise level is high. After all, clusters have been found, the algorithm calculates the average position of the particles in each cluster and returns these positions as the estimated positions of the radiation sources. Estimated positions can then be used for further analysis and work. These positions are provided as an assumption for the user about where the radiation source might be located.

## 4.5 Model comparison

Algorithm benchmarking is a key discipline in data science these days. Its goal is to determine which algorithm can solve a given problem more accurately and quickly, which can lead to more efficient problem-solving and a reduction in the time required for a human to apply the algorithm. Several variants of models for the particle filter have been developed for this thesis. The three most notable models have already been introduced in Sec. 4.3. In the following paragraphs, these models are compared and their positive and negative properties are discussed. The comparison is done by running all three filters in a simulated situation that might occur in a real world application. In order to perform the comparison, different situations had to be recorded as rosbags. An application was created, that can play these rosbags and simultaneously run different versions of the model for the filter at the same time. Also, this application can use a ROS service to change the model parameters, enable, and reset the model in real-time. The results obtained during the measurement are continuously recorded in CSV[2] files for further processing. The post processing of the recorded CSV files is performed using Python scripts that convert the data into graphs suitable for analysis and accuracy representation of the models

### 4.5.1 One radiation source

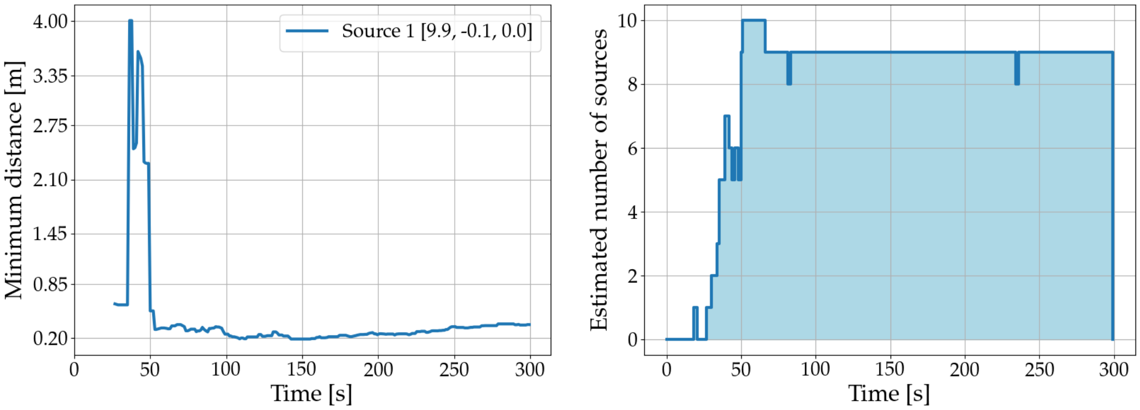
The basic situation, which is demonstrated in the first comparison, is the detection of a single radiation source. The source is static and placed in an empty environment with no obstacles. The source is located approximately 10 meters from the sensor on the UAV. The objective is to detect the location of the source using the incoming photons, which are captured by a simulated Compton camera and processed into Compton cones. All acquired Compton cones are then processed using the uniform sampling Alg. 2. The sampling converts the measurements to a set of points in three-dimensional space. Simultaneously, the UAV scans the space using a LIDAR placed on top of its frame. Next, the depth measurements are fused together with the sampled set of points to form particles for input to the filter.

#### Static UAV

In the first case, the UAV is not moving and just hovers idly near the radiation source for 5 minutes. The Average model is demonstrated as the first model for data filtering in the scenario. The model is run for the whole testing period and its output particles are updated with each new incoming measurement. The particle output from the model is further processed by the estimation clustering algorithm from Sec. 4.4, into the form of possible radiation sources. The results of the scenario can be seen in the following graphs Fig. 4.2.

---

[2] comma-separated values



(a) Graph of the distance of the nearest estimated radiation source to the actual source over time.

(b) Graph showing the number of radiation sources estimated by the filter over time.

Figure 4.2: The results of a single radiation source detection using the Average model (Sec. 4.3.1) and a static UAV.

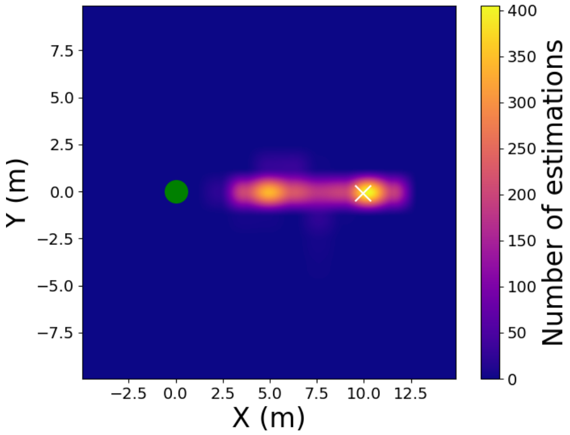
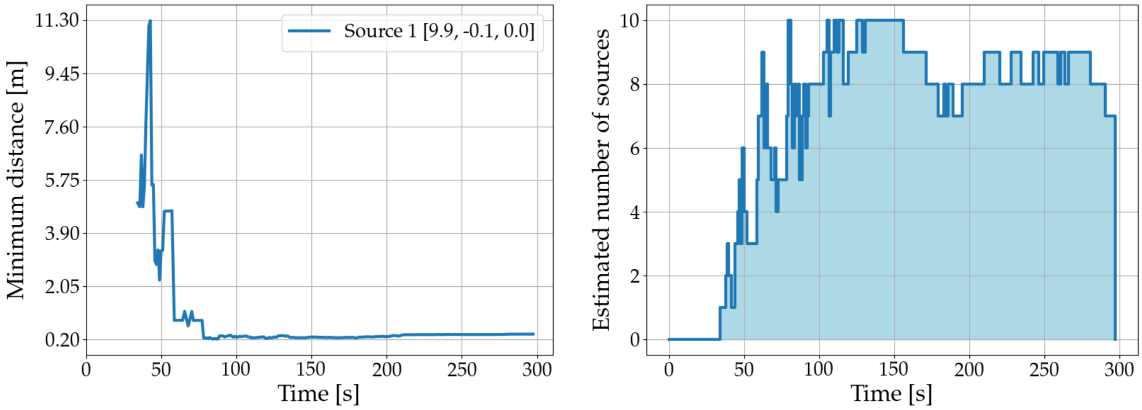


Figure 4.3: Density map of estimated radiation sources for one actual radiation source with static UAV. The density map describes the view from above in two-dimensional space. The white marker represents the real position of the radiation source. The green dot represents the position of the UAV. The color in the map represents the number of points estimated during the measurement. A lighter color of the map means a higher chance of finding a radiation source.

The measurement results appear highly accurate in the first graph Fig. 4.2a. It can be noticed that the estimation of the radiation source by the Average model is highly accurate throughout the measurement. After the initial measurement phase of about 50 seconds of the experiment is over, the distance between the detected radiation source and the real one is never more than 0.5 meters. Fig. 4.2b does not represent the accuracy of the measurement, but it can be noticed that numerous radiation sources are estimated after the initial phase is exceeded. This causes a serious problem in the actual detection of radiation. In Fig. 4.2a it is possible to tell that one of the estimated sources is close to the radiation, but it is not possible to determine which estimated source is the close one without other necessary information. Fig. 4.3 represents wherever the radiation source has been estimated. In a static measurement, all estimated points are positioned in the direction of the radiation source from the UAV's position.



(a) Graph of the distance of the nearest estimated radiation source to the actual source over time.

(b) Graph showing the number of radiation sources estimated by the filter over time.

Figure 4.4: The results of a single radiation source detection using the Surrounding model (Sec. 4.3.3) and a static UAV.

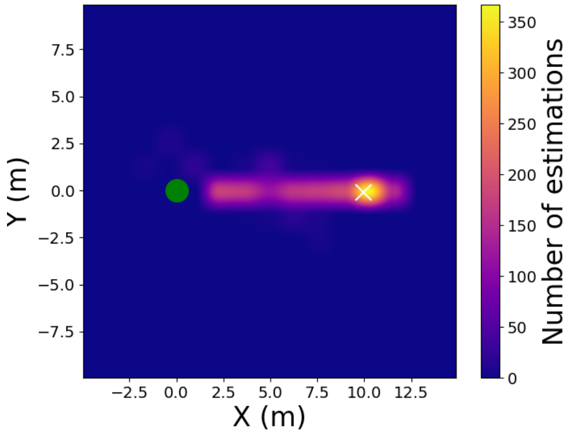
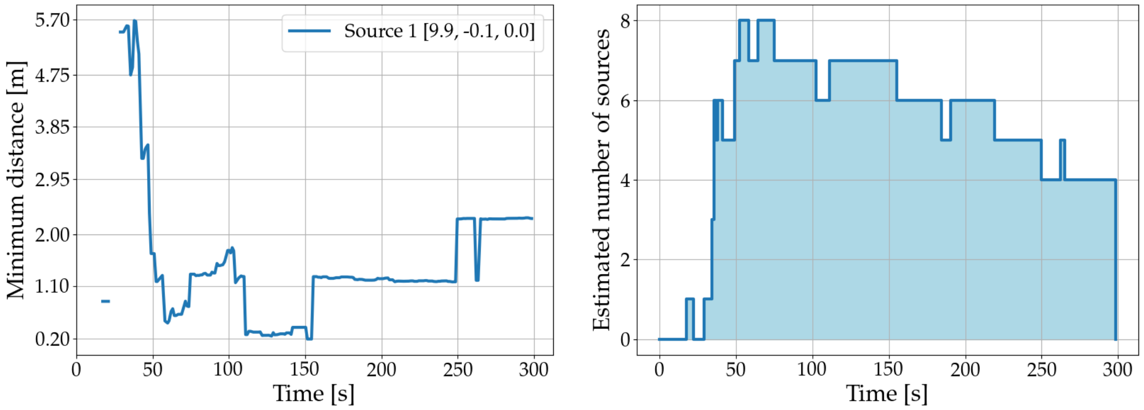


Figure 4.5: Density map of estimated radiation sources for one actual radiation source with static UAV. The density map describes the view from above in two-dimensional space. The white marker represents the real position of the radiation source. The green dot represents the position of the UAV. The color in the map represents the number of points estimated during the measurement. A lighter color of the map means a higher chance of finding a radiation source.

When using the Surrounding model, it is possible to notice the similarity to the previous model. From Fig. 4.4, it can be noted can be deduced that there is also a number of estimated radiation sources and it is not possible to determine which one represents the real source of radiation. A difference between the Average model and the Surrounding model can notice the changes in behavior in the initial phase. In contrast to the Average model, the initial phase lasts longer but slowly converges over time. As with the Average model testing in the previous case, the estimation graph Fig. 4.5 shows that all estimations are in the direction of the radiation source.



(a) Graph of the distance of the nearest estimated radiation source to the actual source over time.

(b) Graph showing the number of radiation sources estimated by the filter over time.

Figure 4.6: The results of a single radiation source detection using the Worst model (Sec. 4.3.2) and a static UAV.

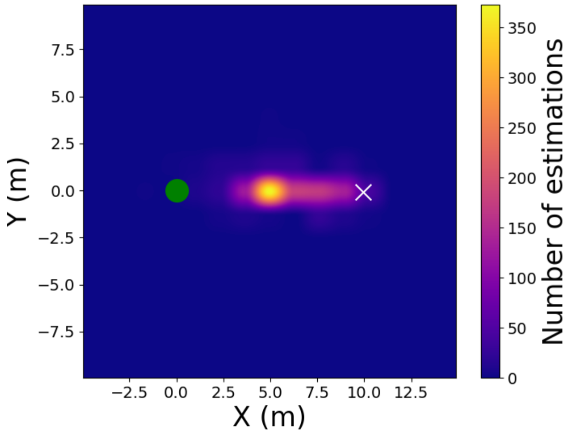


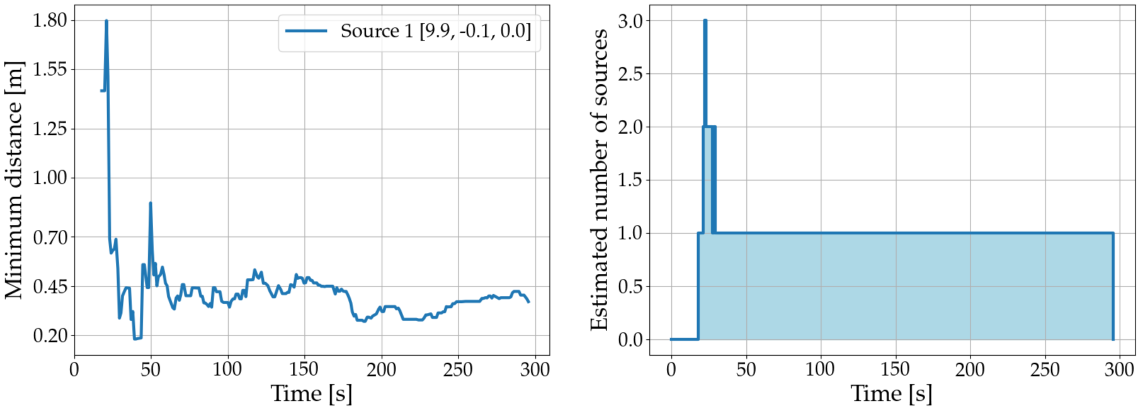
Figure 4.7: Density map of estimated radiation sources for one actual radiation source with static UAV. The density map describes the view from above in two-dimensional space. The white marker represents the real position of the radiation source. The green dot represents the position of the UAV. The color in the map represents the number of points estimated during the measurement. A lighter color of the map means a higher chance of finding a radiation source.

The last model tested for filtration is the Worst model. The results in Fig. 4.6 are quite different from the other models. The minimum distance measurement in the graph Fig. 4.6a starts to decrease, but with one new measurement the distance increases above 1 meter and the number of estimated sources in the Fig. 4.6b decreases. From the observed number of estimated sources which start to decrease over time and the increased minimum distance, it is reasonable to assume that the estimated sources start to converge to a certain location which is not our real radiation source. Same as for testing other models with static UAV, the estimation graph Fig. 4.7 shows that all predictions are from UAV in the direction of the radiation source.

In the summary of the tests of all models it is possible to notice one thing in common. All models estimated more radiation sources than the actual number of sources. This can make the environment look much more dangerous than it actually is. This problem is caused by the frequent sampling of Compton cone points in close proximity to the UAV. Therefore, the model cannot determine the actual position of the radiation source, since all the Compton cones originate in the same area and their overlapping areas are much larger. However, under these difficult circumstances, it is possible to determine the direction in which the radiation source may be located from the estimated sources and particle output of the model. This statement can be seen in each of the graphs containing the positions of the estimations.

### Moving UAV

In this scenario, the environment is similar, there is one source of radiation that the UAV is trying to locate. The source is located in the same place as for a stationary UAV. The difference is that the UAV is moving and actively estimating on the fly. The movement of the UAV forms a circular trajectory centered at the origin of the coordinate system. The trajectory is positioned three meters above the ground with a circular radius of ten meters. This trajectory allows the UAV to approach and move away, and it even passes over the source. All three models are tested, one simulation run takes almost 5 minutes. The results of all three models in this scenario can be seen in Fig. 4.8, 4.9 and 4.10

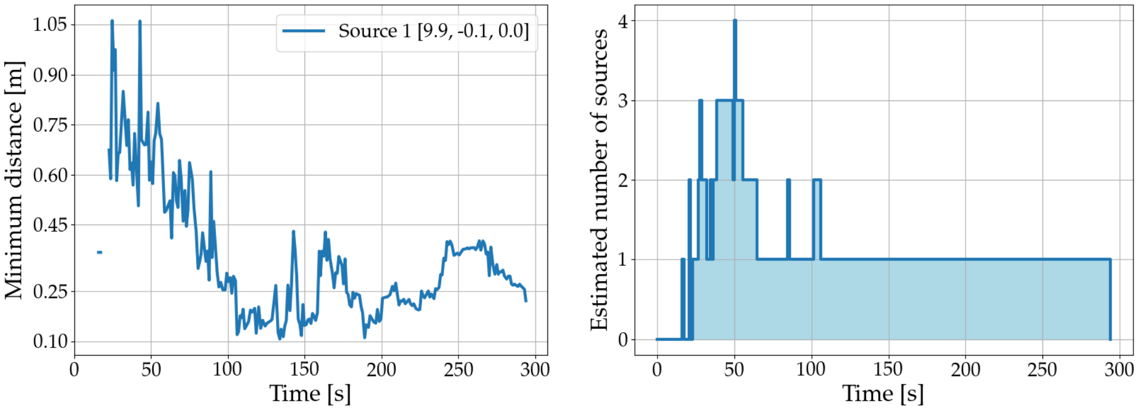


(a) Graph of the distance of the nearest estimated radiation source to the actual source over time.

(b) Graph showing the number of radiation sources estimated by the filter over time.

Figure 4.8: The results of a single radiation source detection using the Average model (Sec. 4.3.1) and a moving UAV.

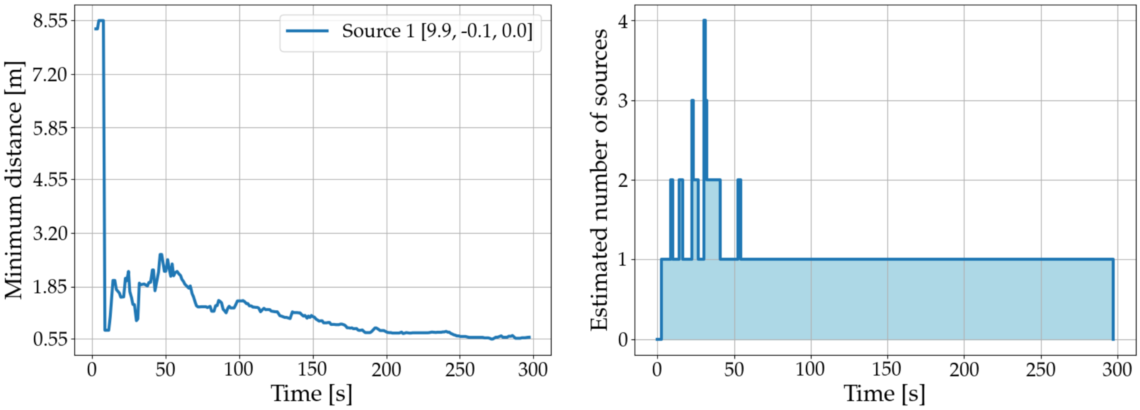
As shown in Fig. 4.10a and Fig. 4.10b, the testing results for the moving UAV are significantly more applicable with respect to the number of simulated radiation sources. Hence, when comparing model results from static measurements, the number of estimated sources not only agrees with the real number of sources, but at the same time offers a detection accuracy of less than 1 meter. This ensures unambiguous localization and identification of the radiation source. A better comparison of the moving and static scenarios is offered by Fig. 4.11. This indicates that a Compton camera moving around the source clearly helps to



(a) Graph of the distance of the nearest estimated radiation source to the actual source over time.

(b) Graph showing the number of radiation sources estimated by the filter over time.

Figure 4.9: The results of a single radiation source detection using the Surrounding model (Sec. 4.3.3) and a moving UAV.



(a) Graph of the distance of the nearest estimated radiation source to the actual source over time.

(b) Graph showing the number of radiation sources estimated by the filter over time.

Figure 4.10: The results of a single radiation source detection using the Worst model (Sec. 4.3.2) and a moving UAV.

identify the location of a single source as contrasted to a stationary Compton camera. The combination of a portable Compton camera and a fast-moving vehicle such as a UAV can have a distinct advantage over a radiation detector by itself.

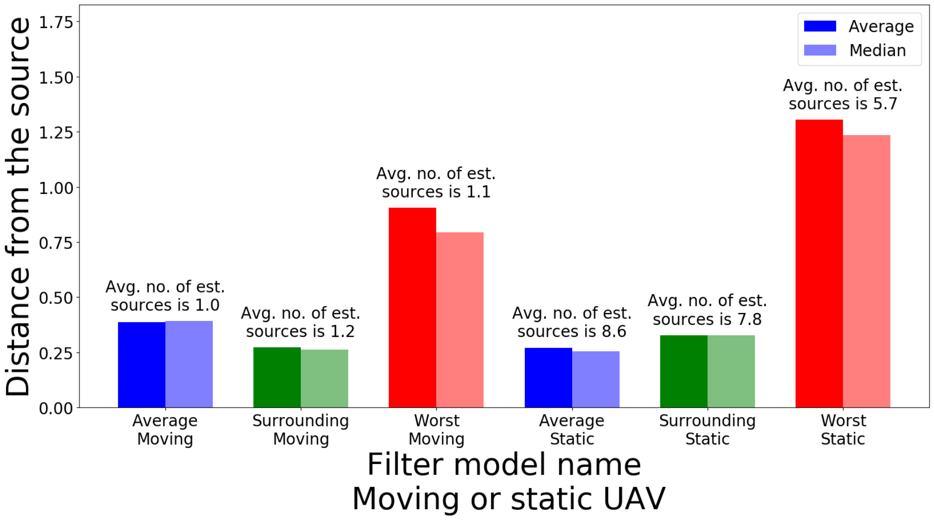


Figure 4.11: The graph compares a moving scenario with a static scenario where the model is trying to detect the position of a single source of radiation. Each bar in the graph represents almost five minutes of measurements with a particular model. The bars consist of two parts. The first represents the mean distance of the closest estimated position to the ground truth across the entire scenario. The second is the median of the closest distances over the entire scenario. Above each bar is the mean number of sources estimated by the radiation model during the measurement.

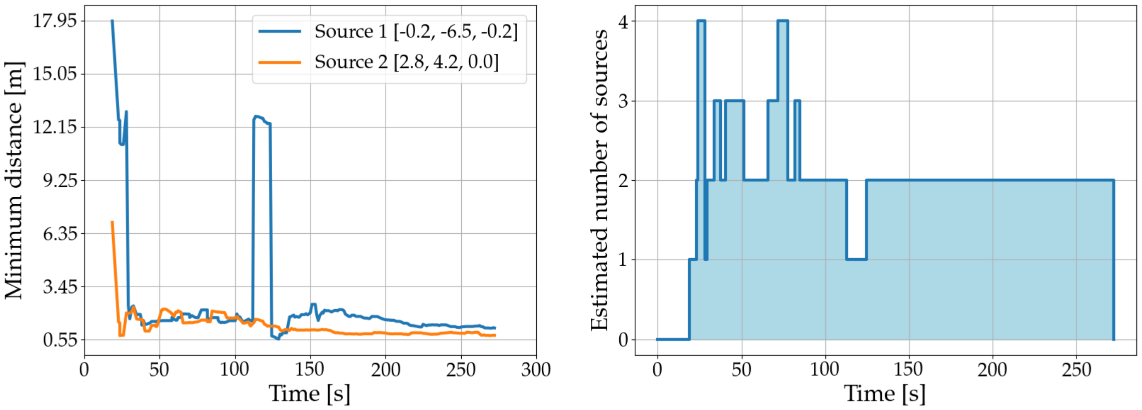
### 4.5.2 Two sources of radiation

The scenario is arranged to detect the positions of two radiation sources instead of one. The sources are deployed in an unobstructed area with a distance between the sources of more than eleven meters. The radiation measurement is similar to the previous case. The UAV has a Compton camera and a LIDAR on its frame, which return measurements of the environment and these are processed by Uniform Sampling into particles suitable for filter input. The simulation was run for all three models Average, Surrounding and Worst model. Static measurements did not provide sufficiently accurate information about the source location when a single radiation source was already tested. Detection with a moving UAV provides much more accurate information which can be better used, therefore the model testing is done only when the UAV is moving.

#### Moving UAV

The scenario where two sources of radiation are detected by a moving UAV is described in this section. The radiation sources are located in the same environment as in the case of one source, with no obstacles placed in the environment. The UAV moves along a circular trajectory at a height of three meters above the ground. The center of the trajectory is at the origin of the coordinate system and the radius of the trajectory is ten meters, which provides sufficient area for the detection of radiation sources. All three filter models, as in the previous situations, are tested and compared as follows.

Testing of the average model was accurate to an error of approximately 1 meter. From

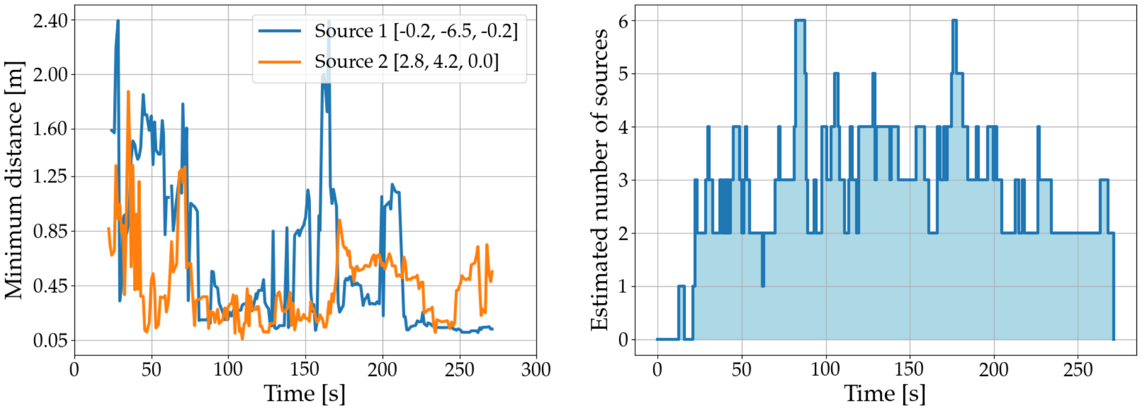


(a) Graph of the distance of the nearest estimated radiation source to the actual source over time.

(b) Graph showing the number of radiation sources estimated by the filter over time.

Figure 4.12: The results of two radiation sources detection using the Average model (Sec. 4.3.1) and a moving UAV.

the graph Fig. 4.12b it can be seen that after the initial state, the correct estimated number of sources was obtained. The model in motion improves source localization and eliminates unlikely hypotheses. Also, Fig. 4.12b presents the information that the estimated sources are located no further than 2 meters from the real sources during the initial phase. Over time, the model achieves even better results and the distance difference between the estimated sources and the real sources gradually decrease below 1 meter. The Average model would clearly indicate where the radiation source might actually be located.

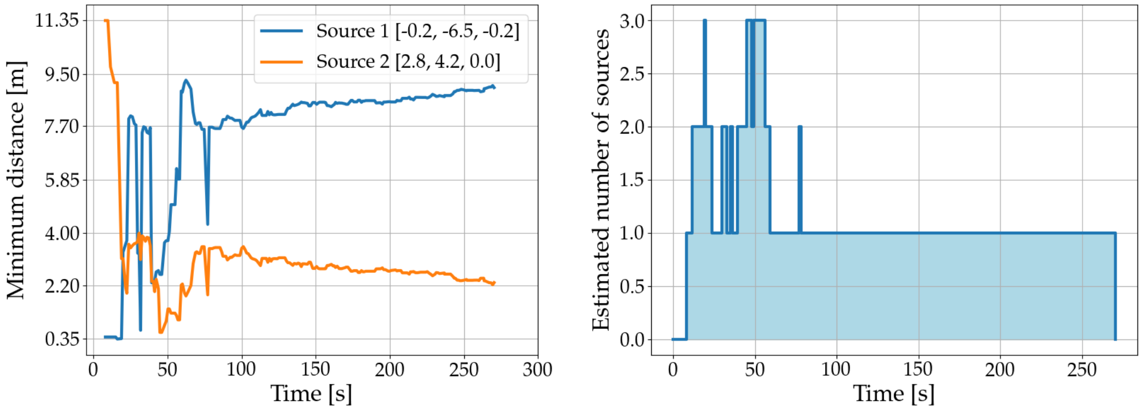


(a) Graph of the distance of the nearest estimated radiation source to the actual source over time.

(b) Graph showing the number of radiation sources estimated by the filter over time.

Figure 4.13: The results of two radiation sources detection using the Surrounding model (Sec. 4.3.3) and a moving UAV.

The Surrounding model, unlike the Average model, generates a larger number of estimated radiation sources according to the Fig. 4.13 in this scenario. On average, this value is approximately three sources throughout the measurement. This is better than underestimating the number of radiation sources and increasing the risk of radiation exposure. Detection is significantly more accurate, obtaining source location detections to less than a tenth of a meter at the expense of more false-positive detections. At the end of the measurement, the number of sources even becomes stable and equal to two, with still high accuracy. In contrast to the previous models, the Surrounding model has been the most successful.



(a) Graph of the distance of the nearest estimated radiation source to the actual source over time.

(b) Graph showing the number of radiation sources estimated by the filter over time.

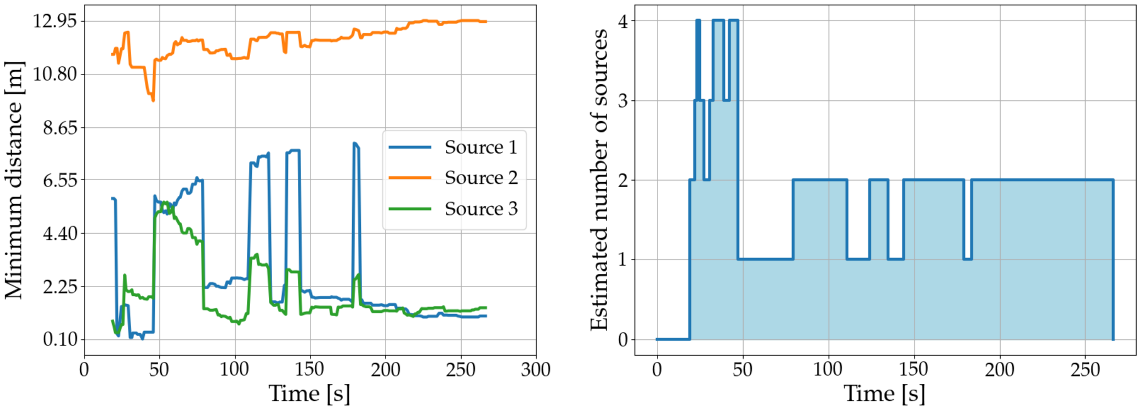
Figure 4.14: The results of two radiation sources detection using the Worst model (Sec. 4.3.2) and a moving UAV.

The Worst model, in the case of more than one radiation source detection stopped working and detecting correctly. It can be seen in Fig. 4.14b. After the initial phase, the model stops estimating multiple sources and begins to converge to a single source. Over time, it can be seen in Fig. 4.14a that how the model chooses one of the real sources and starts to converge its estimation to that source. However, the process requires a lot of time and results in incomplete estimation. Due to a limited battery capacity of a UAV, this approach is not feasible for practical use. Unknowledge of multiple sources is also inadmissible for multi-source detection in the safety constraints and human protection. Therefore, in this case, the model becomes inoperable.

### 4.5.3 Three sources of radiation

The scenario focuses on the detection of three sources of radiation in the vicinity of the UAV. As in previous scenarios, the UAV uses LIDAR and a Compton camera in addition to its necessary systems for functionality. The sensor data is processed into a suitable form by the Uniform sampling method and passed to the particle filter using the model under test. Unlike the other scenarios, this time it is tested in a way where the UAV searches for sources only when moving. The trajectory of motion that the UAV follows consists of a circular path 3 meters above the ground with a radius of 10 meters. The results are therefore depending on the correct detection of the number of sources and the accuracy of the estimation. The radiation sources are distributed in an empty space without obstacles, at irregular distances from each other to simulate a realistic possible radiation detection scenario.

The first filter tested for the three source scenario is the Average model. In Fig. 4.15 it can be seen that the Average model was able to successfully find only two of the three radiation sources that were in its vicinity. This is mainly shown in Fig. 4.15b in which the

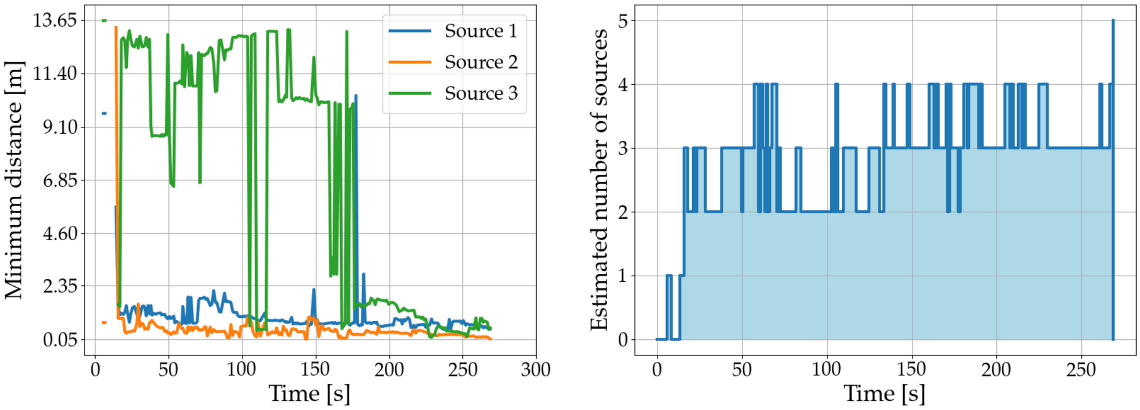


(a) Graph of the distance of the nearest estimated radiation source to the actual source over time.

(b) Graph showing the number of radiation sources estimated by the filter over time.

Figure 4.15: The results of three radiation sources detection using the Average model (Sec. 4.3.1) and a moving UAV.

number of estimated sources drops to one after the initial phase, but after a while it reached the value of two. This kind of change can clearly affect graph Fig. 4.15a, where the minimum distance for *Source 1* has increased and for all others, the distance is calculated only with the available estimated sources. The filter is able to identify a multi-source scenario, but the incorrect determination of the number of sources is unusable in practical terms as it increases the risk of radiation exposure.

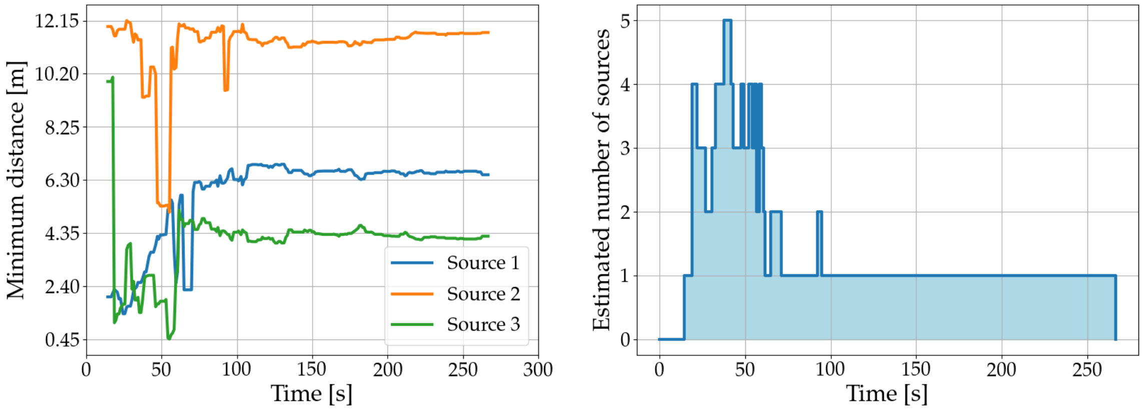


(a) Graph of the distance of the nearest estimated radiation source to the actual source over time.

(b) Graph showing the number of radiation sources estimated by the filter over time.

Figure 4.16: The results of three radiation sources detection using the Surrounding model (Sec. 4.3.3) and a moving UAV.

In the first 125 seconds of the measurement, the model predicts only 2 sources of radiation. It can be seen in Fig. 4.16b that the third source is added several times, but it is estimated at an incorrect position, as can be seen in Fig. 4.16a. A clear change occurs after 175 seconds, where a third hypothesis is initialized close to the actual source. This is also reflected by an increased confidence in the number of estimated sources. The model accuracy results are satisfactory and all three sources are accurately identified with a position error of less than 1 meter.



(a) Graph of the distance of the nearest estimated radiation source to the actual source over time.

(b) Graph showing the number of radiation sources estimated by the filter over time.

Figure 4.17: The results of three radiation sources detection using the Worst model (Sec. 4.3.2) and a moving UAV.

Using the Worst model in this scenario shows the poorest results. Overall, the model stopped working and estimated the radiation source in the wrong place. The minimum distances in Fig. 4.17a do not approach any of the radiation sources. The number of estimated sources in Fig. 4.17b is only one. All of this can mean with a high probability that the model has started to converge to a random location where no source is located. Not only does the filter return an insufficient number of estimated sources, but also the estimated filter does not present any positive effect for the real implementation.

### 4.5.4 Conclusion

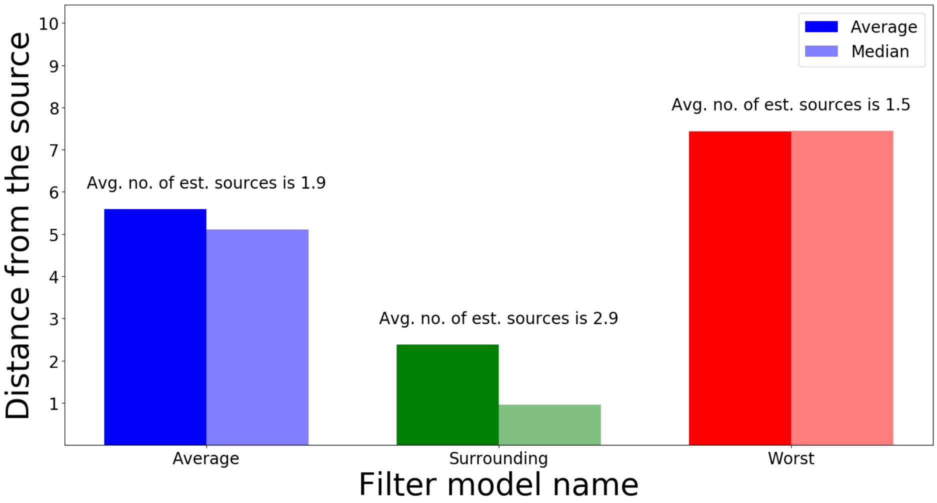


Figure 4.18: The graph compares the use of filter models in a scenario with a moving UAV in which the model estimates the positions of three radiation sources. Each bar in the graph represents five minutes of measurements with a particular model. The bars consist of two parts. The first represents the average distance of the closest estimated position to the ground truth across the entire scenario. The second represents the median of the closest distances obtained over the whole scenario. Above each column is the average number of sources estimated by the radiation model during the measurement.

Overall, it is possible to see how the Average, Surrounding and Worst models perform in different situations and detect different numbers of radiation sources. It tested how the models behave when the UAV is static and moving in the environment. This was achieved by a simple circular trajectory with no human interference in the control of the UAV. It was observed that in the static scenario all models estimate more radiation sources than the actual count. In most cases some of these estimated sources match the actual ones. That is, even if it is not possible to know exactly where the source of the radiation is located, it's at least possible to estimate the direction from which the radiation is coming. This could serve as a direction in which the UAV might move and collect more accurate data in the environment. With a moving UAV, it has been observed that locating a single source for any model does not pose a problem, and the location of a given source can be detected in a noticeably short amount of time, approximately one minute. Detecting the positions of two radiation sources was already exceedingly difficult for the Worst filter and it could not estimate more than one of them. The Average model was able to accurately determine the number of radiation sources after 50 seconds with an accuracy of about 1 meter. For the Surrounding model, there was a problem with estimating a larger number of sources. The model needed more time to get the correct estimate, but its accuracy was still less than 1 meter. The scenario of detecting the positions of three radiation sources was already starting to fail even for the Average model, which estimated only two of them. The Surrounding model needed more time to detect than the one source scenario, but after a few seconds, it successfully estimated both the number and location of the radiation sources. A more detailed view of the scenario is shown in Fig. 4.18, where the statistics for the estimation accuracy of the radiation location and the rounded average number of estimated sources are presented. For the Average model, the number of estimated sources reaches 1.9, for the Worst model 1.5 and Surrounding 2.9 which is the closest to the real number 3. The Surrounding model also achieves the best accuracy with a median minimum distance for all sources of approximately 1 meter. According to Fig. 4.18 and the graphs of all previous scenarios, it is possible to say that the Surrounding model is the most successful of all three models for the scenarios considered in this chapter.

# Chapter 5

# Visualization

In recent years, even recreational UAVs have started to be used as a tool for image recording. They are equipped with a camera that records the picture for the user. The chapter on visualization focuses on the aforementioned imaging characteristics of the thesis. The visualization is performed as a transmission of data from the onboard color camera to the user. In this way, the user acquires a visual cue in the form of an image of the environment that is in front of the UAV. When the user and the UAV interact, visualization can be used for a variety of purposes. In addition, to familiarize the user with the environment, the camera footage can also identify possible radioactive objects by marking them. The identified objects serve as a warning signal, which should not be approached without the necessary protective measures, such as a protective suit. The identification of dangerous objects is based on the results discussed in the radiation source estimation chapter. The results of the data from the previous sections are processed and then transformed as input for the identification of objects in the image. A large part of the object detection is performed using the OpenCV[1] library, suitable for image processing.

## 5.1 OpenCV

OpenCV is an open-source software that deals with image processing and editing. It is used for computer vision, where it provides several algorithms for working with images or videos. Real-time performance is the library's key benefit for this thesis. The library supports image processing in many formats such as JPG, PNG, BMP or for videos AVI format. It is possible to use the library for conversion between formats or to change the representation of color encoding for example from color to a monochronic color scheme. The library provides a number of essential functions and convolutions in image processing. The functions are capable of image blurring and resizing, and the detection of edges, circles, lines or contours in the image.

Furthermore, the OpenCV is able to analyze the color scheme of an image. It also offers several functions for drawing into the image, such as adding primitive geometric objects like a circle, rectangle or even more complex objects, for example, text. OpenCV also supports the creation of GUI, which can be used for real-time parameter setting. Last but not least, it also offers the possibility to create a custom filter, applicable in image convolution. OpenCV is a library created in the C++ programming language, however, it also provides extensive support for other languages such as Java or Python. The OpenCV is a tool, which greatly simplifies working with an image. Also, OpenCV has an active developers' community that contributes to the continuous improvement of the library. The OpenCV library is the clear choice for this thesis because of the features it possesses.

---

[1] `https://opencv.org/`

## 5.2 Object detection

The main section of this chapter is about object detection in a two-dimensional camera image. Object detection is performed by processing the video obtained from the camera. The camera periodically sends messages to a ROS topic. The topic contains messages from the individual video frames that the camera records. These frames are then converted into an OpenCV image and processed individually by an object recognition algorithm.

The object recognition algorithm operates as follows. The code first turns the input color image into grayscale using the *cvtColor* function from OpenCV. Then a median filter with a $3\times3$ matrix size is applied to the grayscale image to reduce noise using the *medianBlur* function. Another option is the use of image dilation to fill in the gaps and make the object boundaries smoother. Although, it is not used by default, the OpenCV GUI allows users to adjust this during runtime. Subsequently, a thresholding algorithm is applied using the OpenCV *thresholding* function with the *THRESH BINARY* and *THRESH OTSU* flags to obtain the optimal image thresholding value. This value is then used as a boundary parameter for the Canny edge detection algorithm using the *Canny* function in OpenCV to detect edges in the image. It identifies the contours of objects in the image from the edge-detected image using the OpenCV $findContours$ function with the *RETR EXTERNAL* and *CHAIN APPROX SIMPLE* flags to extract the shape of the object. *RETR EXTERNAL* discards all contours inside other contours and *CHAIN APPROX SIMPLE* compresses the contour data, keeping only their endpoints, sufficient to represent the contour shape. The contours represent the detected objects and the next step is to wrap them into a geometric shape that represents their location in the image. This is accomplished by using OpenCV's *boundingRect* function, which computes the bounding box of each contour and returns a rectangle representing the object.

## 5.3 Identification of the radiation objects

The output to the user is not just a visual image obtained by the camera. The most important contribution is the identification of possible radioactive sources in the obtained image. To accomplish this task, the object detectors from the previous section must be used. The detector returns a set of objects represented by bounding boxes which it assumes can represent the objects. The next step is to determine which of these objects can be qualified as radioactive. To accomplish this task, it is necessary to obtain coordinates where the radiation might be located in the real world. The coordinates of possible radiation sources are calculated as a list of points from the assumed position of the radiation in Chapter 4. These points are represented in $\mathbb{R}^3$ space in the UAV coordinate system. The points then need to be transformed into the camera coordinate system. The transformation is achieved by using the MRS library[2] based on the TF2 API[3] of the ROS system, which solves various coordinate-type transformations. Before the transformation, it is necessary to check if such a transformation exists and is feasible. The coordinate is then converted into the camera coordinate system. The next step is to assign the transformed points in the image coordinate system to the objects that are detected. The assignment is solved as a comparison of the distance of the point from the object box. If the measured distance is short enough according to the specific value of

---

[2] `https://github.com/ctu-mrs/tf2`

[3] `https://wiki.ros.org/tf2`

around 30 px, the detected object is considered radioactive and is marked in the image. An example of identification can be seen in Fig. 5.1.

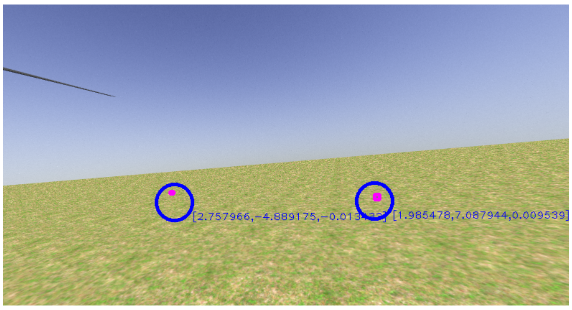


Figure 5.1: Example of radiation source detection in the image from the Gazebo simulator. The bounding boxes of identified radiation objects are bordered in pink. In blue are the circular regions of points where objects are considered radioactive according to the assumptions. World coordinates of the identified objects are also drawn to the image.

## 5.4 3D visualization

Another visual component of this work is the 3D visualization. The 3D visualization is primarily used with simulation in the Gazebo simulator. However, its use can also be applied to the playback of rosbags. The 3D visualization in this work utilizes the ROS Rviz tool. The Rviz tool has full ROS support. The main advantage of Rviz is the ability to take a topic published by a sensor or a topic created by the processing pipeline and displaying the topic data in real time in $\mathbb{R}^3$ space. Typical examples for display are basic objects such as a point in space, but more complex objects such as a UAV model located at certain coordinates are also possible. Rviz provides a graphical user interface that allows to add or change the visualization of an existing topic. It also has features that can be used for visual navigation of the UAV to a certain point or interaction with objects in the simulation. The work uses Rviz tool to display the drone's position and show the processed input from the sensors, notably the Compton cones and the OctoMap. Displaying the measurements helps the observer gain a better understanding of the problem or help in solving the problem. Rviz is currently used to display a set of points generated by the sampling function on a Compton cone fused with depth measurements. Also, Rviz displays the particles processed by the particle filter and the resulting radiation source estimates as color-coded points in space. The C++ library Batch visualizer from ROS package[4], which was created by the MRS group, solves the problem of creating topics for publishing geometric objects such as sets of points, triangles, squares or cubes. Overall, the visualization of the work adds the possibility of additional visual output, which serves to improve the understanding.

---

[4] `https://github.com/ctu-mrs/mrs_lib/tree/dcc494ea1c80fafa51bbc24e80c8f12f6f8b0eed`

# Chapter 6

# Evaluation

This chapter demonstrates the use of all the developed methods and the real deployment of this work. The whole radiation source detection process and the results achieved are discussed. The first part consists of radiation detection in a simulation environment. The simulation environment is configured to resemble a real situation with a complex distribution of objects. The second part deals with radiation detection in a real environment. The thesis is tested on a real device with a simulated radiation source placed in the environment.

## 6.1 Simulation

In this section of the thesis, the evaluation in a simulated environment is presented. The simulation models the detection of two radiation sources in a complex environment resembling a city street. It consists of predisposition, process and results parts.

### 6.1.1 Predisposition

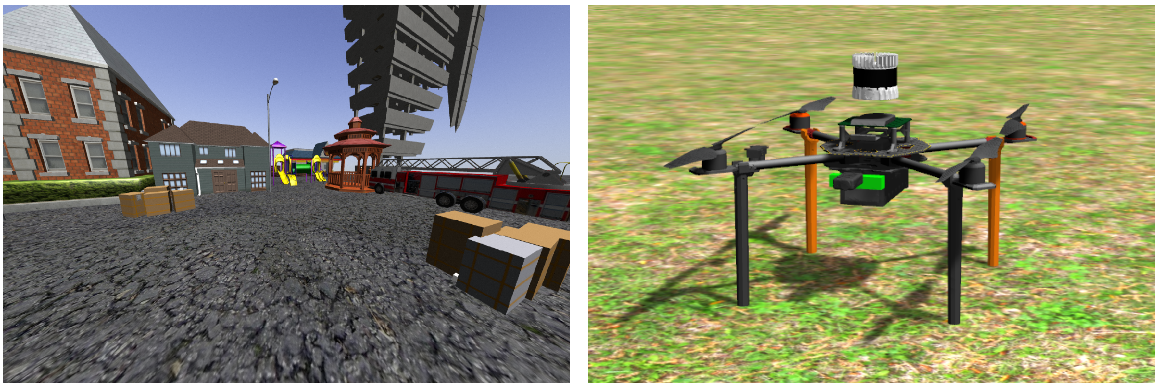
The simulation is performed in the Gazebo simulator, which has sufficient capabilities to meet the requirements necessary for the work. The simulation runs in a newly created Gazebo world where various types of obstacles have been added. Obstacles are represented by buildings, cars, lamps or containers placed in the environment. The created environment can be seen in the Fig. 6.1a. The UAVis the Tarot T650 Fig. 6.1b. The model has four propellers and is approx. 65 cm in diameter. The UAV has simulated sensors placed onboard with the goal of minimizing the difference between the simulation and the real world. One of the sensors is modelled after the Ouster OS0-64[1]. Other sensors are a single-layer Compton camera and a color camera for video footage acquisition. Radiation sources are represented in the Gazebo simulator as white cube-shaped objects. In this case, a strong radioactive source simulating *Cesium-137* was used. The sources are located at different places in the simulation environment. One of the radiation sources is placed on the ground near obstacles and the other is placed on an obstacle 1.3 meters above the ground. A uniform sampling algorithm with a range parameter of 1 meter up to a sampling distance of 15 meters was used to detect the locations of the radiation sources. For processing the particles, the Surrounding model was used for the particle filter to achieve the best flexibility in detection.

### 6.1.2 Process

The process of the simulation is as follows. The Gazebo simulator simulates all parts of the environment. The UAV is placed in the environment and executes its systems together with

---

[1] `https://ouster.com/products/scanning-lidar/os0-sensor/`



(a) Image of the environment in which the simulation was performed, modelling a street in a city.

(b) Image of the UAV model Tarot T650 in Gazebo simulator.

Figure 6.1: The environment and UAV equipment precondition under which the simulation was performed.

the radiation source localization module. In the background of the emulation, the simulation for Compton's camera is being calculated. Data from the Compton camera is generated by a Monte Carlo simulation[2] that evaluates all physical processes inside the detector at the single photon level. The depth camera simultaneously maps the space from the Gazebo simulator and contributes data to the OctoMap. The occupied part of the OctoMap can be seen in Fig. 6.2a. A composite image from the RGB camera with the OctoMap as an overlay is displayed in Fig. 6.2b. The radiation localization system created in this thesis detects and processes the measurements. The UAV moves while capturing the measurements to improve source estimation. This has been proven in the comparison model part (Sec. 4.5).

In the environment where the radiation is located, the UAV flies in autonomous mode in a circular trajectory over the obstacles. After detecting the first possible sources of radiation, the resulting data are displayed on a color camera and suspicious objects are detected and marked as potential radiation sources. The UAV movement continues and new measurements are collected and the accuracy of source localization is improving.

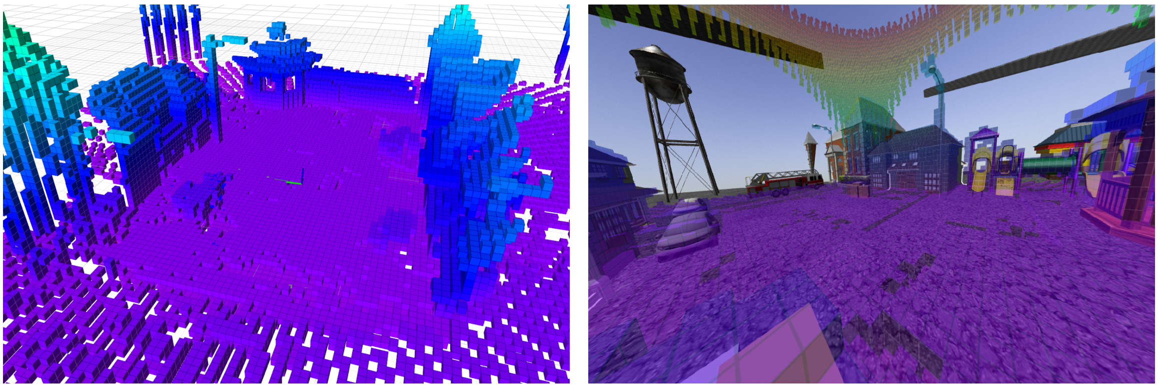
### 6.1.3 Results

After obtaining several Compton cones at different UAV locations, the results started to approximate the real radiation sources. The process of estimation of radiation sources during the whole measurement can be seen in Fig. 6.3a. The first graph shows the accuracy achieved during the whole process, which is below one meter for both radiation sources. Then in Fig. 6.3b, one can see the number of estimated sources over the duration of the whole measurement. Also, useful information about the accuracy of the measurements is provided by the density map (Fig. 6.4) which describes the density of detected positions of the radiation sources estimated during the whole measurement. From the graph, the difference between the area where the radiation was located and where it was not, is clearly recognizable.

The particle filter works as expected in the simulation. The data from the Surrounding Model at a certain point in time are shown in Fig. 6.5. It is already possible to know where the source of the radiation might be by the density of the particles shown in the picture. The last

---

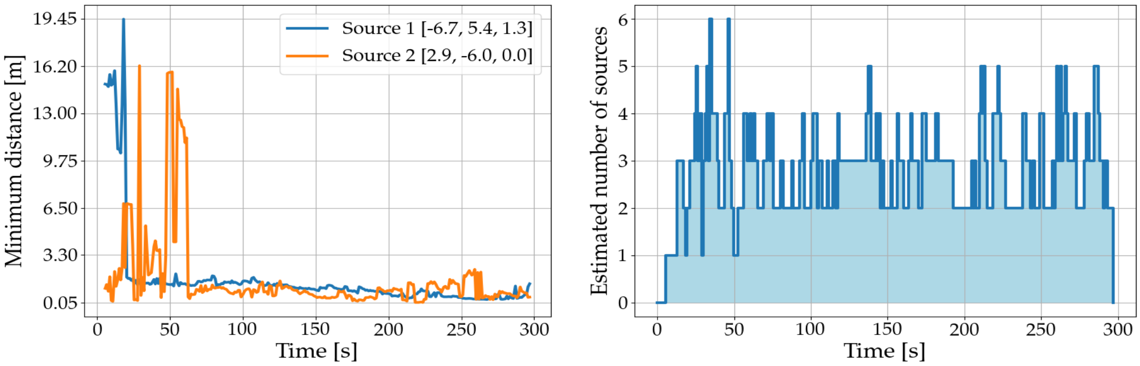
[2] `https://github.com/rospix/gazebo_compton_camera_python`



(a) Part of the OctoMap captured in the simulation and displayed in Rviz.

(b) The octomap depicted together with the image from the camera.

Figure 6.2: An illustration of the OctoMap created in the simulation



(a) Graph of the distance of the nearest estimated radiation source to the actual source over time.

(b) Graph showing the number of radiation sources estimated by the filter over time.

Figure 6.3: The result of the detection of two radiation sources in a complex environment. They show the estimation accuracy of the filtering model named the Surrounding model and its detection capability.

step is to use the estimation algorithm from Sec. 4.4 to achieve clear positions of the radiation sources. In the Rviz visualizer are estimated sources marked as green points. The last step is to work with the color camera. After the estimation of the sources, the transformation of their positions into the camera image is performed, where they are assigned to the objects and then the objects are highlighted. The object recognition environment is no longer as explicit as in the previous case of a flat plane, but still provides sufficient information about the location. In spite of the even more complex situation, the work with the image is shown in Fig. 6.6a and Fig. 6.6b. From Fig. 6.6, several objects can be seen which are considered to be the radiation sources. Not all of the objects show the radiation sources as defined, but each object is in the vicinity of the real source, which can be a good basis for further work.

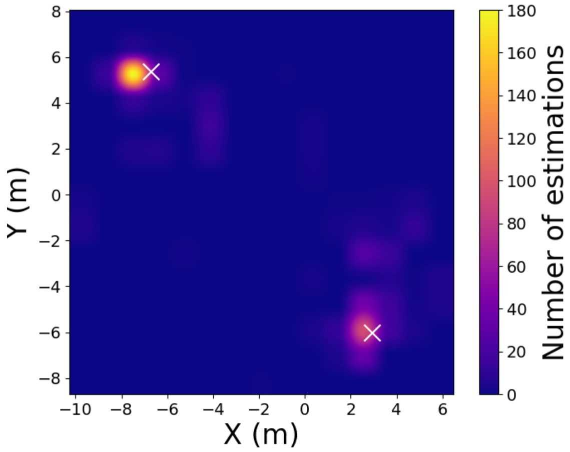
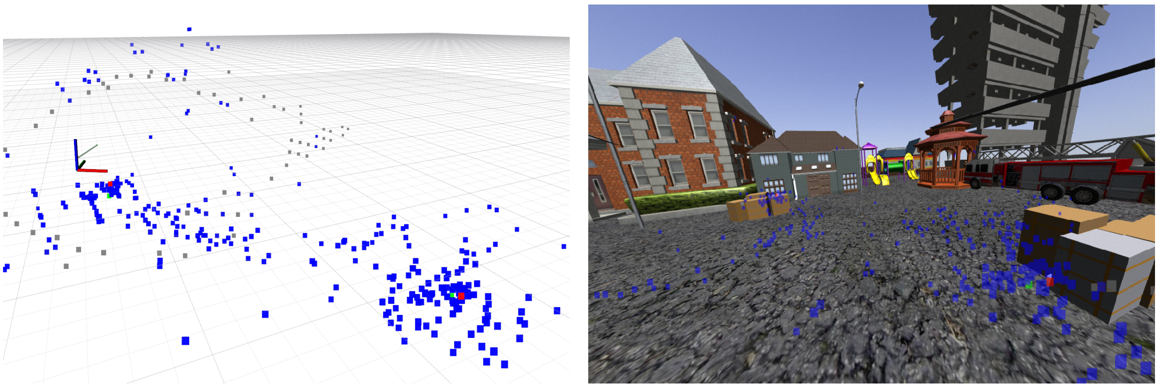


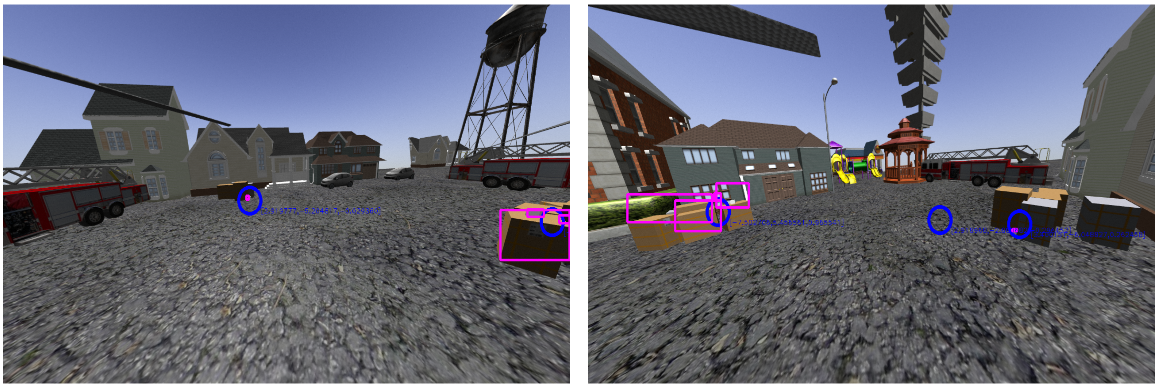
Figure 6.4: Density map of the estimated radiation sources for two actual radiation sources obtained from a Gazebo simulation. Density maps describe the view from above in two-dimensional space. The white marker represents the real position of the radiation source. The color in the map represents the number of points estimated during the measurement. A lighter color of the map means a higher chance of finding a radiation source.



(a) 3D visualization of particles from the particle filter in the Rviz tool.

(b) Visualization of particles from the particle filter displayed in the camera image.

Figure 6.5: Particles from the particle filter used in the detection of radiation sources in the point-color simulation. The blue points represent particles, the grey points are new incoming particles, the green points are the positions of the detected radiation sources and the red points are the real radiation sources.



(a) Image recorded by a camera onboard theUAV during simulation. The first source of radiation is located on the ground and the second source is clearly highlighted on the right side of the image.

(b) Image recorded by the camera onboard the UAV during simulation. The image at this point in time shows three estimated sources of radiation. There are fewer inaccuracies in the identification of objects for a source located above the ground.

Figure 6.6: The images ( a ) and ( b ) show the processed recording at some points in time from a color camera placed on the UAV. The pink frame indicates potentially dangerous objects that could be the radiation sources. The blue circle indicates the estimated positions of the radiation sources in the camera coordinate frame.

## 6.2 Real experiment

The section is concerned with the application and results of work data obtained in a real deployment. The developed methods were tested with a real UAV and its components, sensors. The section consists of the prerequisites, process and results sections which describe the experiment.

### 6.2.1 Prerequisites

The experiment is aimed to verify the usability of the developed methods under real conditions. To achieve this goal, the MRS quadcopter hardware platform [2] built on the Holybro X500[3] frame is used. In addition to the onboard systems the Holybro X500 has, it is also equipped with the Intel NUC[4] with the Linux operating system for the deployment of custom program components. The onboard computer runs the UAV MRS system [5], which provides the UAV state estimation for the radiation localization framework. It is a mini-computer with high computational power and sufficient memory for storing data collected during the experiments. The whole UAV that was used in the experiment is shown in Fig. 6.7a. The UAV is localized, in addition to the nowadays regularly used Global Positioning System (GPS), also with the Real-time Kinematics (RTK)[5] a system that serves as ground-truth for the real experiment. The visual data is collected by a color camera placed on the bottom of the drone in connection with the depth sensor of the Realsense camera.

---

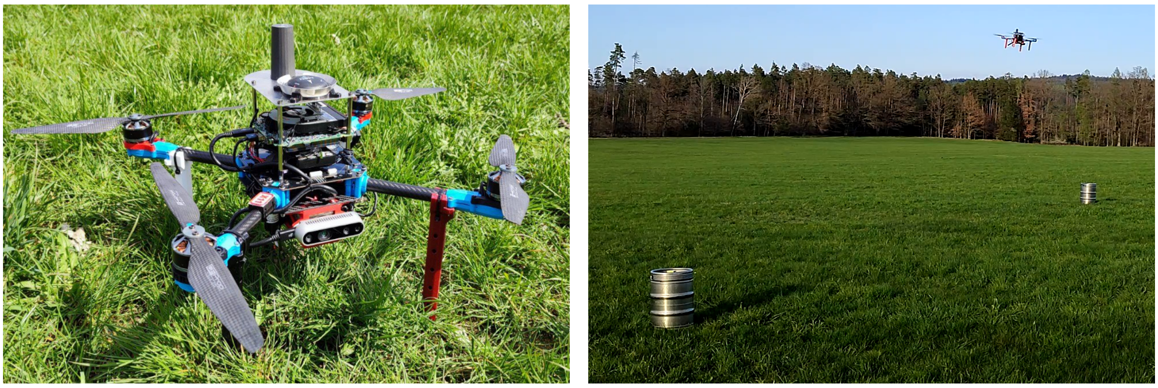
[3] `https://holybro.com/products/px4-development-kit-x500-v2`

[4] `https://www.intel.com/content/www/us/en/products/details/nuc/boards/products.html`

[5] Technology based on the GPS improved by correction data streams from the static station

Due to the hazardous nature of strong radiation sources, and the strict regulations on handling them, it was not possible to use real radioactive material during the experiments. Instead, the previously mentioned radiation simulator is used. The simulator is running directly on the onboard computer of the UAV. The simulated radiation sources are in different locations in the space of the real environment in which the drone moves. The radiation sources are simulating *Cesium-137*, and their activity was set to 1 GBq. The activity of 1 Bq represents one photon emission per second in a random direction. With these settings, the Compton camera yields on average 1 cone per second from a distance of 20 meters.

The experiment was executed in an open flat environment with no obstacles. The environment contains grey barrels that represent the simulated radiation sources. The experiment is carried out in two variants with one and two radiation sources, the results of which are presented in the next section. An example of the two-source environment in which the experiment was carried out is shown in Fig. 6.7b. The radiation detection pipeline uses the same settings as in the Gazebo simulations. The space is sampled using the Uniform Sampling method and the Surrounding model is used as the model for the particle filter.



(a) The UAV model used in a real experiment with all sensors and components based on the Holybro X500.

(b) Example of the environment in which the real experiment was performed with two radiation sources and one flying UAV.

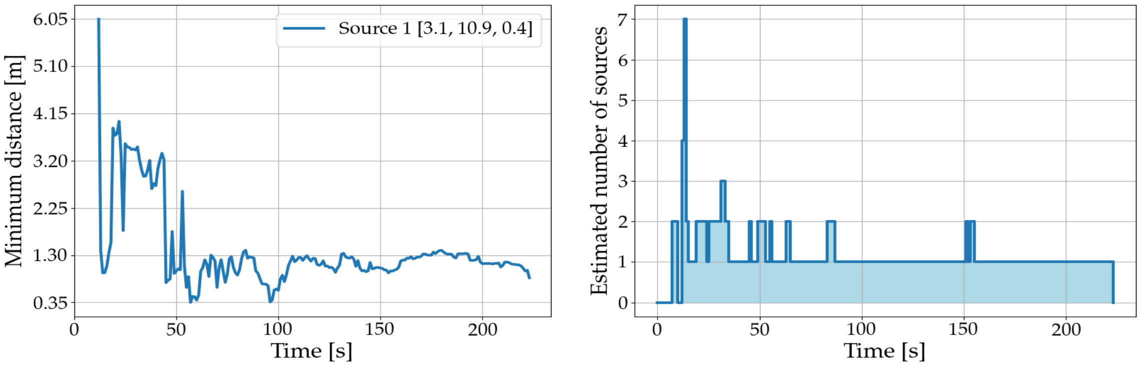
Figure 6.7: The quadrotor platform and the environment in which it was deployed for the experimental evaluation.

### 6.2.2 Process

The process of experimentation in a real implementation is performed in a similar way as in simulation. The UAV is used to detect the radiation positions of the parts created in this thesis. The radiation detection is uploaded to the onboard mini-computer of the UAV and activated in flight by the ROS service. Unfortunately, the UAV used in the experiment was built using an untested platform and its parts did not work correctly during data processing for detection.

A significant delay in the depth camera image was observed and could not be processed in time with the odometry measurements of the device. The onboard computer also experienced power delivery issues and turned off during multiple flights causing a loss of data. Inaccurate data in the position readout of the UAV in space was occurring. To determine the exact positions of the radiation sources and resolve these issues, the depth camera was disabled and the OctoMap of the environment was pre-computed and loaded during the experiments. The voltage fluctuation and subsequent shutdowns of the onboard computer. Instead, the vehicle was controlled manually using the radio transmitter. During the manual flight, the onboard systems were still running and providing data in the ROS environment. The radiation detection system was designed to operate normally under these conditions as well, as it only requires information on the UAV's state and the Compton camera measurements.

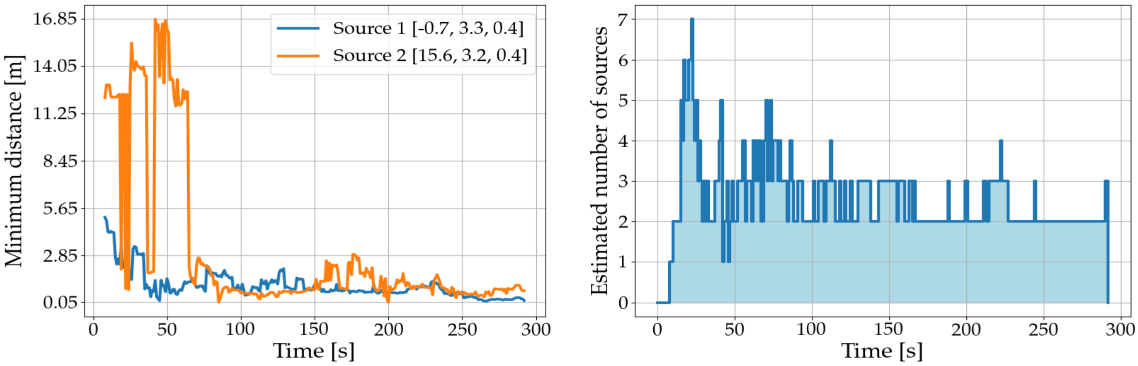
The UAV moved in the radiation source area during the experiment and acquired radiation data. The data was then processed by sampling the Compton cones using the Uniform Method. Next, the data was fused with the OctoMap to get the points located only in the material space and then estimate the radiation sources. Along with the estimation, color camera recording and object detection were performed.



(a) Graph of the distance of the nearest estimated radiation source to the actual source over time.

(b) Graph showing the number of radiation sources estimated by the filter over time.

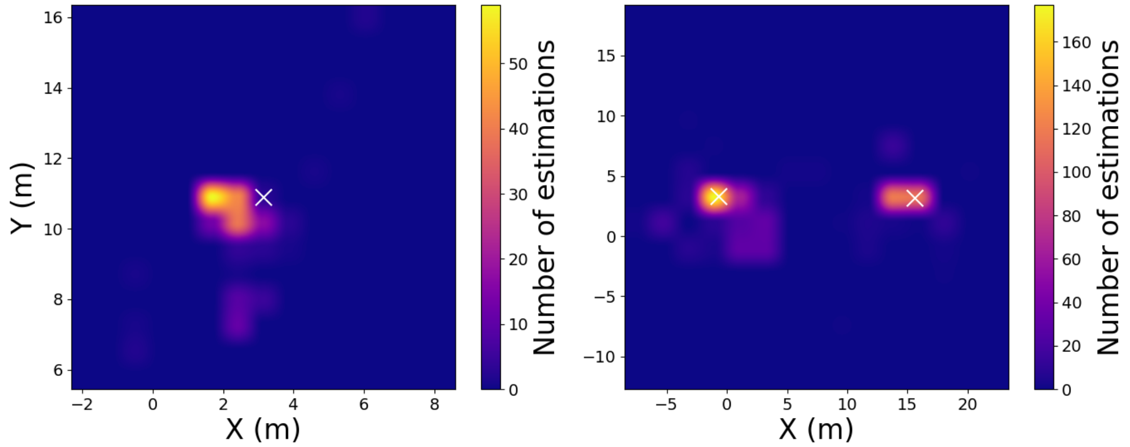
Figure 6.8: The result of the detection of one radiation source in a real environment. They show the estimation accuracy of the filtering model named the Surrounding model and its detection capability.



(a) Graph of the distance of the nearest estimated radiation source to the actual source over time.

(b) Graph showing the number of radiation sources estimated by the filter over time.

Figure 6.9: The result of the detection of two radiation sources in a real environment. They show the estimation accuracy of the filtering model named the Surrounding model and its detection capability.



(a) Density map from the experiment with a single radiation source.

(b) Density map from the experiment with two radiation sources.

Figure 6.10: Density maps of estimated radiation sources for one and two radiation sources obtained in real-world experiments. Density maps describe the view from above in two-dimensional space. The white marker represents ground-truth of the radiation source. The color in the map represents the number of points estimated during the measurement. A lighter color of the map means a higher chance of finding a radiation source.

### 6.2.3 Results

The results of the real experiment depended largely on the accuracy of the UAV position acquisition using GPS or RTK technology, which affects the simulation of the radiation sources. Despite this dependence, the real experiment has demonstrated a sufficient amount of information and the accuracy to detect the location of the radiation source. The results of the source localization are shown in detail throughout the experiment in Fig. 6.8 and Fig. 6.9. Fig. 6.8 shows the results of a single radiation source experiment and graph Fig. 6.9 of two sources. The achieved precision is around 1 meter for one radiation source (Fig. 6.8a) which may be influenced by the localization of the UAV. During this experiment, the system correctly estimated that there is one source of radiation. Fig. 6.9 shows the experiment with two radiation sources. In this experiment, the localization error is also below 1 meter for both sources. The number of sources is fluctuating during the measurement process but stabilizes over time to correctly report two sources. The whole process of all estimated sources for one or two radiation sources is recorded in plots describing the density estimation Fig. 6.10. Where we can see from the picture that the particle filter estimated the radiation sources near the real position most intensively than outside it. Throughout the radiation localization, UAV uses a color camera, which is then processed to visualize the results over Fig. 6.11. This image shows the barrel that was used to represent the simulated radiation source. The barrel is marked in pink to represent the estimated radiation source.

Figure 6.11: Processed image of the color camera when detecting radiation source positions with marked possible threatening object obtained by measurement in real environment test.

## 6.3 Summarization

The evaluation was performed in two variants. The first was a simulation in a complex environment with obstacles and the second was a deployment on a real UAV in an environment without obstacles. In both cases, the radiation localization was running online throughout the entire flight. The results from the evaluation were presented in the previous section. After an initial phase that takes approximately 1 minute, the localization error falls below 1 meter, which can provide sufficient information about the location of the resources for the user. Visually, using the density Fig. 6.10 the estimation placement shows strong agreement with the position of the actual source. The image processing was able to detect the existence of the object and to highlight the correct ones according to the estimated values. The complexity of the environment affects the ability to determine the number of radiation sources, but it does not rapidly affect the accuracy of the measurement.

In addition to the detection part developed in this work, the other components that the UAV requires for its operation have an important role in real deployment. Notably, the Compton cones are projected from the position of the UAV, which is provided by the GPS or RTK positioning systems. All errors in the UAV's self-localization are inherently propagated into the radiation localization process. The detection process returns valid data after a few seconds, which can already be assumed to be a suspicious location, Overall, it can be said that the whole process of detecting the positions of radiation sources was successful with accuracy and speed of convergence acceptable for practical use.

# Chapter 7

# Conclusion

The aim of this work was to design and implement an algorithm for fusion of data from a depth sensor and a radiation detector. The radiation detector was represented by a Compton camera and the depth sensors by a LIDAR or a depth camera. Both types of sensors are located onboard a small compact UAV. The algorithm for fusion is divided into two parts.

In the first part, the output data from the Compton camera, which are in the form of Compton cones, are processed. A parametric equation of a circle in $\mathbb{R}^3$ is created, with the possibility of shifting and resizing through the parameters, to describe the whole cone. The equation is then used by two algorithms to generate a set of points in $\mathbb{R}^3$. Each of the algorithms has its positive as well as negative aspects. A comparison of the developed algorithms is shown in Sec. 3.4. The UAV is moving in an environment with an unknown amount of radioactive emitters and the set of points represents the possible locations of the radiation sources.

The second part deals with the fusion of dosimetry and depth measurements, where the set of points is subsequently filtered and simplified. The depth measurements represent the environment where the UAV operates. The radiation source is assumed not to be in the air and to be a distinct object in the environment. This assumption allows us to filter the set of points using a depth measurement by comparing whether the generated point coincides with the environment in which it is located. Points that do not fulfill the condition are subsequently excluded and the total number of points is reduced to more likely potential radiation locations.

The fusion of depth data into the radiation measurements allows a significant reduction in the number of particles generated by the cone sampling. As a result, the process is much faster and can even be used in real time. Additionally, the removal of very unlikely particles results in a more accurate detection of radiation hotspots. The detection itself is possible by an iterative algorithm, which takes as an input a set of points from the previous fusion step. The algorithm is based on the principle of a particle filter, which evaluates the particles and estimates the most probable locations of the radiation source. Sec. 4.5 is dedicated to the comparison of several evaluation models used in the particle filter algorithm to assign probabilities. The output of the iterative algorithm is a set of particles in $\mathbb{R}^3$ space, which is processed and filtered to extract specific, potentially dangerous locations.

The last task is to identify the dangerous object in the image from the color stream of the onboard camera. This task was performed using the OpenCV image processing library. By using several functions from the library, a simple object detector for the camera image was designed. Points representing potentially hazardous radiation sources are converted into the camera coordinate system and assigned to the detected objects. The final result of this thesis is the marking of hazardous objects on the real-time video recorded by the UAV.

The software pipeline can successfully find more than one radiation source by fusion of dosimetry and depth measurements. The detected positions of the radiation sources start to approximate the actual positions of the sources after a short initialization period, and after one minute reach accuracy of approximately one meter. The result is reported to the user in the form of a video feed with the potential radiation hotspots clearly identified and highlighted.

## 7.1 Future work

Although the work has achieved successful results, there is still scope for improvement in locating the sources of radiation or in navigating the drone. The Compton camera primarily serves as a source of possible radiation directions using Compton cones. In addition to the possible direction of the incoming photon from the radiation source, the Timepix chip inside the Compton camera also allows radiation intensity measurements. The use of the radiation intensity gained from the measurement could greatly help in the localization of radiation sources and in gaining a better knowledge of their abundance during the detection process. It is worth noting, that only a very small fraction of the detected photons undergoes Compton scattering. For the scenario considered in this thesis, the scattering events represent only one percent of all detections. One of the applications is in the weighting part of the particle filter, which would also take into consideration the intensity of the measured radiation at a given location when calculating the weights. Also, if there is an extreme increase in intensity at other locations it would mean the possibility of multiple sources and that could lead to an automatic correction of the detection parameters.

Another possibility is to improve the detection of the objects in the image, for example, using reinforcement learning or using a neural network which could improve the classification of objects identified as radiation sources. Better processing of camera images could better identify radiation sources for complex objects with nested components.

The mobility and navigation of UAVs to autonomously search the environment by adapting to the estimated locations of radiation sources and intensities obtained from measurements is also a suitable area for improvement. Controlling the UAV autonomously over unexplored areas to acquire new data or navigating the UAV in an already traversed area to improve the accuracy of the measurements would provide a fully automatic device to scan the hazardous area. Autonomous control of UAVs could clearly make the work more efficient and improve detection capabilities.

Last but not least, the use of multiple collaborating UAVs is also a suitable step for improvement. The ability to cooperatively search a single area and share the measured radiation data would rapidly accelerate the localization of radiation sources. The opportunities for improvement are limitless and could be further guided by the actual deployment requirements and the available equipment.

# Chapter 8

# References

[1] Z. Chao, R. Komatsu, H. Woo, Y. Tamura, A. Yamashita, and H. Asama, "Estimation of radiation source distribution using structure information for fukushima daiichi nuclear power plant reactor," *2022 IEEE/SICE International Symposium on System Integration (SII)*, 2022. DOI: `10.1109/sii52469.2022.9708811`.

[2] D. Hert, T. Baca, P. Petracek, *et al.*, "MRS Modular UAV Hardware Platforms for Supporting Research in Real-World Outdoor and Indoor Environments," in *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, Jun. 2022, pp. 1264–1273. DOI: `10.1109/ICUAS54217.2022.9836083`.

[3] M. Vrba, Y. Stasinchuk, T. Báča, *et al.*, "Autonomous capture of agile flying objects using UAVs: The MBZIRC 2020 challenge," *Robotics and Autonomous Systems*, vol. 149, p. 103 970, Mar. 2022, ISSN: 0921–8890. DOI: `https://doi.org/10.1016/j.robot.2021.103970`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0921889021002396`.

[4] V. Walter, V. Spurny, M. Petrlik, *et al.*, "Extinguishing real fires by fully autonomous multirotor uavs in the mbzirc 2020 competition," *Field Robotics*, vol. 2, pp. 406–436, Apr. 2022, ISSN: 2771-3989. DOI: `10.55417/fr.2022015`.

[5] T. Baca, M. Petrlik, M. Vrba, *et al.*, "The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles," *J. Intell. Rob. Syst.*, vol. 102, no. 1, pp. 1–28, May 2021, ISSN: 1573-0409. DOI: `10.1007/s10846-021-01383-5`.

[6] T. Baca, P. Stibinger, D. Doubravova, *et al.*, "Gamma radiation source localization for micro aerial vehicles with a miniature single-detector compton event camera," *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2021. DOI: `10.1109/icuas51884.2021.9476766`.

[7] S. K. Bhoi, K. K. Jena, S. K. Panda, *et al.*, "An internet of things assisted unmanned aerial vehicle based artificial intelligence model for rice pest detection," *Microprocessors and Microsystems*, vol. 80, p. 103 607, 2021. DOI: `10.1016/j.micpro.2020.103607`.

[8] D. Smrcka, T. Baca, T. Nascimento, and M. Saska, "Admittance force-based uav-wall stabilization and press exertion for documentation and inspection of historical buildings," pp. 552–559, Jun. 2021. DOI: `10.1109/ICUAS51884.2021.9476873`.

[9] V. Spurny, V. Pritzl, V. Walter, *et al.*, "Autonomous firefighting inside buildings by an unmanned aerial vehicle," *IEEE Access*, vol. 9, pp. 15 872–15 890, Jan. 2021. DOI: `10.1109/ACCESS.2021.3052967`.

[10] D. A. Saikin, T. Baca, M. Gurtner, and M. Saska, "Wildfire fighting by unmanned aerial system exploiting its time-varying mass," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2674–2681, 2020. DOI: `10.1109/LRA.2020.2972827`.

[11] P. Stibinger, T. Baca, and M. Saska, "Localization of Ionizing Radiation Sources by Cooperating Micro Aerial Vehicles With Pixel Detectors in Real-Time," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3634–3641, Apr. 2020.

[12] D. Turecek, J. Jakubek, E. Trojanova, and L. Sefc, "Single layer compton camera based on timepix3 technology," *Journal of Instrumentation*, vol. 15, no. 1, 2020. DOI: `10.1088/1748-0221/15/01/c01014`.

[13] T. Baca, M. Jilek, P. Manek, *et al.*, "Timepix radiation detector for autonomous radiation localization and mapping by micro unmanned vehicles," *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019. DOI: `10.1109/iros40897.2019.8968514`.

[14] U. Jain, R. Tiwari, and W. W. Godfrey, "Multiple odor source localization using diverse-PSO and group-based strategies in an unknown environment," *Journal of Computational Science*, vol. 34, pp. 33–47, May 2019.

[15] Z. Liu and S. Abbaszadeh, "Double q-learning for radiation source detection," *Sensors*, vol. 19, no. 4, p. 960, 2019. DOI: `10.3390/s19040960`.

[16] M. Vrba, D. Heřt, and M. Saska, "Onboard marker-less detection and localization of non-cooperating drones for their safe interception by an autonomous aerial system," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3402–3409, Oct. 2019, ISSN: 2377-3766. DOI: `10.1109/LRA.2019.2927130`.

[17] T. Baca, D. Turecek, R. McEntaffer, and R. Filgas, "Rospix: Modular software tool for automated data acquisitions of timepix detectors on robot operating system," *Journal of Instrumentation*, vol. 13, no. 11, 2018. DOI: `10.1088/1748-0221/13/11/c11008`.

[18] W. Gao, W. Wang, H. Zhu, G. Huang, D. Wu, and Z. Du, "Robust Radiation Sources Localization Based on the Peak Suppressed Particle Filter for Mixed Multi-Modal Environments," *Sensors*, vol. 18, no. 11, p. 3784, Nov. 2018.

[19] Stanford Artificial Intelligence Laboratory et al., *Robotic operating system*, version ROS Melodic Morenia, May 23, 2018. [Online]. Available: `https://www.ros.org`.

[20] D. Turecek, J. Jakubek, E. Trojanova, and L. Sefc, "Compton camera based on timepix3 technology," *Journal of Instrumentation*, vol. 13, no. 11, p. C11022, Nov. 2018. DOI: `10.1088/1748-0221/13/11/C11022`. [Online]. Available: `https://dx.doi.org/10.1088/1748-0221/13/11/C11022`.

[21] P. Martin, S. Kwong, N. Smith, *et al.*, "3d unmanned aerial vehicle radiation mapping for assessing contaminant distribution and mobility," *International Journal of Applied Earth Observation and Geoinformation*, vol. 52, pp. 12–19, Oct. 2016.

[22] X.-B. Tang, J. Meng, P. Wang, *et al.*, "Efficiency calibration and minimum detectable activity concentration of a real-time UAV airborne sensor system with two gamma spectrometers," *Applied Radiation and Isotopes*, vol. 110, pp. 100–108, Apr. 2016.

[23] Y. Sanada and T. Torii, "Aerial radiation monitoring around the Fukushima Dai-ichi nuclear power plant using an unmanned helicopter," *Journal of Environmental Radioactivity*, vol. 139, pp. 294–299, Jan. 2015.

[24] C. Yuan, Z. Liu, and Y. Zhang, "Uav-based forest fire detection and tracking using image processing techniques," pp. 639–643, 2015. DOI: `10.1109/ICUAS.2015.7152345`.

[25] E.-w. Bai, K. Yosief, S. Dasgupta, and R. Mudumbai, "The maximum likelihood estimate for radiation source localization: Initializing an iterative search," *53rd IEEE Conference on Decision and Control*, Dec. 2014. DOI: `10.1109/cdc.2014.7039394`.

[26] H. E. Baidoo-Williams, S. Dasgupta, R. Mudumbai, and E. Bai, "On the gradient descent localization of radioactive sources," *IEEE Signal Processing Letters*, vol. 20, no. 11, 1046–1049, 2013. DOI: `10.1109/lsp.2013.2279499`.

[27] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework basedon octrees," *Autonomous Robots*, 2013, Software available at `https://octomap.github.io`. DOI: `10.1007/s10514-012-9321-0`. [Online]. Available: `https://octomap.github.io`.

[28] K. Buesseler, M. Aoyama, and M. Fukasawa, "Impacts of the Fukushima Nuclear Power Plants on Marine Radioactivity," *Environmental Science & Technology*, vol. 45, no. 23, pp. 9931–9935, Nov. 2011.

[29] J. Fan, W. Xu, Y. Wu, and Y. Gong, "Human tracking using convolutional neural networks," *IEEE transactions on Neural Networks*, vol. 21, no. 10, pp. 1610–1623, 2010.

[30] J. Welle, D. Schulz, T. Bachran, and A. B. Cremers, "Optimization techniques for laser-based 3d particle filter slam," pp. 3525–3530, 2010.

[31] H. Eisenbeiss, "A mini unmanned aerial vehicle (uav): System overview and image acquisition," Oct. 2009.

[32] G. Delaney, S. Jacob, C. Featherstone, and M. Barton, "The role of radiotherapy in cancer treatment: Estimating optimal utilization from a review of evidence-based clinical guidelines," *Cancer: Interdisciplinary International Journal of the American Cancer Society*, vol. 104, no. 6, pp. 1129–1137, 2005.

[33] R. Sim, P. Elinas, M. Griffin, J. J. Little, *et al.*, "Vision-based slam using the rao-blackwellised particle filter," vol. 14, no. 1, pp. 9–16, 2005.

[34] S. Brennan, A. Mielke, D. Torney, and A. Maccabe, "Radiation detection with distributed sensor networks," *Computer*, vol. 37, no. 8, pp. 57–59, Aug. 2004.

[35] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, 2004, pp. 2149–2154.

[36] P. Djuric, J. Kotecha, J. Zhang, *et al.*, "Particle filtering," *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19–38, 2003. DOI: `10.1109/msp.2003.1236770`.

[37] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering methods for practitioners," *IEEE Transactions on Signal Processing*, vol. 50, no. 3, pp. 736–746, 2002. DOI: `10.1109/78.984773`.

[38] M. Montemerlo, S. Thrun, and W. Whittaker, "Conditional particle filters for simultaneous mobile robot localization and people-tracking," vol. 1, 695–701 vol.1, 2002. DOI: `10.1109/ROBOT.2002.1013439`.

[39] J. Segen, "A camera-based system for tracking people in real time," vol. 3, pp. 63–67, 1996.

[40] J. Amanatides and A. Woo, "A fast voxel traversal algorithm for ray tracing," *Proceedings of EuroGraphics*, vol. 87, Aug. 1987.